

Improving Geodatabase Semantic Querying Exploiting Ontologies

Miriam Baglioni^{1,*}, Maria Vittoria Masserotti², Chiara Renso²,
and Laura Spinsanti^{3,**}

¹ IIT, CNR, Via Moruzzi 1, 56100, Pisa, IT
baglioni@iit.cnr.it

² KDDLab, ISTI, CNR, Via Moruzzi 1, 56100, Pisa, IT
{masserotti,renso}@isti.cnr.it

³ IES, JRC, European Commission, Ispra, IT
laura.spinsanti@jrc.ec.europa.eu

Abstract. Geospatial semantic querying to geographical databases has been recognized as an hot topic in GIS research. Most approaches propose to adopt an ontology as a knowledge representation structure on top of the database, representing the concepts the user can query. These concepts are typically directly mapped to database tables. In this paper we propose a methodology where the ontology is further exploited mapping axioms to spatial SQL queries. The main advantage of this approach is that semantic-rich geospatial queries can be abstractly represented in the ontology and automatically translated into spatial SQL queries.

Keywords: semantic geospatial query, ontologies, spatial database, spatial SQL.

1 Introduction

The exponential growth of positioning technologies coming from the widespread use of GPS personal devices, for example embedded in smart phones, tend to produce a huge amount of geographical referred data. This, in turn, calls for methods and techniques capable of managing and querying this large quantity of geo-referenced data. Moreover many non-expert users are becoming aware of the huge potentiality of geographic information in their everyday life. While data management recent developments in spatial and spatio-temporal databases converge towards some well-defined proposals [OraSpa, PostGIS, Guting05], the query capabilities in terms of semantically enhanced user query language have not produced so far any standardized approach. In this context, proposals to facilitate the access to geographical data to the non-expert user may range from advances graphical user interfaces and visual analytics to natural language processing. In the latter approaches having support for semantic geospatial queries is essential and usually these proposals employ an ontology [Gru08] as a knowledge representation level on top of a spatial database. The objective of the ontology layer is to explicitly represent the concepts the user can mention in the query.

* The work was carried out at Dep. Computer Science, Univ. of Pisa, 56100, Pisa, IT.

** The work was carried out at LBD Lab EPFL, EPFL-IC-LBD, CH-1015 Lausanne, CH.

In the last decade, ontologies have gained increasing interest in the GIS community [Mark06, Fonseca02, Spaccapietra04], because they can be used to create and exploit data standards as well as human computer interfaces and to solve heterogeneity/interoperation problems. The relevant literature exploits ontologies to map data sources to explicit ontology concepts [Bishr08] or in geographical information retrieval techniques [Mata09]. The use of ontologies as a middle layer between the user and the database adds a conceptual and semantic level over the data, and allows the user to query the system on semantic concepts without having any specific information about the database at hand [Peuquet02]. Despite of the efforts to create sharable data, many Geographical Information Systems (GIS) evolve from numeric cartography integrating remote sensing and digital images, typically skipping any design and modeling phase. Therefore, quite often they lack both metadata and the conceptual schema, thus losing part of the semantic geographical information. An effect of having an ontology layer linked to a geographical database is to increase not only the number of (geographical) concepts the user can mention in the query - including all the ontology concepts that are abstractions of the database tables - but also the “abstraction level” in term of semantic definitions. To better clarify this point let us consider for example a database storing tables of urban objects such as schools, hospital, universities. Assume that these tables are mapped to urban ontology concepts and subsumed in the ontology by a *Building* concept representing all the spatial objects in the domain which are buildings. This means allowing the user to mention the Building concept in the query abstracting away from the specific kind of building such as schools, hospitals etc.

Given this scenario the main contribution of the present work is to map to the database not only the ontology classes but also the ontology concepts defined by *axioms*, the ontology implicit formal concept definitions. For example, the concept *FloodRiskBuildings*, may define the class of buildings which are *inside* a flood risk area. When these axioms represent spatial relations (in a given form, see Section 4) we provide a method to automatically translate them into spatial database materialized views. The consequence is that the user may query the database not only in the stored tables, but also in complex spatial queries abstractly represented in the ontology and stored as materialized views.

The proposed methodology intends to exploit the already existing domain ontologies where concepts are implicitly defined by axioms, as a conceptual representation of spatial SQL queries. Hence, the semantics of the underlying geographical database is enhanced by explicitly representing information which is implicit in the data (such as the spatial relations) and which can be retrieved by specific spatial SQL queries. We automatically map these ontology axioms into appropriate spatial SQL queries stored as materialized views in the database. We believe that natural language processing techniques may benefit from the introduced methodology since there is a need to have a formal definition of the concepts that may be mentioned in the query. This further increases the expressive power of the query system: in fact, it increases the number of concepts the user can query with respect to the classical ontology database mapping.

Paper is organized as follows. Section 2 presents the approach, while Section 3 shows in details the semantic enhancing process. Then, Section 4 presents the algorithm and Section 5 the related work. Finally, Section 6 draws some conclusions and future work.

2 The Overview of the Approach

In this section we give the overall view of the proposed approach, firstly introducing the main idea of the method with a simple example, then describing some background concepts and presenting the methodology.

