An Ontology for XML Schema to Ontology Mapping Representation

Kai Yang, Robert Steele and Amanda Lo
University of Technology, Sydney
{kayang|rsteele|amanda}@it.uts.edu.au

Abstract:
This paper addresses the problem that exists in the context of XML to Ontology translation. We firstly discuss the problem regarding the loss of information during roundtrip transformation between XML and Ontology followed by the proposal of a mapping representation ontology for modeling concept mappings defined between XML schema and Ontology. Our goal is to enable bidirectional data conversion between XML and Ontology as well as achieving seamless XML data translation through ontology mediation.

1 Introduction

In recent years, XML has become the de facto standard for data exchange on the web, especially in the E-Business domain. XML schema as an emerging data structure definition standard has also reached a wide acceptance. Organizations from the same application domain define and advocate different data schemas for similar purposes, as a result, XML data transformation is required for data exchange between heterogeneous and independently developed applications or systems.

Current works in the area of XML document transformation mainly focus on direct XML-XML translation. In other words, XML document structured under the source schema is directly transformed into an instance of the target schema based on the concept mappings defined between the source and target schema. However, when dealing with a large number of heterogeneous XML data sources, such approach is less scalable and requires $N!$ number of schema mappings to be defined among different sources (where N stands for the number of data sources). To reduce the number of interactions required among different systems, some researches introduced ontology as a mediator for data exchange [16, 14]. In those works, local schemas are mapped to a predefined global ontology. During data transformation, documents in one schema format are transformed into the global ontology format, which will then be transformed into the target schema format. Ontology mediated approach reduces the number of system interactions from $N!$ to N. Hence it is more scalable and feasible for data exchange among large number of heterogeneous data sources.

However, differences between XML schema and Ontology prevents data structural information defined in XML schema to be precisely described using ontology [10], likewise, XML Schema is unable to accurately model the concepts and relations defined in ontology. These differences lead to possible information loss during the transformation from XML document to ontology instance or vice versa, thus they become a major obstacle for ontology mediated XML data exchange.

To overcome this problem, and to enable bidirectional transformation between XML document and ontology instance, this paper proposes a mapping representation ontology for describing concept mappings defined between XML schema and ontology. Our primary goal is to encapsulate a sufficient amount of information in order to compensate the loss of information that has occurred during data translation process. The objectives of this paper include: i) To analyze the problems that exist during forward and backward XML document to ontology document translation. ii) Introduce the mapping representation ontology and show how it can be used to assist two way translations between XML and ontology documents. iii) Introduce an ontology mediated data exchange approach based on the mapping representation ontology. The remaining sections of this paper are organized as follow: Section 2 shows related works in this area. The challenges of bidirectional XML to ontology transformation are discussed in Section 3. The mapping representation ontology is described in Section 4. Section 5 introduces the ontology mediated XML data transformation approach, followed by the conclusion and future work in Section 6.

2 Related Work

Several languages have been developed for the mutual conversion between different XML formats [9] and conversion between XML format to other data formats such as plain text and HTML [4, 3, 8]. However, none of them specifically focus on the bidirectional transformation between XML format and ontology format. biXid [9] is a language proposed for the bidirectional transformation between different XML data formats. biXid is based on programming-by-relation and it allows both non-linear pattern variable and full ambiguity. XSugar [3] is another language specifically designed for bidirectional conversion between XML and text formats, and it emphasizes on the reversibility of data transformation. Similar to XSugar, our research focuses on the reversibility of XML to Ontology transformation, and we are aiming to enable roundtrip translation between XML document and ontology. However, due to the differences exist between XML schema and ontology, the challenges faced by our research is different from XSugar. Since our research emphasizes on the expressive power of ontology, the utilization of ontology will enhance the accuracy of the transformation between XML and ontology format.

