USING ONTOLOGIES TO OVERCOMING DRAWBACKS OF DATABASES AND VICE VERSA: A SURVEY

Fatima Zohra Laallam[1], Mohammed Lamine Kherfi[2] and Sidi Mohammed Benslimane[3]

[1] Laboratory of Electrical Engineering (LAGE), Kasdi Merbah university Ouargla, Algeria
laallam.fatima_zohra@univ-ouargla.dz
[2] Laboratory LAMIA, Department of mathematics and computer Science, University of Québec à Trois-Rivières, Canada
kherfi@uqtr.ca
[3] Laboratory EEDIS, Computer Science Department, University of Sidi Bel-Abbès, Algeria
benslimane@univ-sba.dz

ABSTRACT

For a same domain, several databases (DBs) exist. The emergence of classical web to the semantic web has contributed to the appearance of the notion of ontology that have shared and consensual vocabulary. For a given, it is more interesting to take advantage of existing databases, to build an ontology. Most of the data are already stored in these databases. So many DBs can be integrated to enable reuse of existing data for the semantic web. Even for existing ontologies, the relevance of the information they contain requires regular updating. These databases can be useful sources to enrich these ontologies. In the other hand, for these ontologies more than the ratio 'size of the instances on the size of working memory' is large more than the management of these instances, in memory, is difficult. Finding a way to store these instances in a structured manner to satisfy the needs of performance and reliability required for many applications becomes an obligation. As a consequence, defining query languages to support these structures becomes a challenge for SW community. We will show through this paper how ontologies can benefit from DBs to increase system performance and facilitate their design cycle. The DBs in their turn suffers from several drawbacks namely complexity of the design cycle and lack of semantics. Since ontologies are rich in semantic, DBs can profit from this advantage to overcoming their drawbacks.

KEYWORDS

Databases, Ontologies, Survey.

1. INTRODUCTION

Ontologies are currently at the heart of many applications. They aim to support knowledge management and reasoning on this knowledge, with a view of semantic interoperability between people, between machines and between men and machines. For a machine to be able to process an ontology, the ontology has to be written in a language that the machine can understand.

Several of ontology languages have been developed (RDF, RDFS, DAML, OWL, etc.) To be manipulated, ontologies are needed to be in whole in the memory, during program running. When the quantity of knowledge is important, the performance of the system decreases. To preserve this

DOI : 10.5121/cseij.2013.3201

performance, a good solution may be reached by making use of database (DB) techniques when manipulating ontologies. Databases are used to structure and store a very large quantity of data, and making it easily accessible and with a high performance. Using DBs for the storage of ontologies allows manipulating them from disk rather than from the main memory. Thus, the coupling of a Database (DB) and an ontology can solve this problem. Several studies exist on this context.
Databases consist of a set of entities connected to one another by relationships. At the opposite of ontology concepts, these entities take their meaning at the level of the DB. Moreover, a large part of the semantics of data is lost during the transformation of the conceptual model into the logical model what causes many problems in data integration and retrieve of information. With the birth of the notion of ontology [38], several studies using ontologies have been carried by different communities (particularly the Semantic Web (SW) community and the DB community) to propose solutions to those problems. [48].

Our paper is divided into ten sections. After giving an introduction in section 1, in the second section, we present the DBs. In the third section, we give the main notions about ontologies. In section four, we give a comparison between a conceptual data model (CDM) and an ontology. Thereafter, in section five, we show drawbacks of DBs and how ontologies can remedy these. After that, we show drawbacks of ontologies and how DBs can remedy them. In section six, we talk about the use of ontologies in information systems. Then, in section seven, we identify new concepts resulting from the merging of DBs and ontologies. In section eight, we give the difference between two concepts namely: DBs based on ontologies and ontologies bases on DBs. In section nine, we identify the main lines of research on complementarity DB-Ontology. Then, we focus on those concerned usefulness of DBs for ontologies. For each axis, we specify its objective, the underlying issues and we give a state of the art on existing work. In section ten, we end our paper with a discussion and conclusion.

2. DATABASES (DBs)

2.1. Definition

DBs are structures that can store a large quantity of data. These data can be easily accessed by multiple users, simultaneously. The apparition of DBs dates from the 1960s, when have appeared the first Database Management Systems (DBMS). These systems are software that support the creation and access to the DB and ensure data integrity, reliability, efficiency and competition.
There are several types of DBs. Currently, the most known are: relational DBs, object DBs and object-relational DBs.

2.2. Life cycle of a DB.

The life cycle of a DB is the cycle of development that a DB goes through during the course of its life.

The life cycle of a DB includes four steps:

a) - Cycle of abstraction for designing a DB

The design of a DB consists of building abstract data models reflecting the DB at a conceptual level, using a design method such as MERISE [94], OMT [60], UML [96], etc.

To design a DB, the designer must make a detailed study of a perceived reality (expression of needs). This study is a very delicate and time-consuming. It requires months of work in collaboration with domain users. Furthermore, for the same application, two designers can produce different CDMs. Once designed, the DB conceptual models are then translated into logical models then into physical model to be implemented on machine. These models can't be understood by users because a large part of the semantics of data is lost during transformation.

b) - Implementation of data

Implementation consists of introducing designed models in the machine, to exploit them.

c) - Use (query, updating)

Databases are used in every field of data management where users need to consult and update data safely and with high efficiency, especially if there is a large quantity of data. Thus, there are databases in various fields: medical, industrial, administrative, etc.
A DB may be local. In such a case, it is usable on one machine by a unique user. It may also be distributed. In this case, the information is stored on a remote machine and accessible via network.

d) - Maintenance (correction, evolution)

After a database has been created and exploited, a maintenance task should be accomplished from a moment to another. A database at the physical level lacks ~~of~~ semantics. Thus, the maintenance tasks, involving updating and/or evolution of its structure, become very costly if the conceptual data models are not available.

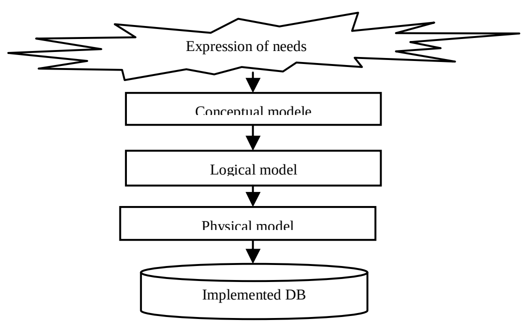

Figure1: Cycle of abstraction for designing a DB

3. ONTOLOGIES

3.1. Definition

In philosophy, ontology is part of metaphysics. It is the study of being as being, i.e. the study of general properties of what exists.

In artificial intelligence, several definitions for the ontology notion have been proposed [38], [15], [40], [6].

We can define an ontology as follows:

An ontology is a formal specification of a conceptualization of a common domain independently of a particular application. It is used by people, databases, and applications that need to share information on a domain [89], [56]. Ontologies are used in the fields of artificial intelligence, knowledge engineering, information systems and the Semantic Web and are developed for variety of domains [88].

3.2. Why use ontologies?

Ontologies are used [67] :

1. To share common understanding of the structure of information among people or software agents (One common knowledge and data model).
2. To enable reuse of domain knowledge (Define once, use in many applications in the same field)
3. To make domain assumptions explicit (Define classes, relationships and instances)
4. To separate domain knowledge from the operational knowledge
5. To analyze domain knowledge: What is the meaning of associations between objects

3.3. Components of an ontology

The main components of an ontology are concepts, relations, instances and axioms [84], [34].

- **Concepts***:* a concept represents a set or class of entities or `things' within a domain.

