On the Definition of Generic Multi-layered Ontologies for Urban Applications

Djamal Benslimane, Eric Leclercq, Marinette Savonnet, Marie-Noëlle Terrasse, Kokou Yétongnon

Université de Bourgogne,
LE2I Equipe Ingénierie Informatique,
BP 47870, 21078 DIJON Cedex France
e-mail : {Djamal.Benslimane,Eric.Leclercq,Marinette.Savonnet,Marie-Noelle.Terrasse,Kokou.Yetongnon}@u-bourgogne.fr

Abstract

Cooperation of information systems is essential for providing decision support for urban management applications. This involves sharing data across collections of the heterogeneous information systems that are used to manage large urban infrastructures. The objective of this work is to define a spatial ontology to describe key features of urban applications, providing a foundation for semantic reconciliation among heterogeneous spatial information sources. We propose a multi-layered ontologies definition framework consisting of ontology layers which are composed of a generic functional structure and one or more domain ontologies. The functional structure embodies general ontological concepts described as abstract data types. The domain ontologies are created by specializing the properties and constraints of the functional structure. Inter-ontologies relationships are defined to integrate information across functional ontological layers and used to query multiple domain ontologies.

1 Introduction

Interoperability is essential for many urban management applications and decision support systems. It involves sharing and reusing data from various heterogeneous information systems that are used to manage urban infrastructures, ranging from transportation systems to electric power, telecommunication and railroad networks. Cooperation among these information systems is required to provide support for applications in which decision making involves accessing and combining information from multiple heterogeneous sources. For example,

Preprint submitted to Elsevier Preprint

planning road repair works typically require discovering and collecting relevant information to determine the impact of the proposed project on existing resources in the location of the projected work.

The development of interoperable information systems has been plagued with major problems including: 1) conflicts arising from the data models and types used to represent traditional and spatial information (Laurini 1998, Devogele et al. 1998); 2) semantic discrepancies among components that are designed, managed and operated independently (Sheth 1999); and 3) lack of tools to allow integrated access to shared information (Včkovski 1998). Exchanging and sharing information require providers and receivers of data to agree on what is the meaning of the information and on what is the specification of the operations that are used to process it. This can be done by defining a reference context or set of terms on which designers can carry out reconciliation of semantic discrepancies. Often the semantics or contextual information (including design assumptions and undocumented data types) associated with information sources is not explicitly specified in database schemas, leading to incorrect interpretation and use of the content of information sources. Several methods can be used to make the semantics of an information system explicit, including meta-attributes, textual documentation, meta-data and ontologies.

There are several approaches for enabling information exchange and sharing among diverse systems, resulting in the introduction of different methods for designing interoperable information systems, particularly federated databases. The most basic is to translate one information system (both schema and data) to another system. Early federated GIS use this method, relying on vendor specific tools to carry out data translations. Sometimes common data formats are used to provide reference semantics to minimize information loss. One approach is to use standards (data and processing services) to achieve interoperation. For example in the GIS realm, OpenGIS consortium (OpenGIS 1996) defines data types and functionalities to allow information sharing among spatial systems. Another approach is to use integration techniques to merge collections of information sources into federated databases (Sheth & Larson 1990). Schema-based federated databases use global federated schema to integrate local information systems while language-based federated systems provide interoperation through extended query languages that are capable of accessing and querying remote systems (Litwin et al. 1990). Finally mediation-based interoperation (Tomasic et al. 1998, Bishr 1998) uses mediators to reconcile semantic differences of local information systems.

Objective and contributions

The research cited above has shown the importance of semantic reconciliation in allowing data sharing among heterogeneous information systems. Despite this extensive research effort on the interoperability of traditional information systems, spatial information interoperation and cooperation, especially ontology-driven systems and applications, have receive much less attention. In this paper we focus on ontology-based interoperation of spatial information systems. As part of the project ISIS (Leclercq et al. 1999) — an ongoing research project on semantic interoperability at the University of Bourgogne — we have developed a methodology for defining ontologies for spatial information systems and applications.

Ontologies are emerging as an important tool for constructing sharable and reusable knowledge repositories and supporting their interaction. This importance stems from the fact that ontologies define common representation terms that provide mutual understanding of an application domain among groups of users. They describe concepts and relationships that a group of information systems can use as a semantic basis on which they can communicate and exchange data. For example, a data provider can use the terms of a shared ontology to describe its objects, allowing a potential data receiver to properly interpret the semantics associated with the data provider's content. Likewise, a data receiver can use a shared ontology to specify its requests and interpret returned results. Moreover, ontologies allow formal and declarative descriptions of the common terms, allowing for automatic or semi-automatic reasoning on shared data of a domain.

Our main concern in defining a spatial ontology is to allow dynamic construction of domain (or application) ontologies to represent common semantics of GIS, particularly urban management information systems.

We address ontology-based semantic reconciliation problems from a more general point of view, focusing not on defining the content of a fixed ontology for a specific spatial information system, but on providing an architecture and the corresponding generic ontologies that a designer can specialize in order to describe domain specific applications. The key features of the approach are: 1) it provides a multi-layered ontologies definition framework in which each layer consists of a generic functional structure that can be instanciated to define domain ontologies; 2) it uses abstract data types (ADT) to specify the generic functional structure of the ontology layers. The abstract data types are ontological concepts which can be specialized to define concepts and relations for domain ontologies; and 3) it uses inter-ontologies relationships to allow integration of information from the different functional systems of the multi-layered ontologies.

The remainder of the paper is organized as follows. Section 2 presents the motivations and architecture for ontology-based systems. Section 3 discusses background and related issues of ontologies. An overview of the methodology for defining multi-layered ontologies for urban information systems is presented in section 4. In section 5, several examples are given to illustrate using the methodology to specify an ontology for urban management applications. Finally, section 6 concludes the paper.

2 Motivation : architecture for ontology-based applications

Figure 1 shows an architecture for supporting urban management applications, with several information sources and a shared ontology. The heterogeneous data sources include GIS and traditional databases. The GIS are used to model information related to 1) roads and traffic in a given urban area; 2) water pipes and sewage systems; and 3) power plants, gas and electric networks. One traditional database contains information on land use and ownerships and the other includes data on health districts and hospitals locations. The shared ontology defines the common terms of the application domain. To express the semantics of their objects, each spatial data source defines ontology mapping information between local object descriptions and the semantic descriptions in the ontology.