Another direction in data exchange is the utilization of global ontology and this approach is known as the ontology mediated approach. Ontology mediated data exchange employs a global ontology for transforming data between different formats. Projects such as the European 'HARMO-TEN' project [6] and SOIRA [19] are typical examples of ontology mediated data exchange. The 'HARMO-TEN' project [6] aims to create an electronic space for tourism stakeholders so that all businesses in the marketplace are able to exchange their information in a seamless manner. Their integration process consists of two phases: the *customization phase* and the *cooperation phase.* The *customization phase* focuses on defining mappings between the local schema and the global ontology. The *cooperation phase* covers instance data translation. During the *cooperation phase*, local data instance is transformated into a local ontology instance, which is then transformed into a global ontology instance. The SOIRA architecture [19] also uses ontology mediated XML document transformation. Instead of translating all local schemas into local ontology, they proposed to translate the global ontology into a global XML schema, and then map local XML schemas to the derived global XML schema. The semantic mapping process is performed base on a set of similarity calculation including name, data type and structure. However, semantic mappings are established base on syntactical similarity rather than semantic similarity, and problems such as granularity and structure difference are ignored. Besides, neither project addressed the problem of information loss during XML to ontology translation.

R. Steele and A. Yu have revealed the strengths and limitations of current researches on XML schema to UML translation [18], their finding is used as a valuable resource for our research. Other works focusing on the problem of XML to ontology transformation include the JXML2OWL framework [15] and the XML to OWL transformation approach [2]. While the former approach is based on concept mappings defined at the schema level, the latter approach is based on mappings defined at metadata level. However, neither approach is reversible or deals with the scenario of ontology to XML transformation.

In contrast, our research specifically focuses on the problem of data conversion between XML format and ontology format and our contributions can be summarized as follows:

i) The analyses of problems exist during translation between XML document and ontology instance.

ii) The development of a mapping representation ontology for representing concept mappings defined between XML schema and ontology. The mapping representation ontology is capable for capturing unique information defined in both XML schema and ontology.

iii) The proposal of an ontology mediated XML transformation approach to enable seamless data exchange among heterogeneous XML sources.

3 XML Schema and Ontology

Previous studies already revealed the differences between XML schema and Ontology [5, 12, 10, 7], while XML schema provides rich syntax and structure definitions for data modeling, ontology, on the other hand, allows more sophisticated semantic modeling of a particular domain. M. Klein, et al. [10] gave a more detailed analysis on the differences between OIL and XML schema, this paper further expands on their findings and analyses the possible problems that may occur during the translation between XML and ontology instance.

3.1 XML schema and Ontology Comparison

In this paper, we generally divide the differences between XML schema and Ontology into three groups: data type, structure and relation. Table 1 shows the major differences between XML schema and Ontology, their similarities are not covered.

	XML Schema	Ontology
Data type	XML schema supports large number of built-in data types including string, boolean, decimal, float, date, etc. A complete list of datatypes is documented in [1].	Some ontology languages such as RDF and OIL only supports limited number of data types. Others such as DAML and OWL allows use of XML Schema datatypes by referring to the datatype URI.
Structure	XML schema uses nested data structure, where each element can be mixed with other simple, complex or mixed elements. The top-most element is considered as the root element of the concept hierarchy. Upper elements are seen as the parent of its content lower elements.	Ontology supports element composition through properties. Each class can have various datatype properties and object properties. However, ontology is an object-oriented conceptual model rather than a hierarchy of terms or concepts. Therefore, every class existing in the ontology can be seen as the root element.
	XML schema allows the definition of structural constraints, concepts such as sequence is used to describe the order between content items.	Ontology does not support order between properties.
Relation	XML schema only supports inheritance through type derivation (extension or restriction). It does not support multiple-inheritance.	Ontology supports multiple-inheritance, one class can inherit properties from multiple parent classes.
	XML schema does not provide grammars for relation constraints definition.	Ontology supports inheritance on properties, it also provides simple logics on relations such as *transitive* and *symmetric* for reasoning on class.

