ONTOLOGY-BASED INTEGRATION OF XML DATA

Schematic Marks as a Bridge Between Syntax and Semantic Level

Christophe Cruz, Christophe Nicolle
Laboratoire LE2I, UMR CNRS 5158,Université de Bourgogne, BP 47870, 21078 Dijon Cedex – France

Keywords: XML, XML schema, Ontology, Integration, Semantic.

Abstract: This paper presents an ontology integration approach of XML data. The approach is composed of two pillars the first of which is based on formal language and XML grammars analysis. The second pillar is based on ontology and domain ontology analysis. The keystone of this architecture which creates a bridge between the two pillars is based on the concept of schematic marks introduced in this paper. These schematic marks make it possible to establish the link between the syntactic level and the semantic level for our integration framework.

1 INTRODUCTION

Data integration consists in inserting a data set into another data set. In the context of the Web, document integration consists in creating hypermedia links between the documents using URL. Associated documents are multimedia documents which can be text, pictures, videos, sounds or any other file format. In the context of database integration, integrated data coming from several information systems makes information more complete and more relevant with a more global objective use. For example the biomedical data source is known to be hyper-linked because the description of objects suggests several hyperlinks, allowing the user to "sail" from one object to another in multiple data sources. Indeed there are on the Web more than one hundred genetic databases, two hundred twenty-six data sources of molecular biology, etc. (A. Baxevanis, 2000),(D. Benson et al., 2000). A second example of activity which is more and more based on the use of complex data source integration is the field of geographical information management. Just as biological and medical data, geographical data are heterogeneous and distributed. These data are for example weather (France: www.meteofrance.com) or cartographic (road maps: www.viamichelin.com). In fact data integration of various sources will thus make it possible to carry out more complex, precise requests and improve the various information systems available. The objective of data integration is to benefit from the diversity of information available and to benefit from the rise of new Web technologies. Moreover the re-use of data and services make it possible to optimize the costs for the acquisition and the maintenance of information. Finally, the management and the cost of the data-processing resources are distributed among the whole of the data suppliers.

2 BACKGROUND

In the context of the Web, XML technologies became a headlight technology for data structuring and data exchanges. Many systems using XML as databases integration have a mediation approach (A. Pan et al.,2002), (D. Draper et al., 2001), (M. J. Carey et al., 2000), (A. Cali et al., 2001). The evolution of the Web technologies changed the integration problem of information. In fact the XML contribution to define not only integration schemas but also the definition languages of the corresponding models reduced considerably the problems related to the structural and the syntactic heterogeneity. The contribution of the Web technologies related to the service oriented architectures solved partially the problems of the localization and the data access allowing the design of interoperability architectures on a greater scale (K. Aberer et al., 2001). Nevertheless, during the integration data process and the integration services there remain many problems related to semantic heterogeneity.

In order to support the action of agents, knowledge has to represent the real world by reflecting entities and relations between them.

Cruz C. and Nicolle C. (2006).
ONTOLOGY-BASED INTEGRATION OF XML DATA - Schematic Marks as a Bridge Between Syntax and Semantic Level.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 30-37
DOI: 10.5220/0001245400300037

Therefore knowledge constitutes a model of the world and agents use their knowledge as a model of the world. On the one hand the integration of different entities is possible when the semantic of the entities and the relations is identical. In addition, to model the semantic of knowledge as well as the structure where this knowledge is stored, it is necessary to reach a higher conceptual level. For that knowledge representation is independent of knowledge use. Thus knowledge representation and inferential mechanisms are dissociated (N. Guarino et al., 1994). On the other hand, domain conceptualization can be performed without ambiguity only if a context of use can be given. In fact a word or a term can designate two different concepts depending on the particular context of use (B. Bachimont et al., 2000). The semantic of knowledge is strongly constrained by the symbolic representation of computers. Therefore N. Guarino (N. Guarino, 1994) introduced an ontological level between the conceptual level and epistemological level. The ontological level forms a bridge between interpretative semantics in which users interpret terms and operational semantics in which computers handle symbols (T. Dechilly and B. Bachimont, 2000). In fact the problem consists in defining an upper conceptual level to handle the semantic in XML documents for their integration. This level will define a semantic framework leading to the integration of XML data by the use of an ontology.

