Extending the ANSI/SPARC Architecture Database with Explicit Data Semantics: an Ontology-based Approach

Chimène Fankam, Stéphane Jean, Ladjel Bellatreche, and Yamine Aït-Ameur

LISI/ENSMA and University of Poitiers - BP 40109, 86961 Futuroscope, France
{fankamc,jean,bellatreche,yamine}@ensma.fr

1 Introduction

The database (DB) design process follows the traditional ANSI/SPARC architecture proposed by Bachman [**?**]. A conceptual model (CM) is translated into a logical model corresponding to a data specification implemented in a DB system. The physical model defines how data are stored and accessed. External models allow a DB designer to adapting data according to user's requirements. Regarding the semantic exploitation of data models, this architecture has two major drawbacks [**?**]: (1) a strong dependency of models with designers and specific application requirements; (2) a gap between conceptual and logical models that increases with the discrepancy of the conceptual modelling languages.

The maintenance and/or evolution of the CM, that must be consensual when dealing with semantic integration of data sources (semantics and schema conflicts), are in the kernel of these problems. Recently, some works give more importance to CMs by materializing them in a DB [**?**]. In these works, the design of a CM is preceded by the design or by pre-existence of ontology. In this case, both ontology and data are represented in the DB. Such a DB is called an *ontology-based database* (OBDB). Hence our proposition is to extend the ANSI/SPARC architecture to support OBDBs.

2 Ontologies and Databases

This paper is based on our analysis of domain ontologies presented in [**?**]. Ontologies can be combined into a layered model, called the Onion Model and shown on Figure 1. Compared to DB requirements, (1) *Conceptual Canonical Ontologies (CCOs)* can be considered as shared CMs. They may play the role of a global schema in a DB integration architecture. (2) *Non Conceptual Canonical Ontologies (NCCOs)* provide mechanisms similar to views in DBs; non canonical concepts can be seen as virtual concepts defined from canonical concepts. These mechanisms may be used to represent the mapping between different DBs. (3) *Linguistic Ontologies (LOs)* may be used to localize similarities between DB schemas [**?**], to document existing DBs or to improve the DB-user dialog.

With the increasingly amount of data represented as instances of ontology classes (*ontology-based data*), a novel approach for processing these data in DBs,

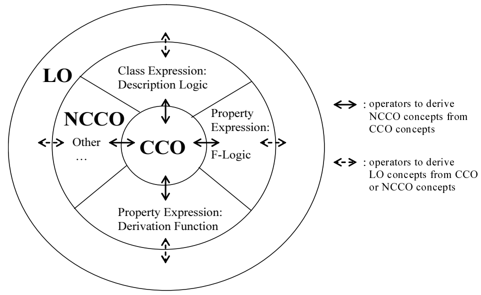

Fig. 1. The Onion Model of domain ontology

called Ontology-Based DataBases, emerges. An OBDB represents explicitly (1) ontologies, (2) data structure or data schema, (3) data, and (4) links between data and their schema and between data and the ontology. The ANSI/SPARC architecture with its three levels (conceptual, logical and external levels) does not support directly OBDB that introduces an additional data access level: the ontological level integrating semantics of data in a DB. As a consequence, we propose an extension of this architecture.

3 New Capabilities of the Proposed Architecture

Figure 2 shows our proposed extended architecture. The CM references the ontology using semantic links. It is defined using a subset of of the ontology that fulfill the application requirements. Compared to the ANSI/SPARC architecture, the proposed database architecture offers new capabilities.

Capabilities resulting from the Onion Model. Our proposed architecture introduces a new level to separate the logical (structure) and the ontological (semantics) representation of data.
Capability 1. *The database management system (DBMS) allows to expressing query at the ontological level independently of the logical representation of data.*
Exploiting non canonical concepts of the NCCO layer is a new capability.
Capability 2. *The DBMS supports the definition of non canonical concepts using canonical concepts of an ontology. Queries shall be expressed using canonical and non canonical concepts.*
For the LO layer, users shall be able to use terms in their own language.
Capability 3. *The DBMS supports the definition and exploitation of linguistic definitions of concepts that may be defined in different natural languages.*
Another characteristic of our proposed architecture is to enforce upward compatibility with the ANSI/SPARC architecture which leads to other capabilities.