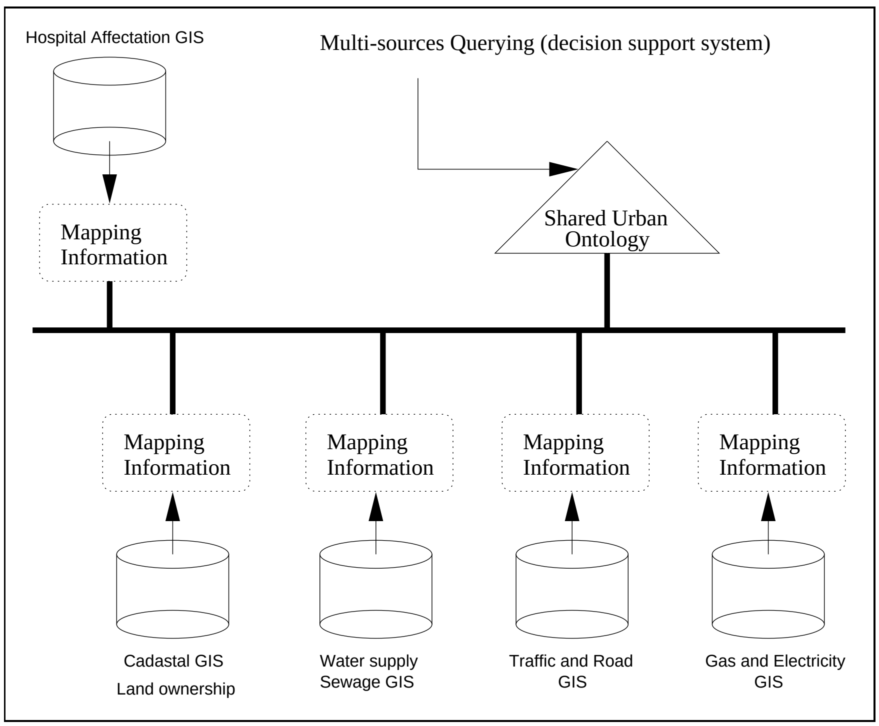

Fig. 1. Consolidation of various information from different data sources

To illustrate some of the issues involved in ontology-based cooperation, consider a scenario where there is a need to construct a new hospital in a given health district. The pre-planning process involves choosing an appropriate location for the new hospital and evaluating the overall cost and impact of the new project on neighboring residential and commercial neighborhoods. The planning staff needs 1) to access the GIS to retrieve relevant information on maps of the locations of existing hospitals, roads accessibility, and electric power and water needs; 2) to combine information from different sources (traditional and spatial) to estimate the costs of existing (land, houses and buildings) properties that must be expropriated in the selected area to build the hospital.

To carry out the pre-planning decisions and draft an initial economic impact plan, two types of operations are required. First, semantic search operations can be used to locate the GIS that match the semantics of the query. The ontology allows the identification of local sources that contain relevant information on parcels, roads, water pipes, sewers and electric plants. The ontology provides support for identifying local data sources that can match the semantics of the query. Second, object retrieval operations can be used to extract relevant objects and convert them to their ontological representation. This involves using the knowledge stored in the ontology to determine the spatial operations that are allowed on the retrieved spatial objects.

Ontology-based interoperation of spatial information systems (including urban management applications) is motivated by the following properties:

- **Precise description of data and resources.** Each information source uses standardized terms of the ontology and its inherent semantics to provide a formal description of its data. Queries based on the agreed upon semantics are less prone to misinterpretation of local information semantics.
- **Support for information localization.** Sharing information requires that a user discovers and locates the relevant data he or she needs. A shared ontology can be used to provide support for the discovery process and to implement the required data access and translation tools.
- **Dynamic support for multiple contexts or interpretation of data.** Traditional schema-based database integration of systems requires costly updates to accommodate new semantics. Using the terms of an ontology as meta-constructs or meta-attributes allows proper dynamic interpretation of the different contexts.
- **Query content dependent interoperation.** Ontologies allow a dynamic interoperation in which the content and the context of queries are interpreted with respect to the ontology to limit exploration of remote sources to those that have information that are consistent with the context of the query.

3 Background

Early research on ontologies (Guarino 1995) in artificial intelligence has focused to a large extent on what ontologies are and how they can be represented. Ontologies are used to identify terms (a vocabulary) that represent the core features of an application domain. They represent common semantics of the domain. Gruber defines an ontology as an explicit specification of a conceptualization, that is an abstract and simplified representation of real world entities (Gruber 1993). An ontology also provides a formal representation for the concepts of an application and the relationships among them, thus capturing the intended meaning of the terms of the domain of interest.

To represent ontologies, different models have been identified. Informally, an ontology can be represented by a classification of terms. Natural language-like descriptions are used to give the intended meaning of the terms. Formally, different models have been used to describe ontological terms including:

- KIF (Knowledge Interface Format) (Genesereth & Fikes 1992) is based on full predicate logic with lisp syntax. It is intended for portable ontology and does not provide for inference. Ontologies specified in KIF can be translated in other languages like OQL and LOOM by using Ontolingua Translator (which is built on top of KIF) (Farquhar et al. 1996, Fikes et al. 1997).
- Terminological models like LOOM (MacGregor 1988), CLASSIC (Borgida et al. 1989) use terms (concepts and roles) to represent domains. Concepts are classes of objects in the domain and roles are binary relationships between objects. Concepts can be created from existing terms via operators on roles and concepts. Classification of the terms is carried-out based on generalization relationships between terms.
- RDF (Resource Description Framework) (RDF 1998) is a data model and support mechanism for representing meta-data of schemas. It is a graph-based data model using a class system organized in hierarchies, grouped into schema typically authored for a specific purpose (Namespace). Each element is identified by exactly one namespace. It uses XML (XML 1998) language (eXensible Markup Language) to exchange and process meta-data between different applications. RDF allows the description and exchange of meta-data schema but doesn't provide tools to facilitate their construction.

Ontologies have been defined and used in several domains to provide data conversion and understanding. We present below a brief review of some of the relevant approaches. In (Weinstein 1998), Weinstein uses a formal ontology model to describe bibliographic relations. The proposed ontology is used to generate a knowledge-base of meta-data from a sample of the MARC (MAR n.d.) description standard. The model used to implement the ontology is a variant of the description logic model (LOOM). The values and attributes from the MARC records are mapped into LOOM instances.

In the domain of multi-agent systems, Jones (Jones 1998) defines shared ontology to represent message passing between agents. The terms of the ontology specify the meaning of concepts that intelligent agents use to exchange queries and results. Two categories of terms compose the ontology: primitive terms are undefined while non-primitive terms are defined by other terms (primitive or non-primitive). To reduce semantic conflicts, the primitive terms are selected from standard vocabularies which are top-level ontology, resulting in two-level shared ontologies defined over a top-level ontology. In the domain of medical applications, Gennari et al in (Gennari et al. 1995) propose a different approach of ontology definition. The focus is not on specifying the content of an ontology, but on providing tools to support collaborative development and construction of a common vocabulary. They define a web-based ontology server to allow browsing and editing of a common controlled vocabulary for medical applications. The model used to represent the ontology is Ontolingua.

The next two approaches are from spatial information systems domain. Wariyapola et al (Wariyapola et al. 1999) develop an ontology and a meta-data model for distributed spatial systems. It can be used to locate, retrieve and visualize information about coastal ocean environment. The focus of this work is on identifying issues related to the definition of meta-data. The model used is an object-oriented model that allows the authors to combine three existing meta-data standards in an extensible environment called Warwick framework. In addition, web-based tools are provided to facilitate the design of the shared ontology. In the second approach, Coenen et al (Coenen et al. 1996) present an ontology for spatial reasoning using a tesseral representation of space. The solution is based on (tesseral) address systems which is used to linearize space to allow single dimension reasoning on a small set of terms. The authors propose three types of ontologies related to three basic concepts, namely tesseral address, spatial objects and constraints. Tesseral address is the low-level ontology. Its terms are used to specify the spatial ontology. And finally topological constraints specify relations between spatial objects.

4 Overview of a methodology for defining a multi-layered ontology

The justification for a multi-layered ontology for urban applications centers on viewing them as systems that integrate multiple abstraction layers, each representing generic spatial functionalities. In this section, we present the multi-layered spatial ontology, its key requirements and the relationships between the layers of the ontology.

Gruber's definition of formal ontologies provides a theoretical foundation for describing domains. A conceptualization of a domain depends on thematic points of view and the abstraction process used to represent the real world.