The implementation of an ontology is a mapping stage between the system elements and their ontological "counterparts". Once this mapping has been carried out, the representation of elements in the ontology is regarded as a meta-data diagram. The role of a meta-data diagram is double (B. Amann and D. Partage, 2003). On the one hand it represents an ontology of the knowledge shared on a domain. On the other hand it plays the role of a database schema which is used for the formulation of requests structured on meta-data or to constitute views. This principle is applied to ontology based data integration using domain ontology to provide integration structures and request processes to these structures. According to (I. F. Cruz et al., 2004), (M. Klein, 2002), (L. V. Lakshmannan and F. Sadri, 2003) data integration consists in defining rules of mapping between information sources and the ontological level. The principle consists in labeling source elements and thus providing semantic definition to elements compared to a consensual definition of the meaning. This phase is inevitably necessary because this information was not added to the document during its creation. Moreover an XML schema defines only the structure of associated XML documents. However, an XML schema contains tacit knowledge that can be used to define ontology by extracting a set of elements and properties whose meaning will be defined for a more global use.

Section 3 describes a general view of our method based on two pillars (ontology and formal language). The keystone of our method is the concept of schematic marks. This section describes this concept by a formal way. Section 4 defines a set of integration rules based on the schematic marks.

3 METHOD OVERVIEW

Our integration solution consists in connecting various levels of semantic and schematic abstraction. This solution is articulated in two stages. The first stage relates to the semantic formalization of the writing rules to define an XML grammar. This formalization will enable us to define the components of a generic ontology. The second stage relates to the definition of the ontologization mechanisms of the semantic elements from a specific XML grammar to obtain an ontology of domain. The concepts and the relations of the domain ontology are then defined starting from the elements of the XML schema. These mechanisms make it possible to identify some concepts and relations common to several XML schemas. Consequently ontology makes it possible to link the concepts and the relations by amalgamating the attributes of the common elements which are semantically identical. The domain ontology will be extended and then modified to represent the semantic of several XML schemas relating to a particular domain. To specify the "semantic elements" of an XML schema it is first necessary to identify and mark them. We call these schematic marks. They will be used to establish links between the structure of an XML document and its semantic definition. Those schematic marks represent the structural level of the integration system.

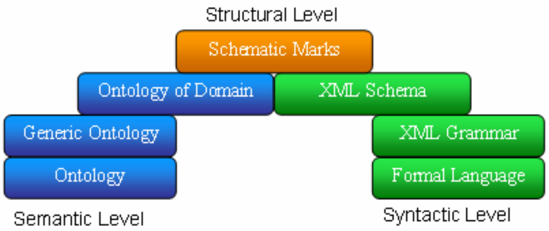


Figure 1: The two pillars and the keystone of our method.

To specify the semantic of the XML schema elements it is necessary to identify and mark them using schematic marks. These marks will be used to establish links between the structure of the XML document and its semantic definition. This section presents first of all the formalization of XML documents using formal languages. In addition this section outlines a formalization of XML grammars using the formal languages on which the principle of schematic marks is based. This underlines the fact that XML grammars generate languages of Dyck. According to definite properties of Dyck languages the concepts of factor and schematic mark are defined.

3.1 XML Document Formalization

An XML document is composed of text and opening tags associated to closing tags. Some of these tags are at the same time opening and closing tags. In fact, empty tags define the sheets of the XML trees. One of the properties of an XML canonical document is to be composed of only opening or closing tags. Empty tags can be exchanged by opening and closing tags without any problems for the XML parser. Consequently, any XML file having empty tags has an equivalent without empty tags. This property is syntactic because it does not appear in the grammatical rules formalizing the structure of the document. Starting from this information some definitions are expressed.

