THE PLIB ONTOLOGY-BASED APPROACH TO DATA INTEGRATION

Guy Pierra
Laboratory of Applied Computer Science (LISI)
National Engineering School for Mechanics and Aerotechnics (ENSMA), Poitiers
86960 Futuroscope Cedex - Prance
pierra@ensma.fr

Abstract This paper presents an overview of the integration approach developed in the PLIB project (officially ISO 132584 "parts Library"). This approach is based on (1) an ontology model, (2) an ontology mapping model, and (3) an ontology-based database model (OBDB).

1. Introduction

The semantic heterogeneity has been identified as the most challenging issue of data integration since it requires to understand the relationship between data and real world objects to be able to map various conceptualization often based on various point of views. In the traditional data integration approach, domain semantics is encoded in a procedural form. More recent approaches (Wache, 2001) explicitly represent domain semantics through ontologies. The commonality of most of these approaches is that data integration is considered *a posteriori,* once the data sources have been set up.

The goal of this paper is to give an overview of the *a priori* approach developed over the last 10 years in the PLIB project to provide for automatic integration of engineering component databases and catalogues (ISO, 2004). The PLIB approach is based on three ideas, leading to the development of three resources. (1) In a number of structured activity domains, a precise shared vocabulary is already existing to allow person-to-person communication. It should be possible to make this vocabulary computers sensible in the form a shared ontology. A *context-explicit ontology model,* known is the PLIB ontology model, has been developed. (2) The main characteristic of human activity is to innovate, this means to create specific extensions of existing concepts and practices. An *on-*

tology mapping model allowing a user to define its own ontology as a specialization of a shared ontology has been developed. (3) Integrating data require human and/or computer understanding of data meaning. We have developed a new database model, called *Ontology-Based Database (OBDB)* where each database contains its own ontology.

We presents an overview of these three models in the three following sections.

2. Overview of PLIB ontology

The role of a PLIB ontology is twofold. First it is intended to support user interfaces at the knowledge level, both for graphical access and for queries. Second, it provides for automatic integration. PLIB ontologies are: (1) object-flavored (the word is captured by classes and properties), (2) property-oriented (class are only defined when required to define the domain of some properties), (3) conceptual (each entry is a concept defined by a number of facets, both formal and informal, and not a term), (4) multilingual (each entry is associated with a globally unique identifier (GUI), words used in some facets may appear in any number of languages) (5) formal (the PLIB ontology model is specified in EXPRESS); (6) modular (an ontology may reference another ontology); (7) consensual (both the model (Kashyap et al., 1996), and domain ontologies result from standardization consensus).

2.1 Minimizing context-sensitivity

Importance of a context representation for the semantic integration of heterogeneous database was already underlined by a number of researchers in multidatabase systems, both at the schema definition level (Kashyap et al., 1996), and at the property value level (Goh et al., 1999). To ensure the feasibility to reach a consensus on an ontology definition (Pierra, 2003), a PLIB ontology minimizes context sensitivity: (1) definition context explication is done by associating with each properties, the higher class where it is meaningful and with each class the properties applicable to each class; (2) value context explication is done by associating with each *context-dependent property* value its evaluation context represented as a set of *context parameter*-value pairs (properties whose values are not context-sensitive are called *characteristic properties*), (3) value scaling explication is done at the schema level by associating each quantitative property type both with a dimensional equation and with a unit, and (4) to avoid context bias when choosing the properties associated with a class, each class is associated, at the ontology level, with

all the properties that are essential for its instances, at least in a very broad context.

2.2 Formal definition of PLIB ontologies

Formally, a single PLIB ontology may be defined as a 6-tuple : $O =< C, P, IsA, PropCont, ClassCont, ValCont >$, where: (1) C is the set of classes used to describe the concepts of a given domain; (2) P is the set of properties used to describe the instances of C. P is partitioned into P_{val} (characteristics properties), P_{fonc} (context dependent properties) and P_{cont} (context parameters); (3) $IsA : C \rightarrow C$ is a partial function, the semantic of which is subsumption; (4) $PropCont : P \rightarrow C$ associates to each property the higher class where it is *meaningful* (the property is said to be *visible* for this class); (5) $ClassCont : C \rightarrow 2^P$ associates which each class all the properties that are applicable to every instances of this class (rigid properties); (6) $ValCont : P_{fonc} \rightarrow 2^{P_{cont}}$ associates to each context dependent properties the context parameters of which its value depends.