The design of ontologies for interoperable urban information systems must take into account variations in the views (conceptualizations) of an application domain modeled by different information systems. These views may vary in levels of detail or the meaning associated with the terms that are used to represent domains. An ontology therefore can provide a reference semantics or basis on which the information systems can reconcile differences when conflicts arise in their views of an application domain.

The underlying principles and key features of the multi-layered spatial ontology described in this work are:

- **Levels of ontologies:** In (Guarino 1997), Guarino classifies ontologies by considering two criterias: level of detail and level of dependence. The level of detail of the ontology determines how close it is to the intended meaning of the vocabulary. Very detailed ontologies contain large number of explicit meanings while simple ontologies contain a reduced number of generic terms that can be expanded by implicit rules that are accepted and understood by a community of users. The level of dependence determines whether an ontology is defined for a task or a general domain. Several types of ontologies can be distinguished: 1) top-level ontologies consist of general concepts independent of a particular domain or task; 2) domain ontologies describe vocabularies relevant to a generic domain; 3) task ontologies are relevant to a particular task; and 4) application ontologies are composed of concepts derived from upper level ontologies.
- **Multi-layered functional view of GIS:** Spatial information systems are often characterized as integrated systems that combine different functionalities including data storage, databases capabilities and specific spatial processing and operations. As a result, they can be viewed as comprising several abstract layers, each defining a generic set of functionalities. For example, Voisard (Voisard & Schweppe 1994, Voisard & Schweppe 1998) proposes a design methodology in which multiple layers are used to abstract GIS operations and provide processing services for other layers. Urban information systems which typically require interoperation of heterogeneous information systems can also be described by thematic layers that represent different urban infrastructures. For instance, infrastructures such as highway networks, transportation systems, electric power networks can be described by a generic functional layer based on graph terminology and graph traversal operations. Similarly, two-dimensional spatial objects (surface) can be used

to define a generic functional layer to represent and manage land occupation, buildings, parks, zoning districts of a city.

- **Generic and sharable ontology:** There exists a significant amount of standard vocabulary and meta-data format for describing spatial information. Reusing these meta-information aims to avoid duplicating the effort and cost invested in their development. To be shared by a large community of users, an ontology must be constructed collaboratively by the intended users. This requires a trade off between 1) defining a large extensive ontology containing all possible concepts and relationships of a domain and 2) a generic ontology comprising a reduced number of concepts that a large community can agree on. The generic ontology is more feasible than the large ontology and can provide general core concepts and functionalities for generating domain specific vocabularies, which can be done by specializing the concepts and relations of the generic ontology, but requires a clear specification of the intended semantics associated with the terms of the ontology and the rules for deriving other concepts.

4.2 The multi-layered ontologies

To meet the above design requirements, we draw upon the work of Guarino to define a multi-layered ontologies for semantic interoperability of spatial information systems. It is a functionality-based solution consisting of a set of inter-related ontologies layers, each corresponding to a specific spatial functional abstraction. When a layer is viewed individually as a spatial processing domain, it exhibits a set of specific features, functionalities and semantics. Inter-ontology relations are used to represent semantic connections between the layers. These relations are used to map concepts in one layer to one or more concepts in other layers.

Figure 2 depicts the general architecture of multi-layered ontologies. It includes a top level ontology and one or more ontology layers. The top-level ontology which is similar to the one defined by Guarino in (Guarino 1997) represents general concepts including time, person or address that are common to several functional domains. For example, the spatial concept localization can model different coordinate systems such as latitude/longitude, XYZ coordinates, polar coordinates, relative location, and postal addressing localization system.

An ontology layer is organized in two levels:

- **The functional level** corresponds to a high level abstract view of the operations (functionalities) of the ontology layer. Typically, it is a generic ontology represented by an abstract functional structure consisting of high level ontological concepts and corresponding abstract functional descrip-

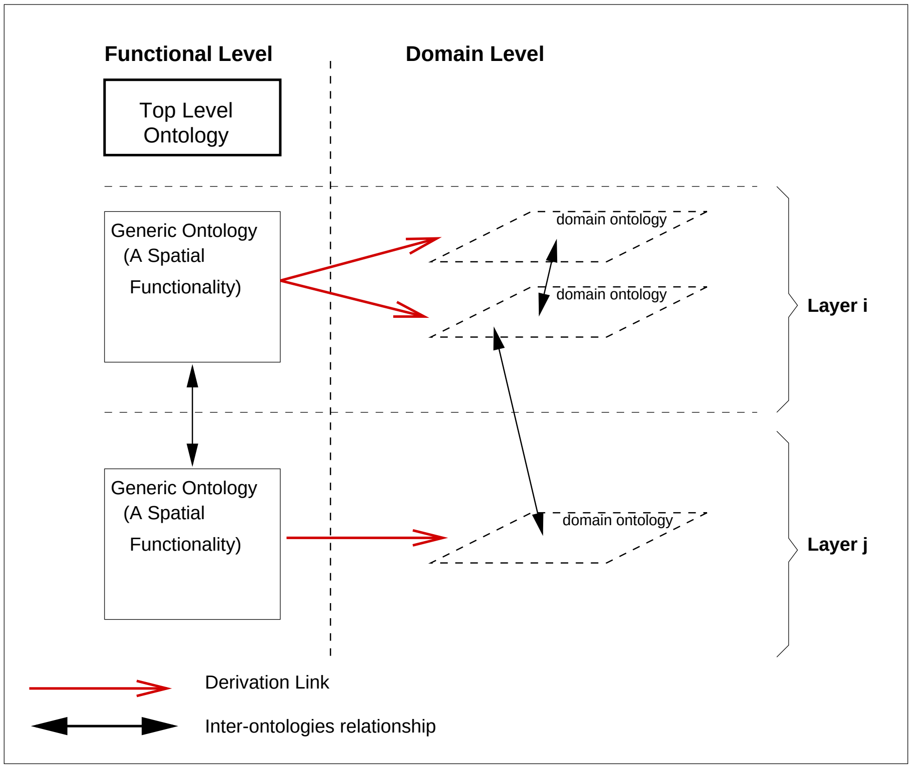

Fig. 2. Overview of multi-layered ontologies

tions, which are used to define operations and specify constraints that must be in the domain ontologies. These definitions provide abstract semantic interfaces and are not based on structural descriptions. For example, a functional level description of urban networks (water, traffic, railroads etc..) may consist of generic nodes (without structural descriptions), generic links and traversal functions to model the flow of goods or objects through the network.

- **A domain level** consists of one or more domain ontologies that are consistent with the functional level of the ontology layer. Domain ontologies represent the semantics of real world objects. They are used to specialize or instantiate the components of the functional level. Constructing a domain ontology involves using a derivation mechanism to select subsets of operations, parameters and constraints from the generic functional components and to specialize them to represent the characteristics of an application domain. For example, in figure 2 the derivation mechanism, represented by gray arrows, can be used to specialize the generic network ontology to construct water and road domain ontologies.

In this section we present the methods for formulating the content of an ontology. The representation of a layer is based on several concepts: 1) abstract data types; 2) generic functional classes (GFC); 3) domain level classes (DLC); and 4) an extended object oriented model.

4.3.1 Functional level ontology representation

At the functional level, ontological concepts are represented by abstract data types (ADT). In our approach, each ADT is over-specified (i.e. it contains all the possible operations)(Burstall & Goguen 1977, Wirsing 1990). Its axioms are given by algorithmic specifications as formalized in (Loeckx 1987). ADT specify the most generic functional behavior of ontology concepts. They convey the idea that the semantics of a domain can be specified through a semantic interface (operations defined by the ADT to manipulation objects of the domain of interest). Furthermore, ADT are used to specify interactions (or the behavior) of the operations.