Definition 1: An *alphabet* is a finite set of symbols recorded Σ. These elements are called *letters*. In this paper most of the time it will be written: $\Sigma = \{a, b, \ldots\}$. The *size* $|\Sigma|$ of an alphabet Σ equals the number of its elements.

Definition 2: A *word* or a *sentence* on Σ is a sequence of letters coming from this alphabet. The word « wall » is a sequence of letters from the alphabet $\Sigma = \{a, b, \ldots, z\}$. It is said by convention that the word *empty* is the null size word. It is written: ε. The set of all words that are possible to be written on the alphabet Σ is written: Σ^*. $\Sigma^+ = \Sigma^* - \{\varepsilon\}$.

Definition 3: A *formal language* L is a set of words on Σ^*. A *language* L on the alphabet Σ is called regular (A regular language is a language of kind 3 in the Chomsky hierarchy) only if it is generated on the alphabet Σ and if it is defined by a regular expression. It means that the set of regular languages on Σ is exactly the set of languages recognizable by finite state automaton on Σ. In others words for each finite state automaton it is possible to associate regular expressions that define identical languages recognized by the same automaton and reciprocally.

We have just seen that a language L is made of words generated starting from an alphabet. If in an XML document we just consider the tree structure without taking into account the values of the tags' attributes, then the set of the XML documents which it is generable from a XML schema define a language. Moreover the set of the tags of XML documents defined by XML schema represents a part of the alphabet on the language. It defines only one part because the alphabet can contain letters not used by the language.

Definition 4: A formal grammar can be defined as a mathematical entity on which we can associate an algorithmic process to generate a language. It is defined as a quadruplet $G = \langle N, T, P, S \rangle$ in which:

- T written also Σ is the terminal alphabet on G.
- N the non terminal alphabet on G.
- $V = N \cup T$ is an alphabet composing the whole symbols of G.
- P is a set of production rules or regular expressions.
- $S \in N$ is the start symbol on G.

$N = \{S, A\} \qquad T = \{a, b\}$

$P = \{(S \to AA), (S \to \varepsilon), (A \to aa), (A \to bb)\}$

$S \to AA \mid \varepsilon$
$A \to aa \mid bb$

$G_1 = \langle N, \Sigma, P, S \rangle$

Example 1: G_1 grammar.

Example 1 shows a grammar for which the generated words are «aaaa» «aabb» «bbaa» «bbbb» and «». The language is composed of four words and the empty word.

3.2 Formal Grammar and XML Grammar

We saw in the preceding section the concept of regular language and that of formal grammar. This section presents formal grammars and XML grammars by making connections between them. These grammars have the characteristic to have a final vocabulary composed of opening tags and closing tags.

Definition 1: For a given set A composed of opening tags and corresponding closing tags, an XML document is a word composed from the alphabet $T = A \cup \bar{A}$.

For the moment we just take into account the syntactic structure. An XML document x is well formed if only one tag is root and if the tags are correctly imbricated.

Definition 2: A document x is well formed if x is generated by production rules from a language of Dyck on $T = A \cup \bar{A}$. A language of Dyck is a language generated by a context-free grammar where $a_n \in A$ and $b_n \in \bar{A}$:

$$S \to SS \mid \varepsilon \mid a_1Sb_1 \mid a_2Sb_2 \mid \ldots \mid a_nSb_n \ \ with\ n \geq 1$$

This definition corresponds to the terminology and the notation used in the XML community. A language is indeed a set of XML documents which can be generated starting from a grammar. These grammars of XML languages are called "Document Type Definition" (DTD). The axiom of grammars is qualified DOCTYPE and the whole of the production rules is associated to a tag ELEMENT (Example 2). A tag ELEMENT is made up of a type and a model. The type is simply the name of the tag and the model is the regular expression for the element.

$$P = \{(S \to a(S \mid T)(S \mid T)\bar{a}), (T \to bT\bar{b}), (T \to b\bar{b})\}$$

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT a ((a | b), (a | b))>
<!ELEMENT b (b*)>
```

Example 2: Similarity between a grammar and a DTD.

All of the production rules corresponding to the grammar can also be represented using a XML schema which can be translated out of a DTD (Example 3). DTD only defines the relations between the various components of a document contrary to the XML schema that defines also data types.