Concepts fall into two kinds:

1. *Primitive (canonic) concepts* are those which only have necessary conditions (in terms of their properties) for membership of the class.
2. *Defined (non-canonical) concepts* are those whose description is both necessary and sufficient for a thing to be a member of the class.

- **Relations*:*** describe the interactions between concepts or a concept's properties. Relations also fall into two broad kinds: *Taxonomies and Associative* relationships.

-

1. *Taxonomies* organize concepts into sub-super-concept tree structures. The most common forms of these relations are: *Specialization* relationships commonly known as the `is a kind of' relationship, and *Partitive* relationships which describe concepts that are part of other concepts.
2. *Associative* relationships are used to make links between concepts across tree structures.

- **Axioms:** are used to constrain values of classes or instances. In this sense, the properties of relations are kinds of axioms. Axioms, however, include more general rules. Their inclusion in an ontology may have several aims: define the meaning of the components; define restrictions on the value of attributes; set restrictions on the arguments of a relation; check the validity of a specific information; infer new information.

- **Instances**: are the `things' represented by a concept. They represent the extensional definition of an ontology.

4. COMPARISON BETWEEN ONTOLOGY AND CONCEPTUAL DATA MODEL (CDM) OF DB

After having introduced the DBs and ontologies in the previous section, we note that the ontologies as the CDM are a reality of a domain at a conceptual level. Then what is the difference between the two? In what follows, we will compare between ontologies and the CDMs. Ontologies and schemas both provide a vocabulary of terms that describes a domain of interest. However:

- DB schemas often do not provide explicit semantics for their data. Semantics is usually specified explicitly at design-time, and frequently is not becoming a part of a DB specification, therefore it is not available. Ontologies however are logical systems that themselves obey some formal semantics, e.g., we can interpret ontology definitions as a set of logical axioms.[70], [75]
- In the context of web, both ontologies and DBs are used but DBs are more suitable for the classic Web while ontologies are more appropriate for the semantic Web.
- Ontologies specify a representation of data modeling at a level of abstraction above diagrams of a specific database (logical or physical), so that the data can be exported, translated, interrogated and unified for all systems developed independently.
- The formal character of the ontologies can apply the operations of reasoning on ontologies. This is absent in the CDM.
- All instances of an ontology classes cannot initialize the same properties. They do not necessarily have the same structure as for tuples of a table.

In [13], Benslimane et al compare between ontology and CM as follow:

Motivation: one of the main goals of ontology is to standardize the semantics of existing concepts within a certain domain. While an ontology represents knowledge that formally specifies a shared/agreed understanding of the domain of interest, CMs describe the structural requirements for the storage, retrieval, organization, and processing of data in information systems such that the integrity of the data is preserved.

Usage: ontology plays a significant role at run-time to browse ontology concepts to form semantically correct queries, and perform some advanced reasoning tasks [40]. So, ontology is sharable and exchangeable at run-time, while CDMs are off-line model diagrams [44] and their queries are usually to retrieve a collection of instance data [71].

Evolution: generally an ontology is a logical and dynamic model that can deduce new knowledge relations from the stored ones, or check for its consistency. However, CMs are static and are explicitly specified at design, but their semantic implications might be lost at implementation-time.

Model Elements: in ontology, elements can be expressed either by their names or as Boolean expression in addition to using axioms such as cardinality/type restrictions, or domain/range constraints for classes or properties. On the other hand, CMs are concerned with the structure of data in terms of entities, relationships and a set of integrity constraints. For example primary key and functional dependences play very important roles within DBs, but this is not always the case in the ontology since it concentrates more on how the concepts are semantically interrelated.

Other differences may exist between ontology and DB. They are the basis of the contribution that can provide one to fill the drawbacks of the other. This is the subject of the following section.

5. DRAWBACKS OF ONTOLOGIES AND HOW CAN DBs HELP OVERCOMING THEM AND VICE VERSA

5.1. Drawbacks of ontologies and how can DBs help overcoming them

It was found that the ontologies suffer from the following deficiency: Larger is the ratio 'size of the instances on the size of the working memory' more difficult is the management of these instances in memory. This drawbacks is related to the size of the storage memory (working memory). To update an ontology, data recorded before will be loaded in memory, then rewrite them completely at the end of session. This processing lacks flexibility and performance (in terms of running time), when the size (number of instances) of the ontology becomes considerable. For increased flexibility and performance of systems based on ontologies, we use a store triples (triple store, ie, a database). This store will allow the structuring of a large number of instances. This allows better management of data. The Instances will be stored in a previously installed DB and will be accessed by a DBMS connected to the ontology through a connector. The advantage is that we do not overwrite all the time all the data and we have the possibility of making requests on what was added.

5.2. Drawbacks of DBs and how can ontologies help overcoming them

It was found that DBs suffer from a number of shortcomings, namely:

- The CDM answers a particular application need, it can't be reused for another different application need.
- Complexity of the design cycle of a DB.
- Difficulty of data integration especially when CMs are different (according to their points of view or according to their type). [22]
- Obligation to know the names of entities and their attributes represented to formulate a query that may be semantically right. [31]

After study and observation, researchers have discovered that ontologies can play an important role in filling these drawbacks. In what follows, we will detail each drawback and then explain how the ontology can remedy this:

– Objective of modeling

CDM prescribed information that must be represented in a particular computer system to face a given application need. In contrast, an ontology describes the concepts of a domain of a rather general point of view, regardless of each specific application and any system in which it is likely to be used. Elements of the domain knowledge represented by ontology and relevant for a particular system can be extracted from the ontology by the system designer without the latter having to rediscover them.

– Atomicity of concepts

Unlike CDM, where each concept only makes sense in the context of the model in which it is defined, in an ontology, each concept of an ontology is associated with an identifier to reference it from any environment, regardless of the ontology in which it was defined.

CDM can extract, from an ontology, only concepts (classes and properties) relevant to its target application and organize them in a quite different way from their organization in the ontology, the reference to the ontological concept allowing define precisely the meaning of the referencing entity.

– Consensuality

In contrast to CDM, a domain ontology representing the concepts in a consensus form to a community. Therefore, all members of the community can use this ontology and have easy access to the information in the field. Similarly, the semantic integration of all systems based on the same ontology can be easily realized if the references to the ontology are explained.

Mix well in a system 'ontology and DB' avoids all the problems of integration and non-interoperability between systems.

– Non-canonical represented information

Unlike the CMs that use a minimal language and non-redundant information to describe domain ontology, ontologies use, in addition to primitive concepts (canonical), non-primitive (defined or non-canonical) concepts that provide alternative access to the same information.

6. THE USE OF ONTOLOGIES IN INFORMATION SYSTEMS

Many works on the ontology domain have clarified the concept of ontology but the capabilities and benefits of Information Systems (IS) based on this technology are still unclear. Therefore, the usage of these in industry and services is not widespread [47]. One of the most interesting usages of ontologies in Information System is ontology-based data access. To the usual data layer of an information system we superimpose a conceptual layer to be exported to the client. Such a layer allows the client to have:

- A conceptual view of the information in the system which abstracts away from how such information is maintained in the data layer of the system itself [106], [48], [78], [79].
- More alternative for expressing queries [31].
-

7. BIRTH OF NEW CONCEPTS

Due to the merging of DBs and ontologies, new concept has emerged in the recent years: Ontology based on database (OBDB). It is a database model that allows both ontologies and their instances to be stored (and queried) in a single database. The DB was initially empty. This interests SW community. The same word is used to name a database based on ontology (DBBO) which interests DBs community, where data (tuples) of the DB are based on an ontology. Their meanings are explicitly defined by reference to an ontology. Each word (table or attribute) in the DB is linked to the underlying concept (Semantic Indexing). This ontology concepts can be linked to others concepts. There more concepts, but only vocabularies: synonym dictionaries, thesaurus, and taxonomies. This extension will allow the user to expand their research using synonyms, hypernyms and hyponyms (extension semantics).