Axioms specify that: (1) IsA defines a single hierarchy, (2) visible and applicable properties are both inherited, and (3) only visible property may become applicable.

EXAMPLE 1 *Figure 1 (a) presents a single ontology. Class hierarchy is represented by indentation. P = {mass}. The mass properties applies to hardware and components, but not to software and simulation models. mass is visible at the level of resources : PropCont(mass) = resources, with a definition s. t. "the mass of a resource that is a material object". It becomes applicable in hardware and components: ClassCont (hardware) = {mass}; ClassCont (component) = {mass}.*

3. Inter-ontology mappings

PLIB does not assume that all data sources use the same ontology. Each data source may build its local ontology without any external reference. It may also build it based upon one or several reference ontologies (i. e., standard ones). A class of a local ontology may be described as *subsumed* by one or several other class(es) defined in other ontologies. This means that each instance of the former is also instance of the latter. This relationship is named *case-of*. Though case-of relationship the subsumed class may either import properties (their GUI and definitions are preserved) or map properties (the properties are different but they are semantically equivalent) that are defined in the referenced class(es). It may also define additional properties.

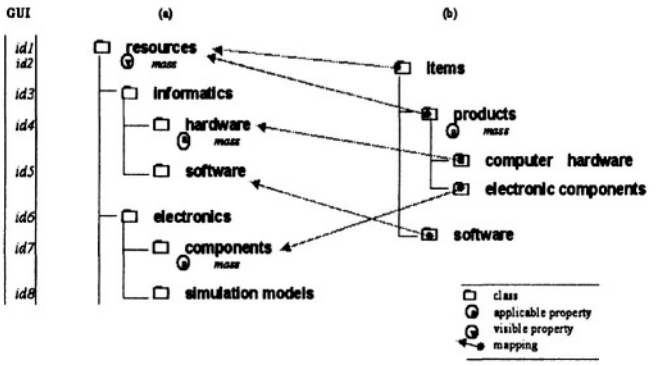

Figure 1. An example of a reference ontology (a) and of an user defined ontology (b)

A PLIB ontology O_m that includes mapping onto one (or several) other ontologies (called a *PLIB ontology*) may be formally defined as a couple: $O_m =< O, M >$, where : O is a single PLIB ontology, and $M = \{m_i\}$, is a mapping defined as a set of mapping objects.

Each mapping object has four attributes : $m =< domain, range, import, map >$, where: (1) $domain \in C$ defines the class that is mapped onto an external class by a *case-of* relationship; (2) $range \in GUI \subset \{string\}$ is the globally unique identifier of the external class onto which the $m.domain$ class is mapped; (3) $import \in 2^p$ is a set of properties visible or applicable in the $m.range$ class that are imported in $ClassCont(m.domain)$; (4) $map \subset \{(p, id) \mid p \in P \wedge id \in GUI \subset \{string\}\}$ defines the mapping of properties defined in the $m.domain$ class with equivalent properties visible or applicable in the $m.range$ class. The latter are identified by their GUIs.

EXAMPLE 2 *Figure 1 (b) present a (user-defined) ontology mapped on a reference ontology (a).* C = {*items, products, computer hardware, electronic components, software*} *and* P = {*mass*}. $M = m1, m2, m3, m4$ *with* $m_1 = (item, id1, (), ())$; $m_2 = (products, id1, (id2), ())$; $m_3 =$ *(computer hardware, id4, (), ())* ; $m_4 =$ *(electronic components, id7, (), ()). We note that no properties are mapped, they are all imported.*

Axioms (1) and (2) for single ontologies hold. Axiom (3) states that only imported or visible properties may become applicable. As shown by example 2, the structure of a (user) ontology may be quite different from the one of a standard ontology she references. Nevertheless, a

system storing the user ontology $< O, M >$ may automatically answer queries against the standard ontology(ies) on which O is mapped.

4. Ontology-based database

We call *ontology-based data base* (OBDB) a database (1) that explicitly represent an ontology, possibly including a mapping onto other ontologies, (2) whose schema refers to the ontology for each of its represented entities and properties and (3) whose each data may be interpreted in a consistent way using the meaning defined for the corresponding ontology entry. An OBDB is not required to populate either all the classes of its ontology or all the properties defined for a given class.