```
    <?xml version="1.0" encoding="UTF-8"?>
            <!ELEMENT ROOT (A | (B, C))>
  <?xml version="1.0" encoding="UTF-8"?>
  <xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
     <xsd:element name="A"/>
     <xsd:element name="B"/>
     <xsd:element name="C"/>
```

```
        <xsd:element name="ROOT">
            <xsd:complexType>
                  <xsd:choice>
                      <xsd:element ref="A"/>
                      <xsd:sequence>
                          <xsd:element ref="B"/>
                          <xsd:element ref="C"/>
                      </xsd:sequence>
                  </xsd:choice>
            </xsd:complexType>
        </xsd:element>
     </xsd:schema>
```

Example 3: Similarity between DTD and W3C XML schema.

In this section we saw reminders on the formal languages by making parallels between a formal grammar and an XML schema. We know that an XML schema is a formal grammar that generates a language of Dyck and that it has consequently the properties of a language of Dyck. The following section describes the properties of grammars that generate languages of Dyck and introduces the definition of schematic marks.

3.3 Factor and Schematic Marks

The properties of the languages of Dyck were the subject of studies undertaken by J Berstel (J. Berstel and L. Boasson, 2000). By drawing parallels between XML grammar and the languages of Dyck, J Berstel defines the concept of factor. According to the lemma 3.3 of J.Berstel if G is an XML grammar on $T = A \cup \bar{A}$ generating a language L with a non terminal vocabulary X_a and $a \in A$ then for each $a \in A$ the language generated by X_a is a set of factors of words in L which are languages of Dyck starting by the letter a : $L_G(X_a) = F_a(L)$

A given language:
$$L = \left\{ ca(b\bar{b})^{n_1}\bar{a}a(b\bar{b})^{n_2}\bar{a}\ldots a(b\bar{b})^{n_k}\bar{a}\bar{c} \mid \mathrm{n}_{1,2}\ldots_{k} > 0 \right\}$$
then
$$F_c(L) = L,\ F_b(L) = \left\{b\bar{b}\right\}^*,\ F_a(L) = \left\{a(b\bar{b})^*\bar{a}\right\}$$

Example 4: Factors of the language L.

This means that a language is factorizable in an under language and a factor of a language of Dyck is a language of a language of Dyck. Consequently an under tree of an XML document can be generated by a factor of the language of Dyck to which the XML document belongs.

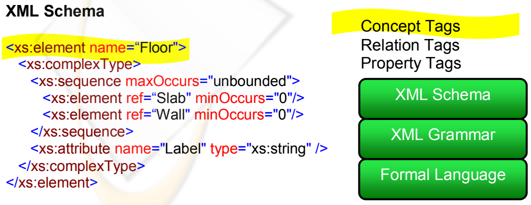
According to the corollary 3.4 of J. Berstel there is only one factor for an XML grammar $F_a(L)=L$. This means that there is only one 'father' tag for all others. This tag is the root. Consequently there is only one factor $F_a(L)=L$ where a is the root.

According to the same corollary 3.4: For a given word w of the language L there is a unique factorization $w=au_{a_1}u_{a_2}\dots u_{a_n}\bar{a}$ with $u_{a_i}\in D_{a_i}$ for $i\in 1,\dots n$. D_{a_i} is a language of Dyck starting by a_i and $D_{a_i}\subset D_A$. The *trace* of a word w is defined by a word $a_1a_2\dots a_n$. The *surface* $a\in A$ in the language L is a set $S_a(L)$ of all traces of the words of the factor $F_a(L)$. The notion of surface is used by Berstel to demonstrate the following proposition:

Proposition 1: For each XML language L there is only one reduced XML grammar generating L.

This means that all the languages of an XML grammar are generated by the same reduced XML grammar. This independently of the values of tag's attributes in the documents. And if we only take into account the syntactic structure of XML documents. 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 grammar XML generating this language. This proposal makes it possible to introduce the concept of schematic mark. A reduced grammar does not have any useless non terminal vocabulary which is, in general, not the case for grammars generating languages of Dick. But an XML schema does not use unnecessary tags, so an XML schema does not use unnecessary non terminal vocabulary.

Definition 3: A schematic mark is a mark on an XML schema to identify a production rule.



