Ontology-based heterogeneous XML data integration

Christophe Cruz
ACTIVe3D-Lab, 2 rue René Char, BP 66606,
21066 Dijon Cedex, France
christophe.cruz@khali.u-bourgogne.fr
Christophe Nicolle
Le2i, UMR CNRS 5158, Université de Bourgogne,
BP 47870, 21078 Dijon, France
cnicolle@u-bourgogne.fr

Abstract: *In this paper we present an ontology-based method for formalizing the implicit semantic and we suggest mechanisms to semantically integrate XML schemas and documents as well. After a survey of database interoperability, we present our semantic integration approach by explaining the nature of ontology. The article then presents our integration method for XML data and schemas using a generic ontology.*

Categories and Subject Descriptors
D.3.2[Language Classifications]: H.2 [Database Management]; H.2.3 [Languages]

General Terms
Ontology, XML

Keywords: Information Systems, Ontology, Data Integration, XML

Received 30 Oct. 2004; Reviewed and accepted 30 Jan. 2005

1. Introduction

The kinds and the size of information systems are increasing ever more rapidly these days. Currently, the main issue is not how to create and use these information systems, but how to integrate and allow them to interoperate. For more efficient data use, the integration of information systems has to allow users to formulate more relevant requests and needs to provide more complete answers. As an example, biomedical information is particular in that it is hyperlinked, because biomedical object descriptions offer a set of links to multiple information sources. Consequently, users can navigate through different information systems to find the needed data. We can count more than a hundred genetic databases, two hundred twenty six data sources on molecular biology, etc. (Baxevanis, 2000 and Benson 2000 et al). Since the beginnings of information system history, various suggestions for systems and data integration have been proposed. In this area, XML[1] is quickly becoming a universal support for the representation of shared information (text, sound, image, etc.) on the Internet. For example, the manipulation of pictures with XML is performed at various levels. At the most basic level, it is possible to insert a picture into an XML document by using a tag containing the address of the picture file. At higher levels, there are specific XML languages for modeling and handling 2D (SVG[2]) and 3D (X3D[3]) pictures, video (MPEG-7[4]), architectural conception (BlisXML[5]), health-related images (HL7[6]), etc.

In terms of data integration, the use of XML resolves the problems of structure and syntactic heterogeneity. The use of specific translators written in XSL[7] to convert XML documents from a source format to a target format resolves the problems of schematic heterogeneity. Nevertheless, semantic heterogeneity problems are not resolved by XML definition. To do so requires the definition of the tacit semantic carried out in the XML documents. In this paper we present an ontology-based method to formalize this implicit semantic and we suggest mechanisms to semantically integrate XML schemas and documents.

The following is divided into five parts. First, we survey the issue of database interoperability. After this first section, we introduce our semantic integration approach by explaining the nature of ontology. Third, we present our method of integration for XML data and schemas using a generic ontology. Finally, a conclusion describes our progress in this method of integration.

2. Background

Database interoperability issues have been extensively studied in the past. Several approaches, including database translation, distributed systems, federations, multi-base language, mediation and ontology, have been proposed to bridge the semantic gaps among heterogeneous information systems.

The database **translation** approach is a point-to-point solution based on direct data mappings between pairs of information systems. The mappings are used to resolve data discrepancies among the systems (Yan L.L., Ling T.W., 1992). The database translation approach is most appropriate for small scale information processing environments involving a reduced number of participants. The number of translators grows with the square of the number of components in the integrated system. For example, consider two information systems IS1 and IS2, the corresponding translators must be placed between the information systems. Information is exchanged by converting their format using translators.

standardization approach, the information sources use the same model or standard for data representation and communication. The standard model can be a comprehensive metamodel capable of integrating the requirements of the models of the different components (Atzeni P., Torlone R., 1997). The use of a standard metamodel reduces the number of translators needed to resolve semantic differences (this number grows linearly with the number of components). However, the construction of a comprehensive metamodel is difficult; the manipulation of high level languages is complex; and there are no unified database interfaces. In our example, travel agencies must define a common model to export their data. A centralized information system can be built to replace the original information systems (IS1, IS2). The global centralized schema is a combination of all the data contained in IS1 and IS2.