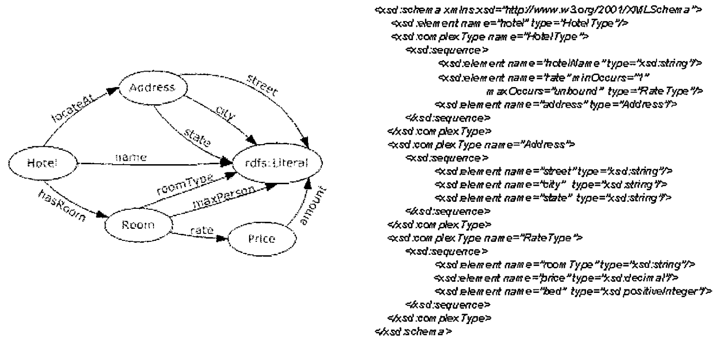
Table 1: XML schema and Ontology Difference

Due to these fundamental differences, we argue that it is not feasible to translate a XML document completely into an ontology instance or vice versa without loss of structural or semantic information.

3.2 Translation between XML and Ontology

We denote the translation of XML to Ontology as $T_{x \rightarrow o}$ and the translation of Ontology to XML as $T_{o \rightarrow x}$. Given an XML input X_i, the output of $T_{x \rightarrow o}$ is expressed as $O_o = T_{x \rightarrow o}(X_i)$.

Similarly, given an Ontology input instance O_i, the output of $T_{o \to x}$ is written as $X_o = T_{o \to x}(O_i)$. Additionally, given an input XML document X_i, $T_{x \to o}$ is said to be reversible if and only if $X_i = T_{o \to x}(T_{x \to o}(X_i))$. Alternatively, given an input ontology instance O_i, $T_{o \to x}$ is reversible if and only if $O_i = T_{x \to o}(T_{o \to x}(O_i))$.



```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="hotel" type="HotelType"/>
   <xsd:complexType name="HotelType">
      <xsd:sequence>
            <xsd:element name="hotelName" type="xsd:string"/>
            <xsd:element name="rate" minOccurs="1"
                     maxOccurs="unbound" type="RateType"/>
           <xsd:element name="address" type="Address"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="Address">
      <xsd:sequence>
           <xsd:element name="street" type="xsd:string"/>
           <xsd:element name="city"  type="xsd:string"/>
           <xsd:element name="state"  type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="RateType">
      <xsd:sequence>
           <xsd:element name="roomType" type="xsd:string"/>
           <xsd:element name="price" type="xsd:decimal"/>
           <xsd:element name="bed"  type="xsd:positiveInteger"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

a.) Ontology schema for Hotel Price

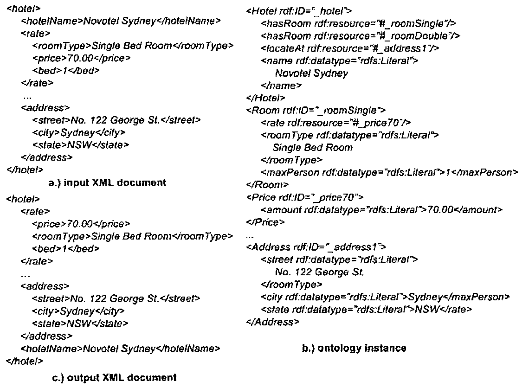
b.) XML schema for Hotel Rate

Figure 1: Schemas for Hotel Rate

To find out the possible problems that may occur during the translation process this paper uses the hotel rate data structure as an example. This example is generically representative of the bidirectional XML to ontology translation challenges. Figure 1.a shows the hotel rate ontology in a graph form. Figure 1.b shows an XML schema equivalent to the hotel rate ontology. Using the conventional mapping representation approach [15], concept mappings between the given schema and ontology can be expressed as a 2-tuple *(Ontology Concept URI, XPath Expression)*, where *XPath Expression* represents the XML schema element or attribute and the *Ontology Concept URI* represents the corresponding concept defined in the ontology. E.g. the concept mapping between the element address and the class *Address* is documented as *(Address, /hotel/address)*.

Following $T_{x \to o}$ proposed in [15], an XML document (Figure 2.a) is transformed into ontology format (Figure 2.b) written in OWL. The generated ontology instance conforms to the ontology schema defined in Figure 1.a. However, during $T_{o \to x}$ different forms of XML documents can be generated, each following a unique format. Figure 2.c shows one XML instance generated from $T_{o \to x}$. This XML document is not equivalent to its original format and it does not comply with the XML schema defined in Figure 1.b. Hence, we say that the XML to ontology transformation process proposed in [15] is not reversible. Same experiment is applied to other XML to Ontology translation approaches including [2, 15, 17, 12] to test their reversibility, however, none of them support bidirectional transformation between XML and Ontology.

Through the analysis of result from previous experiment, three factors are detected, which lead to the failure of roundtrip transformation: i) Element order is ignored during $T_{x\rightarrow o}$, and this is due to the difference between XML schema and ontology as discussed in Section 3.1. ii) Data type information is also lost during $T_{x\rightarrow o}$, as most ontology only uses primitive data types. iii) Most importantly, no information captured by the concept mapping can be used by $T_{o\rightarrow x}$ to compensate the loss of information occurred during $T_{x\rightarrow o}$, consequently, none of the selected approaches allows consistent roundtrip translation from XML to Ontology.