```
<xs:element name="Floor">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="Slab" minOccurs="0"/>
      <xs:element ref="Wall" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Label" type="xs:string" />
  </xs:complexType>
</xs:element>
```

Figure 2: Definition of a schematic mark for the floor.

Those schematic marks allow to make links between an XML schema element and a factor in a corresponding XML language and, as a result, all XML documents that can be generated by the XML schema. In the following example the elements which are marked define a concept. Consequently, the schematic marks allow to provide links between the concept and these instances. Thus, the schematic marks can also mark a relational element and an attribute element. Those concept elements, relational elements and attribute elements are used to define the domain ontology of the XML schema. The schematic marks are used to link the component of the ontology and the element of the XML schema. At the ontology level the schematic marks are said to be the semantic definition of the XML schema elements that allow to identify the elements that have a tacit semantic and are not defined formally.

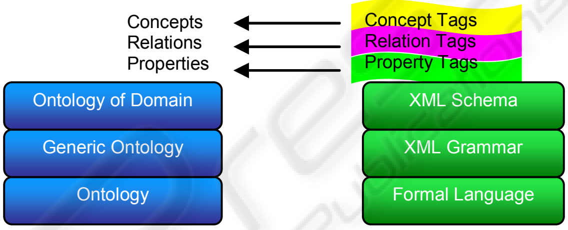


Figure 3: Definition of schematic marks.

The following section shows that several schematics marks of various XML schemas can have the same semantics defined by a common ontology. In this case, the element properties corresponding to the schematic marks are integrated within the same concept defined in ontology. It is called semantic integration of XML schema.

4 INTEGRATION RULES

This section is outlined in three parts. The first part presents the formalization of the XML grammar construction rules. This formalization enables us to release a set of terms which will be used to build our generic ontology. This generic ontology allows to define the elements of the domain of each XML schema to be integrated. The second part presents the definition of generic ontology. This generic ontology is used as model to create domain ontologies. This formal structure makes it possible to index the tacit knowledge contained in the XML schema. This knowledge is then explicit because it raises any ambiguity on the interpretation of the XML document terms generated starting from an integrated XML schema.

4.1 Schematic Formalization

This section presents a set of definitions and rules to formalize the construction of XML grammars and to define the concepts, relations and attributes of our generic ontology.

Definition 1: The Factor $F_a(L)$ of a language L is a *conceptual factor* if it defines a concept. For example, the factor $F_{Building}(L)$ is a concept because it defines the concept *Building*.

Rule 1: If the conceptual factor $F_a(L_1)$ of a language L_1 and the conceptual factor $F_b(L_2)$ of a language L_2 handle the same semantic of a common concept then the intension of the concept is define with the help of the conceptual factors $F_a(L_1)$ of L_1 and $F_b(L_2)$ of L_2.

Definition 2: The Factor $F_a(L)$ of a language L is a *relational factor* if it defines a relation. For example, if a wall contains a door then the relational factor $F_{Contain}(L)$ is a relation because it defines a relation between a wall and a door.

Rule 2: If the factor $F_a(L_1)$ of a language L_1 and the factor $F_b(L_2)$ of a language L_2 handle a common semantic of a relation then the intension of the relation is define with the help of the relational factors $F_a(L_1)$ of $F_b(L_2)$ of L_2.

Definition 3: The Factor $F_a(L)$ of a language *L* is an *attribute factor* if it defines one or several proprieties of a concept or a relation. For example, if a wall has a geometrical shape then the attribute factor $F_{Geometry}(L)$ is an attribute because it defines the geometry of a wall.

Rule 3: If the factor $F_a(L_1)$ of a language L_1 is an *attribute factor* then it is integrated in the intension of the concept or the relation. For example if a wall has a thermical propriety the geometry attribute factor $F_a(L_1)$ is integrated in the intension of the concept *wall*. The attribute factor can be integrated in several intensions of concept and relation.

