XML-IS: ONTOLOGY-BASED INTEGRATION ARCHITECTURE

Christophe Cruz and Christophe Nicolle
Laboratoire Le2i (UMR CNRS 5158),Université de Bourgogne, Faculté des Sciences Mirande
Aile de l'Ingénieur, BP 47870, 21078 Dijon Cedex, France

Keywords: XML, XML schema, Knowledge, Ontology, Information retrieval, Integration, Semantic.

Abstract: This paper presents an architecture that aims at integrating XML documents. Today, information exchanges in business processes widely use XML to format data. But this is done without a common process management. For some applications it is necessary to archive data for future processes (error detections, logs, and knowledge enquiry data mining). For this reason our architecture allows users to retrieve information from automatically integrated XML. This is not possible without the help of a semantic level definition which is build upon XML schemas. To reach this goal we use the knowledge defined in XML schemas to share a global ontology of domain. At a final step this ontology is used in the retrieval process by end-users to search and retrieve information.

1 INTRODUCTION

One of the most impressive capacities of new information and communication technologies is to generate data. Every day thousands of new Web sites and databases are available. However, it is done without taking into account the future life of those data. As a consequence, the reusability of data is the weakness of these new information sites. This is the reason why search engines are very popular. Moreover, the search on the Web consists in using the appropriate words that are associated with the context. But you cannot surf on a knowledge level that drives you directly to information looked for. In fact, this knowledge should permit a better search and a better reusability.

During the last seven years, XML has known an incredible and extensive use for systems of data exchange and data sharing. XML is used to define most of the Web information (videos, images, 3D scenes, domain information, office data, etc.) Many systems using XML as database integration have a mediation approach (Pan, 2002), (Draper, 2001), (Carey, 2000), (Cali, 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. Moreover, 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 (Aberer, 2002). Nevertheless, during the integration data process and the integration services there remain many problems related to semantic heterogeneity. So, a formal description of the semantic shared should avoid ambiguities when it is defined in relation to a domain of knowledge. Hence, the reuse of information in a specified domain should be improved.

2 RELATED WORK

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 (Amann, 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 the XML-IS architecture to provide integration structures and request processes to these

Cruz C. and Nicolle C. (2008).
XML-IS: ONTOLOGY-BASED INTEGRATION ARCHITECTURE.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 273-277
DOI: 10.5220/0001525102730277

structures. According to (Cruz, 2004), (Klein, 2002), (Lakshmannan, 2003) data integration consists in defining rules of mapping between information sources and the ontological level. These rules consist in adding semantic annotation to source elements and thus provide semantic definition to elements compared to a consensual definition of the meaning.
Compared with the approach of SAWSML (Martin, 2007) XML-IS is also a method to annotate `<xs:element>` tags contained in XML schema, but it is used to identify the tags in an XML document validated by this XML schema. In SAWSDL the annotation process consists in identifying the input and output formats of the Web Services which are specified by an XML schema. Here XML documents validated by annotated XML schemas are indexed by an automatic process. Our architecture should be able to annotate WSDL and SOAP index documents as it is done with XML schemas and XML documents. Actually, it was not designed for it, as SAWSDL was not designed to index SOAP documents.
The next section gives a general view of our architecture based on the field of ontologies and formal languages. This section introduces the notion of the semantic mark which is the basis of our architecture. Section 4 describes the implementation of XML-IS by explaining the data model, the handling system and the retrieval system.

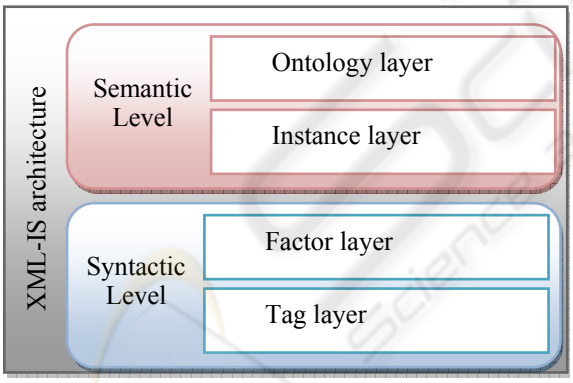

Figure 1: this figure presents the architecture of XML-IS composed of four layers. The ontology layer and the instance layer which define the semantic level. The factor layer and the tag layer define the syntactic level. Semantic marks are used to link the Semantic level to the Syntactic level via the ontology layer and the factor layer. The tag layer is used to index XML documents. The ontology layer is used to request XML-IS.

