Ontology Enrichment and Automatic Population From XML Data

Christophe CRUZ, Christophe NICOLLE
Laboratory Le2i
Université de Bourgogne
B.P. 47870, 21078
Dijon CEDEX, France

{christophe.cruz, cnicolle}@u-bourgogne.fr

ABSTRACT

This paper presents a flexible method to enrich and populate an existing OWL ontology from XML data. Basic mapping rules are defined in order to specify the conversion rules on properties. Advanced mapping rules are defined on XML schemas and OWL XML schema elements in order to define rules for the population process. In addition, this flexible method allows users to reuse rules for other conversions and populations.

1. INTRODUCTION

XML data is composed of XML documents and XML schemas which validate the corresponding XML documents. Actually, an XML schema contains the knowledge that the application has to share through data exchange. However, XML covers only the syntactic level, but doesn't support the semantic level. The semantic level describes the meanings between the input and output. The syntactic level is a set of rules that allows to create a sentence, which will give the computer a set of instructions in order to complete a particular task. The lexical level deals with input device dependencies in which the user will specify the exact syntax [18]. In the context of computer and information sciences, an ontology defines a set of representational primitives which allow to model a domain of knowledge or discourse [21]. Ontologies are widely used to capture and organize knowledge concerning a particular domain. The knowledge defined in ontologies is used as an index to retrieve specific data [1], to infer new knowledge [2], to semantically annotate multimedia data [3], to find out Web Services automatically [4], or to match knowledge with other knowledge for a more general purpose.

XML schemas contain the knowledge of a domain that was specified by the author. This specification is only syntactic without any semantical definition. This is due to the fact, that XML data are used to exchange data between processes that were developed for this data. In order to permit the exploitation of the knowledge contained in XML schemas and instances, we propose an ontology enrichment and an automatic population process from XML data based on a manual mapping of XML schemas. Ontology enrichment is the activity of extending an ontology by adding new elements (e.g. concepts, relations, properties, axioms) [3]. The enrichment process in our solution consists in annotating knowledge which is contained in XML schemas in order to define the ontology schema [23]. The knowledge extract here is manual and is done by the user who is the only one who knows which part of the XML schema will be required in future processes. The ontology population process is the activity of adding new instances to an ontology [3]. According to [9, 10, 11, 12], data integration can be undertaken by defining rules of mapping between information sources and the ontological level. These rules consist in adding a semantic layer to source elements and thus provide semantic definition to elements with regard to a consensual definition of the meaning. Our process consists in mapping XML schema elements to an existing ontology in order to enrich the ontology and to automatically populate it. Then the populated ontology can be used to find out new knowledge, to extract a new XML schema with its associated XML document. The following section discusses about previous work done on this subject. Section three presents the principles of our method which is based on formal languages. Section four shows our method to enrich the ontology by matching schemas and to populate by an automatic process.

VLDB '08, August 24-30, 2008, Auckland, New Zealand.

2. Previous work

Some work has been done specifically on the translation of XML schemas into OWL ontology [5, 6, 14, 15, 16, 17] (e.g. table 1). In table 1, "XSLT" means that the method uses an XSL style sheet such as XML data for the conversion process. "Automatic XSD mapping" means that the user cannot intervene in the mapping process between XML schema elements and the OWL ontology. "Automatic XML instances" means that if instances are generated by the studied method then the process is automatic. "Multiple XSD integration" means that the project allows users to integrate several XML schemas to an OWL ontology. "OWL-DL" means that the generated ontology is a description logic ontology which allows inference and consistency checking. "Mapping in RDF" means that the method uses RDF to specify the mapping between schemas. The last column implies that if the value is "yes" then XML schemas are mapped to an OWL ontology and the instances of the XML schemas are translated in an RDF document. It means that instances are not OWL instances. The last row of the table presents the properties of our method. Our method doesn't use XSLT because the process is too complex to be used with an XSLT processor (e.g. regex). The mapping is done manually but the population of the ontology is automatic. We also allow the user to integrate several XML schemas into an existing OWL-DL ontology. In addition, in order to specify the mapping between schemas and the ontology, we use the RDF language in order to permit an advanced management of mapping