8. DIFFERENCE BETWEEN DATABASES BASED ONTOLOGY (DBBO) AND ONTOLOGY BASED DATABASE (OBBD)

For DBs community, provide the DBs by ontologies this gives birth to a new DB model called *ontology based database* (OBDB) that we propose to call *databases based ontologies* (DBBO). This concept of DBBO will not to be confused with the notion of *ontologies based on DB* (OBDB) which interests semantic web community and has as main objective allowing the storage of a large quantity of instances (section 9.1).

A DBBO has as main objective enriching DBs in semantic [17], [31], [23], [24], [106], [48], [78], [79], [3], [49]. This new concept has two main characteristics. First, it allows managing both ontologies and data. Second, it permits to associate each data with the ontological concept which defines its meaning. Such a concept is a data source that contains:

- A database a priori (structure and tuples);
- An ontology (local) for the semantic indexing of DB;
- Possibly, the local ontology references to external ontologies;
- A relationship between each data and the ontological notion that defines the sense.

According to our reading, it was found that the notion of OBBD is used interchangeably to refer to database based ontology (DBBO) or an ontology based database (OBDB). For both structures, this can provide a level knowledge user interface and query a DB without knowing its exact pattern [79]. In our opinion, it must do the difference between the two:

- OBDB structure the instances of the ontology in a DB to increase the effectiveness of ontological systems. Thus, instances of the ontology will be stored in a structure of a DB (DB without tuples). The interpretation of these instances is not structural (tuple), it is semantics. This interests the SW community (Section 9.1)

- DBBO is to have ontological layer above the DB to increase the semantics of a DB to facilitate information retrieval, integration, maintenance, etc. Ontology in this case has no instances a priori. This interests the DBs community. It is the subject of this current axis. The following table shows the difference between the two concepts.

	OBDB	**DBBO**
Community	SW	DB
Data Interpretation	Semantic	Structural a priori
Existence of databases a priori	No	Yes
Storage	Instances	Tuples
Goal of merging ontology and DB	Using the functionalities of DBMS	Increase of semantic
Level data access	knowledge level	Logical model, a priori
Modification of the DB structure	No	Yes

Table1. Comparison between OBDB and DBBO

9. USING ONTOLOGIES TO OVERCOMING DRAWBACKS OF DATABASES AND VICE VERSA: THE MAIN AXES OF RESEARCH

The DBs are widely used. They serve to store large quantity of data. These data are managed in a simple, efficient and effective manner by a DBMS.

The use of DBs in the field of ontologies offers an important benefit. DBs allow to store high quantity of ontology data.

The modeling and use of ontologies in the field of database creation and data integration offers several benefits. Ontologies describe domains on a high abstraction level and can therefore be easily understood by and discussed with domain experts without detailed database knowledge. The strengths of ontologies lie especially in the possibility to build a consistent and formal vocabulary, which cannot only be used for the definition of the structure and meaning of data stored in a database, but also be reused, to interoperate with and build applications based on this vocabulary [88].

Recent years have seen the appearance of several works of rapprochement between databases and ontologies to fill the drawbacks of one relative to the other.

In our study of this domain, we found that the major axes of research relating to this merger are organized around five main axes, which may be grouped in two families:

Usefulness of ontologies for DBs:

- Ease of retrieval of information in DBs, by increasing their semantic [17], [31], [8], [9], [3],
- Facilitate the design of DBs, from ontologies,
- Facilitate the integration of DBs,
-

Usefulness of DBs for ontologies:

- Design of ontologies, taking advantage of the existence of several DBs in the same field of ontologies,
- Management of a large number of instances in an ontology.

We present in this section axes related to the usefulness of DBs for ontologies. For each of these axes, we will present it, underlying issues and cite relevant research.

9.1. Management of a large number of instances in an ontology

9.1.1. Objective of the axis

The goal of this axis of research is to allow easy management of an ontology instances when they are in large quantities. We will try to answer in this section the following questions:

- What is the problem if the number of instances in an ontology becomes large?
- How DBs can participate in solving this problem?
- What are the existing contributions to realize this solution?

9.1.2. Problem of ontologies

For ontologies more than the ratio '*size of the instances on the size of working memory*' is large more than the management of these instances, in memory, is difficult (the constraint of incompatibility with the MC treatment).

Finding a way to store these instances in a structured manner to satisfy the needs of performance and reliability required for many applications becomes an obligation. As a consequence, defining query languages to support these structures becomes a challenge for SW community.

9.1.3. Using a DB for solving the problem of ontologies.

In a DB, we can store and managing a large quantity of information without affecting the ease of data management. For SW community, linking ontology to a database allows to get benefit from the functionality of DBMS. The use of DB in ontological systems allows the structuring of a large number of instances. Ontologies and instances which they describe are stored in a DB. This leads to a better data management. The instances will be stored in a DB and be accessed by an appropriate DBMS which is connected to ontology. In these data, the interpretation is not structural (tuple), it is semantics. This allows providing a user interface at the knowledge level and queries a DB without knowing the exact pattern [79].

9.1.4. Existing contributions

In the context of SW, several structures have been proposed to support both ontologies and DBs. We propose to call them ontology based on DB (OBDB) (not to be confused with the notion of database based ontology (DBBO) which interests DB community. Several approaches have been proposed for storing both ontologies and ontology individuals in a database, to get benefit of the functionalities offered by DBMSs (query performance, data storage, transaction management, etc.) [24], [25], [59], [1], [2], [61], [16], [32], [36], [63], [80], [73], [81], [58], [51], [52], [33], [101], [42],[62], [100].

These approaches differ according to:

- The supported ontologies models (RDF, OWL ...)
- The database schema used to store ontologies (intentional metadata layer) represented by this model;
- The database schema used to represent instances of the ontology (extensional layer).

Most of the works are based on:

- ✓ **RDF model** because The core of the Semantic Web is built on this data model [87]
- ✓ **Le relational model** because relational database management systems (RDBMS) are simple and have repeatedly shown that they are very efficient, scalable and successful in hosting types of data which have formerly not been anticipated to be stored inside relational databases. RDBMSs have shown also their ability to handle vast amounts of data very efficiently using powerful indexing mechanisms [Sak 10].

In the following table, we present some examples of work that exist in this context

Work	Structure	Ontology model	Database scheme
[2]	RDF suite	RDFs	Object-Relational
[16]	Sesame	RDFs	Various
[73]	DLDB	DAML+OIL	Relational
[105]	rdfDB	RDF	Relational
[82]	RDFStore	RDF	Relational
[14],	RDQL	RDF-S/OWL	Relational
[91]	PARKA-DB	RDF	Relational
[97]	KAON	RDF	Relational
[25]	OntoDB	Various	Object-Relational

Table2. Examples of work on OBDB

9.1.4.1. Approaches of representations of OBDBs

The births of OBDBs were nearly in the 2000s. They were not designed to explain the semantics of data in a database in the usual sense, but to provide a solution to persistent for data from the Web and described by conceptual ontologies represented in a particular model of ontologies [1], [2], [16], [32], [36], [63], [73], [81], [58], [51], [52], [33], etc. As a result, most architectures are moving away from the traditional architecture of databases. These architectures are usually based on very fragmented data representation schemes. They are influenced by the data structure on the Web (RDF structure especially). This type of architecture does not exploit the structural information and typing which is traditionally available in the logic models of databases.