3 ARCHITECTURE

Our proposal consists in connecting various levels which are composed of the semantic and the schematic levels. To specify the semantic of the XML schema elements it is necessary to identify and mark them. These marks will be used to establish links between both levels. The syntactic level concerns the structure of the XML document and the semantic level concerns its semantic definition (cf. Fig. 1).
The properties of a language of Dyck were the study subject undertaken by J. Berstel (Berstel, 2000). By drawing parallels between XML grammar and languages of Dyck, J. Berstel defines the concept of factor. According to the lemma 3.3 of J. Berstel, a language is factorizable into an under language and a factor of a language of Dyck is a language of Dyck. Thus, a subtree of an XML document can be validated by a factor of a language of Dyck. This 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 "semantic mark". A semantic mark is a mark on an XML schema that makes it possible to identify a production rule (e.g. a factor). This production rule corresponds to tag `<xs:element/>` in a XML schema. A tag `<xs:element/>`, which is marked, is linked to the definition of a concept, or a relation, or an attribute. The semantic annotation by a semantic mark on a production rule permits to annotate automatically all tags in XML documents which are validated by the XML schema marked. Those concepts, relations and attributes are used to define the domain ontology of the XML schema. The feature needed for our ontology is available in the OWL specification. Consequently, we use Jena to store the OWL ontology defined on XML schema and the OWL instances on the corresponding XML documents. OWL specifications make it possible to define concepts (`owl:class` and `rdf:subClassOf`), relations (`owl:ObjectProperty`), and attributes (`owl:DataTypeProperty`) for several domain ontologies. For instance, the concept Heater is common to both domain ontologies coming from two different XML schemas. The pooling of this concept from several domain ontologies makes it possible to integrate several XML schemas. The instance layer composed of instances of the ontology layer is automatically linked to the tag layer which is composed of the tags from XML documents validated by annotated XML schemas.

The next section describes the implementation of the different layers and underlines the fact that several semantic marks from various XML schemas can have the same semantic defined by a common ontology. Hence, the element properties corresponding to the semantic marks are integrated within the same concept defined in ontology.

4 IMPLEMENTATION

This section describes the implementation of XML-IS architecture. The first part shows the data model of the systems. The second part describes the handling system. The third part presents the retrieval model based on the data model.
The semantic level is described by an ontology that represents the implicit knowledge on XML schemas which are defined by the users. The syntactic level is described by an ontology that represents the explicit knowledge on an XML schema defined by our architecture.

4.1 Data Model

Concerning the semantic level, the ontology layer is made of a super class `Concept`, a super ObjectProperty `Relation`, the super DataTypeProperties `Attribute` and `SimpleAttribute`. An `Attribute` is a tag that defines an attribute for a father tag. A `SimpleAttribute` is a tag attribute. For instance, `id` is an attribute of the tag `div`, `<div id="1342">`. The `SimpleAttribute` can be used to define the `Attribute` tag that makes it possible to integrate XML documents from different XML schemas. Every "semantic mark concept" generates automatically a new class which is a subclass of `Concept`. Every new XML document validated by an integrated XML schema generates automatically new instances of the new class for every tag referenced in the ontology. This process is identical to every semantic mark `relation`, `attribute` and `simpleAttribute`. Concerning the syntactic level, the factor layer represents the factors selected by the user to define the semantic marks. Those factors are defined by the following super DataTypeProperty `Factor_Concept`, `Factor_Relation`, and `Factor_Attribute`. We also defined a property `SchemaXML` to keep a link between Factors and Schemas. The tag layer is composed of the concept, the relation and the attribute instances which keep links between XML schemas, XML documents and the ontology.

4.2 Handling System

The schema integration process is composed of four steps, the parsing step, the schematics marks selection step, the semantic validation step and the covering validation step.

The Parsing Step. XML schemas are also XML documents. Thus, they are also trees from which we generate a special tree called Schema Tree (Fig. 2 (1)). The nodes of the Schema Tree are generated from the tags `<xsd:element>`. The attributes of the nodes are generated from the tags `<xsd:attribute>`.