The problem statement is given in an urban scenario describing hospitals, schools, and flood risk areas of the city stored in a spatial database, as depicted below:

Table 1. The Flood Risk example

HOSPITAL			
ID	Name	The_geom	Helicopter
231	Careggi	POLYGON(10.589,47.779,....,WGS84)	Y
254	S.Maria	POLYGON(10.579,47.783,....,WGS84)	N

SCHOOL				
ID	Name	Classes	Category	The_geom
45	L.daVinci	52	Secondary	POINT(10.583,47.775,....,WGS84)
83	Mazzini	15	Primary	POINT(10.577,47.789,....,WGS84)

FLOOD RISK BASIN		
ID	Code	The_geom
26	AZ34	POLYGON(10.581,47.749,....,WGS84)
42	3SX9	POLYGON(10.579,47.782,....,WGS84)

Then consider the natural language query: *Which are the public buildings in a flood risk area?* The information to answer this query is actually stored in the spatial database, since we have information about hospitals and schools (which are public buildings) as well as the flood risk areas. However, public buildings have no correspondent database table where to issue a spatial SQL query. This means a semantic interpretation step has to be applied to the query: the user must query the HOSPITAL and SCHOOL tables and, from the resulting records, she/he has to select the ones which have an *intersect* spatial relationship with the flood risk basin of the city. Obviously, to express these queries the user needs to have the knowledge of both the database structure and the appropriate spatial operator to apply, depending on the type of the referred spatial objects.

The general objective of the present work is to introduce a methodology to support the automatic translation of spatial semantic information represented in an ontology into spatial SQL. The knowledge contained in the ontology can be of two kinds. First, knowledge is organized in a taxonomy of geographical places so that the more general concepts are represented as top concepts, compared to the more specific concepts, correspondent to database tables, represented as “leaves” of the taxonomy. Secondly, we assume some concepts are defined as a result of a spatial relation between two spatial objects and formalized as axioms. The central point of the work is to show that it is possible to map geographic places of interest, conceptually expressed by ontology axioms, to a query over the actual data stored in a geographical database. The advantage of using an ontology rely on the fact that the user can query the data without having the knowledge of the structure of the underlying database, or the spatial SQL syntax.

2.1 Background

Ontologies have certainly become a research topic in several disciplines, ranging from philosophy, geography, geomatics up to machine learning and artificial intelligence. The definition given by [Gru08] is used to define ontology as “a technical term denoting an artifact that is designed for a purpose, which is to enable the modeling of knowledge about some domain, real or imagined”. Such ontologies determine what can be represented and what can be inferred about a given domain, using a specific formalism of concepts. Formally, an ontology is a 5-tuple $O := \{C, R, HC, rel, A0\}$, where C is a set of concepts, which represent the entities in the ontology domain; R is a set of relations defined among concepts; HC is a taxonomy or concept-hierarchy, which defines the *is_a* relations among concepts ($HC(C1, C2)$ means that $C1$ is a sub-concept of $C2$), $rel: R \rightarrow C \times C$ is a function that specifies the relations on R . Finally, $A0$ is the set of axioms expressed in a logical language, such as first order logic.

Ontology languages are formal languages used to construct ontologies. They allow the encoding of knowledge about specific domains and often include reasoning facilities that support the processing of that knowledge. Among all the ontology languages, we considered Web Ontology Language (OWL), that is a well known standard arisen from the Semantic Web and it is now a W3C recommendation [OWL]. An interesting feature of OWL is that it relies upon a family of languages known as Description Logics (DL) that provide a deductive inference system based on a formal well founded semantics [Baad03]. The basic components of DL are concepts (classes) and roles (properties), termed as TBox, and individuals (instances), termed as ABox. Concepts describe the common properties of a collection of individuals and roles are binary relations between concepts. Furthermore, a number of language constructs, such as intersection, union and role quantification, can be used to define new concepts and roles, by means of axioms.

2.2 Methodology

Our approach is based on the use of an ontology to represent the domain of interest and as a middle layer between the user and the data. The connection between the ontology and the database can be done by an explicit mapping between database tables and ontology concepts using a correspondence table. The mapping of concepts to database tables can be done in several ways, such as manually, or automatically extracting the ontology from the database [Baglioni07, Vital09]. The ontology – to – database mapping is deeply covered in the literature, and details of the different approaches can be found for example in [Barrasa04, An06, Li05].

Let us assume to have a geographical database containing the following tables: Hospital, School, Street, Square, PedestrianArea, CityCenter, River and FloodRisk Basin. And let us suppose to have the domain described by the ontology in Figure 1. As a matter of terminology, we refer to the following definitions to describe the possible ontology-to-database mappings:

- **Direct:** There is a one to one correspondence between an ontology concept and a database table. Considering the database tables listed above, and the ontology of Figure 1, there is direct correspondence for the ontology concepts Hospital, School, Street, Square, PedestrianArea, CityCenter, River and FloodRiskBasin.

- Indirect:** The concept is an antecedent in the hierarchy (a super-class) of a concept that has a direct connection to the database. Considering the database tables listed above, and the ontology of Figure 1, it is possible to indirectly refer the PublicBuilding, District, Place, Building, UrbanObject and GeographicObjects concepts.