Capabilities for Preserving Compatibility with the ANSI/SPARC Architecture. The SQL language is the standard defined to manipulate data. As a consequence, compatibility with the SQL language is required.

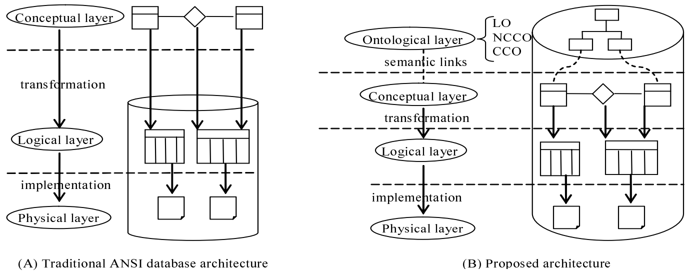

Fig. 2. Proposed extension of ANSI/SPARC architecture

Capability 4. *The DBMS permits the manipulation of data at the logical level preserving SQL compatibility.*
To optimize query processing at the ontological level, the DBMS shall provide an access to the lower level, i.e., the conceptual level.
Capability 5. *The DBMS handles access to data at the conceptual level from the ontological level.*
These capabilities have been implemented in the OBDB OntoDB2 introduced in [**?**]. In particular, OntoDB2 supports the definition of non canonical concepts (capability 2). To our knowledge, this feature is not available on most existing OBDBs, it is described in next section.

4 Representation of Non Canonical Concepts

The non canonical concepts constructions permit a richer semantic expression and characterization of the data stored in the DB. According to their origin, these constructors can be grouped in three categories (1) constructors of defined classes (Union, Intersection, Restriction) and properties (inverse, symmetric, transitive). They come from Description Logic; (2) logical rules issues from Frame Logic. They require a rule based reasoning engine to deduce new facts from existing ones; (3) Algebraic expressions (e.g, `diameter = 2*radius`). They come from the data processing community. They require an interpreter for algebraic expressions. Thus, a challenge is now to define a flexible OBDB architecture allowing using these various constructors.
From a DB point of view, non canonical ontologies introduce redundancy requiring specific treatments. The difficulty to have a kernel (non-deductive) DB covering the Onion Model is increased. As a consequence, to design such a DBMS, each level of the Onion Model must be managed specifically.
- **Defined Classes.** As stated before, non canonical classes (union, intersection, restriction) instances will be computed using the view mechanism (with triggers) that can be used to compute union, intersection or selection on a set of data.
- **Algebraic Characteristics of Properties.**
(a) Symmetry: we propose to automatically materialize a posteriori all data.

(b) Transitivity: to avoid overhead due to transitive closure, we assume at the beginning that all data are materialized. When a new relation concerning a transitive property is added, the new facts are materialized by a trigger in a non recursive manner. The trigger works as follows. When a new pair `P(x,y)` is added, for each existing pair `P(i,x)` in the DB, a new pair `P(i,y)` is added.
- **Algebraic Expressions**. Like for defined classes, the value of a derived property is computed by evaluating its expression (may be encapsulated by a view).

5 Conclusion and Future Work

In this paper, we have presented a new database architecture extending the traditional ANSI/SPARC architecture with the semantic of data : the ontology layer. Each concept references, by its unique identifier, the semantic definition available in ontologies. The resulting DB is an ontology-based database (OBDB). OBDBs support access to data at the knowledge level. They also provide help to the DB designer by defining the conceptual model as a fragment of ontologies. Notice that proceeding this way preserves an upward compatibility with the traditional ANSI/SPARC architecture. Indeed, when references to the ontology are omitted, we obtain the classical database architecture.

The developed architecture together with the associated exploitation language OntoQL has been implemented in the OntoDB database. Some demos are available on `http://www.plib.ensma.fr/plib/demos/ontodb/index.html`.

As future work, we plan to extend this approach to other software architectures and principally we intend to study how Architecture Description Languages (ADL) can be extended with an ontological layer. We believe that this approach scales up to other software architectures and helps to reduce the heterogeneity of software architectures and software architecture models.