A hierarchy of concepts (see figure 4) is associated with each ADT, representing specialization relationships among objects of domains that are constructed or derived from the ADT. For instance, an ADT used to abstract the functional level description of network domain ontologies for railways and traffic networks applications contains two generic concepts *Node* and *Link*: *Node* will be associated with a hierarchy which depicts relationships between domain level concepts representing railways station, rail-crossing or cities; and similarly a hierarchy of railways-domain-level concepts that tracks, bridges, tunnels will be associated with the generic concept *Link*. In this example, the general operations in the semantic interface of the ADT includes 1) graph-based operations such as path traversal, inter-nodes distance evaluation and path cost optimization; and 2) general topological relations as connectivity or connected subgraph search.

ADT are used to create Generic Functional Classes (GFC) consisting of concepts that share the definition and properties of a functional ontology. A GFC can use other GFC, which are named sort in the ADT theory. A GFC is a tuple $gfc = \langle Name, S, Op, Ax, Co, Map \rangle$ where $Name$ is the name of gfc, S is the set of sort used in gfc, Op is a set of functions symbols (called operations), Ax is a set of axioms, $\langle S, Op \rangle$ constitutes the signature, $\langle \langle S, Op \rangle, Ax \rangle$ is the specification of the ADT used to build the *GFC*, Co is the list of the constraints name which describe the properties of gfc, $Map = \{f, f : Co \rightarrow Op\}$ is a set of mapping functions from Co to Op which associate to a constraint its describing operations.

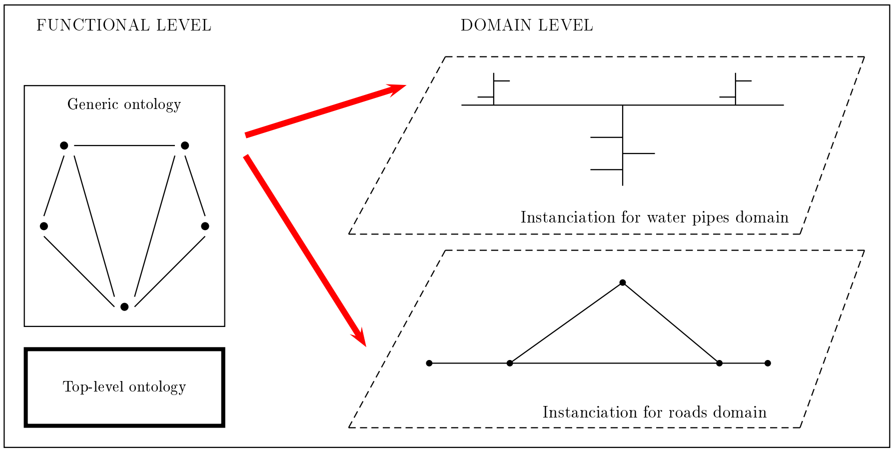

Fig. 3. Example of an ontology layer

4.3.2 Domain level ontology representation

At the domain ontology level, an object oriented-based reference model is used to describe ontological concepts. Domain Level Classes (DLC) are used to specialize generic functional classes for particular application domains. A DLC is an abstract class with no extension (it has no real object instances) and is defined by: $dlc = \langle Name, IS, Op, Ax, In \rangle$ where $Name$ is the name of dlc, IS is a set of instanciation links between generic component and specific objects, Op is a subset of operations derived from the operations of the $GFC :: Op$ of which dlc is an instance or specialization, Ax is the set of axioms associated to Op, In is a set of invariants which are used to express semantic characteristics or properties of the domain.

Figure 3 depicts an example of an ontology layer: the functional level describes a generic ontology for networks which is derived in the domain level in two domain ontologies for water pipes and roads.

As stated above, a DLC is defined by selecting constraints and relevant operations from the description of a concept of the generic ontology. Consider the example of ontological layer shown in figure 3. The ontology layer consists of the networks functional level and two domain level ontologies for water pipes and roads domains. The corresponding instanciation process for defining the domain level classes is shown in figure 4. Dashed arrows are instanciation links while plain arrows depicts classic specialization links between DLC. To define the DLC, the generic components of the network GFC are mapped to specific components of the application domains. For example, in figure 4, the node GFC is therefore instantiated by two specific nodes: *car-traffic* and *water-pipes*.

Each DLC is mapped to an Object Class (OC) which represents low level information systems. In interoperable systems, they are mapped to wrapper classes, which are used to encapsulate local information sources. An OC is a

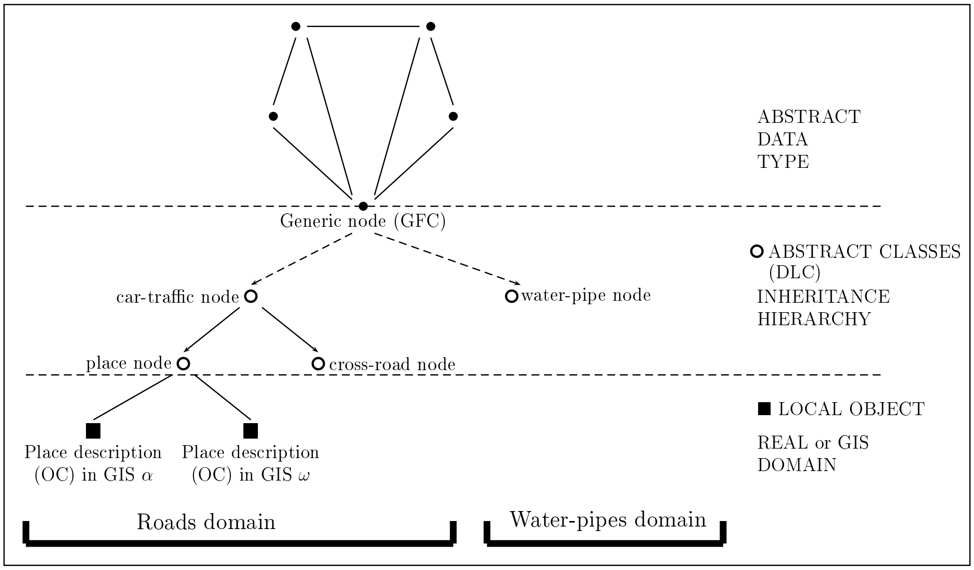

Fig. 4. Concept hierarchy of a generic node

tuple $oc = \langle Name, AttList, MethList, Cor\rangle$ where $Name$ is the name of *oc*, *AttList* is the list of the attributes belonging to *oc*, $MethList$ is the list of the methods define for *oc*, Cor is a set of mapping between operations inherited from GFC implemented by methods in *oc*.

4.4 Inter-Ontologies relationships

Inter-ontologies relationships (called ontological relations) are spatial relationships among objects from the same ontology layer or from different layers. These relations can be defined at both functional and domain levels. At the domain level, an ontological n-ary relation $r \in R$ is defined by a tuple $r = (OC_1, OC_2, ...OC_n)$ where $OC_i \in GFC \cup DLC$ are ontological concepts. For example, in figure 5 the dashed arrows between the two domain ontologies of $Layer1$ state the fact that an object (a pipe) in the *Water-pipes* application is at the same location (or address) as an object (a street) in the *Road network* domain. Likewise, plain links between the ontologies in $Layer1$ and those in $Layer2$ state the fact a health district or cadastral parcels may include crossroads (intersection of two or more streets).