```
<hotel>
    <hotelName>Novotel Sydney</hotelName>
    <rate>
        <roomType>Single Bed Room</roomType>
        <price>70.00</price>
        <bed>1</bed>
    </rate>
    ...
    <address>
        <street>No. 122 George St.</street>
        <city>Sydney</city>
        <state>NSW</state>
    </address>
</hotel>
```

a.) input XML document

```
<hotel>
    <rate>
        <price>70.00</price>
        <roomType>Single Bed Room</roomType>
        <bed>1</bed>
    </rate>
    ...
    <address>
        <street>No. 122 George St.</street>
        <city>Sydney</city>
        <state>NSW</state>
    </address>
    <hotelName>Novotel Sydney</hotelName>
</hotel>
```

c.) output XML document

```
<Hotel rdf:ID="_hotel">
    <hasRoom rdf:resource="#_roomSingle"/>
    <hasRoom rdf:resource="#_roomDouble"/>
    <locateAt rdf:resource="#_address1"/>
    <name rdf:datatype="rdfs:Literal">
        Novotel Sydney
    </name>
</Hotel>
<Room rdf:ID="_roomSingle">
    <rate rdf:resource="#_price70"/>
    <roomType rdf:datatype="rdfs:Literal">
        Single Bed Room
    </roomType>
    <maxPerson rdf:datatype="rdfs:Literal">1</maxPerson>
</Room>
<Price rdf:ID="_price70">
    <amount rdf:datatype="rdfs:Literal">70.00</amount>
</Price>
...
<Address rdf:ID="_address1">
    <street rdf:datatype="rdfs:Literal">
        No. 122 George St.
    </roomType>
    <city rdf:datatype="rdfs:Literal">Sydney</maxPerson>
    <state rdf:datatype="rdfs:Literal">NSW</rate>
</Address>
```

b.) ontology instance

Figure 2: Sample XML Document and Ontology Instance

4 Ontology based Mapping Representation

As explained above, the problem of information loss, especially loss of order and datatype information, is hardly prohibitable during data translation. For this reason, this paper proposes a compensation approach to enable bidirectional translation between XML and Ontology. We argue that an adequate amount of concept mapping information can be used to compensate the information lost during single-trip translation. Given an XML schema X, an Ontology schema O and a set of concept mappings defined between X and O is denote as M_{xo}, the logical constraints defined by the schema X is defined as set $\sum_x$ covering all syntactical and structural definitions, while logical constraints defined by the ontology O is defined as $\sum_o$ covering all concepts, properties and relations. In addition, the logical information encapsulated in M_{xo} is defined as $\sum_M$. The problem of information loss occurred during the translation between XML and Ontology is denoted as $\sum_{loss} = \neg(\sum_x \cap \sum_o)$, and to compensate the loss of information the condition $\sum_{loss} \subseteq \sum_M$ has to be meet so that $\sum_x \subseteq (\sum_o \cup \sum_M)$ and $\sum_o \subseteq (\sum_x \cup \sum_M)$.

Due to the expressive power of ontology, many researches use ontology for representing concept mappings [13, 19]. Influenced by these works, we propose an ontology based approach for representing mappings between XSD and ontology. The representation ontology is capable for capturing unique information from both XSD and ontology so that it allows consistent roundtrip data transformation between XML and ontology. Similar to the hierarchy defined in XDM (XQuery and XPath Data Model), we treat the structure of concept mapping as a recursive node-leaf structure. The goal is to retain structural and order information defined by XML schema, while capturing the semantic information defined in ontology.