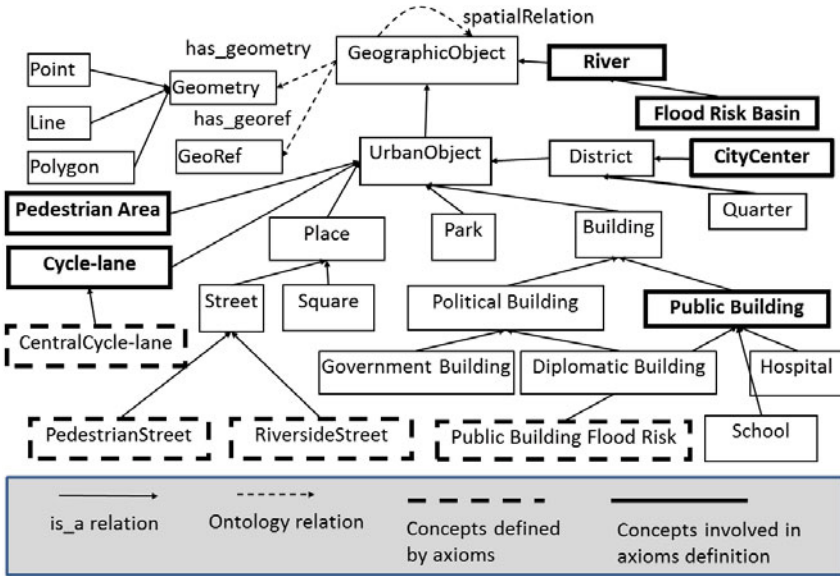


Fig. 1. The Urban Ontology

- Implicit:** The concept is defined by an axiom and it is not directly connected to the database. In the example it is possible to implicitly refer the PublicBuilding FloodRisk, PedestrianStreet, and RiversideStreet concepts.

The main contribution of this paper is a methodology to automatically map implicit concepts to SQL queries. Axioms are expressed in OWL DL describing properties the entities must have to belong to the concept.

In the ontology in Figure1, boxes in dotted line represent the concepts defined by axioms. The form of axioms we are assuming here use a spatial relation between two geographical objects to define a new concept. For example, the dotted boxes in Figure 1 are defined as follows:

- Public Building Flood Risk* be the Public buildings located *inside* the Flood Risk basin
- PedestrianStreet* be a Street located (at least partially) *inside* the PedestrianArea
- RiversideStreet* be a Street located along the River (within a distance of maximum 200m)
- CentralCycle-lane* be the Cycle-lane that intersects the CityCenter.

These descriptions can be formalized in OWL DL axioms having the general form:

SpecializationConcept \equiv *Concept1* and (*spatialRelation* some *Concept2*)

Again, consider the concept *PublicBuildingFloodRisk* that can be expressed combining *PublicBuilding* with *FloodRiskBasin* through the spatial property *intersect*. As an OWL axiom, this can be formalized as:

PublicBuildingFloodRisk \equiv *PublicBuilding* and (*intersect* some *FloodRiskBasin*).

A schema of the approach is illustrated in Figure 2, where the ontology is analyzed by an *Enhancing Module* in order to discover the axioms that can be mapped to SQL queries. The *Enhancing Module*, given a domain ontology and a geographical database, produces two outcomes: a number of SQL queries (stored in the DBMS as materialized views) which are derived by ontology axioms and a new *DataBase Ontology* (DBOntology). The DBOntology is obtained from the original domain ontology selecting only the concepts that have a direct, indirect and implicit connection to the database. Therefore, the DBOntology acts as a knowledge representation structure which is the basis of the query module, representing all the concepts that the user can mention in the query. The query module could be a complex natural language module (like in [Baglioni08, Baglioni09]) or a simpler interface depending on the application. The query module details are not reported in this paper.

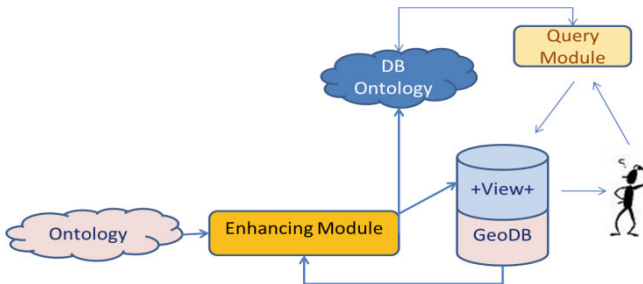


Fig. 2. The proposed methodology

An example of DBOntology resulting from the application of the *Enhancing Module* to the ontology of Figure 1 and the database tables listed above, is reported in Figure 3. Here, the white solid rectangles represent the classes of the ontology that define the spatial topological relations. The spatial relations considered here are inspired by the OpenGIS [OGIS] standard and are: *equals*, *disjoint*, *touches*, *within*, *overlaps*, *crosses*, *intersect*, *contains*. These relations are modeled in the ontology as sub-properties of *spatialRelation* (see Figure 1). The dark solid rectangles represent the classes of the ontology that have either a *direct* or an *indirect* mapping to the database. The white shattered rectangles represent concepts that have an *implicit* mapping to the database.

From Figure 3, we can notice that *Public Building* is a super-class of the two concepts *Hospital* and *School*, which, in turn, have a direct correspondence to database tables. The *Public Building Flood Risk* concept is defined by an axiom stating that a flood risk public building is a kind of *PublicBuilding* that *intersects* the

FloodRiskBasin. Since FloodRiskBasin has a direct map to a database table, the axiom related to the concept PublicBuildingFloodRisk can be translated to a database query. In the same way, Riverside Street and PedestrianStreet concepts can be mapped to spatial SQL queries. It is worth noticing that some concepts of the domain ontology have no mapping with the data (i.e Cycle-lane, DiplomaticBuilding, Park) therefore they will not appear in the final DBOntology. For the same reason the CentralCycle-lane concept, defined by an axiom, is excluded by the algorithm because it refers to the concept Cycle-lane that is not represented in the database.

As mentioned above, we base our approach upon OWL DL, a W3C standard arisen from research on the Semantic Web based on Description Logics [Baad03]. However, the kind of axioms that we consider have a specific simple form to ensure an easy translation into spatial SQL. Indeed, mapping OWL into SQL is a complex task object of research since many years and it is out of the scope of the paper presenting all these approaches (see for example [Acci05, Calvanese09]). The main difference among literature approaches and the present work is that we use OWL axioms only from a *syntactical* point of view in order to translate into the appropriate SQL query. We are not interested in the *reasoning* aspects of the ontology and axioms. The OWL fragment we consider here has not relations to the inference power of the reasoning engine.