Federated systems consist of a set of heterogeneous databases in which federation users can access and manipulate data transparently without knowledge of the data location (Sheth A.P., Larson J.A., 1990). Each federation database includes a federated schema that incorporates the data exported by one or more remote information system. There are two types of federations. A tightly coupled federation is based on a global federated schema that combines all participant schemas. The federated schema is constructed and maintained by the federation administrator. A loosely coupled federation includes one or more federated schema that are created by users or the local database administrator. The federated schema incorporates a subset of the schema available in the federation. This approach rapidly becomes complex as the number of translators required grows. In our example, the existing information systems are completely operational for local users. Only shared data are integrated into the federated schema. The federated system is only made up of data that IS1 and IS2 want to exchange.

Language based **multi-base systems** consist of a loosely connected collection of databases in which a common query

[1] XML : http://www.w3.org/XML/
[2] SVG : http://www.w3.org/Graphics/SVG/
[3] X3D : http://www.web3d.org/x3d/
[4] MPEG-7 : http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
[5] BLIS XML : http://www.blis-project.org/BLIS_XML/
[6] HL7 : http://www.hl7.org/
[7] XSL : http://www.w3.org/TR/xsl/

language is used to access the contents of the local and remote databases (Keim D.A., Kriegel H.P., Miethsam A., 1994). In contrast to the distributed and federated systems, the burden of creating the federated schema in this approach is placed on the users, who must discover and understand the semantics of the remote databases. In our example, the various companies have to define a global common language (Q) to query their information systems (IS1, IS2). This solution is well adapted for information systems that are based on the same family of data models and does not require complex query translators.

The **mediation** approach is based on two main components: mediator and wrapper. The mediator is used to create and support an integrated view of data over multiple sources. It provides various services to support query processing. For instance, a mediator can cooperate with other mediators to decompose a query into sub queries and generates an execution plan based on the resources of the cooperating sites. The wrapper is used to map the local databases into a common federation data model. The wrapper component provides the basic data access functions (Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J., Widow J., 1995). In our example, a translator, which acts as a wrapper, is placed between the conceptual representation of the mediator and the local description of each information source.

Recently, **Web-services** have been proposed as a method to address some of the challenges of web-based integrated systems. A Web-service can be viewed as a set of layers contained in a stack. The layers are dynamically defined according to user needs and are called through a set of Internet protocols. The protocols are different from those used in various network architectures. However, in all Web-service architecture, a base set of protocols is always used (W3C, 2001). This base set is composed of: SOAP (SOAP, 2003), WSDL (WSDL, 2001) and UDDI (UDDI, 2003). They allow for the discovery, description and exchange of information between Web-services. In our example, IS1 and IS2 web-services are invoked by distant systems and combined with distant applications to manage local information. Information is exchanged in the XML format. Nevertheless, the combination of web services requires the definition of semantic analysis based mechanisms. Thus, many solutions using the combination of web-services are based on ontology.

The **ontology** based interoperability approach uses ontology to provide an explicit conceptualization of the common domain of a collection of information systems (Benslimane , Leclercq , Savonnet, Terrasse, Yétongnon., 2000). Ontology defines a common vocabulary that can be employed by users from different systems. The construction of ontology for a domain is a difficult task and often requires merging existing overlapping ontologies. The interoperability solutions based on ontology describe the semantics of information rather than its organization or its format. In our example, the companies have to define ontology to capture the semantics of their domain of activity. The ontology approach can be used with the interoperability approaches listed above.