Formally, an OBDB is a quadruplet $OBDB =< O_m, I, Sch, Pop >$, where: (1) O is a PLIB ontology $(O_m =< O, M >)$; (2) I is the set of instances of the database; (3) $Sch : C \rightarrow 2^P$ associates to each ontology class c_i of C the properties which are effectively used to describe the instances of the class c_i; (4) $Pop : C \rightarrow 2^I$ that associates to each class (leaf class or not) those own instances. The following axiom, that states that only applicable properties for a class may be used for describing instances of this class, holds :

$$\forall c_i \in C, Sch(c_i) \subset Applic(c_i) \quad (1)$$

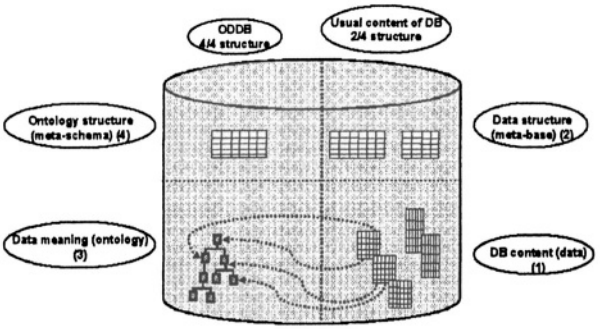

Figure 2. The OntoDB model for OBDB

The OntoDB architecture model we have proposed for OBDB, and prototyped in various environments, defines four different parts (see figure 2). The *ontology part* contains ontology definition as instances of the ontology model (that may be PLIB or any other model represented as a set of objets). In order to make the system generic with respect to

ontology model evolution, the *meta-schema part* records, in a reflexive meta-model, both the ontology model (or models) and its own structure. The *data part,* contains description of object instances (belonging to the ontology domain) described in terms of ontology class belonging and ontology property values. But, unlike individuals of description logic that may be described by any number of class belonging and by any existing properties (if they are not associated with specific constraints) thus making difficult storage indexing, in the OntoDB model instance data must obey to two assumptions. (A1) Each instance must belong to one only class, called its *base class* (and to all of its superclasses). (A2) Each instance may be only described by properties that are applicable property for its base class. With these two assumptions, each class may be associated with a view of which each row describes an instance that defines this class as its base class, and of which columns are the subset of applicable properties that were selected to constitute the schema of this class. Finally, the *meta-base part* (that allows in any DBMS to record data schema) is used for recording how the above view is defined on terms of table structure.

5. Conclusion

The PLIB integration approach allows both: (1) an autonomy of the various data sources, each one having its own ontology, and (2) an automatic integration. To the best of my knowledge, It is the first approach that fulfil these two conditions. An increasing number of B2B e-commerce actors are moving in the direction of PLIB.

References

L. Bellatreche, G. Pierra, D. Nguyen Xuan and H. Dehainsala, An Automated Information Integration Technique using an Ontology-based Database Approach. in *Proc. of 10th ISPE Int. Conf. on Concurrent Engineering*(2003)

C.H. Goh, S. Bressan, S. Madnick, M. Siegel, Context Interchange: New Features and Formalisms for the Intelligent Integration of Information, *ACM Transactions on Information Systems* 17(3) (1999) pp. 270-293.

ISO 13584-25. 2004. *Industrial Automation Systems and Intregration - Parts Library - Part 25: Logical model of supplier library with aggregate values and explicit content,* (ISO, Geneva, 2004)

V. Kashyap and A. Sheth, Semantic and schematic similarities between database objects: a context-based approach, *The VLDB Journal,* 5 (1996) pp. 276-304.

G. Pierra, Context-Explication in Conceptual Ontologies: The PLIB Approach. in *Proc. of 10th ISPE Int. Conf. on Concurrent Engineering* (2003)

H. Wache, T. Vgele, U. Visser, H. Stuckenschmidt, Ontology-Based Integration of Information: A Survey of Existing Approaches, in *Proc. of the IJCAI-01 Workshop: Ontologies and Information Sharing,* Seattle, WA, (2001) pp. 108-117