At the functional level, the relations are used to associate generic functional concept in one layer to one or more concepts in another level. These relations can be instantiated to define domain level inter-relationships among the corresponding domain ontologies. Figure 5 shows a relationship between the generic functional structure of networks and coverage ontologies. Note also that in this case the spatial operation described by a relation link is inherited by the domain ontologies derived from the functional ontologies that are associated by the relation.

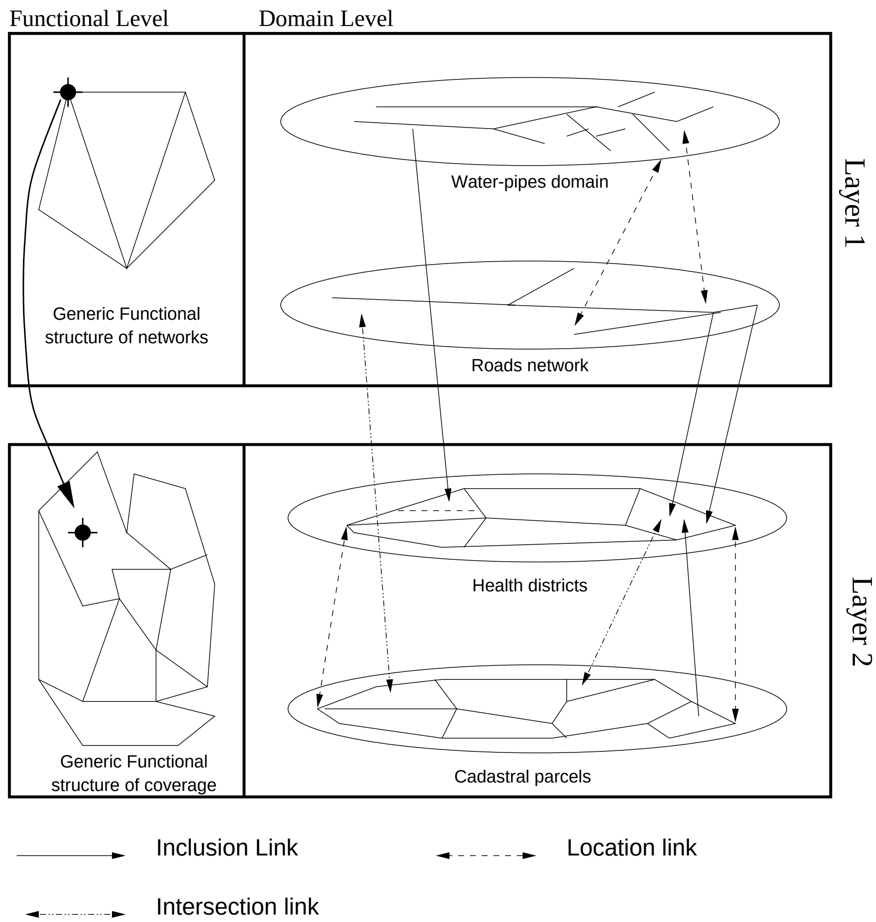

Fig. 5. Inter-ontologies relationships

5 Multi-layered ontologies for urban information management systems

In the previous section an ontology definition methodology has been identified to take into account the abstraction levels and the associated semantics of spatial information systems when ontology-driven applications are designed. In this section we show how to apply the methodology to describe two urban applications domains: 1) urban utilities networks (traffic, water, power lines ...) and 2) land use (buildings, public parks, health districts ...).

5.1 *Example : ontology structure for a given type of query*

Consider the following query, denoted by Q, for choosing an appropriate location (and related information) for a new hospital in a health district. The decision is based on a set of characteristics that the selected location must have.

Find a parcel or a group of contiguous parcels (and their types and owners) with a surface superior to 25 acres, located less than 1 km from a highway and with at least 3 carriage-way accesses. Then, determine the sub-networks of water, electric and sewer pipes that cross the parcels.

The relevant domain information can be retrieved from the distributed decision support system shown in figure 1, including the following data:

- Cadastral information on parcels, including their geometry, type and owner. In this domain, the required operations are those that deal with spatial topological relationships such as neighbor, spatial inclusion, therefore spatial abstraction the domain.
- Information on car-traffic, water-pipes, electric and sewer systems. The required domain operations are classical graph-based operations. Thus, a spatial abstraction of networks is to be used.
- Top-level ontology is required for providing general information on parcels, owners and other entities of the application domain.

Figure 6 presents a multi-layered ontology which provides support for processing the Q query, consisting of a top level ontology, two generic ontologies (networks and coverage) and the corresponding domain ontologies derived from networks (i.e. Car-Traffic, Water-pipes, Sewer-pipes and Electric) and from coverage (i.e. Cadastral Parcels).

5.2 *An example multi-layered ontology for urban network infrastructures*

5.2.1 *Generic ontological model for urban networks*

To create domain ontologies for network-based urban applications, we must first define a generic ontological model to represent the inherent functionalities and abstractions of urban networks. The functionalities are expressed through a generalized graph-based structure (figure 7) that combines functionalities of graph and hyper-graphs (Harel 1988, Harel & Naamad 1996).

It contains three main components. The first component which is shown in figure 7.a is a functional structure consisting of two generic components (*nodes*