3. Semantic Approach of Integration

In order to support the action of agents, knowledge has to represent the real world by reflecting entities and the relations between them. Therefore, knowledge constitutes a model of the world and agents use their knowledge as a model of the world. Our method of integration is based on the semantics of knowledge carried out by information. Thus, the integration of different entities is possible when the semantics are identical. In addition, to model the semantics of knowledge as well as the structure where this knowledge is stored, it is necessary to rise to a high conceptual level. Knowledge representation is independent of knowledge use. Thus, knowledge representation and inferential mechanisms are dissociated (Guarino, 1994a). 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 particulare context of use (Bachimont, 2000). The semantics of knowledge are strongly constrained by the symbolic representation of computers. Therefore, N. Guarino (Guarino 1994b) introduced an ontological level between the conceptual level and epistemological level. The ontological level forms a bridge between interpretative semantics where users interpret terms and operational semantics where computers handle symbols (Dechilly, 2000).

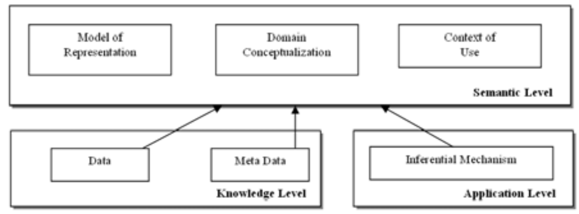


Figure 1. Description of information levels
The knowledge and the application level use the semantic Level

This section is composed of three sections. The first section defines what ontology is. The second part describes the various types of ontologies. The third part explains how to define the semantics of XML data.

3.1. The notion of ontology

A general consensus defines the role of ontologies in the following formula: “An ontology is a specification of a conceptualization” (Gruber, 1993). Only general processes of knowledge representation define ontologies, and they step in after the conceptualization phase. This definition consists in representing the knowledge of a specific domain. “A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose” (Gruber, 1993) N. Guarino considers ontologies as formal and partial specifications of conceptualization (Guarino, 1995). Ontologies are formal because they are expressed in logical form. Ontologies are partial, because no conceptualization can be entirely formalized, either because of design ambiguities, or because the representation language of ontologies cannot entirely represent them. Operational formalisms have a weak interpretation tolerance. Indeed, ambiguities pose problems for the treatments, obliging them to transform from an abstract ontology to a completely unambiguous formal ontology.

The general process of knowledge representation is characterized by three stages: Conceptualization, Ontologization and Operationalization.

- The conceptualization stage consists in identifying the domain of knowledge. It is subdivided into two stages. First of all, it is necessary to sort knowledge which is specific to the domain from that which is only the expression of knowledge in this same domain. Then, choices must be expressed on the nature of even conceptual elements of extracted knowledge (concepts, relations, properties, rules, constraints, etc). Then a semantic standardization is necessary (Bachimont, 2000). The problems of conceptualization include the description of tacit knowledge which can only be done at the time phase of ontology use. Indeed, at the time of knowledge conceptualization, tacit knowledge is never expressed, because they go from oneself for all (Leclere, 2002). Once concepts and relations are identified by their terms, it is necessary to describe semantics by indicating known instances, links that they maintain and their properties. It is then necessary to formalize the conceptual model obtained during the phase of ontologization.
- A partial formalization will make it possible to build an ontology. In order to respect the general objectives of ontologies, T. Gruber proposes five criteria that guide the process of ontologization (Gruber, 1993). Clearness and objectivity must be independent of any choice of implementation. The axioms must be coherent, consistent from the logical point of view. Ontology must be extensible; this means that it must be possible to extend it without modification. Encoding postulates have to be as tiny as possible, to ensure good portability. The vocabulary must be minimal so that the expressivity of each term is maximal. The respect of the domain semantics must be ensured by an ontological commitment, a concept suggested initially by (Gruber, 1993) as a criterion to use a shared specification of vocabulary. An ontological commitment is the guarantee of coherence between an ontology and a domain, but not the guarantee of ontological completeness. N. Guarino defined the ontological commitment as the relation between a logical language and

a set of semantic structure definitions. The direction of the concept is given by its extension in the interpretation context of the language. These semantic and ontological commitments must be guaranteed by a semantic structuring of knowledge. This structuring is all the more necessary to fill the formal gap between conceptualized knowledge and the formalism to represent them into machine. Ontologization leads to the hierarchical construction of concepts, relations and attributes. Once the model is structured, it should be translated into a semi-formal language of ontology representation. Among the languages of representation developed at the conceptual level, three major models may be distinguished: languages containing frames[8], logics of description and the model of the conceptual graphs (following section). Some of these languages or the languages using these models are already operational and the ontologies expressed in these formalisms can be directly used into machine. In other cases, an operationalization is necessary.