The Semantic Marks Selection Step. During the marking of the first XML schema, the user defines the first ontology of domain. For this, the user defines marks if the elements are concepts, relations or attributes. During the marking of the other XML schemas, the user defines other ontologies of domain, but if a tag `Concept` or a tag `Relation` or a tag `Attribute` is already defined then the user selects the corresponding entity (Fig. 2 (3)).

The Semantic Validation Step. When the user has finished the semantic marks selection step then it is necessary to validate this marking. Indeed, the user is free to define the semantic marks but some rules must be observed for the semantic coherency. For instance, a `Relation` tag must have an `Concept` tag as father tag. An entity `Relation` must have a set not equal to zero of a tag `Concept`. An tag `Attribute` can be linked to any entity.

The Covering Validation Step. Once the semantic validation is done, the covering validation must be undertaken. Indeed, every element of the element tree is not necessarily selected by the user. For the integration it is necessary to link every entity. Consequently, the root node must at least be marked. Thus, the tree is represented by clusters. The nodes without marks are associated to the cluster of the ancestor which handles the semantic mark (Fig. 4).

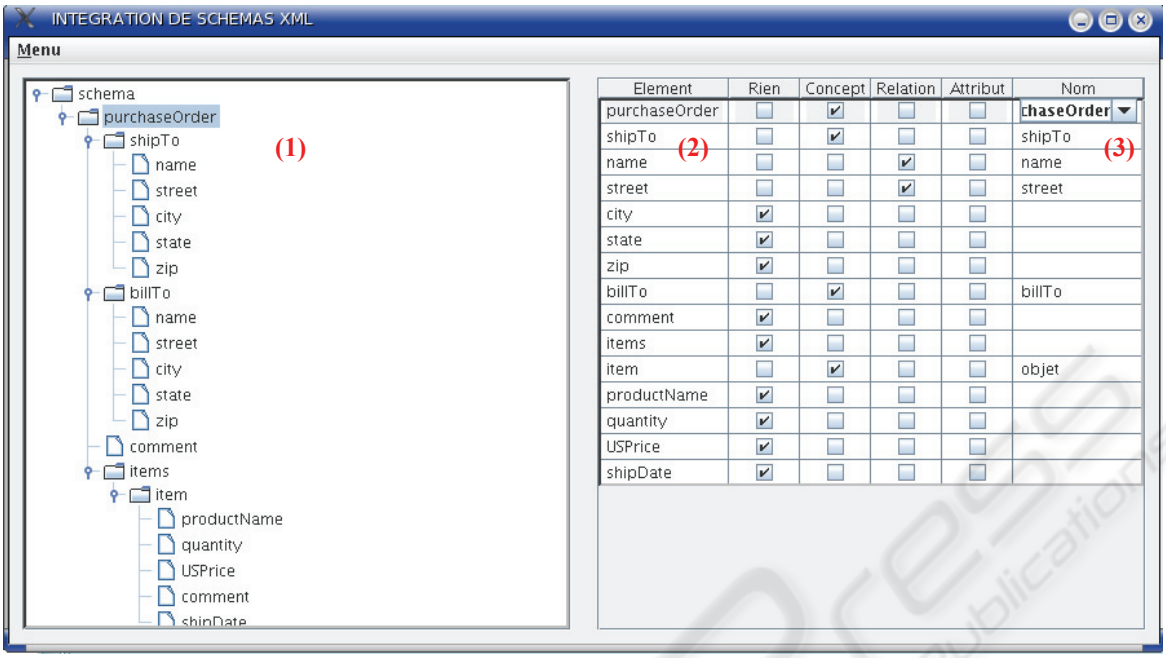

Figure 2: This figure represents the graphic interface for the semantic definition of entities. In (1) the user can see and select an entity. And in (2) he defines the kind of selected entity (rien~nothing). With the comboBox in (3) the user can write the name of the new concept, relation, attribute or select a previous name (nom~name).

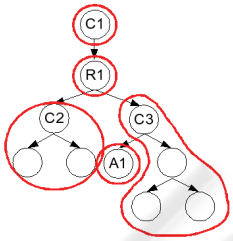

Figure 3: This figure shows the clustering of non-marked nodes. The tree represents clustered entities by red lines. The nodes without marks are associated to the cluster of the ancestor (C2 and C3) which handles the semantic mark.