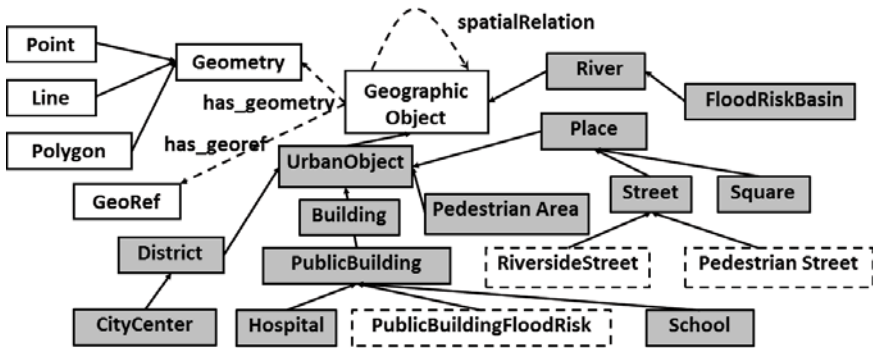


Fig. 3. The DBOntology where only the concepts mapped to the database are represented

More in general, we can say that allowed axioms define specialization classes by restricting the spatialRelation property (or its sub properties) with the “exists” operator (also called *some* in OWL). Indeed, this logical expression may be mapped to a SQL SELECT statement involving the spatial database topological relations. We explicitly refer to a spatial relation which can be immediately one-to-one mapped in a spatial operator.

In the next section we introduce the details of the semantic enhancing process with an exhaustive explanation of all the possible cases which can be found and processed.

3 The Semantic Enhancing Process

The Enhancing module is responsible for both the selection of the domain ontology classes that can be solved as an SQL query, the production of the corresponding

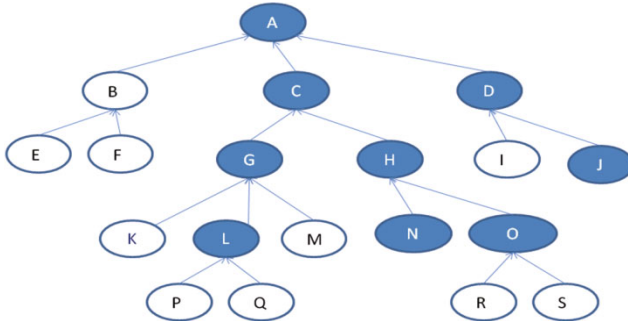


Fig. 4. An example of a generic ontology schema *Onto*

materialized views and contextually, the construction of the *DBOntology*. We illustrate the process by means of an example.

Consider a generic ontology schema, indicated as *Onto*, shown in Figure 4. Dark circles represent the ontology concepts *directly or indirectly* connected to the database, whereas white circles represent concepts *implicitly* connected to the database. Concepts that are not directly, indirectly, or implicitly mapped to database tables are not depicted in the figure. The enhancing process firstly considers all the concepts in *Onto* that have a *direct* connection to the database. They correspond to the set of concepts {J, L, N, O}. Among them, we select their direct children and sibling that have an *implicit* mapping, shown in Figure 4 in white color. Hence, the selection of the concept O will cause the selection of the nodes R and S since both are direct children of O. The concept L has both siblings and children and will cause the selection of the concepts {K, M, P, Q}. The concept N does not have any direct children, and the only sibling is already part of the dark concepts of *Onto* (direct and indirect mapped), so nothing else is selected. Therefore, the set of the concepts in *Onto* selected in this first step as potentially mapped to SQL queries is {I, K, M, P, Q, R, S}.

In the next step the enhancing process checks if these selected concepts can be associated to materialized views. To do this we must ensure that: (i) the axiom that defines the concept can be expressed by a spatial SQL query and (ii) the concepts defined by axiom are already in the *DBOntology* which has been built.

The *DBOntology* is built iteratively, and at the first step it is composed of all the dark concepts in *Onto*. The Enhancing Module checks the white concepts (implicit mapped), and when axioms are defined in terms of a spatial relation with a concept belonging to the *DBOntology* iteratively built so far, this new concept incrementally becomes part of the new *DBOntology*. This process is recursive and finite since the number of considered concepts is finite. This process results in the construction of the *DBOntology* that becomes the knowledge representation structure mapped to the spatial database where both direct, indirect and implicit connection to the database is explicitly represented.

We can define formally this process as follows.

Given a domain Generic Ontology $O_G=(C_G,R_G,rel_G,H_G,A_G)$, the *DBOntology* $O_{DB}=(C_{DB},R_{DB},rel_{DB},H_{DB},A_{DB})$ is defined as follows:

- The set of ontology concepts is defined recursively starting from the concepts in C_G so

$$C_{DB}^0 = C_G$$

$$C_{DB}^n = C_{DB}^{n-1} \cup \{c \in C_{DO} \mid \exists c' \in \text{Leaf}(C_G) \wedge (h_{DO}^{-1}(c) = h_{DO}^{-1}(c') \vee H_{DO}(c, c')) \wedge c' \neq c \wedge \exists a \in A_{DO}. c \in \text{Conc}(a) \wedge \forall c'' \in \text{Conc}(a). c'' \neq c \wedge c'' \in C_{DB}^{n-1}\}$$

where

 - C_{DO} are the concepts in *Onto*,
 - $\text{Leaf}(C) = \{c \in C \mid \neg \exists c' \in C. H(c, c')\}$,
 - $h^{-1}: C \rightarrow C. h^{-1}(c) = c' \Leftrightarrow H(c', c)$
- $A_{DB} = \{a \in A_{DO} \mid \forall c \in \text{Conc}(a). c \in C_E \wedge \forall r \in \text{relations}(a). r \in \text{SpatialRelation}\}$

where

 - A_{DO} is the set of axioms in *Onto*
 - $\text{Conc}(a) = \{c \in C \mid a \in A, c \in a\}$ is the set of concepts that define the axiom a
 - $\text{relations}(a) = \{r \in R \mid a \in A, r \in a\}$ is the set of relations that define the axiom a
 - *SpatialRelation* is the set of the considered spatial relation
- $R_{DB} = R_G$
- $\text{Rel}_{DB} = \text{rel}_G$
- $H_{DB} = H_G \cup \{(c, c') \mid c \in C_G \wedge c' \in C_{DB} \wedge h^{-1}(c') = c\}$