- This operationalization consists in implementing an ontology to make it possible for a machine to handle a knowledge domain. However, whereas many of the languages using the models authorize the expression of inferential knowledge, few are implemented to make possible the handling of this knowledge. The model of conceptual graphs is an exception, because knowledge representation in graph form makes it possible to implement reasoning by formal operations on graphs (comparisons, fusions, etc). If the language is not operational, it is necessary either to implement this language, or to transcribe ontology in an operational language. Finally, the operationalized ontology is integrated into a machine within a handling system.

3.2. Typology of ontologies

Ontologies are defined relative to a general process, which aims at the representation of knowledge. There exists a very large variety of ontologies resulting from conceptualizations of problems (Figure 2). Domain ontologies belong to the biggest category. These ontologies describe a set of vocabularies and concepts modeling a knowledge domain. The objective of these ontologies is to create models of domain objects. These ontologies are a kind of knowledge meta-model in which concepts and properties are used for the definition of object types. Task ontologies form the second category of ontologies and are used to conceptualize tasks specific to a domain. These tasks are tasks of diagnosis, tasks of planning, tasks of design, tasks of configuration, etc. These ontologies govern a set of vocabularies and concepts, which describes a resolution structure of the problems inherent in the tasks and independent of the domain. Domain ontologies and task ontologies are located on the same conceptual level. Generic ontologies imply a higher level of abstraction while application ontologies are more specific than domain and task ontologies. This means that ontologies of a higher level can be used by ontologies of a lower level. Consequently, generic ontologies convey knowledge which aims to solve generic problems, or generic knowledge usable by various domain ontologies. Application ontologies, on the other hand, are used in applied problems of knowledge interfering with task and domain ontologies. The concepts specified by ontologies of application often correspond to the roles played by the entities of the domain while carrying out some activity.

3.3 Ontologies and XML

This section presents our work on ontology based data integration. The implementation of an ontology requires a stage of mapping between the system elements and their ontological “counterparts”. Once this correspondence is made, the element representation of the system in the ontology is regarded as a diagram of meta-data. 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 ontology based data

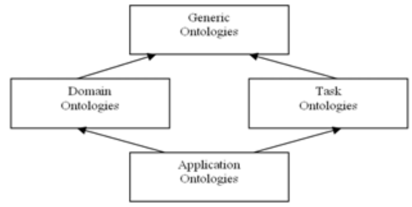


Figure 2. Typology of ontologies

integration using domain ontologies to provide integration structures and request processes to these structures. According to (Cruz, 2004, Klein 2002, Lakshmannan, 2003), data integration consists in defining rules of correspondence between information sources and the ontological level. The principle consists in labeling source elements, and thus providing a semantics to elements compared to a consensual definition of the meaning. This phase is inevitably necessary, because such information was not added to the document at the time of its creation. Moreover, the XML schema defines only the structure of associated XML documents. However, the XML schema can be used to define ontology by extracting a set of elements and properties whose meaning will be defined for a more global use.

4. Methodology of Integration

Our solution to integration consists in connecting various levels of semantic and schematic abstraction. This solution is articulated around 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 a domain ontology. 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, ontologies will make it possible to link the concepts and relations by amalgamating the attributes of the two common elements (c.f. Figure 3). The domain ontology will be extended and then modified to represent the semantics of several XML schemas relating to a particular domain.