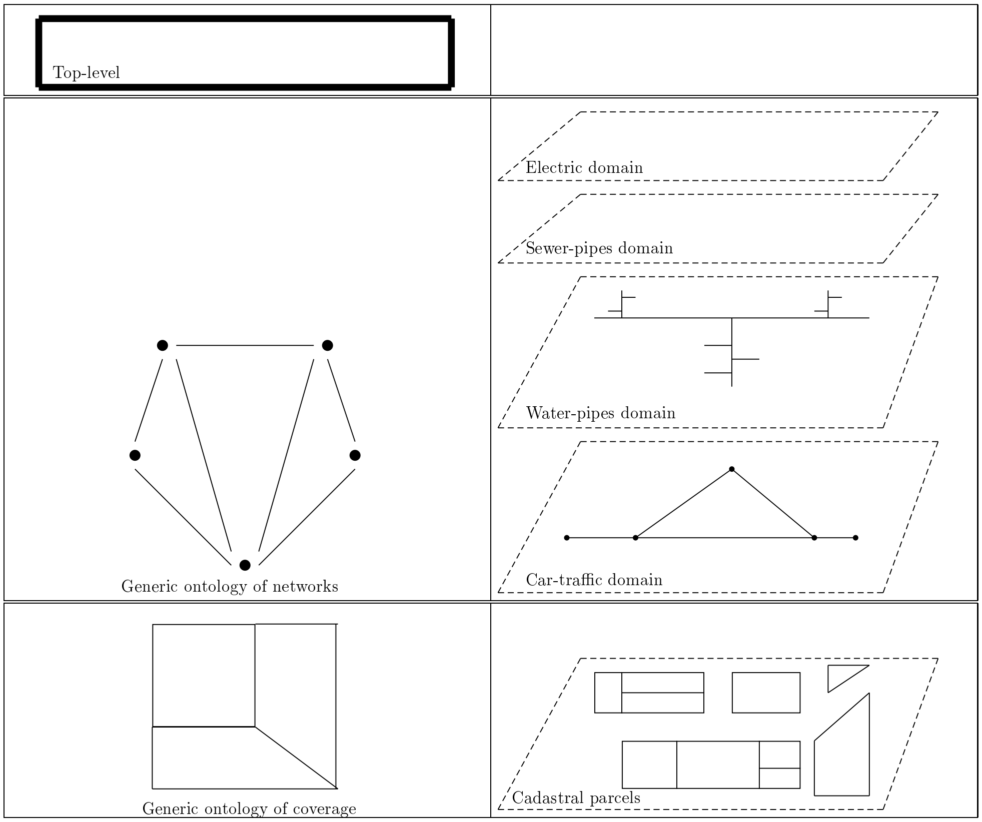

Fig. 6. Ontology layers for the query Q

and *links*) that model generalized urban network functionalities without any implication of application domain. Its interface contains all possible operations that are consistent with network functionalities. The second component (shown in 7.b) is a set of constraints that can be associated with the functional abstraction. They define properties of the components (e.g. link orientation,) or the whole network structure (connectedness of a network). The third component shown in figure 7.c provides a set of operations. The operations are not detailed but are globally presented through packages (sets of operations such as Path optimization packages including algorithms as Ford, Bellman-Kalaba, Dijkstra's).

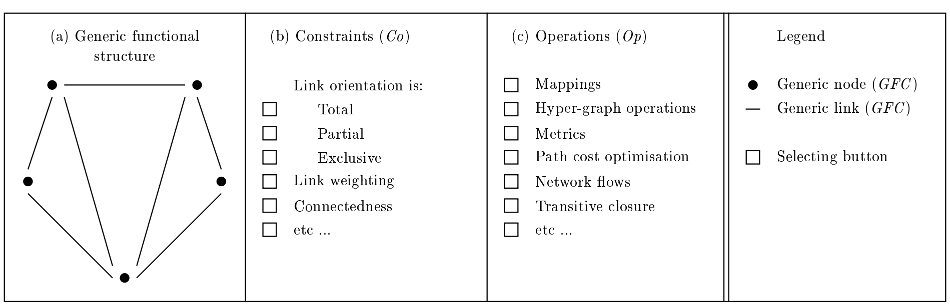

Fig. 7. Generic ontological model of urban network

5.2.2 Domain ontologies for urban networks

Using the above generic ontological model to define domain ontologies involves instanciating the generic nodes and links and specifying an appropriate set of constraints and operations to reflect the characteristics of an application domain. Figure 8 shows two instanciated domain ontologies, namely a Road domain and Water-pipes domain. The instanciation of a generic ontological model requires the following steps:

- **Fixing constraints from the generic structure:** This step and the next are inter-dependent steps which are used to choose constraints and operations to adapt a generic ontology to an application domain. The selection of the appropriate domain dependent constraints is carried out in this step. For example, the following constraints (see figure 8.b1) are retained for the Road domain ontology in figure 8: a partial orientation which states that only one way traffic is allowed on the streets, link weighting to represent allowed flow of traffic. A different set of constraints (see figure 8.b2) is chosen for the Water-pipes domain. Contrary to the Road domain, a total orientation is assumed in this case to state the fact that water can only flows in one direction in the pipes.
- **Choosing relevant operations:** This step is used to select the subset of operations a domain ontology can allow. It takes into account the set of constraints selected in the above step, which may invalidate the choice of some operations. For example, if links are not weighted some metrics become irrelevant. Some operations defined in the generic structure may be coherent with the domain constraints but are useless in the application domain (see figures 8.c1 and 8.c2). For example, any operation for which link orientation is irrelevant is also will not be valid in Water-pipes application domain.
- **Instanciating generic components:** Links and nodes in the generic urban network structure are generic classes "without context". The contextual information are represented when a domain concept is defined and mapped to the generic component. All the instanciated concepts of a generic components are organized in a concept hierarchy of which the generic component is the root. Figure 9 presents the hierarchies corresponding for the generic components *node* and *link*, classify concepts of the ontology domains Road and Water-pipe.

  Generic links and nodes within both Road domain and Water-pipes domain are shown in figure 9. Table 1 gives an example of class definitions for part of the hierarchy associated with *generic_node*. Class invariants (logic expression of characteristics (Meyer 1987)) are used to differentiate an inheritant class from its ancestor classes. The root classes of the hierarchies (generic_component) provide support for defining general function packages (Ford algorithms in the example). The inheritance mechanism makes it possible to define the general functions at the appropriate level. For example (figure 10), the processing function *evaluate_average_of_waiting* is described

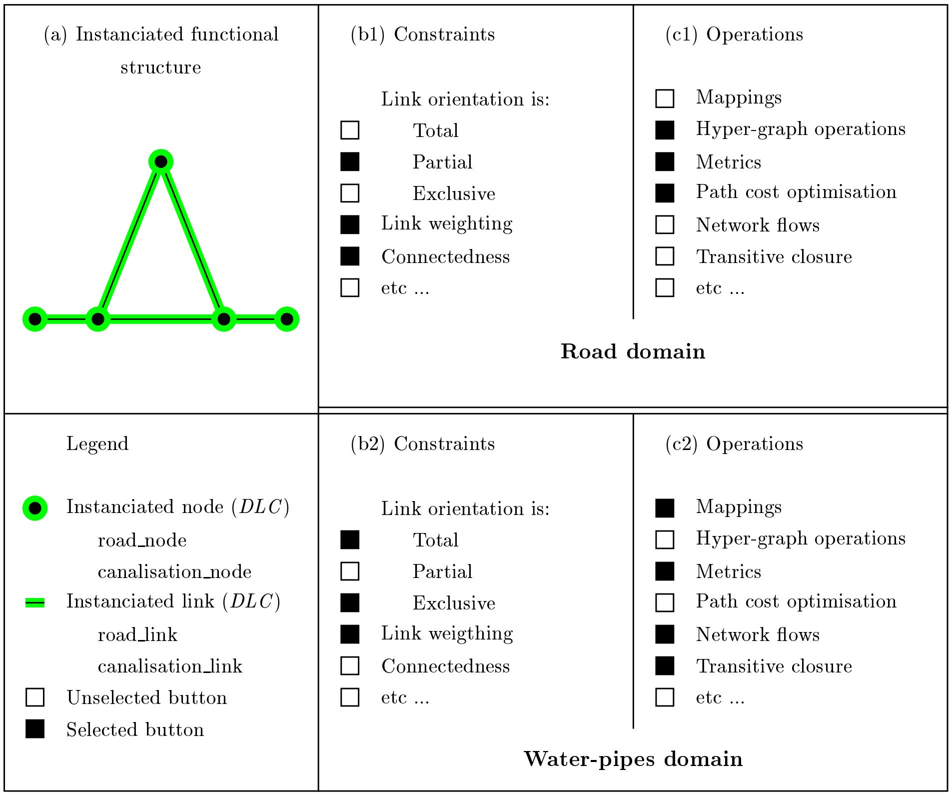

Fig. 8. Selecting constraints and operations for Road and Water-pipes domains

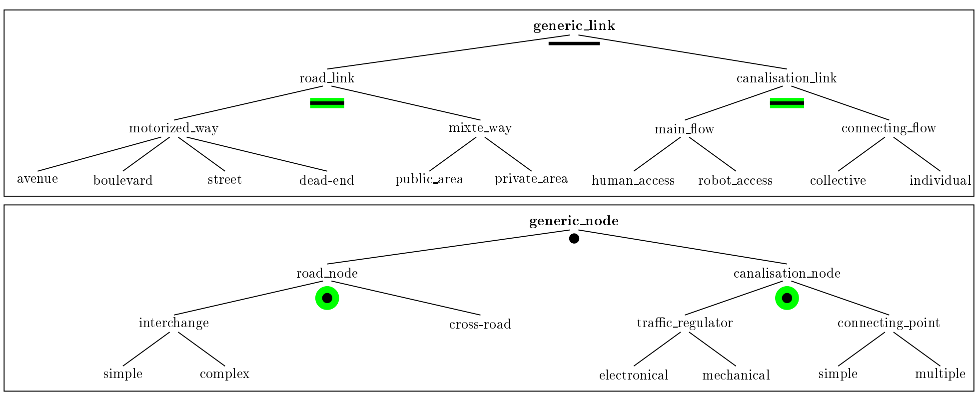

Fig. 9. Class hierarchies for the Road and Water-pipes domains

on the class *road_node* in terms of a virtual manipulating function called *get_waiting_time*, which is defined at a low level class *cross_road* and associated to actual data or function in every concerned GIS.