rules. Finally, the instances of the ontology are obviously defined in the model of the ontology schema. The approach in [16] allows the mapping of XML schemas to an existing ontology and appropriately generates rules that automatically transform XML instance documents to instances of the mapped ontology. The ontology is generally richer than the automatically mapped XML schema ontology. This is a manual process to map XML schemas to an existing OWL ontology. In order to generate the XSL document, mapping rules are defined. This approach is the first that takes into account two issues. First, it must be ensured that duplicate instances are detected in the XML document. Second, a complete support of the restrictions, such as maximal and minimal cardinality of properties over classes, is impossible to guarantee. Some solutions to those issues are given, such as appropriate warnings. Concerning the schema mapping, the method is powerful but lacks advanced functionality, such as the fusion of properties when schemas are more complex than the ontology schema. For instance, the first names are defined by separate tags in an XML schema and are defined in only one tag in a second schema. Thus, a rule can be defined to fuse the first names of a person. Our method allows also to define advanced rules of transformation in RDF that can be reused for other mappings. Consequently, we have extended the solution described in [16] by mapping several XML schemas in an existing OWL ontology, which consists in defining mapping rules between each XML schema to a common existing OWL ontology. In addition, we allow users to define partial mapping of XML schemas in order to enrich and populate relevant data for the desired data management process. Furthermore, we define the transformation rules in RDF that allows a more flexible and a more fine-grained rule definition in order to allow a partial reuse of annotated XML schemas and data type conversion.

Table 1. This table summarizes all properties of studied projects

Papers	XSLT	Automatic XSD mapping	Automatic XML instances	Multiple XSD integration	OWL-DL	mapping in RDF	XSD2OWL XML2RDF
[14] Ferdinand & al.	no	yes	yes	no	yes	no	yes
[05] García & al.	no	yes	yes	no	no	no	yes
[15] Bohring & al.	yes	yes	yes	no	yes	no	no
[16] Rodrigues & al.	yes	no	yes	yes	yes	no	no
[17] Anicic & al.	no	no	no	no	yes	yes	no
Cruz & Nicolle	no	no	yes	yes	yes	yes	no

3. Principle

The principle of our solution consists in annotating and linking various levels which are the semantic level (OWL schema) and the schematic level (XML schema). The method is articulated in two steps. The first step relates to the formalization of "mapping" rules between an XML schema and an OWL XML schema. The "mapping" rules make it possible to enrich an existing ontology of domain from concepts and relationships conceptually present in XML schemas. This can be realized by a machine. Semantically it will not be richer than the XML schema. Actually, a rich semantic mapping cannot be done by a machine for the moment. In addition, an ontology makes it possible to link the concepts and the relations from several schemas by amalgamating the attributes of common entities with the help of an identical semantic (figure 1 (E)).

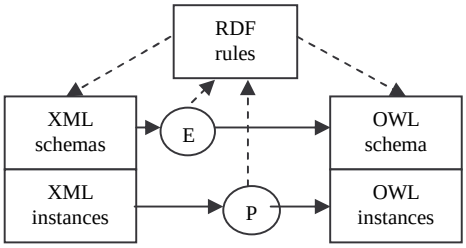


Figure 1. Enrichment and population of an existing ontology.

This step relates also to the definition of "basic mapping" rules in order to translate complex data such as sub trees into simple or complex attributes of a class. The second stage consists in populating the ontology previously enriched from XML documents validated by the mapped XML schema (figure 1 (P)). The population has to follow some rules such as the imitation of attribute cardinalities and unique instances. Consequently, "basic mapping" rules have to model and specify restrictions on attributes. This principle will be described in the next section. In order to specify the relevant elements of an XML schema for the enrichment process, it is necessary to identify and mark these elements. These marks are called "schematic marks" and are external RDF annotations of XML structures. (RDF rules in fig. 1). We do not use the term "annotation" to designate a mark because we make the distinction between internal annotations that are added to the document and external annotations that are stored in a repository outside the annotated document Actually, the "schematic marks" are a specific RDF graph that defines external annotations on XML schemas. In figure 1, (E) is the process of enriching an ontology and (P) is the process of populating the ontology. These two processes use an RDF graph as rules to enrich and to populate the ontology. The rules in RDF are defined during the mapping process. The following section presents a brief formalization of schematic marks.