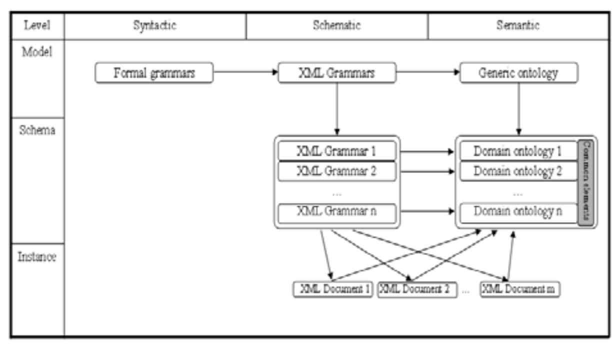


Figure 3. Common elements of domain ontologies

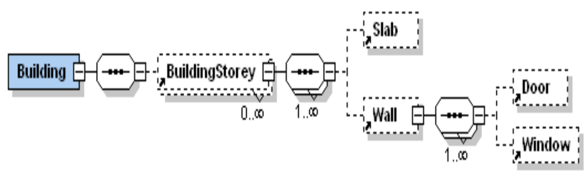
To specify the element semantics of an XML schema, it is first necessary to identify and mark them. We call these schematic marks. They will be used to establish bonds between the structure of an XML document and its semantic definition.

4.1. XML Grammars and XML documents

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 can be likened to the leaves of trees. These tags form a tree structure whose writing rules are collected in an XML schema. Example 1 shows an XML schema which validates the XML document Example 2. If one marks the elements of the XML schema using a schematic mark by binding them to a semantic definition, then the implicit semantics of the XMLschema becomes explicit. Consequently, the elements of an XML document, whose schema was marked, can be recognized in a semantic way.

[8] Frame languages: http://hopl.murdoch.edu.au/findlanguages.prx?NodeID=2231240&which=ByMyCat

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
                                              attributeFormDefault="qualified">
<xs:element name="Building">
       <xs:complexType>
           <xs:sequence>
                <xs:element ref="BuildingStorey" minOccurs="0" maxOccurs="unbounded"/>
              </xs:sequence>
         <xs:attribute name="Label" type="xs:string" use="optional"/>
       </xs:complexType>
    </xs:element>
       <xs:element name="BuildingStorey">
              <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" use="optional"/>
              </xs:complexType>
       </xs:element>
       <xs:element name="Slab">
              <xs:complexType>
                 <xs:attribute name="Label" type="xs:string" use="optional"/>
              </xs:complexType>
       </xs:element>
       <xs:element name="Wall">
              <xs:complexType>
                     <xs:sequence maxOccurs="unbounded">
                            <xs:element ref="Door" minOccurs="0"/>
                            <xs:element ref="Window" minOccurs="0"/>
                     </xs:sequence>
                     <xs:attribute name="Label" type="xs:string" use="optional"/>
              </xs:complexType>
       </xs:element>
       <xs:element name="Door">
              <xs:complexType>
                     <xs:attribute name="Label" type="xs:string" use="optional"/>
              </xs:complexType>
       </xs:element>
       <xs:element name="Window">
              <xs:complexType>
                     <xs:attribute name="Label" type="xs:string" use="optional"/>
              </xs:complexType>
         </xs:element>
       </s:schema>
```



Example 1. Building XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<Building xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents\batiment.xsd" Label="Ecole">
<BuildingStorey Label="Etage 1">
             <Slab Label="Dalle Plafond"/>
             <Slab Label="Dalle Sol"/>
             <Wall Label="Mur 1">
                    <Window Label="Fenêtre 1"/>
                    <Window Label="Fenêtre 2"/>
                    <Window Label="Fenêtre 3"/>
                    <Door Label="Porte 1"/>
             </Wall>
             <Wall Label="Mur 2">
                    <Window Label="Fenêtre 4"/>
             </Wall>
```

```
<Wall Label="Mur 3">
                    <Window Label="Fenêtre 5"/>
              </Wall>
              </BuildingStorey>
 <BuildingStorey Label="RDC">
 <Slab Label="Dalle Sol"/>
<Wall Label="Mur 4">
       <Window Label="Fenêtre 6"/>
       <Window Label="Fenêtre 7"/>
       <Door Label="Porte 2"/>
       <Door Label="Porte 3"/>
 </Wall>
 <Wall Label="Mur 5"><
       <Window Label="Fenêtre 8"/>
       <Window Label="Fenêtre 9"/>
 </Wall>
</BuildingStorey>
</Building>
```

Example 2. XML document respecting building XML schema

4.2 Semantic definition of the schematic marks

In the preceding section, we detailed how to distinguish two elements of an XML schema thanks to schematic marks. We must now define the semantics of these schematic marks. Ontologies are logical systems that incorporate semantics. The formal semantics of knowledge representation systems allows us to interpret ontological definitions as a set of logical axioms. Consequently, the definition of semantic marks is defined within an ontology. In this way, we will be able to handle the elements defined in grammars by their semantics. The following section presents the principal defining concepts of an ontology: concepts and relations.