- **Expressing relationships between objects:** This step provides tools to express constraints on objects using rules and relations. An *anchor* is a participant concept of a relation. An anchor is considered as a particular object that has to be attached to another object. Furthermore anchors which provide connection points to other layer are an essential concept to related different views of the same real world object.

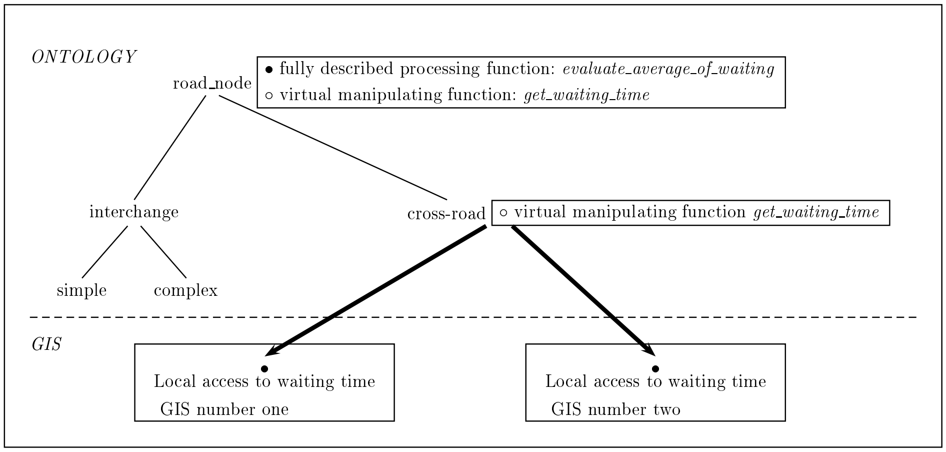

Fig. 10. Differing node instanciation

5.3 Generic ontological model for coverage

Coverage is a generalized structure which can be used to model 2D surface covered by polygons (as classically determined by points and segments in most GIS models). It uses a single generic object called *generic shape*. Coverage is more general than tessellation insofar as it is not a partition (some parts of 2D surface may be not covered). The associated set of constraints determine properties of 1) coverage (such as total or partial coverage, with or without intersection of shapes ...) and 2) shapes themselves (regular or not, presence or absence of holes, islands...).

Figures 11.a, 11.b, 11.c show the generic ontological model of *Coverage*. Figure 11 also shows three domain level ontologies that are created by instanciation of the generic ontology *Coverage*. Two of the domain ontologies are similar, they represent land occupation by Buildings and Public parks. Both are based on disjoint coverages with irregular shapes, but differ in granularity and coverage density. The selected sets of constraints are shown in figures 11.b1 and 11.b2 while the operations are given in figures 11.c1 and 11.c2. The third domain ontology represents Health-district which is a total coverage with irregular shapes. See figures 11.b3 and 11.c3 for the corresponding set of constraints and operations. The hierarchy of concepts corresponding to the component *generic shape* is shown in figure 12.

5.4 The Multi-Layered Ontologies

Using the above ontologies layers (functional components and domain level ontologies) to process the example query requires the construction of a multi-layered ontology in which a critical phase is the expression of multi-layer con-

Table 1
Class definitions for part of the Road domain ontology

CLASS *generic_node*
ANCESTOR:
INVARIANT:
{
get_idf: generic_node $\rightarrow$ idf
put_idf: idf $\rightarrow$ generic_node

Ford_minimal_path: generic_node, generic_node $\rightarrow$ set_of_paths
Ford_maximal_path: generic_node, generic_node $\rightarrow$ set_of_paths
... }

CLASS *road_node*
ANCESTOR: generic_node
INVARIANT: get_number_of_entry + get_number_of_exit $\geq$ get_arity
{
get_arity: road $\rightarrow$ integer
put_arity: integer $\rightarrow$ road

get_number_of_entry: road $\rightarrow$ integer
put_number_of_entry: integer $\rightarrow$ road

get_number_of_exit: road $\rightarrow$ integer
put_number_of_exit: integer $\rightarrow$ road

get_level_number: road $\rightarrow$ integer
put_level_number: integer $\rightarrow$ road

get_waiting_time: cross_road $\rightarrow$ integer
put_waiting_time: integer $\rightarrow$ cross_road

get_average_of_waiting: cross_road $\rightarrow$ integer
evaluate_average_of_waiting: cross_road $\rightarrow$ cross_road
... }

CLASS *cross_road*
ANCESTOR: road_node
INVARIANT: get_arity=4 and get_level_number=1
{
get_waiting_time: cross_road $\rightarrow$ integer
put_waiting_time: integer $\rightarrow$ cross_road
... }

straints and properties. Rules and anchor connections provide the foundation for expression these constraints. Consider the following properties (table 2):

Property 1 states the fact that "Every building must have access to at least one road". This property is implemented by an anchor point named *Access* and the corresponding virtual function *access_road* which determines for each building a set of roads. A reverse virtual function *access_building* is associated with anchor point R_{Access} to determine the set of buildings connected to a road.

Property 2 states the fact that "There is a correspondence between street segments and building postal addresses". This property is implemented by an anchor point $R_{Address}$ and the corresponding virtual functions giving, on the one hand end buildings for a given road segment, and, on the other hand, the set of addresses between two buildings. Note that the *address* concept has to be defined in the top-level ontology.

Property 3 states the fact that "There is generally one and only one road between two neighboring but non adjacent buildings". This property is implemented by an anchor point $R_{Topology}$ and two corresponding virtual functions. The former is a boolean function, which is true if the rule is verified