3.1 Factor and Schematic Marks

The properties of Dyck's languages were the subject of studies undertaken by J. Berstel [19]. By drawing parallels between XML grammar and the languages of Dyck, J. Berstel defines the concept of "factor". This notion means that a language is factorizable in an under language and a factor of a Dyck's language is a Dyck's language. As a consequence, an under tree of an XML document can be generated by a factor of a Dyck's language to which the XML document belongs. According to the corollary 3.4, the notion of surface is used by Berstel to demonstrate the following proposition: For each XML language *L* there is only one reduced XML grammar generating *L*. This proposal implies that if a factor were defined on an XML language then this factor would

correspond to a production rule of the reduced XML grammar that generates this language. A reduced grammar does not have any useless non terminal vocabulary. An XML schema does not contain unnecessary tags, so an XML schema does not use unnecessary non terminal vocabulary. Consequently, an XML grammar is necessarily its own reduced grammar. This proposal makes it possible to introduce the concept of schematic mark. A **schematic mark** is a mark on an XML schema that identifies a production rule. Each tag of an XML instance which has the same name was produced by the same production rules. These marks are used in RDF rules to identify the production rules of the XML schema and are specified in XPath. (e.g. xsd:schema, xsd:element, xsd:attribute, etc.)

3.2 Semantic of the schematic marks

An OWL grammar is also an XML grammar which can be marked as an XML schema. An ambiguous point has to be underlined. In figures 1 and 2, the OWL schema concerns the schema part of the ontology. However, OWL has an XML schema in order to validate OWL documents [20]. In this section we focus on the XML schema of the OWL XML schema. This allows us to mark an OWL ontology, such as an XML schema in order to define the semantic of XML schema with OWL ontologies. Consequently, we are able to mark production rules of the OWL ontology. (e.g. owl:Ontology, owl:Class, rdfs:subClassOf, owl:Objectproperty, etc.). OWL uses most of the built-in XML schema data types. References to these data types are by means of the URI reference for the data type, http://www.w3.org/ 2001/XMLSchema. The following data types are recommended for the use with OWL, xsd:string, xsd:normalizedString, xsd:Boolean, etc.

4. Enrichment and population method

This section describes how the enrichment and the population of an OWL ontology are managed from XML schemas. The method is based on the definition of schematic marks, mapping rules and advanced mapping rules (fig. 2). The first part describes the schema marking in order to annotate the element of XML schemas. It relates to the XML schema but also to the OWL XML schema. The second part presents the mapping step which is composed of the conversion rules, the ontology enrichment process and the ontology population process.

4.1 Schema marking

An RDF graph is used to annotate each XML schema. These marks are specified to keep all information in a graph that is required during the mapping with the OWL schematic mark step (fig. 2). The TriG Syntax is used to ease the explanations of how we employ schematic marks with XML schemas. Example 1 shows how to mark a first schema and how to specify an id in order to avoid duplicate OWL instances. The “bmr” name space is used to identify the basic mapping rules. In addition, “bmr:xpath” defines an element in the XML schema as does “bmr:xpathId” with the addition that the latter acts as an identifier. A unique property on an XML element is welcome to define an identifier.

```
# TriG RDF mapping
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix bmr: <http://www.example.org/BasicMappingRules#> .
@prefix amr: <http://www.example.org/AdvancedMappingRules#> .
:XSDMarksSchema_1 {
:student                bmr:xpath    "/univ/students/student" .
:studentName            bmr:xpath    "/univ/students/student/name" .
:prof                   bmr:xpath    "/univ/profs/prof" .
:profName               bmr:xpath    "/univ/profs/prof" .
:profAge                bmr:xpath    "/univ/profs/prof/age" .
:studentAge             bmr:xpath    "/univ/ students/student/age" .
…
:studentId              bmr:xpathId    "/univ/students/student[@idNum]" .
:prof Id                bmr:xpathId    "/univ/profs/prof[@idprof]" . }
```

Example 1: schematic mark definitions on an XML schema.