4.3 Concepts and Relations

A concept is characterized by three components which are one or several terms, a concept and a set of objects (Uschold, 1995). The notion, also called the intension of the concept, contains the semantics of the concept. The intension of the concept corresponds to the properties, the attributes, the rules and constraints of an object's class. For example, width and the height describe the intension of the concept "wall ". At the same time, the attributes of a schematic mark characterize the intension of the concept defining the element. Moreover, the intension of a schematic mark integrating several schemas is made up of all the element attributes of various XML schemas. Thus, the concept models the semantics of the elements in various XML diagrams.

A concept can be regarded as a class whose instances form the extension of the concept. By analogy, the extension of a concept corresponding to a schematic mark is the set of under trees of XML documents validated by its grammar. If the concept integrates several schematic marks, then the extension is made up of various under trees from XML documents produced by various XML schemas. Moreover, if two XML under trees from two XML documents of different XML schemas correspond to the same concept instance, then these under trees are integrated in this same instance. We note two levels of integration here. The schema integration level where the attributes are gathered in the intension of the same concept and the data integration level where values of the attributes are gathered in the same instance of the concept.

Like a concept, a relation includes one or several terms, a notion and a set of objects. The relations are also organized into a hierarchical structure using properties of subsumption. The intrinsic properties of a relation are algebraic properties (symmetry, reflexivity, transitivity) and cardinality, which defines the number of possible relations between instances. Moreover, the notion of intension and extension defined for concepts is also defined for relations. Thus, relations can be marked in an XML schema. Nevertheless, an XML document defines two kinds of relations. The first kind of relation is defined by a tag, and the second kind of relation is defined by a direct relation between a tag father and a tag son. In this case, the relation is an implicit one that the system will define as a relation, but without XML definition.

The cognitive structure storing information must not only allow schema integration, but also data integration. To do so, the structure must be able to store the new concepts carried out using the schematic marks, but also instances of the concept containing the values of the attributes. The notion of intension and extension thus makes it possible to define the semantic structure of XML data. The conceptualization which is carried out to describe semantic information is on a higher level of abstraction. Consequently, the ontology is of a generic type, because it makes it possible to dynamically define new concepts without having to redefine the cognitive structure. The new stored concepts form a domain ontology describing the semantics of the XML schema.

4.4 Integration level

To integrate an XML schema with a generic ontology, it is necessary to define rules. These rules of integration make it possible to share the properties of various XML schemas. This level of integration is called schema integration, because it gathers in the same concept or relation various properties defining the intension of a concept or a relation. On this level of integration comes to be added a second level of integration. This is named integration of data. Indeed, the first level defines the concepts, relations and attributes which will be instanced in the second level of integration.

The implementation of the generic ontology in the form of classes allows for both levels of integration. In fact, these classes represent the notions of concept, relation and attribute, and will be instanced to allow for the definition of domain ontologies. Consequently, a domain ontology corresponds to an XML schema and if XML schema elements are semantically identical, then domain ontologies have common elements.

Rules for integration form an integral part of the system conceptualization. They provide a vocabulary forming the taxonomy of our system. This vocabulary is composed of the words: *concept*, *relation*, attribute, *conceptual mark, relational mark, attribute mark*, *semantic element*, and *relational element*. Each one of these words defines a concept of generic ontology and will correspond to a class.