The rules and definitions 1, 2 and 3 make it possible to define the conceptual factors, the relational factors, and the factor attributes of a language. These factors correspond to schematic marks carried out the elements of an XML grammar. A schematic mark in an XML grammar corresponds to a factor in the language. By defining these factors as conceptual, relational or as attributes, we give them a semantic. This semantic is already carried by the elements but this marking allows to make it explicit. Moreover, the definition of their attributes constitutes the intension of the concept or the relation and thus this improves their definition. By amalgamating the attributes of the diagrammatic marks of various XML schemas through the same concept or the same relation we carry out an integration of concepts or relations. This integration constitutes the integration of an XML schema. The following rules and definitions define the particular cases.

Rule 4: If $F_a(L)$ is a relational factor of a language *L* then $F_a(L)$ links a conceptual factor ‘father’ to a set of conceptual factor ‘sons’.

Rule 5: If the trace of a conceptual factor $F_a(L)$ is composed of a conceptual factor $F_b(L)$ then the link between the two conceptual factors $F_a(L)$ and $F_b(L)$ is a relational factor $F_{rab}(L)$. For example if the conceptual factor *wall* has a conceptual factor ‘son’ *door* then there is a relational factor between the conceptual factor *wall* and the conceptual factor *door*.

Rule 6: If the trace of a conceptual factor $F_a(L)$ is composed of a conceptual factor set $F_\alpha(L)$ with $\alpha \in \mathrm{A}$ (alphabet $T = A \cup \overline{A}$) and if the semantic of the relation is the same then the link between $F_a(L)$ and the set $F_\alpha(L)$ is a relational factor $F_{r\alpha}(L)$, having for conceptual factor ‘father’ $F_a(L)$ and for conceptual factor ‘son’ set $F_\alpha(L)$. For example the conceptual factor *floor* has a conceptual factor *wall*, a conceptual factor *column*, a conceptual factor *beam* and a conceptual factor *slab*. If the signification of the link between a floor and the elements *wall*, *column*, *beam* and *slab* is the same then the relational factor between the element *wall* and the other elements are of the same kind.

Definition 4: The intension of a concept is composed of a set of schematic marks on XML grammars. Those schematics marks are connected to several conceptual factors of the language generated by the grammars.

Definition 5: The extension of a concept is composed of set of instances. In the present case those instances are called semantic elements having a trace which is a set of XML trees.

Definition 6: The intension of a relation is composed of a set of schematic marks on the XML grammars. Those schematic marks are connected to a relational factor of the language generated by the grammars.

Definition 7: The extension of a relation is composed of a set of instances. In the present case those instances called relational elements have a trace which is a set of XML trees.

Definition 8: A factor defines a concept or relation or an attribute of an element from an XML schema. Consequently, a mark is a conceptual factor or a relational factor or an attribute factor.

Rule 7: If two instances of a concept represent the same object or the same relation then they can be identified as equal. For example, an object *wall* has a set of thermical properties and another object *wall* has a geometrical shape definition. Those two walls are the same object if they have an identifier that allows to identify them as the same object.

These definitions and rules give a set of vocabulary that composes the taxonomy of our system. This vocabulary is composed of the following words: *concept, relation, attribute, conceptual factor, relational factor, attribute factor, semantic element,* and *relational element*. Each word defines a concept of our generic ontology and is linked to a class.