Recently, several architectures of OBDB have been proposed [23] [29] [48] where the implementation of the data based ontological approaches the structure of traditional databases. Under certain hypotheses of typing and structuring, these OBDBs allow, on the one hand, obtaining better performance in query processing and on the other hand, they allow indexing of existing databases with ontologies. Thus, the term tends toward DBBO.
The OBDBs can store, in a database, ontologies and instances which they describe according to four approaches:

- **The RDF warehouse approach (generic):** in this approach a large quantity of data is stored as RDF triples "subject - predicate - object" [105], [1], [95].

This representation is independent of the model ontologies supported in OBDB. The instances are directly related to an ontology without being structured by a significant pattern for a user database.

Triplet		
Sujet	**Prédicat**	**Objet**
http://rdfs.org/sioc/ns#Forum	rdf:type	rdfs:Class
http://rdfs.org/sioc/ns#Forum	rdfs:label	Forum
http://rdfs.org/sioc/ns#Forum	rdfs:comment	A discussion area …
http://rdfs.org/sioc/ns#Forum	rdfs:subClassOf	http://rdfs.org/sioc/ns#Container
http://rdfs.org/sioc/ns#has_host	rdf:type	rdf:Property
http://rdfs.org/sioc/ns#has_host	rdfs:label	has host
http://rdfs.org/sioc/ns#has_host	rdfs:comment	The Site that hosts this Forum
http://rdfs.org/sioc/ns#has_host	rdfs:domain	http://rdfs.org/sioc/ns#Forum
http://rdfs.org/sioc/ns#has_host	rdfs:range	http://rdfs.org/sioc/ns#Site
…	…	…

Figure 2: an extract from the generic representation of an ontology [48]

- **The virtual RDF warehouse approach:** The instances schema (really stored as tuples) is designed independently of an ontology. Mapping rules can make the connection between ontology and instances in order to make explicit the data schema. This keeps the data stored in the DB in their structural forms and have a virtual and semantic view of these data, instead of transforming data into ontology language (RDF, OWL ...) [14].

- **The specific approach:** in this approach, the representation of the OBDB is dependent on the supported ontology model. The schema of instances can be built from the ontology. It consists of a representation of the ontology model in the relational or object-relational model supported by the DBMS underlying at OBDB. This representation is adopted by several OBDBs as: Sesame [16], RDFSuite [2], DLDB [73], PARKA [91], KAON [97].

In this approach, despite the specificity of the used ontological models, many OBDB provide the ability to import and export data to other ontologies models. For example, Sesame can export these data in OWL, KAON in RDF. This is the current trend of OBDBs which advocates the separation between ontologies and data (instances).

Class			
Id	**URI**	**label**	**Comment**
1	http://rdfs.org/sioc/ns#Container	Container	An area in…
2	http://rdfs.org/sioc/ns#Forum	Forum	A discussion…
3	http://rdfs.org/sioc/ns#Site	Site	The location of …

SubClassOf	
sub	Sup
2	1

Property					
Id	**URI**	**label**	**comment**	**Domain**	**Range**
11	http://rdfs.org/sioc/ns#has_host	has host	The Site …	2	3

Figure 3: An extracted example from of described ontology in RDFS. [48]

- **The OBDB approach**: Instances can be structured as a scheme in a traditional DB. The elements of this scheme are then linked to an ontology to define the semantics. Unlike DBBO that interprets the data focusing (at first) to their structure, OBDB interprets data (instances) semantically regardless of their structure at the DB [48].

-

Approach	**OBDB**	**DBBO**
Virtual RDF warehouse approaches	X	
The RDF warehouse approaches	X	
Specific approaches	X	
DBBO approaches		X

Table3. Approaches of Birth of new concepts (DBBO and OBDB)

9.1.4.2. Usual language for OBDB

Actually, several tools for managing OBDB are available (e.g., Protégé2000). Usually, ontologies and their ontology individuals manipulated by these tools are stored in the main memory, during the execution. Thus, for applications manipulating a large quantity of ontology individuals, query performance becomes a new issue [22]. In this context, several query languages have been developed: SWRL [43], RQL [2] [16], SPARQL [80], D2RQ, SquishQL [105], OWL-QL [32], RDQL [14], OntoQL [49], [45], [46]. We propose to call them ontology languages based on DB (OLBDB). Each language is specific to a particular type of ontology model. SWRL allows querying data and ontologies modeled in OWL. RQL language is intended for data and ontologies modeled in RDF-Schema. SPARQL relate to those modeled in RDF. Languages SOQA-QL [102] and OntoQL [49] allow querying ontologies and data regardless of the used ontology model.

We have presented in this axis the notion of OBDB that allows storing ontologies and theirs instances in a DB in order to managing a large number of instances and to benefiting of DBMS functionalities. We have presented the approaches used to represent OBDB and the used languages to manipulate these structures.

9.2. Design of ontologies from BDs

9.2.1. Objective of the axis

This axis interested, essentially, the web semantic community. It concerns the domain of reverse engineering. In [20], Elliot J. Chikofsky and James H. Cross II define this term as follow:

“Reverse engineering is the process of analyzing a subject system to:

- Identify the system’s components and their interrelationships and
- Create representations of the system in another form or at a higher level of abstraction.”

For the same domain, several DBs exist. The emergence of classical web to the semantic web has contributed to the appearance of the notion of ontology that can have shared and consensual vocabulary. For a given, it is more interesting to take advantage of existing databases, to build an ontology. Most of the data are already stored in these databases. So many DB can be integrated to enable reuse of existing data for the semantic web.

Even for existing ontologies, the relevance of the information they contain requires regular updating. These databases can be useful sources to enrich these ontologies [41].

9.2.2. Methods and approaches of extracting and enriching ontologies from database schemas

With the development of Web technology, data-intensive Web pages, which are usually based on relational databases, are seeking ways to migrate to the ontology based Semantic Web. This migration calls for reverse engineering of structured data (conceptual schema or relational databases) to ontologies.

There exist many works on reverse engineering on extracting entity-relationship and object models from relational databases. However, there exist only a few approaches that consider ontologies, as the target for reverse engineering [53], [18], [19], [86], [90], [41], [28], [72]... A majority of the work on reverse engineering has been done on extracting a conceptual schema (such as an entity-relationship) from relational databases. The primary focus has been on analyzing key correlation. Data and attribute correlations are considered rarely. Thus, such approaches can extract only a small subset of semantics embedded within a relational database.

To extract additional hidden semantics, these approaches can require, in additional to analysis of conceptual schema, other analysis such as tuples and queries users. Many works exist, in this context.

The method presented in [50] aims to translate a relational model into a conceptual model with the objective that the schema produced has the same information capacity as the original schema. The conceptual model used is a formalization of an extended Entity-Relation model. The method starts transforming the relational schemas into a form appropriate for identifying object structures. Some cycles of inclusion dependencies are removed, and some relation schemas are split.

After the initial transformations, the relational model is mapped into a conceptual schema. Each relation model gives rise to an object type, and the inclusion dependencies give rise to either attributes or generalization constraints, depending on the structure of the keys in each relation schema. The iterations with the user are needed during the translation process. For each candidate key, a user must decide whether it corresponds to an object type of its own, and for each inclusion dependency where both sides are keys, a user must decide whether it corresponds to an attribute or a generalization constraint.

The method presented in [53] uses the database schemas to build an ontology that will then be refined using a collection of queries that are of interest to the database users. The process is interactive, in the sense that the expert is involved in the process of deciding which entities, attributes and relationships are important for the domain ontology.