4.3 Retrieval Model

In this section we present how to retrieve information from XML-IS. To define our retrieval system we use the notion of context. This notion is inspired by the following citation of Recanati (Recanati, 1993): "The meaning of a word like 'I' is a function that takes us from a context of utterance to the semantic value of the word in that context, which the semantic value (the reference of 'I') is what the word contributes to the proposition expressed by the utterance." For instance the word "Beetle" has a different semantic value in the context "Vehicle" than in the context "Biology". Indeed, in the context "Biology" the word "Beetle" refers to the semantic value "Insect" and in the context "Vehicule" it refers to the semantic value "Car". From this citation we have defined the following function:

$$Y = f(X) \quad (1)$$

f: is the word
X: is the context
Y: is the semantic value of the word f in the context X

By analogy, a word is an instance (an instance of Concept or an instance of Relation or an instance of Attribute). Y is the value of the attributes taken into account in the context. The context X is defined by the following set:

$R = \{ r \mid r \in$ the set of ***Relation*** $\}$
$A = \{ a \mid a \in$ the set of ***Attribute*** $\}$
$S = \{ s \mid s \in$ the set of ***AttributeSimple*** $\}$
$X = \{ r, a, s \mid r \in R'$ and $R' \subset R,$
$a \in A'$and $A' \subset A,$
$s \in S'$ and $S' \subset S \}$

The context can be seen as a semantic filter because only a subset of attribute values is kept in a Semantic Element. Actually, the definition of a context allows us to select a set of instances `Concepts` according to a set of `Relations`, `Attributes` and `AttributeSimples`. In addition, the

set of `Relations` is used to select a set of instances of `Concepts`. Indeed, an instance of `Concept` is selected if it is a son or a father of an instance of the `Relation` defined in the context. Thus, if the instance of `Concept` is selected then the semantic value of the word is a multiset (e.g. bag) of `Attribute` values and a second multiset of `AttributeSimple` values according to the context. Moreover, if the instance of `Concept` is not linked to an instance of `Relation`, which is not an instance of one of the `Relations` defined in the context, then the set of semantic values is empty.

These previous definitions allow us now to introduce the notion of "Contextual Tree". A Contextual Tree is the result of a request to our information system XML-IS. These requests are semantic filters that give an adapted view of information. A request Rq is defined in the following manner:

$$\Omega = \{x \mid x \in \text{the set of Context}\} \quad (2)$$
$$C = \{c \mid c \in \text{the set of } \textbf{\textit{instance\_Concept}}\}$$
$$Rq = \{ x, y \mid x \in \Omega' \text{ and } \Omega' \in \Omega \text{ and } |\Omega'| = 1, \; y \in C' \text{ and } C' \subset C\}$$

Concerning the definition of a context we add a rule for the validation of the request. If the contextual tree resulting from the request is not a hierarchical non-cyclic graph then the context use is not well defined. It means that the `Relation` selected for the definition of Context generates a Contextual Tree which is not a tree. To resolve the issue of a bad defined Context the user has to select fewer Relations. For instance:

R' = { "placement" }, A' = { "shape" }, S' = { "id"} (3)
C' = { c is the set of ***Instance_Concept*** which is defined by the class ***Concept*** "wall", "slab" and "pipe"}

$$\Omega_{Geo} = \{ r, a, s \mid r \in R' \text{ and } R' \subset R, \; a \in A' \text{ and } A' \subset A, \; s \in S' \text{ and } S' \subset S\}$$
$$Rq = \{ x, y \mid x \in \Omega_{Geo}, y \in C' \text{ and } C' \subset C\}$$

The result of this request Rq is the selection of all instances of Concept defined by the instances of Concept called "wall", "slab" and "pipe" which are linked by the instance of `Relation` defined by the entity `Relation` called "placement". The semantic value is the geometrical value of shape. The result is an XML document that describes the shapes of walls that belong to the building.

5 CONCLUSIONS

In this paper, we have presented our method to integrate XML data and to retrieve XML information through a defined context of use. The objectives were reached with the introduction of the semantic marks. The XML-IS system was tested on a set of XML grammar schemas as well as on a set of XML documents associated with each XML schema.

REFERENCES

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

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

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

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

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

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

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

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

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

Martin, D., Paolucci, M., Wagner, M., 2007, *Towards Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective*, In OWL-S Experiences and Future Developments Workshop at ESWC 2007, June 2007, Innsbruck, Austria.

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

Recanati. F., 1993, *Direct Reference: From Language to Thought*. Blackwell Publishers, Oxford, UK.