- The class *concept* is defined by properties modeled by its intension and composed of schematic marks.
- The class *relation* is defined by properties modeled by its intension and composed of schematic marks.
- The *semantic* and *relational elements* are classes allowing to instance objects from XML documents.
- Objects from the class *semantic element* related to an instance of the class *concepts* form the extension of the instance of *concept*.
- Objects from the class *relational element* related to an instance of the class *relation* form the extension of the instance of *relation*.
- Instances of the classes *conceptual marks* and *relational marks* refer to the schematic marks on the XML schema. These marks are XML data extracted from the XML schema.

We presented in this section the classes necessary for the definition of a generic ontology. These classes can be regarded as meta-classes whose instances are classes defining domain ontologies. The meta-class concept allows the definition of classes such as a class "wall" specifying the attributes of the concept "wall" in an ontology of the building trade, for example. These meta-classes

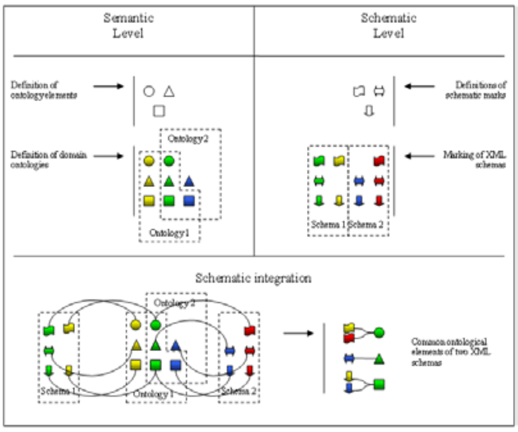


Figure 4. Schematic integration level

preserve the bond between the classes and the instances allowing a rapid search in the information system using either the meta-data or the data directly.

5. Conclusion

In the more specific framework of a language oriented object, the correspondence between an intension and an extension is fixed. If the modeling of the intension corresponding to the concept of a class is finished, the structure of the objects generated starting from the model cannot evolve. It is then only possible to modify the values of the instances. The ontological approach makes possible the addition of new structures of data to the classes, as well as the addition of new data to the existing objects. This is possible by adding new properties to the intensions of the concept. This contribution does not call into question the initial structure, but simply makes it possible to enrich it during its cycle of use. This evolutionary design of the cognitive structure aims to integrate the descriptions of objects using different XML catalogues provided by Web Services. Indeed, each service is described by an XML document and information is transported by a flow of XML data.

Until now, it has only been a question of syntactic validation of data by XML grammars. The question of semantic validation has not yet been tackled. The importance of certain attributes of the intension for the validity of a concept has not been fully clarified in the literature. Indeed, some properties are essential for the characterization of a concept and their suppression would mean the suppression of the concept. On the other hand, the integration of several intensions into the same concept makes it possible to approach the concept in a contextual way. If a context is characterized by the source domain of the intension, then the suppression of an intension does not call into question the concept in the other contexts. Consequently, an information system based on an ontology integrating XML data can evolve in terms of its cognitive structure while adding or by removing intensions with the concepts. The contextual approach is advantageous for the installation of processes allowing the retrieval of data from the information system [VAN04]. This contextual approach will not be developed here as it is beyond the framework of the present article.

References

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

Bachimont, B. (2000). 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, 305-323.

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

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

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).

Dechilly, T., & Bachimont, B. (2000). Une ontologie pour éditer des schémas de description audiovisuels : extension pour l'inférence sur les descriptions. in IC'2000.

Gruber, T. (1993). A translation approach to portable ontology specifications, Knowledge Acquisition 5(2), 199-220.

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

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

Guarino, N., Giaretta, P. (1995). Ontologies and knowledge based, towards a termonological clarification, in Mars N., eds., Towards very large knowledge bases: knowledge building and knowledge sharing, IOS Press, 25-32.

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).

Leclere, M., Trichet, F., & Fürst, F. (2002). Operationalising domain ontologies : towards an ontological level for the SG family, in Foudations and Applications of Conceptual Structure, contributions to the International Concference on Conceptual Structures (ICCS'02), Bulgarian Academy of Sciences.

Uschold, M., & King, M. (1995) Towards a methodology for building ontologies, in Proceeedings of the workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI'95.