- The class *concept* is defined by properties represented by its intension which is composed of a list of conceptual factors.
- The class *relation* is defined by properties represented by its intension which is composed of a list of relational factors.
- The *semantic* and *relational elements* are classes allowing the instantiation of objects coming from XML documents.
- The whole object of the class *semantic element* linked to an instance of the class concepts represents the extension of the instance of *concept*.
- The whole object of the class *relational element* linked to an instance of the class *relation* represents the extension of the instance of *relation*.
- The instances of the class *conceptual factors* and the class *relational factors* are references to the schematic marks on XML grammars. Those marks are XML documents extracted from XML grammars. The traces are XML documents extracted from XML documents to integrate.

We saw until now the definitions and the rules for the integration of XML schemas using a generic ontology. These rules make it possible to share the properties of various XML schemas. This level of integration is called schema integration level because it gathers in the heart of the same concept or relation various properties defining the intension of a concept or a relation. On this level of integration follows a second level of integration. This level is called data integration level. The first level defines the concepts, the relations, and the attributes which will be instanced on the second level of integration using the schematic marks.

4.2 Generic Ontology

Previous sections have presented a partial formalization of XML grammars which makes it possible to build a generic ontology. This generic ontology allows to define a set of domain ontologies corresponding to various integrated XML schemas.

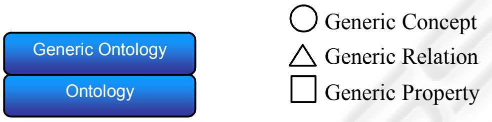


Figure 4: Def. of the generic concepts of our generic ontology.

Once implemented this generic ontology makes it possible to define the concepts, the relations, and the properties of several domain ontologies.

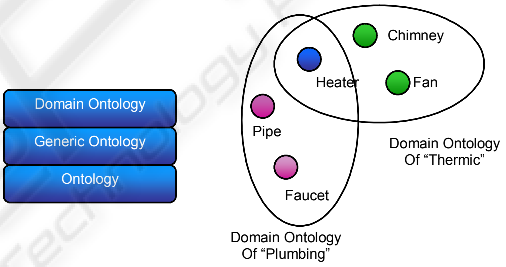


Figure 5: Definition of two domain ontologies and integration of the concept "heater".

In the example figure 5 the concept *Heater* is common to both domain ontologies coming from two different XML schemas. The pooling of this concept in two domain ontologies makes it possible to integrate two XML schema.

Among the languages of representation developed at the conceptual level there are three principal types of models: languages containing frame (M. Kifer et al., 1995), logics of description (D. Kayser, 1997), and the model of the conceptual graphs (J. Sowa, 1984). To define our ontology we chose the language containing frame because the development related to the Web services which will be used in our architecture is carried out in the programming language directed for JAVA objects.

5 CONCLUSION

After a syntactic approach of XML data this paper presented an ontology based on the semantic approach of data XML. The study of the semantic structure of XML grammars and ontologies has inspired the construction of classes representing an integration ontology of data generated by XML grammars. This cognitive structure has many advantages of which the most important one is to be evolutionary as well on the level of the data as of the definition of the data structures. The field of ontologies made it possible to achieve the goal which is an integration structure of XML data. Nevertheless, the treatment of XML grammars for the semantic extraction of knowledge realized by the user in a manual way thanks to the knowledge of XML schema is not without defects. This extraction causes problems in the decomposition of the XML schemas. How to choose the level of granularity by limiting the problem of the redundancy of information? This case is rather current in complex XML schema where tags refer to tags having an identifier. In this manner a tag can have two 'fathers' so that these tag 'fathers' compete with each other (A. Dekhtyar, 2003). In this case if these links are not factorized the information of the link is duplicated for each semantic element which are referenced. These cases are easily detectable and it is then as easy to help the user to remove this redundancy of information.