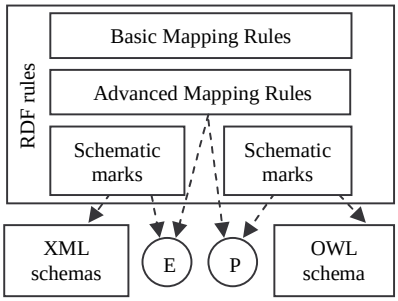


Figure 2. Relationships between XML and OWL schemas.

In figure 2, the processes (E) and (P) are here to show the relationships established with the RDF rules. In fact, the main objective of the figure is to describe the components of the RDF rules. They are composed of the schematic marks on XML schemas and on an OWL XML schema. These marks are used to identify elements required for the mapping process. Basic mapping rules define rules for the conversion of data from XML schemas to OWL. Advanced mapping rules use basic mapping rules in order to define the mapping between XML schemas and OWL. In addition, these rules allow to define new elements in the OWL ontology for the enrichment process and for the population process.

4.2 Mapping steps

4.2.1 Conversion rules

The conversion rules consist in defining rules in order to convert properties that cannot be directly copied to the ontology “datatypeProperty”. The first kind of conversion is simple because it is the conversion of a simple type into another simple type (ex. 2).

```
:conversionBasicRules_1 {
:string2int     bmr:conv        :rule1 .
:rule1          bmr:type        bmr:convSimple .
:rule1          bmr:domain      xsd:string .
:rule1          bmr:range       xsd:int . }
```

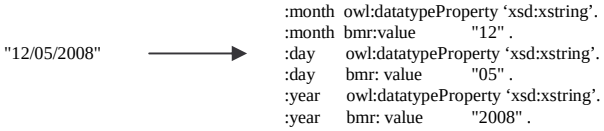
Example 2: A simple conversion from a string into an integer.

In example 2, the rule expresses only the semantic, the process has to be done by a program. A set of conversion methods has to be carried out once and for all and can be reused for complex conversion rules.

The second kind of conversion is complex because it is the conversion of a sub tree from the schema into a simple type into the OWL ontology.

```
<date >     <month>12</ month >
            <day>05</day >                  ⟶    "12/05/2008"
            <year>2008< /year >  </date>
```

The conversion of complex data can also be found in a semi-structured data. For instance, a date can be defined in a text format and could have to be converted in month, day and year format in the ontology.



Those rules are expressed with the help of an RDF graph as Basic Mapping Rules.

4.2.2 Ontology population

In order to generate the population of the ontology, the RDF mapping has to be defined. To achieve this, relationships have to be created between the RDF graph of the XML schema marks, the RDF graph of the OWL schematic marks and the RDF graph of the conversion rules (ex. 2).

```
RDFmapping{
:graph1   amr:convBR          :conversBasicRules_1 .
:graph2   amr:convBR          :conversBasicRules_2 .
:graph3   amr: convBR         :conversBasicRules_3 .
:graph4   amr:owl   Marks     :OWLMarks_1.
:graph5   amr:xsdMarks        :XSDMarksSchema_1 .
:graph6   amr:mappingRules    :mappingRules_1 . }
```

Example 2: RDF graph of comversion rules.

In example 2, we define the RDF graph of mapping. It contains previously defined graphs such as the conversion rules, and schematic marks.

4.2.3 Ontology enrichment

The enrichment consists in defining relationships between XML schematic marks and OWL schematic marks. In order to enrich the ontology, the RDF mapping graph has to contain information about new entities in the ontology that have to be created (ex. 3). The name space "amr" represents the "advanced mapping rules".

```
mappingRules_1 {
:mapRule01      amr:range       :prof, :member .
:mapRule01      amr:domain      :person .
:mapRule02      amr:range       :profId, :memberId .
:mapRule02      amr:domain      :personId .
:mapRule02      amr:basicRule   :string2int .
…
:mapRule06      amr:newDTP      :age .
:age            amr:DTPname     "personAge" .
:age            amr:range       xsd:int.
:age            amr:domain      :person .
:mapRule07      amr:range       :age .
:mapRule07      amr:domain      :profAge .
:mapRule07      amr:basicRule   :string2int .
… }
```

Example 3: Advanced Mapping Rules.