Figure 3 shows the mapping ontology in a graph form, the class *Mapping* is defined to represent atomic mappings between an atomic XSD concept and an ontology entity. Another class *Composite Mapping* is used to represent a node structure, where its sub-nodes are represented by other *Composite Mapping* instances, and its leaves are represented using *Mapping* instances. In addition, the *Mapping* class is associated with an *Atomic XSD Entity* and an *Ontology Entity*, where the *Atomic XSD Entity* retains all information required for the construction of an atomic XML element or attribute. A set of formal definitions is given below.

- A *Composite Mapping* is defined as a 3-tuple $CM = ((M_s|CM_s|EM_s)^*, CE?, OE?)$ where M_s is the sub-mapping; CM_s is the sub-composite mapping; EM_s is the sub-empty mapping; CE is a *Composite XSD Entity* used for capturing structural information of a complex XML schema element; OE is an *Ontology Entity* used for representing class, datatype property or object property from Ontology. An *Empty Mapping* is a *Composite Mapping* with either no CE or OE.

- A *Mapping* is defined as a 2-tuple $M = (AE, OE)$ where AE is an *Atomic XSD Entity* used for capturing concept information of the atomic XML schema element selected for the mapping.

- An *Atomic XSD Entity* is defined as a 3-tuple $AE = (P, N, DT)$ where P is the XPath of AE; N is the name of AE; DT is the data type of AE, which can be any XML schema simple type.

- An *Ontology Entity* (OE) is defined as an entity with reference to other Classes, Data Type Properties or Object Properties from the domain Ontology.

Note that conventional mapping representation approach is considered as ontology-oriented, meaning that for each ontology class, datatype property or object property, a corresponding concept mapping representation is created so that all mappings are linked together through ontology relations [15]. Such approach is beneficial for retaining semantic information defined in ontology, thus is mainly used for XML to ontology translation. In contrast, the proposed mapping representation ontology focuses on the structure defined by XML schema. Starting from the root element, for each composite element a composite mapping is created to model its position. Element sequence is modeled using left-right relation, where element appear in the left mapping has higher order ranking than its right side correspondences. As defined earlier, an *Empty Composite Mapping* is a *Composite Mapping* with no corresponding ontology component or XML schema component. It is used for solving structure or concept differences between XML schema and ontology.

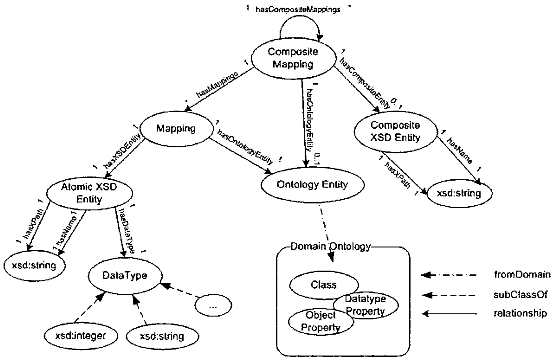


Figure 3: XML Schema and Ontology Mapping Representation Ontology

The following rules are proposed for the creation of mapping representation instance:

1) A *Mapping* instance is created when a mapping is defined between either a xsd:attribute or xsd:element containing no sub-elements or xsd:simpleType and an Ontology Datatype Property.

2) A *Composite Mapping* instance is created when a mapping is defined between either a xsd:complexType or xsd:element containing sub-elements and an Ontology Class. Same set of rules applies to the sub-elements of the composite element.

3) Given two mappings (either common Mappings, Composite Mappings or Empty Mappings) M_1 and M_2, if a relation R exists from OE_1 in M_1 to OE_2 in M_2, then an *Empty Mapping* instance EM_i is created to model R between M_1 and M_2, and with M_1 is the parent of EM_i and M_2 is the child of EM_i. (As shown in Figure 4.a)

4) An *Empty Mapping* instance is created when structure difference occurs between a parent mapping and its child mapping. *Empty Mapping* with ontology class or relation is created to model missing ontology structure or concept. *Empty Mapping* with XML element is used to model missing XML structure or concept. (As shown in Figure 4.b)