Our method was tested in a static way on a set of XML grammar schema as well as a set of documents XML associated with each XML schema. The future objectives of development are double. On the one hand we wish to develop a complete system allowing the integration of XML schema and XML data in a dynamic manner through a graphic interface. This tool will include also tools allowing requests to the system according to our previous work. In addition, we wish to test this method for the integration of Web Services. The Web services are defined using an XML schema defining the contents of SAOP documents. Consequently, our architecture would make it possible to carry out semantic requests on a set of Web Services.

REFERENCES

A. Baxevanis, *The Molecular Biology Database Collection,* Nucleic Acids Research, vol 28, n°1, 2000, p.1-7 http://nar.oupjournals.org/cgi/content/full/27/1/1

D. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, B. Rapp, D. Wheeler, GenBank, *Nucleic Acids Res*, vol. 1, n°28, 2000, p. 15-8, http://www.ncbi.nih.gov/Genbank/

A. Pan, J. Raposo, M. Álvarez, P. Montoto, V. Orjales, J. Hidalgo, L. Ardao, A. Molano, Á. Viña, The Denodo Data Integration Platform, *VLDB*, Hong Kong, China, 2002

D. Draper, A. Y. HaLevy, D. S. Weld, The Nimble XML Data Integration System, *IEEE International Conference on Data Engineering*, April 02 - 06, 2001, Heidelberg, Germany

M. J. Carey, J. Kiernan, J. Shanmugasundaram, E. J. Shekita, S. N. Subramanian, XPERANTO : Middleware for Publishing Object-Relational Data as XML Documents, *The VLDB Journal*, pp 646-648, 2000

A. Cali, G. De Giacomo, M. Lenzerini, Models for Information Integration: Turning Local-as-View into Global-as-View, *Proceedings of the International Workshop on Foundations of Models for Information Integration,* 2001.

K. Aberer, P. Cudre-Mauroux, M. Hauswirth. A framework for semantic gossiping. *SIGMOD Record*, 31(4), 2002

N. Guarino, C. Carrara, P. Giaretta, An ontologie of meta-level categories, in J. Doyle F. S & Torano P., eds., Principles of Knowledge representation and Reasonning, *Morgan-Kauffman*, pages 270-280, 1994.

B. Bachimont, Engagement sémantique et engagement ontologique : conception et réalisation d'ontologie en ingénierie des connaissances, In *Charlet J., Zackland M., Kessel G. & Bourigault D., eds., Ingénierie des connaissances : évolution récentes et nouveaux défis,* Eyrolles, pages 305-323, 2000.

N. Guarino, The ontological level, in R. Casati B. S. & White G., eds, *Philosophy and the cognitive sciences*, Hölder-Pichler-Tempsky, 1994.

T. Dechilly, B. Bachimont, Une ontologie pour éditer des schémas de description audiovisuels, extension pour l'inférence sur les descriptions, In *Actes des journées francophones d'Ingénierie des Connaissances* (IC'2000), 2000.

B. Amann, Du Partage centralisé de ressources Web centralisées à l'échange de documents intensionnels, *Documents de Synthèse*, 2003.

I. F. Cruz, H. Xiao & F. Hsu, An Ontology-based Framework for Semantic Interoperability between XML Sources, In *Eighth International Database Engineering & Applications Symposium* (IDEAS 2004), July 2004.

M. Klein, Interpreting XML via an RDF schema. In *ECAI workshop on Semantic Authoring, Annotation & Knowledge Markup* (SAAKM 2002), Lyon, France.

L. V. Lakshmannan, F. Sadri, Interoperability on XML Data, In *Proceeding of the 2nd International Semantic Web Conference* (ICSW'03), 2003.

M. Kifer, G. Laussen, J. Wu, Logical foundations of object-oriented Land frame-based languages, In *journal of the ACM*, 1995.

D. Kayser, La représentation des connaissances, *Hermès*, 1997.

J. Sowa, Conceptual structures: information processing in mind and machine, *Addison-Wesley*, 1984.

A. Dekhtyar, I. E. Iacob, A Framework for Management of Concurrent XML Markup, International Conference on Conceptual Modeling, *in ER 2003*, pages 311-322, 2003.

J. Berstel, L. Boasson, *XML Grammars* MFCS 2000: 182-191