It is iterative in the sense that the process will be repeated as many times as necessary.

The process has *two stages*. In the *first one*, the database schemas are analyzed in detail to determine keys, foreign keys, and inclusion dependences. As a result of this process a new database schema is created, and by means of reverse engineering techniques, it is content is mapped into the new ontology. In the *second stage*, the ontology constructed from the database schemas has to be refined to better reflect the information needs of the user and can be used to refine the ontology.

This approach [86] proposes to automate the process of filling the instances and their attributes' values of an ontology using the data extracted from external relational sources. This method uses a declarative interface between the ontology and the data source, modeled in the ontology and implemented in XML schema. The process allows the automatization of updating the links between the ontology and data acquisition when the ontology changes.

The approach needs several components: *an ontology* (containing the domain concepts and the relations among them), *the XML schema* (is the interface between data acquisition and the ontology), and an *XML translator* (to convert external incoming relational data into XML when it is necessary).

This approach [90] tries to build light ontologies from conceptual database schemas using a mapping process. To carry out the process, it is necessary to know the underlying logical database model that will be used as source data. The approach has the following five steps to perform the migration process:

1. Capture information from a relational schema through reverse engineering.
2. Analyze the obtained information to built ontological entities by applying a set of mapping rules.
3. Schema translation. In this step the ontology is formed by applying the rules mentioned in the previous step.
4. Evaluate, validate and refine the ontology.
5. Data migration. The objective of this step is the creation of ontological instances based on the tuples of the relational database. It has to be performed in two phases: first, the instances are created, and in the second phase, relations between instances are established.

In [4]I. Astrova proposes a novel approach, which is based on an analysis of key, data and attribute correlations, as well as their combination. The approach is based on the idea that semantics of the relational database can be inferred, without an explicit analysis of the relational schema (i.e. relations, key and non-key attributes, functional and inclusion dependencies), tuples and user queries. Rather, these semantics can be extracted by analyzing HTML forms (both their structure and data) because data is usually represented through HTML forms for displaying to the users. The semantics are supplemented with the relational schema and user "head knowledge" to build an ontology. The approach uses a relational schema and HTML forms as input, and goes through three basic steps: (1) analyzing HTML forms to extract a form model schema that expresses semantics of a relational database behind the forms, (2) transforming the form model schema into an ontology ("schema transformation"), and (3) creating ontological instances from data contained in the forms ("data migration").

In [55] S. Krivine & all present a set of tools which are used for reverse engineering of structured data to OWL ontologies. S. Krivine & all see that the records cannot always be considered as instances of concepts but sometimes they are considered as concepts. These concepts maintain, in particular, subsumption relationships need to make in the ontology. So they proposed an extension of the tool RDBToOnto in order to prevent instantiating records of some database tables and reproduce hierarchies of concepts. The created ontology is guided both by knowledge on the data and also by the use envisaged for the ontology.

In [41], M. M. Hamri and S. M. Benslimane present a semi-automatic approach for ontology enrichment from UML class diagram corpus of a specific domain. The goal of the approach is to extract the most relevant classes in a corpus of UML class diagram on the same domain of conceptualization, in order to subsequently inject them into domain ontology in order to enrich it. This enrichment approach consists of four stages: the selection of relevant classes using techniques usually applied to texts (calculation of frequency), validation, generation of global class diagram, and enrichment of the ontology.

Many works of construction of ontologies from UML model exist. In [41], there is a good state of the art.

10. DISCUSSION AND CONCLUSION

After a deep analysis, we have found that DBs are useful for ontologies and vice versa. In this paper, we have concentrated on the usefulness of DBs for ontologies. The main objective of our paper was to present the drawbacks of ontologies and show how DBs can drawback them essentially performance degradation due to the large quantity of instances. Also, we have talked about the usefulness of ontologies for DBs and identified the main axes related to this domain but the presentation of existing approaches using ontologies in favor of DBs will be a future work.

In this paper, the focus has been made on concepts related to these two concepts (ontology and DB), the difference between the two, major axes on this complementarity and new concepts which emerged as a result of this. After a deep analysis, we found that the collaboration between these two disciplines (DB and ontologies) can be done in five main axes: facilitate the design of DB, increase their semantics for information retrieve, allowing their integration, design ontologies from existing DBs and allow storage of large number of instances in ontological systems. We have focused on areas concerning the use of DBs to overcome the drawback of ontologies. For each axis, we have tried to present the underlying issues and cover the approaches used to solve these problems. The reader with no knowledge of the domain and those who want to know more will find all regarding the complementarity between DBs and ontologies.
A considerable effort has been effected to:

- Identify the axes resulting from the rapprochement between DBs and ontologies (table 5),
- Represent existing approaches using DBs in favor of ontologies
- Explain the concept of DBBO which interest DB community and which is very recent,
- Give the difference between DBBO and structures concerning semantic web community that we have proposed to call OBDB. In the literature, there is confusion between these two concepts, since the birth of DBBO is a consequence of the works done before in the context of semantic web (table1).
- Give the difference between CDM and ontology (table 6).

Drawback	DBs	Ontologies	Complementarity between DBs and ontologies to overcoming drawback
Answers to a book of specification and it can't be reused for another different application need.	×		Design DBs from ontologies
Complexity of the design cycle	×		Design DBs from ontologies
Difficulty of data integration	×		Create ontologies from DBs using reverse engineering
Obligation to know names of entities and their attributes to formulate a query	×		endowed BDs of ontologies to increase their semantic
Difficult to maintain	×		/
Performance degradation due to the large quantity of information		×	Endowed ontologies of DBs to benefit from the functionality of DBMS
Create ontologies from scratch		×	Design of ontologies from DBs

Table 4. drawbacks overcame due to the complementarity between DBs and ontologies.

Axis	Community	Techniques	Sub-axes	Related works
Facilitate the integration of DBs	DB, SW	Mapping, reverse engineering	Alignment of ontologies	[27], [98]. [81], [85], [74], [66], [64], [26], [93].
Management of a large number of instances in an ontology.	SW	DBMS	New query languages and new structures (OBDB).	[79], [24], [59],[1], [2], [61], [16], [32], [36], [63], [80], [73], [81], [58], [51], [52], [33], [101], [42] [23], [29], [48], [95], [102], [14], [91], [97], [22],[16], [105],[49].
facilitate of retrieval of information in DBs	DB	reverse engineering	New query languages and new structures (DBBO)	[17], [31], [3], [47], [106], [48], [76], [77], [78], [79], [23], [24], [49], [29], [65], [102].
Design of ontologies	SW	Mapping, reverse engineering		[4], [41], [55], [90], [86], [53], [50], [20].
Facilitate the design of DBs, from ontologies	DB	Mapping		[99], [13], [11], [98], [54], [74], [Sut 93], [Mic 94], [69], [23], [88], [30], [104].

Table 5. the axes resulting from the rapprochement between DBs and ontologies

	MCD	Ontologie
Relation with applicative needs	Dependent	Independant
More appropriate for	Classic Web	Semantic Web
Meaning of concepts	Contextual	Independent of context
Concepts can be referenced from any environment	No	Yes
consensus for a community	No	Yes
Concepts	Canonic	Canonic and non-canonic
Model	Static and off-line	Dynamic and on-line
The semantic	Is lost during the implementation	Is Preserved during the implementation
Use	Stockage & recherche	Sharing, exchange, reasoning

Table 6. Comparison between CDM and ontology

REFERENCES

[1] R. Agrawal, A. Somani, and Y. Xu: Storage and Querying of E-Commerce Data, Proc. VLDB 2001, Roma, Italy, available as http://www.vldb.org/conf/2001/P149.pdf

[2] Alexaki, S., Christophides,V., Karvounarakis, G., Plexousakis, D., andTolle, K. (2001). The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases. In Proceedings of the 2nd International Workshop on the Semantic Web, pages 1–13.