In this example, "prof" and "member" are mapped to the class person with the data type property personId (with a conversion rule :string2int) and personName. In this example, the property age for a person is created in order to enrich the ontology and the age of the professors is mapped to this new data type property. Object properties and classes can be created in the same way by using "amr:newOP" and "amr:newClass".

5. Conclusion

We have presented a flexible method to enrich and populate an OWL ontology for the integration of XML data. Basic mapping rules and advanced mapping rules are defined by users and can be reused for other conversions and populations of ontologies. The RDF rules can be used to automatically extract from XML schemas some elements that can be converted in order to help users during the mapping.

6. REFERENCES

[1] García, R., Celma, O.: Semantic Integration and Retrieval of Multimedia Metadata, ISWC, Galway, Ireland, 2005.

[2] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/, 2004.

[3] Castano, S., Espinosa, S., Ferrara, A., Karkaletsis, V., Kaya, a., Melzer, S., Moller, R., Montanelli S., Petasis, G.: Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology, (IWOD) ESWC, Innsbruck, Austria, 2007.

[4] Martin, D., Paolucci, M., Wagner, M.,: Towards Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective, ESWC, June, Innsbruck, Austria, 2007.

[5] García, R., Celma, O.: Semantic Integration and Retrieval of Multimedia Metadata, ISWC, Galway, Ireland, 2005.

[6] Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches, VLDB, Hongkong, Aug. 2002.

[7] Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++, SIGMOD Conference, 2005.

[8] Rahm, E. and Bernstein, P.: A survey of approaches to automatic schema matching. The VLDB Journal 10, 334-350, 2001.

[9] Cruz, I. F., Xiao, H., Hsu, F., 2004.: An Ontology-based Framework for Semantic Interoperability between XML Sources, IDEAS, 2004.

[10] Klein, M., 2002.: Interpreting XML via an RDF schema. In ECAI workshop on SAAKM, Lyon, France, 2002.

[11] Lakshmannan, L. V., Sadri, F.,: Interoperability on XML Data, ICSW, 2003.

[12] Cruz, C., Nicolle, C.: Ontology-Based Integration of XML data, Webist, Setubal, Portugal, pp. 30-37, 2006.

[13] Huynh Quyet Thang, Vo Sy Nam, XML Schema Automatic Matching Solution, International journal on Information Systems Science and Engineering, vo.l 4, number 1, 2008.

[14] Matthias Ferdinand and Christian Zirpins and D. Trastour: Lifting XML Schema to OWL, in: Koch, Nora and Fraternali, Piero and Wirsing, Martin (Hrsg.): Web Engineering, ICWE 2004, Munich, Germany, July 26-30, 2004.

[15] Bohring, H.; Auer, S.: Mapping XML to OWL Ontologies. Leipziger Informatik-Tage (LIT 2005), Sep. 21-23, 2005, Lecture Notes in Informatics (LNI).

[16] Rodrigues, T., Rosa, P. and Cardoso, J.: Mapping XML to Exiting OWL ontologies, International Conference WWW/Internet 2006.

[17] Anicic, N., Ivezic, N. and Marjanovic, Z.: Mapping XML Schema to OWL, Enterprise Interoperability, Springer London, 2007.

[18] Bowers S, Delcambre L.: Representing and Transforming Model-Based Information, In Proceedings of the Workshop on Semantic Web at ECDL-00, Lisbon, Portugal, 2000.

[19] Berstel, J., Boasson, L.,: XML Grammars, MFCS 2000, , 2000.

[20] OWL XML schema, http://www.w3.org/, 2007.

[21] Thomas, C.: Conceptual, Semantic, Syntactic, and Lexical Model, 2002.

[22] Gruber, T.: Ontology, to appear in the Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2008.

[23] Faatz, A., and Steinmetz, R.,: Precision and recall for ontology enrichment. In Proc. of ECAI-2004 Workshop on Ontology Learning and Population, Valencia, Spain, Aug. 2004.