For each new concept added to the *DBOntology*, a materialized view representing the SQL query correspondent to the axiom defining the concept is added to the database.

As far as the axiom translation into materialized views is concerned, we distinguish different cases, depending on the structure of the selected node. There are three cases: the concept has siblings, the concept has children, and the concept has both siblings and children. We give the intuition of each case with an example, whereas the algorithm is illustrated in the next section.

Case A – Siblings. Consider the *PublicBuildingFloodRisk* concept that does not have a direct link to the database tables, and its siblings classes, *Hospital* and *School*, are directly associated to database tables. The axiom translation results in the creation of a materialized view whose data are the union of the subset of the *Hospital* and *School* records located in the flood risk areas.

Reminding that the OWL axiom defining the *PublicBuildingFloodRisk* concept is *PublicBuildingFloodRisk* \equiv *PublicBuilding* and (*intersect some FloodRiskBasin*), the associated SQL view is:

```
CREATE VIEW C PublicBuildingFloodRisk as
  (SELECT * FROM Hospital h, FloodRiskBacinhc
   WHERE intersect(h.the_geom, hc.the_geom)
   UNION1
   SELECT * FROM School b, FloodRiskBacinhc
   WHERE intersect(b.the_geom, hc.the_geom))
```

¹ Notice that the use of the UNION operator to include data selected from both tables.

Case B – Children. Let us consider the ontology depicted in Figure 5A and let PedestrianStreet be defined as a Street whose geographical coordinates are located within a PedestrianArea. The correspondent axiom is defined as $PedestrianStreet \equiv (Street \text{ and } (within \text{ some } PedestrianArea))$. Similarly, suppose RiversideStreet is defined as the streets whose geographical coordinates have a *touch* relation with a river and the correspondent axiom be $RiversideStreet \equiv (Street \text{ and } (touch \text{ some } River))$. In this example, the concept Street is directly mapped to the database. The axiom translation creates three new materialized views that: (1) select the records of the Street table in a *within* relationship with a PedestrianArea and (2) the records of the Street table located in a *touch* relationship with the River. An additional materialized view, called ComplementStreet, is created to collect the records of the table Street which are not included in the two created views. As a consequence, the union of the records belonging to the three views corresponds to the original Street table, but now we can distinguish between the three cases. The corresponding DBOntology is depicted in Figure 5B. The SQL statement for the creation of the Complement class is the following

```
CREATE VIEW ComplementStreet as
SELECT * FROM Street
WHERE not exist (select * from PedestrianStreet)
AND not exist (select * from RiversideStreet)
```

Case C – Merge siblings and children. The selected concept has both siblings and children. Both the previous steps are applied in this case.

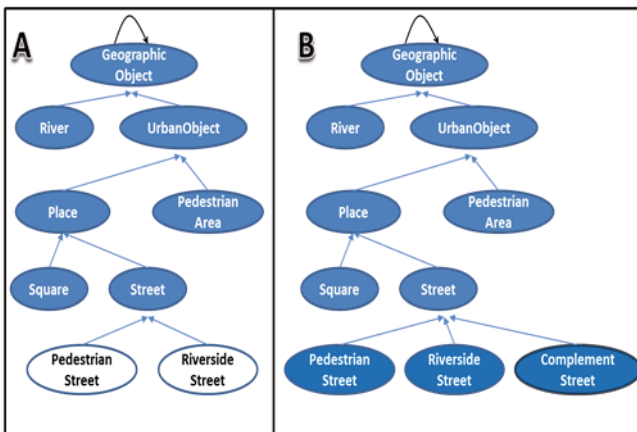


Fig. 5. A) fragment of the urban ontology. PedestrianStreet and Riverside street are define by axioms and not mapped to the database B) Fragment of the corresponding DBOntology. Street concept has been mapped to the database by means of its children PedestrianStreet and Riverside Street. A new concept complement has been added to map the remaining records.

4 The Algorithm of the Enhancing Module

In this section we present the details of the Enhancing Module algorithm. The algorithm is structured in two main phases. The first phase consists in (1) selecting all the domain ontology concepts with a *direct* mapping to database tables, and (2) selecting all the concepts in the domain ontology that are siblings or children of the concepts found at point 1). We call the resulting ontology the DBOntology at step zero (DBO₀). The leaves of this ontology are the concepts mapped directly to database tables. The second phase consists in expanding, as much as possible, the DBO₀ with the input domain ontology concepts defined by axioms, following the rules presented in previous section. At each step n , the algorithm selects all the Domain Ontology concepts that are direct children or siblings of the selected DBO_{n-1} leaf, when they are not included in the DBO_{n-1}. From these concepts the algorithm selects only the ones whose axioms can be solved by an SQL query. This means that we must be sure that all the concepts mentioned in the axioms belong to DBO_{n-1}. In other words these concepts are directly associated to database tables or already materialized as views. In the latter case the algorithm creates the corresponding SQL views, inserting the new classes with a *is_a* relationships in the DBO_{n-1} thus producing the DBO_n. The process stops when the set of concepts to be materialized is empty, or there is no change from one step to the next one (in this case the remaining concepts will be discarded). The DBO_n constructed by the process is then returned as the final DBOntology.