To assist the understanding of above rules, Figure 4 shows a mapping representation instance created between the hotel rate schema (Figure 1.b) and the hotel rate ontology (Figure 1.a).

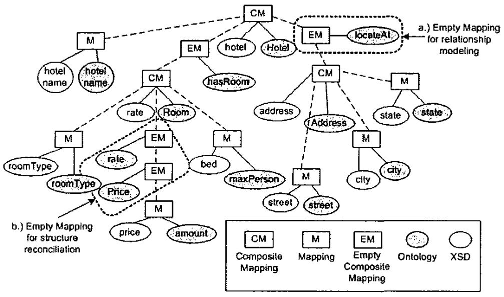


Figure 4: Mapping Representation Instance

To simplify the graph, data type information is ignored. Here, we will further explain rule No.4 and will demonstrate how the problem of structural difference between the schema and ontology is solved by applying rule No.4. As defined in the hotel rate schema, the XML element '/hotel/rate/price' can be mapped to the property 'amount' from the class 'Price'. However, no mapping has been defined for the class 'Price', which causes a problem when translating 'price' into 'amount'. To solve this problem, two *Empty Composite Mappings* are created, as shown in Figure 4. The *Price* mapping is used to link the property 'amount' with the class 'Price' and the *rate* mapping is used to link the class 'Price' with class 'Room'. Both mappings are *Empty Composite Mapping* with no corresponding XML schema components.

The generated mapping instance captures information defined in both XML schema and ontology. It integrates the ordered tree structure defined by XML schema together with the graph structure defined by ontology. All ontology entities (documented in gray circles) are linked together following the constraints $\sum_o$ defined by the domain ontology. At the same time, all schema entities (documented in white circles) are inter-connected following the logical constraints $\sum_x$ defined by the XML schema, sequence information is captured using left-right rules, where left hand side element has a higher order ranking than the right hand side elements. Hence, we conclude that the logical information encapsulated in the concept mapping $\sum_M$ is a superset of the information loss occurred during data translation, and the captured information can be used to compensate information loss occurred during single-trip translation.

5 Ontology mediated XML Transformation

Based on the mapping representation ontology introduced in Section 4, this section proposes an ontology mediated XML transformation approach to enable seamless data exchange between heterogeneous XML data sources. Figure 5 shows the ontology mediated XML transformation approach. Initially, a set of mappings are defined between various local schemas and the global domain ontology. Each set of concept mappings is represented using a mapping representation instance. During instance level translation, an XML document is firstly transformed into the ontology format, which will then be translated into target XML format.

A Java based framework is under development for automating bidirectional data translation. In the framework, a mapping ontology instance is unmarshalled into a group of inter-connected Java classes, and each class is used as a temporary data container during structure reconciliation. When an XML document is passed to the program, different element values are stored into their corresponding data containers depend on mapping rules defined. Then these data container are reunited into a new structure which is compliant to the defined domain ontology. Similar scenario occurs during the reverse translation process.

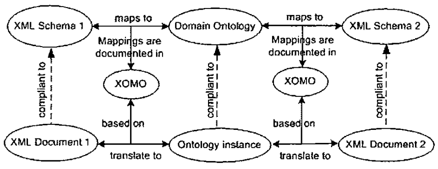


Figure 5: Ontology Mediated XML Transformation

6 Conclusion and Future Work

This paper analyzed the problems encountered in the context of data translation between XML and ontology. Due to the fundamental difference between XML schema and ontology, and the inadequacy of conventional mapping representation approaches, the problem of information loss, particularly sequence and datatype information, is inevitable when translating XML document to ontology instance. To solve this problem, we proposed a compensation approach and introduced a mapping representation ontology to capture unique information defined in both XML schema and ontology, so that during backward translation all information lost in the forward translation process can be compensated using the information captured by the mapping representation. We also introduced an ontology mediated XML transformation approach for XML data exchange.