[3] Ariane Assélé Kama, Giovanni Mels, Rémy Choquet, Jean Charlet et Marie-Christine Jaulent. Une approche ontologique pour l'exploitation de données cliniques. IC 2010, Nîmes : France (2010).

[4] I. Astrova, Reverse Engineering of Relational Databases to Ontologies, In: Proceedings of the 1st European Semantic Web Symposium (ESWS), LNCS 3053 (2004) 327–341.

[5] Asunción Gómez-Pérez, David Manzano-Macho. Deliverable 1.5: A survey of ontology learning methods and Techniques. Universidad Politecnica de Madrid, 2003.

[6] Asunción Gómez-Pérez "Ontological engineering", Ontological Engineering: IJCAI'99.

[7] James Bailey, François Bry, Tim Furche, and Sebastian Schaffert, Web and Semantic Web Query Languages: A Survey, Reasoning Web, 2005 – Springer.

[8] Mustapha Baziz, Nathalie Aussenac-Gilles, Mohand Boughanem, "Exploitation des Liens Sémantiques pour l'Expansion de Requêtes dans un Système de Recherche d'Information » XXIème Congrès INFORSID, 2003.

[9] Mustapha Ba&ziz, Mohand Boughanem, Nawel Nassr, « La recherche d'information multilingue : désambiguïsation et expansion de requêtes basées sur WordNet », Proceedings of the Sixth 2003.

[10] Ladjel Bellatreche, « Intégration de sources de données autonomes par articulation a priori d'ontologies ». Actes du XXIIe Congrès INFORSID, Biarritz, pp 283-298, 2004

[11] Bellatreche l., Xuan d., Pierra g., Hondjack d., « Contribution of Ontology-based Data Modeling to Automatic Integration of Electronic Catalogues within Engineering Databases », Computers in Industry Journal, vol. 57, no 8-9, 2006.

[12] Sidi Mohamed Benslimane and all. « construction automatique d'une ontologie à partir d'une base de données relationnelle: approche dirigée par l'analyse des formulaires HTML. INFORSID'06, Hammamet, Tunisie pp. 991-1010.

[13] S.M. Benslimane, M Malki, D. Bouchiha. "Deriving conceptual schema from domain ontology: A Web application reverse-engineering approach". The International Arab Journal of Information Technology (IAJIT), 7(2):, ISSN 1683-3198, 2010.

[14] D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. In the 3rd International Semantic Web Conference (ISWC 2004).

[15] Borst,W. N. (1997). Construction of Engineering Ontologies. PhD thesis, University of Twente, Enschede.

[16] Broekstra et al., 2002] Broekstra, J., Kampman, A., and van Harmelen, F. (2002). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Horrocks, I. and Hendler, J., editors, Proceedings of the 1st International Semantic Web Conference (ISWC'02), number 2342 in Lecture Notes in Computer Science, pages 54–68. Springer Verlag.

[17] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo,Maurizio Lenzerini, Antonella Poggi, Riccardo Rosati. Ontology-based database access. 2007

[18] Cerbah, F. (2008a) Learning Highly Structured Semantic Repositories from Relational Databases - RDBtoOnto Tool, in Proceedings of the 5th European Semantic Web Conference(ESWC 2008), Tenerife, Spain, June, 2008.

[19] Cerbah, F. (2008b) Mining the Content of Relational Databases to Learn Ontologies with deeper Taxonomies. Proceeding of IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI'08) and Intelligent Agent Technology (IAT'08), Sydney, Australia, 9-12.

[20] Elliot J. Chikofsky, James H. Cross II, Reverse Engineering and Design Recovery: A Taxonomy. IEEE Software, January 1990.

[21] J. Conesa, X. de Palol, A. Olive. "Building Conceptual Schemas by Refining General Ontologies," 14th International Conference on Database and Expert Systems Applications - DEXA '03, Czech Republic, pp. 693-702, 2003.

[22] Hondjack Dehainsala. « Base de données à base ontologique". Proc. du 23ème congrès Inforsid, 2004.

[23] Hondjack Dehainsala, Guy Pierra, Ladjel Bellatreche, Yamine Aït Ameur Conception de bases de données à partir d'ontologies de domaine : Application aux bases de données du domaine technique. In Proceedings of 1ères Journées Francophones sur les Ontologies (JFO'07),pp. 215–230

[24] Hondjack Dehainsala, Guy Pierra, and Ladjel Bellatreche, OntoDB: An Ontology-Based Database for Data Intensive Applications. In Proceedings of the 12th International Conference onDatabase Systems

for Advanced Applications (DASFAA'07), Lecture Notes in Computer Science, pp. 497–508. Springer.

[25] Hondjack Dehainsala, Guy Pierra, and Ladjel Bellatreche, Benchmarking data schemes of ontology based databases. Proc. of Database Systems for Advanced Applications (DASFAA'2007), Bangkok, Thailand, April 9-12, 2007. Lecture Notes in Computer Science Volume 4277, 2006, pp 48-49. Springer.

[26] Dejing Dou, Paea LePendu, "Ontology-based integration for relational databases", SAC '06 Proceedings of the 2006 ACM symposium on Applied computing Pages 461-466.

[27] Gayo Diallo, Une Architecture à Base d'Ontologies pour la Gestion Unifiée des Données Structurées et non Structurées. Thèse de docteur de l'Université. Université Joseph Fourier – Grenoble I, France. Décembre 2006.

[28] G. Dogan and R. Islamaj, Importing Relational Databases into the Semantic Web,http://www.mindswap.org/webai/2002/fall/Importing_20Relational_20Databases_20into_20the_20Semantic_20Web.html (2002).

[29] Chimène Fankam, Stéphane Jean, Guy Pierra, Ladjel Bellatreche. Enrichissement de l'architecture ANSI/SPARC pour expliciter la sémantique des données: une approche fondée sur les ontologies. ACAL 2008.

[30] Chimène Fankam & all, "SISRO : Conception de bases de données à partir d'ontologies de domaine", TSI Volume 28, pages 1 à 29, 2009.

[31] Ines Fayech et Habib Ounalli. « Une Ontologie De Domaine Pour L'enrichissement Sémantique D'une Base De Données ». SETIT 2009. Tunisie.

[32] Richard Fikes, Patrick Hayes, and Ian Horrocks. "OWL-QL – A Language for Deductive Query Answering on the Semantic Web", , ScienceDirect, Elsevier. Dec 2004.

[33] D. Florescu and D. Kossmann. A performance evaluation of alternative mapping schemes for storing xml data in a relational database. Technical Report 3680, INRIA Rocquencourt, France, 1999.

[34] François RASTIER, ONTOLOGIE(S). Revue d'Intelligence artificielle, 2004, vol. 18, n°1, p. 15-40.

[35] H. El-Ghalayini, M. Odeh, R. McClatchey, "Deriving Conceptual Data Models from Domain Ontologies for Bioinformatics," The 2nd international conference on Information and Communication Technologies from Theory to Application ICTTA'06, Damascus, Syria, 2006.

[36] Gregory Karvounarakis and all, RQL: A Declarative Query Language for RDF. www'02 Proceding of the 11th international conference on WWW Pages 592-603.

[37] Michael Gruninger and Jintae Lee. Ontology applications and design. Communications of The ACM February 2002/Vol. 45, No. 2

[38] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220.