Let us go through the algorithmic part. Let us recall the formal definition of the ontology given in Section 2.2, and introduce some basic definitions and acronyms used through the algorithm. DBO indicates the DBOntology, DO the Domain Ontology, CA stands for Concept Axiom indicating the set of Domain Ontology concepts defined by axioms and implicitly mapped to database tables. Finally CC indicates ConceptComplement, of the concept to be complemented. A number of predefined functions are used, such as *DefiningConcept DC(c)*: is a function which, given a concept defined by an axiom c , returns the set of concepts mentioned in the axiom. *hO* indicates the Hierarchy of the ontology O and *hO(c)* returns the set of children of c , whereas *hO-I(c)* returns the concept father of c . Eventually, *Leaf(O)* is a function that, given an ontology O , returns the set of its leaves. Analogously, *Node(O)* returns the set of node concepts in the ontology O . The function *has_axiom(c)* checks whether a given concept is defined by an axiom. The pseudo-code is illustrated below.

The Enhancing algorithm

```
//Selection of the possible nodes for the enhancing process
CA = {}
Leaf = Leaf(DBO) //takes the DBO leaves
Repeat
  c ∈ Leaf
  c' = hdo-1(c) // takes the father of the node
  Leaf \ {c}
  forall c'' ∈ hdo(c').c'' <> c // the set of c's
```

```

//siblings in DO
    if has_axiom(c'') then
//If the concept is defined by an axiom an does not belong to
the set of DBO nodes and it is not already present in the set of
leaves, it is added to the Caset//
        if not ((c'' ∈ Nodes(DBO)) or
            (c'' ∈ Leaf)) then
            CA ∪ {c''}
forall c'' ∈ hdo(c) // the set c's children
    if has_axiom(c'') then
        CA U{c''}
until Leaf = {}
//Selection of the concept to be materialized, construction of
the materialized view, update of the DBO, and update of the DB
if needed
Repeat
    CA_tmp=CA
    Forall c ∈ CA.
//takes the concept needed to define the new class
    c' = DC(c) \ hdo-1(c)
    if c' ∈ DBO then
        CA\{c}
        Create_view(c)
        add_to_DBO(c)
        c'' = hdo-1(c)
        if c'' ∈ Leaf(DBO) and not c'' ∈ CC then
            CC U {c''}
Until CA = CA_tmp
For all c ∈ CC Remove_and_Complement(c)

```

The definition of the function *add_to_DBO(c)* follows:

```

add_to_DBO(c) =
Leaf = Leaf(DBO)
Repeat
    c' ∈ Leaf
    Leaf \ {c'}
// the father of concept c' in the HC relation
    c'' ∈ hdo-1(c')

```

```

//c is sibling of c' if  $HC_{DO}(c, c')$  holds
  if  $HC_{DO}(c, c')$  then
} //adds the new concept to the set of DBO concepts
   $C_{DBO} \cup \{c\}$ 
//adds the father relationship in DBO
   $HC_{DBO} \cup \{(c, c')\}$ 
//c is child of c' if  $HC_{DO}(c, c')$  holds
  If  $HC_{DO}(c, c')$  then
     $C_{DBO} \cup \{c\}$ 
     $HC_{DBO} \cup \{(c, c')\}$ 
until  $HC_{DO}(c, c')$  or  $HC_{DO}(c, c')$ 

```

The function *create_view(c)* builds the materialized views as an output of the Enhancing Module. In this code we introduce new functions 1) *concept_name(c)* returns the name of the database table the concept *c* is mapped on, 2) *Axiom(c)* returns the axiom that defines the concept *c*, 3) *get_predicate(axiom)* returns the spatial predicate contained in the axiom, 4) *Leaves_O(c)* returns all the leaves children of the concept *c* in the ontology *O*. Furthermore, the function *DBConnection* provides the connection to the spatial DB and executes the query. The *create view(c)* function definition follows.

```

create_view(c) =
//takes the father of c and the other concept in the Enhanced
Ontology needed to define c.
Conc = DC(c)
// takes the set of concepts the father can be rewritten in
ExpFa = {}
 $\forall c' \in Conc$ 
  if  $HC_{DO}(c, c')$  then
    father = c'
//verify if father has to be expanded
  if not isMappedToTable(father) then
    Leaves = LeavesDBO(father)
    For all  $c \in Leaves$ 
      If isMappedToTable(c) then
        ExpFa  $\cup$  {c}
      Else
        ExpFa = {father}
    Else
//takes the leafs of the concept c'
  Leaf = LeavesDBO(c')

```

```

//production of the query for each Leaf related to the axiom
    Q = {}
    Forall f ∈ ExpFa
        Forall c ∈ Leaf
            Q ∪ {"SELECT * FROM
                " &concept_name(f) & "AS a, "
                &concept_name(c) & "AS b
                WHERE " &
                get_predicate(Axiom(c)) &
                " (a.the_geom, b.the_geom)"}
//creation of the view
query = empty string
q ∈ Q
query = "CREATE VIEW " &concept_name(c) &
        " VIEW as ( " & q
Q\{q}
For all q ∈ Q
    query = query & "UNION " & q
query = query & ")"
//connection to the DB and view materialization
DBConnection(query)