In this paper, we use ontology as a trade-off between accuracy and efficiency of the transformation process. Future works in this area include further investigation into improvements and extension to the currently existing languages such as XSLT, XDuce to enhance the efficiency of the proposed transformation approach. A potential area is to translate mapping representation instance into XSL document to allow fast transformation execution. Another task is to compare the performance of the proposed XML transformation approach with other existing XML transformation approaches.

References

[1] Paul V. Biron, Kaiser Permanente, and Ashok Malhotra. Xml schema part 2: Datatypes second edition. 2004.

[2] Hannes Bohring and Soren Auer. Mapping xml to owl ontologies. `http://www.informatik.uni-leipzig.de/~auer/publication/xml2owl.pdf`.

[3] Claus Brabrand, Anders Moller, and Michael I. Schwartzbach. Dual syntax for xml languages. *Elsevier Science*, 2007.

[4] James Clark. Xsl transformations (xslt). Nov 1999.

[5] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of xml and rdf. *IEEE Internet Computing*, 2000.

[6] Mirella Dell'erba, Oliver Fodor, Wolfram Hopken, and Hannes Werthner. Exploiting semantic web technologies for harmonizing e-markets. *Information Technology & Tourism*, 2005.

[7] Yolanda Gil and Varun Ratnakar. A comparison of (semantic) markup languages. `http://www.isi.edu/expect/web/semanticweb/paper.pdf`.

[8] Haruo Hosoya and BenjaminA C. Pierce. Xduce: A typed xml processing language. *ACM Transactions on Internet Technology*, 2003.

[9] Shinya Kawanaka and Haruo Hosoya. bixid: A bidirectional transformation language for xml. *ICFP'06*, Sep 2006.

[10] Michel Klein, Dieter Fensel, Frank van Harmelen, and Ian Horrocks. The relation between ontologies and xml schemas. 2000.

[11] Michel C. A. Klein. Interpreting xml documents via an rdf schema ontology. *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, 2002.

[12] Patrick Lehti and Peter Fankhauser. Xml data integration with owl: Experience & challenges. *Proceedings of the 2004 International Symposium on Applications and the Internet*, 2004.

[13] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. Mafra - an ontology mapping framework in the context of the semantic web. *Workshop on Knowledge Transformation for the Semantic Web (KTSW)*, Jul 2002.

[14] Tova Milo and Sagit Zohar. using schema matching to simplify heterogeneous data translation. *Proceedings of the 24rd International Conference on Very Large Data Bases*, 1998.

[15] Toni Rodrigues, Pedro Rosa, and Jorge Cardoso. Mapping xml to existing owl ontologies. `http://dme.uma.pt/jcardoso/Research/Papers/WWW-Internet06-JXML2OWL.pdf`.

[16] Hong Su, Harumi Kuno, and Elke A. Rundensteiner. Automating the transformation of xml documents. *Proceedings of the 3rd international workshop on Web information and data management*, 2001.

[17] David Trastour, Matthias Ferdinand, and Christian Zirpins. Lifting xml schema to owl. *ICWE*, 2004.

[18] Augustin Yu and Robert Steele. An overview of research on reverse engineering xml schema into uml diagrams. *Proceedings of the Third International Conference on Information Technology and Applications*, 2005.

[19] Lin Zhang and Jin-Guang Gu. Ontology based semantic mapping architecture. *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 4, Aug 2005.