[39] Tom Gruber. "Toward principles for the design of ontologies used for knowledge sharing", 1995.

[40] N. Guarino, "Formal ontology and information systems," In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS'98), Ternto, Italy, pp. 3-15, 1998.

[41] M. M. Hamri, S. M. Benslimane, Vers une approche d'enrichissement d'ontologie à base de schémas conceptuels. Journées francophones sur les ontologies (JFO), December 2-4, 2009, Poitiers, France.

[42] Harris, S. and Lamb, N. and Shadbolt, N. 4store: The design and implementation of a clustered RDF store. 5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009), pages 94-109, 2009.

[43] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. Available from http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.

[44] M. Jarrar, J. Demy, R. Meersman , "On Using Conceptual Data Modeling for Ontology Engineering", Journal on Data Semantics, LNCS 2800, Springer, No. 1, pp. 185-207, 2003.

[45] Stéphane Jean, Guy Pierra and Yamine Ait-Ameur. « OntoQL: An exploitation language for OBDBs ». (2005).VLDB Ph.D. Workshop, 41–45.

[46] Jean, S., Aït-Ameur, Y., and Pierra, G. Querying ontology based database using OntoQL (an Ontology query language). In Proc. of Ontologies, DataBases, and Applications of Semantics. (2006).

[47] Jean, S., G. Pierra, et Y. Aït-Ameur (2007). Domain Ontologies : a Database-Oriented Analysis, Volume 1 of Lecture Notes in Business Information Processing, pp. 238–254. Springer Berlin Heidelberg.

[48] Stéphane JEAN, « OntoQL, un langage d'exploitation des bases de données à base ontologique », doctorat thesis, 2008, France.

[49] Stéphane Jean, Yamine Aït-Ameur, Guy Pierra "A Language for Ontology-Based Metamodeling Systems" Advances in Databases and Information Systems. Lecture Notes in Computer Science Volume 6295, 2011, pp 247-261.

[50] Johannesson P. (1994) A Method for Transforming Relational Schemas into Conceptual Schemas. In 10th International Conference on Data Engineering, Ed. M. Rusinkiewicz, pp. 115 -122, Houston, IEEE Press, 1994.

[51] G. Karvounarakis. A Declarative RDF Metadata Query Language for Community Web Portals. Master's thesis, University of Crete, 2000.

[52] G. Karvounarakisa, A. Magganarakia, S. Alexakia, V. Christophidesa, D. Plexousakisa, M. Schollb, K. Tolle. Querying the Semantic Web with RQL. In Computer Networks, Volume 42, Issue 5, August 2003, Pages 617–640.

[53] Kashyap, V. (1999). Design and Creation of Ontologies for Environmental Information Retrieval. Twelfth Workshop on Knowledge Acquisition, Modelling and Management Voyager Inn, Banff, Alberta, Canada. October, 1999.

[54] Kim W., Seo J., « Classifying Schematic and Data Heterogeneity in Multidatabase Systems », Computer, vol. 24, no 12, 1991, p. 12–18, IEEE Computer Society Press.

[55] S. Krivine & all. « Construction automatique d'ontologie à partir de bases de données relationnelles: application au médicament dans le domaine de la pharmacovigilance ». IC2009. Hammamet, Tunisie.

[56] Laallam Fatima Zohra, Modélisation et gestion de la maintenance dans les systèmes de production. doctorat thesis, 2007, Annaba, Algeria.

[57] M. Lenzerini. Data integration: A theoretical perspective. In Proc. of PODS 2002, pages 233–246, 2002.

[58] J. Liljegren. Description of an RDF database implementation. Available at. http://infolab.stanford.edu/~melnik/rdf/db.html

[59] Liu, B. and Hu, B. HPRD: a high performance RDF database, International Journal of Parallel, Emergent and Distributed Systems. Vol. 25. N°2, pages 123-133, 2010.

[60] Lorensen W., Rumbaugh J., Blaha M., Premerlani W., Eddy F. Modélisation et conception orientées objets. Masson edition, 1997. 516 pages.

[61] Luo, Y. and Picalausa, F. and Fletcher, G.H.L. and Hidders, J. and Vansummeren, S. Storing and indexing massive rdf datasets. Journal of Semantic Search over the Web, Springer. Pages 31-60, 2012.

[62] A. Magkanaraki et al. Ontology Storage and Querying, Technical Report No. 308, April 2002, Foundation for Research and Technology Hellas, Institute of Computer Science, Information System Laboratory.

[63] S. Melnik. Storing rdf in a relational database. Available at http://WWW-DB.stanford.edu/~melnik/rdf/db.html. Last modification: Dec 3, 2001.

[64] Mena E, Illarramendi A, Kashyap V, Sheth A (2000) OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. International journal of Distributed and Parallel Databases (DAPD). Vol. 8. Number 2. Pages: 223-271. ISSN 0926-8782. April, 2000.

[65] Mizoguchi-Shimogori Y., Murayama H., Minamino N., "Class Query Language and its application to ISO13584 Parts Library Standard", In 9th European Concurrent Engineering Conference, ECEC 2002, Modena, Italy, 2002, p. 128-135.

[66] S. Mustière & all, " Appariement de schémas de BD géographiques à l'aide d'ontologies déduites des spécifications ", Workshop EGC 2007, pp 23-28.

[67] Natalya F. Noy and Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

[68] Natalya F., "Semantic integration: a survey of ontology-based approaches", ACM, Volume 33 Issue 4, December 2004, pp 65 – 70.

[69] NAVAS-DELGADO I., ALDANA-MONTES J. F., « A Design Methodology for Semantic Web Database-Based Systems », ICITA '05 : Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2,Washington, DC, USA, 2005, IEEE Computer Society, p. 233–237.

[70] N. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. Knowledge and Information Systems, 2002.

[71] N. Noy, M. Klein, "Ontology Evolution: Not the same as Schema Evolution," Knowledge and Information Systems, vol. 6 no. 4, pp. 428-440, 2004.

[72] Nyulas C. O'connor, M. TU, S. (2007), Data Master – a Plug-in for Importing Schemas and Data from Relational Databases into Protégé, (http://protege.stanford.edu/conference/2007/presentations/10.01_Nyulas.pdf).

[73] Pan, Z., and Heflin, J. Dldb: Extending relational databases to support semantic web queries. In Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems (2003).