```

Finally, the definition of the `remove_and_complement(c)` function follows:

```

Remove_and_complement(c)=
//selection of the children of c in the extended
// enriched ontology
Leaf = LeavesDBO(c)
query = "CREATE VIEW Complement" &concept_name(c)
        & " as SELECT * FROM " &concept_name(c) &
        " WHERE not exist "
l ∈ Leaf
query = query & "((SELECT * FROM " &
                concept_name(l) & ") "
Leaf\{l}
Forall l ∈ Leaf
    query = query & " AND not exist
        (SELECT * FROM " &concept_name(l) & " ) " &
query = query & ")"
DBConnection (query)
DBConnection("DROP TABLE " &concept_name(c))

```

5 Related Work

The partnership between ontologies and GIS has seen a growing interest in the last decade [Fonseca02, Mark06, Vidal09, Zaki09]. The role of ontologies in geographical information science can be manifold. A well known topic is Geographical Information Retrieval (GIR), where the ontology supports spatial querying, as witnessed for example by the results of the SPIRIT project [Jones05], where methods for ontology-based spatial query expansion for geographical search engines were studied. For example, in this context, [Cardoso07] proposes a method for the geographical expansion of queries exploiting spatial relationships. Another work [Mata07] proposes the use of an ontology to decide where and what should be searched from different data sources.

Several approaches share with our work the use of ontology for querying geographical databases. However, to the best of our knowledge, no approaches explicitly map ontology axioms to spatial SQL materialized views.

A recent approach [Peachavanish07] proposes a methodology that exploits multiple ontologies for the interpretation of geospatial queries. Compared to our approach, they propose a mediation between ontologies that we are not considering, and, in general, their approach is more conceptual and not based directly on querying database tables. The work in [Torres05] proposes an ontological semantic layer to query a geographical database and in this is similar to our proposal. In particular, their approach allows different community users to access the same geographic database. However, they do not consider specifically the problem of representing the location of a geographical object, neither the translation from ontology axioms to spatial SQL queries which we are handling here.

Similar in the objective, to our approach, the work in [Lüscher08] that aims at enriching the semantics of geodatabase for enhanced user queries. However, compared to the proposed methodology, the authors propose a complementary approach, since they infer semantic information about spatial objects exploiting pattern recognition techniques, that we are not considering here.

The work of [Zhao08] shares with our approach the use of ontologies as a query interface towards spatial data, but the focus there is on data integration and they don't consider the use of a domain ontology to further enhance the geospatial semantics of queries, neither they use a mapping of ontology axioms to spatial SQL queries.

The problem of mapping between ontologies and relational databases is faced by a broad range of literature works. Some example – but the list is not exhaustive, are [Barrasa04, An06, Li05] however, it is out of the scope of the paper to provide a survey on them. Some of these approaches particularly tackle the problem of translating axioms into rules such as [Vasilecas06, Vasilecas07]. However the main differences between these approaches and ours rely on the fact that they do not deal with the spatial domain, and the axioms are not translated in (spatial) SQL materialized views, but in business [Vasilecas06] or ECA rules [Vasilecas07] represented as database triggers. Several other approaches have their roots in the database integration, and they typically map mapping Description Logics formulas to SQL, such as [Levy99] or [Calvanese08].

6 Conclusions and Future Work

In this paper we propose a methodology for mapping ontology axioms to SQL queries on a geodatabase. This approach enhances the usual ontology – to – database mapping,

providing support for an automatic translation of ontology axioms to a spatial SQL query. The advantage of this proposal mainly relies on the possibility to support the automatic (or semi-automatic) translation of natural language spatial queries into appropriate database queries stored as materialized views. As a consequence, this methodology increases the number of (ontology) concepts that may be referred in the user queries thus semantically enhancing the query process to a geodatabase.

The advantage of the proposed approach is many-fold: at first it allows to abstract away the domain concepts with respect to the underlying database, thus allowing to reuse the same domain ontology with different database in the same domain. Secondly, we can enhance the user query capabilities exploiting ontology concepts that do not have a classical direct mapping to the database. Finally, the ontology is expressed in a standard Semantic Web technology, such as OWL: it makes easier the use of different domain ontologies and permits different semantical interpretation of the same geographical database.

Some interesting open issue we intend to explore in the future are related to the limitations that we have posed to the present solution. For example, relaxing the limitations on the form of the axioms and generalizing the use of nested relations and object properties.

References

- [Acci05] Acciarri, A., Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Palmieri, M., Rosati, R.: QuOnto: Querying ontologies. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence, AAAI 2005 (2005)
- [An06] An, Y., Borgida, A., Mylopoulos, J.: Discovering the Semantics of Relational Tables through Mappings. In: Proc. of the 21th Nat. Conf. on Artificial Intelligence, AAAI 2006 (2006)
- [Baad03] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The description logic handbook: Theory, implementation and applications. Cambridge University Press, Cambridge (2003)
- [Baglioni07] Baglioni, M., Masserotti, M.V., Renso, C., Spinsanti, L.: Building Geospatial Ontologies from Geographical Databases. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 195–209. Springer, Heidelberg (2007)
- [Baglioni08] Baglioni, M., Giovannetti, E., Masserotti, M.V., Renso, C., Spinsanti, L.: Ontology-supported Querying of Geographical Databases. Transactions of GIS 12(s1), 31–44 (2008)
- [Baglioni09] Baglioni, M., Masserotti, M.V., Renso, C., Soriano, L., Spinsanti, L.: A Tool for Extracting Ontologies from Geographical Databases. In: SEBD 2009, Convegno Italiano di Basi di Dati, Camogli, Genova, Italy (2009)
- [Barrasa04] Barrasa, J., Corcho, O., Gómez-Pérez, A.: R2O, An Extensible and Semantically Based Database-to-Ontology Mapping Language. In: Second Workshop on Semantic Web and Databases, SWDB 2004 (2004)
- [Bishr08] Bishr, Y.: The Geospatial Semantic Web: Applications. In: Shekhar, S., Xiong, H. (eds.) Encyclopedia of GIS. Springer, Heidelberg (2008)
- [Calvanese08] Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: View-Based Query Answering over Description Logic Ontologies. In: KR 2008, pp. 242–251 (2008)