[74] Christine Parent & Stefano Spaccapietra, « Intégration de bases de données: Panorama des problèmes et des approaches », Ingénierie des systèmes d'information Vol.4, N°3, 1996.
[75] Pavel Shvaiko and Jérôme Euzenat, “A Survey of Schema-Based Matching Approaches" Journal on Data Semantics IV, Lecture Notes in Computer Science, 2005, Volume 3730/2005, pp 146-171
[76] Pierra, G., Context-explication in conceptual ontologies: Plib ontologies and their use for industrial data. Journal of Advanced Manufacturing Systems (2004)
[77] Pierra, G., H. Dehainsala, Y. A. Ameur, L.Bellatreche, J. Chochon, and M. Mimoune (2004). Base de Données à Base Ontologique : le modèle OntoDB. Proceeding of Base de Données Avancées 20èmes Journées (BDA’04), 263–286.
[78] Guy Pierra and Hondjack Dehainsala and Yamine Ait Ameur and Ladjel Bellatreche« Base de données à base ontologique: Principe et mise en œuvre », Ingénierie des systèmes d’information, Hermès, 2005.
[79] Guy Pierra Yamine AIT Ameur, Ladjel Bellatreche, H. Dehainsala, S. Jean, C. Fankam, D. N'guyen Xuan ; « Données à base ontologique gestion, interrogation, intégration », 1ère journée francophone sur les ontologies, Sousse, 2007
[80] Edit. E. Prud’hommeaux, A. Seaborne. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. http://www.w3.org/TR/rdf-sparql-query/
[81] Raphael Squelbut, Olivier Curé, “Integrating data into an OWL Knowledge Base via the DBOM Protégé plug-in”, 9th International Protégé Conference July 24-26, 2006, Stanford University, USA.
[82] Alberto Reggiori, Dirk-Willem van Gulik, Zavisa Bjelogrlic. Indexing and retrieving Semantic Web resources: the RDFStore model. in Proc. of the SW AD-Europe Workshop on Semantic Web Storage and Retrieval, 2003.
[83] Riichiro Mizoguchi. « Le rôle de l’ingénierie ontologique dans le domaine des EIAH ». STICEF, Volume 11, 2004.
[84] Robert Stevens, Carole A. Goble and Sean Bechhofer, “Ontology-based Knowledge Representation for Bioinformatics”, Brief Bioinform. 2000 Nov;1(4):398-414.2000.
[85] Rousset MC, Reynaud C: Knowledge representation for information integration. Inf. Syst. 2004, 29:3-22.
[86] Rubin D.L., Hewett M., Oliver D.E., Klein T.E., and Altman R.B. (2002). Automatic data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. In: Proceedings of the Pacific Symposium on Biology, Lihue, HI, 2002 (Eds. R.B. Altman, A.K. Dunker, L. Hunter, K. Lauderdale and T.E. Klein).
[87] Sherif Sakr and Ghazi Al-Naymat. Relational Processing of RDF Queries: A Survey.
[88] Sandra Geisler and all. “Ontology-based system for clinical trial data management”. IEEE Benelux EMBS Symposium, 2007. Cette reference n’existe pas sur le document
[89] J.J.Solari.’OWL Web Ontology Langage Use Cases and Requirements’. http://www.yoyodesign.org/doc/w3c/webont-req-20040210/
[90] Stojanovic, L.; Stojanovic, N.; Volz R. Migrating data-intensive Web Sites into the Semantic Web. Proceedings of the 17th ACM symposium on applied computing (SAC), ACM Press, 2002, pp.1100-1107.
[91] K. Stoffel, M. Taylor, J. Hendler. Efficient Management of Very Large On-tologies.In Proceedings of American Association for Artificial Intelligence Conference (AAAI-97), AAAI/MIT Press 1997.
[92] Kingshuk Srivastava, P.S.V.S.Sridhar, Ankit Dehwal. “Data Integration Challenges and Solutions: A Study”. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 7, July 2012.
[93] Heiner Stuckenschmidt and all. Enabling technologies for interoperability. In Ubbo Visser and Hardy Pundt, editors, Workshop on the 14th International Symposium of Computer Science for Environmental Protection, pages 35–46, Bonn, Germany, 2000.
[94] Tardieu H., Rochfeld A. and Colletti R. La méthode Merise Principes et outil. 352 pages, Eyrolles 2000.
[95] THEOHARIS Y., CHRISTOPHIDES V., KARVOUNARAKIS G., « Benchmarking Database Representations of RDF/S Stores. », Proceedings of the Fourth International Semantic Web Conference (ISWC’05), 2005, p. 685-701.
[96] Vallée F., Roques P. UML en action - De l'analyse des besoins à la conception en Java. 387 pages, Eyrolles edition, 2000.
[97] R. Volz, D. Oberle, B. Motik, S. Staab. KAON SERVER -A Semantic Web Management System. In Proceedings of the 12th World Wide Web, Alternate Tracks - Practice and Experience, Hungary, Budapest. 2003.
[98] WACHE H. and all « Ontology-based Integration of Information - A Survey of Existing Approaches," In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, Vol.

pp. 108-117.
[99] Dr. Waralak V. Siricharoen Ontology Modeling and Object Modeling in Software Engineering. International Journal of Software Engineering and Its Applications Vol. 3, No. 1, January, 2009.
[100] Ye, Y. and Ouyang, D. and Dong, X. Persistent Storage and Query of Expressive Ontology. Third International Conference on Knowledge Discovery and Data Mining, 2010. WKDD'10, pages 462-465.
[101] Zhou S. Exposing relational database as rdf. 2nd International Conference on Industrial and Information Systems (IIS), 2010, pages 237-240.
[102] ZIEGLER P., STURM C., DITTRICH K. R., « Unified Querying of Ontology Languages with the SIRUP Ontology Query API. », Proceedings of Business, Technologie and Web (BTW'05), 2005, p. 325–344.
[103] Ziegler and Klaus R. Dittrich. "Data Integration–Problems, Approaches, and Perspectives". Conceptual Modelling in Information Systems Engineering, pages 39–58.Springer, Berlin, 2007.
[104] Veda C. Storey, Debabrata Dey, Harald Ullrichc, Shankar Sundaresand, "An ontology-based expert system for database design". Data & Knowledge Engineering,Volume 28, Issue 1, 30 October 1998, Pages 31–46.
[105] Edd Dumbill. "Putting RDF to Work". Article on XML.com. August 09, 2000.(http://www.xml.com/pub/a/2000/08/09/rdfdb/)
[106] Jean, S., Aït-Ameur, Y., and Pierra, G. Querying ontology based database using OntoQL (an Ontology query language). In Proc. of Ontologies, DataBases, and Applications of Semantics. (2006).

AUTHORS BIOGRAPHIES

1st author: Fatima Zohra LAALLAM received the B.S. degree in computer science from the Badji Mokhtar university, Annaba, Algeria, in 1991, and the M.Sc. and the Ph.D. degrees in computer science from the Badji Mokhtar university, Annaba, Algeria, in 2003 and 2007, respectively. She is a Professor within the Department of Mathematics and Computer Science, Kasdi Merbah university, Ouargla, Algérie. Her research interests include image processing, databases, and ontologies. She is a member and a cofounder of the LAGE laboratory and the GRAIN research group (Ouargla, Algeria).

2nd author: Mohammed Lamine Kherfi received the B.S. degree in computer science from the Institut National d'Informatique, Algeria, in 1997, and the M.Sc. and the Ph.D. degrees in computer science from the Université de Sherbrooke, Canada, in 2002 and 2004, respectively. Between 1997 and 2000, he was the Head of the General Computer Services Agency. Currently, he is a Professor within the Department of Mathematics and Computer Science, Université du Québec à Trois-Riviéres, Canada. His research interests include image and multimedia retrieval, computer vision, image processing, machine learning, and web-based applications. He has published numerous refereed papers in the field of computer vision and image processing, and holds an invention patent. He has served as a Reviewer for numerous journals and conferences. He is a member and a cofounder of the LAMIA laboratory (Trois-Rivières, Canada) and the GRAIN research group (Ouargla, Algeria).

Dr. Kherfi is the recipient of numerous scholarships and awards from the Université de Sherbrooke between 2000 and 2005, as well as from the Institut National d'informatique between 1992 and 1997. He received the FQRNT excellence postdoctoral fellowship in 2005.

3rd author: Sidi Mohamed Benslimane is a Associate Professor at the Computer Science Department of Sidi Bel Abbes University, Algeria. He received his PhD degree in computer science from Sidi Bel Abbes University, in 2007. He also received a M.S. and a technical engineer degree in computer science in 2001 and 1994 respectively from the Computer Science Department of Sidi Bel Abbes University, Algeria. He is currently Head of Research Team 'Service Oriented Computing' at the Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS. His research interests include, semantic web, web engineering, ontology engineering, information and knowledge management, distributed and heterogeneous information systems and context-aware computing.