- [Calvanese09] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: Ontologies and Databases: The DL-Lite Approach. In: Tessaris, S., Franconi, E., Eiter, T., Gutierrez, C., Handschuh, S., Rousset, M.-C., Schmidt, R.A. (eds.) Reasoning Web. LNCS, vol. 5689, pp. 255–356. Springer, Heidelberg (2009)
- [Cardoso07] Cardoso, N., Silva, M.J.: Query Expansion through Geographical feature Types. In: GIR 2007, Lisboa, Portugal, November 9 (2007)
- [Fonseca02] Fonseca, F.T., Egenhofer, M.J., Agouris, P., Camara, C.: Using Ontologies for Integrated Geographic Information Systems. *Transact. GIS* 6(3), 231–257 (2002)
- [Gru08] Gruber, T.: Ontology. In: Liu, L., Özsu, M.T. (eds.) *Encyclopedia of Database Systems*. Springer, Heidelberg (2008)
- [Guting05] Güting, R., Schneider, M.: *Moving Objects Databases*. Morgan-Kaufman, San Francisco (2005)
- [Levy99] Logic-Based Techniques in Data Integration. In: Levy, A., Minker, J. (eds.) *Logic-based Artificial Intelligence* (1999)
- [Li05] Li, M., Du, X., Wang, S.: Learning Ontology from Relational Database. In: *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, August 18-21 (2005)
- [Lüscher 08] Lüscher, P., Weibel, R., Mackaness, W.A.: Where is the Terraced House? On the Use of Ontologies for Recognition of Urban Concepts in Cartographic Databases. In: Ruas, A., Gold, C. (eds.) *Headway in Spatial Data Handling*, pp. 449–466. Springer, Heidelberg (2008)
- [Mark06] Mark, D.M., Egenhofer, M.J., Hirtle, S., Smith, B.: *UCGIS Emerging Research Theme: Ontological Foundations for Geographical Information Science* (2006)
- [Mata07] Mata, F.: Geographic Information Retrieval by Topological, Geographical, and Conceptual Matching. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) *GeoS 2007*. LNCS, vol. 4853, pp. 98–113. Springer, Heidelberg (2007)
- [Mata09] Mata, F., Levachkine, S.: iRank: Integral Ranking of Geographical Information by Semantic, Geographic, and Topological Matching. In: *IF&GIS 2009*, pp. 77–92 (2009)
- [OGIS] OpenGIS Simple Feature Access,
<http://www.opengeospatial.org/standards/sfa>
- [OraSpa] ORACLE, Oracle Spatial,
<http://www.oracle.com/database/spatial.html>
- [OWL] OWL Web Ontology Language, <http://www.w3.org/TR/owl-features/>
- [Peachavanish07] Peachavanish, R., Karimi, H.A.: Ontological Engineering for Interpreting Geospatial Queries. *Transactions in GIS* 11(1), 115–130 (2007)
- [Peuquet02] Peuquet, D.: *Representations of Space and Time*. The Guilford Press, N.Y. (2002)
- [PostGis] PostGres database– PostGIS extension,
<http://postgis.refractions.net/>
- [Spaccapietra04] Spaccapietra, S., Cullot, N., Parent, C., Vangenot, C.: On Spatial Ontologies. In: *GeoInfo* (2004)
- [Jones05] Fu, G., Jones, C.B., Abdelmoty, A.I.: Ontology-Based Spatial Query Expansion in Information Retrieval. In: Chung, S. (ed.) *OTM 2005*. LNCS, vol. 3761, pp. 1466–1482. Springer, Heidelberg (2005)
- [Torres05] Torres, M., Quintero, R., Moreno, M., Fonseca, F.T.: Ontology-Driven Description of Spatial Data for Their Semantic Processing. In: Rodríguez, M.A., Cruz, I., Levashkin, S., Egenhofer, M.J. (eds.) *GeoS 2005*. LNCS, vol. 3799, pp. 242–249. Springer, Heidelberg (2005)

- [Vasilecas06] Vasilecas, O., Bugaite, D.: Ontology-Based Information Systems Development: The Problem of Automation of Information Processing Rules. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 187–196. Springer, Heidelberg (2006)
- [Vasilecas07] Vasilecas, O., Bugaite, D.: An algorithm for the automatic transformation of ontology axioms into a rule model. In: International Conference on Computer Systems and Technologies. ACM International Conference Proceeding Series, vol. 285 (2007)
- [Vidal09] Vidal, V.M.P., Sacramento, E.R., de Macêdo, J.A.F., Casanova, M.A.: An Ontology-Based Framework for Geographic Data Integration. In: Heuser, C.A., Pernul, G. (eds.) ER 2009. LNCS, vol. 5833, pp. 337–346. Springer, Heidelberg (2009)
- [Zhao08] Zhao, T., Zhang, C., Wei, M., Peng, Z.: Ontology-Based Geospatial Data Query and Integration. In: Cova, T.J., et al. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 370–392. Springer, Heidelberg (2008)
- [Zaki09] Zaki, C., Servières, M., Moreau, G.: Combining Conceptual and Ontological Models for Representing Spatio-Temporal Data and Semantic Evolution in GIS. In: Ontologies for urban development: Future development of urban ontologies, Liège, Belgium, pp. 95–104 (2009)