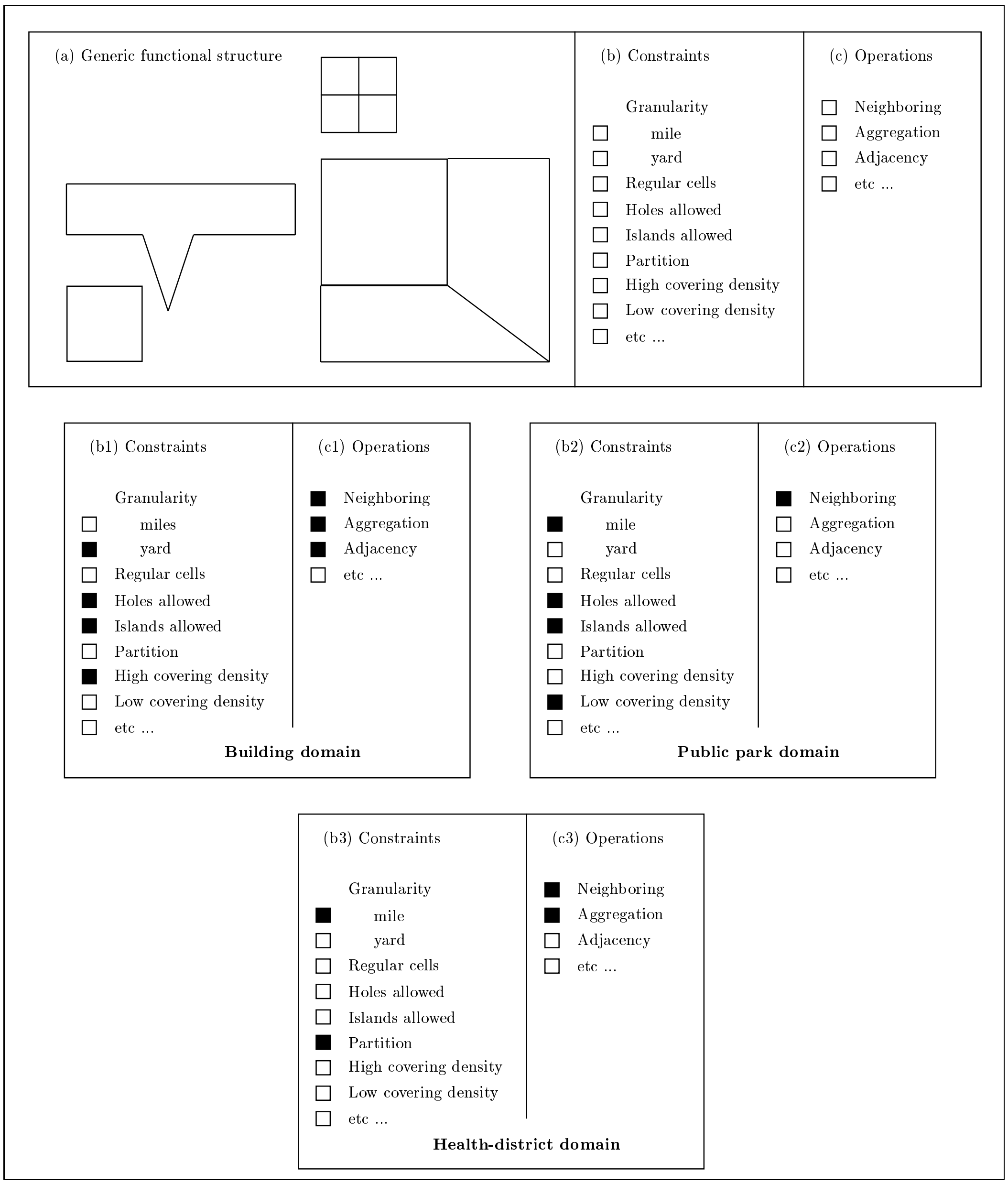

Fig. 11. Generic ontological model and derivation for Coverage

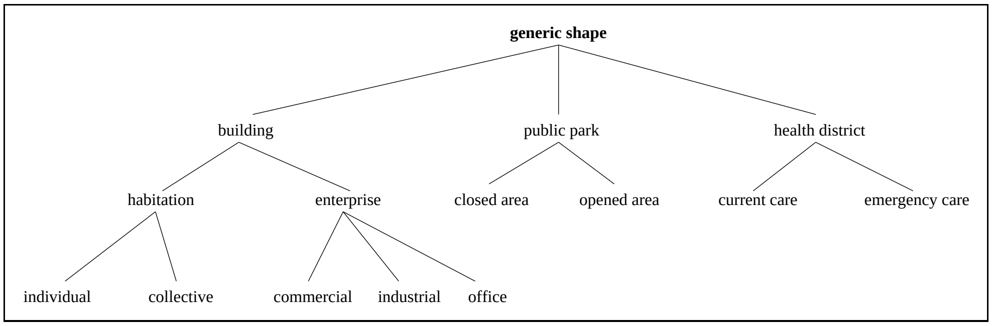

Fig. 12. Class hierarchies for Building, Public park and Health-district domains

for a given pair of buildings. The latter gives the set of roads associated to a given pair of buildings.

Property 4 expresses a common property of water pipe layer and cadastral layer : "Each link in the water pipe network is associated (exactly) one localised shape in the cadastral layer". This shape, called "connecting surface", determines the area in which it is both possible to connect the link and not possible to safely position another link of any type (electric, gas ...). This property is implemented by an anchor point *Connecting_Shape* and the corresponding virtual functions. The function *connect_shape* associates a link to its shape. Its reverse function is named *connect_link*.

Several remarks can be made on the properties. The above properties may only be different in terms of coercivity: obligatory for rule 1, optional for rule 2 and with determined exceptions for rule 3. Furthermore, it is necessary to propose tools to verify the completeness and the coherence of the functions that are used to express correspondences. Another remark concerns the level where rules have are defined. A general property may be expressed on the upper level of an inheritance hierarchy. For example, the rule expressing the existence of a maximum distance between a building and a health-district may be given on the *habitation* class. But the rule expressing the existence of a minimum number of accesses to road depending on the number of apartments has to be given in the *building* class.

5.5 How to use inter-ontology relationships to process spatial queries

To cope with the "new hospital location" request, let us consider two layers. The first layer is derived from networks ontology and represents water pipes. The second layer is derived from coverage ontology and represents the cadastral parcels. Assume we have to test -on water pipes criteria- a potential solution i.e. a set S of contiguous parcels. The proposed algorithm projects the intersection problem onto cadastral layer [1]. Let us supose that each link

[1] Another algorithm may be considered by projecting onto water layer.

Table 2
Anchors definition for classes "buildings" and "roads"

ANCHOR *Access* CORRESPONDING RELATION: R_{Access} DOMAIN LEVELS: Road domain & Building domain SOURCE CLASS: *building* TARGET CLASS: *road_node* CONSTRAINT: connection is mandatory { IMPLEMENTATION: virtual functions *access_road*: building → set of road_node *access_building*: road_node → set of building }
ANCHOR *Address* CORRESPONDING RELATION: $\mathrm{R}_{Address}$ DOMAIN LEVELS: Road domain & Building domain SOURCE CLASS: *road_link* TARGET CLASS: *building* CONSTRAINT: connection is optional { IMPLEMENTATION: virtual functions *extreme_buildings*: road_link → tuple of building *addresses*: tuple of building → list of address }
ANCHOR *Topology* CORRESPONDING RELATION: $\mathrm{R}_{Topology}$ DOMAIN LEVELS: Road domain & Building domain SOURCE CLASS: *building* TARGET CLASS: *road_node* CONSTRAINT: connection with exceptions { IMPLEMENTATION: virtual functions *rule_validity*: building × building → boolean *get_links*: building × building → set of road_link }

of water pipe is associated, through an inter-ontology relationship, a 2D-shape corresponding to its "connecting surface [2]". Figure 13.a presents the parcels (identified by a number from 1 to 7) and the corresponding sub-set of water-pipe networks. On this example, there is no node into parcels 6 and 4. Figure 13.b shows the projection on coverage layer of the connecting surfaces corresponding to the water pipes sub-network. The only problem is parcels 6 and 4 that have no intersection with any connecting surface.

[2] i.e. the surface inside which it is both possible to connect this link and not possible to safely have another link from any type (water, electric, gas ...).

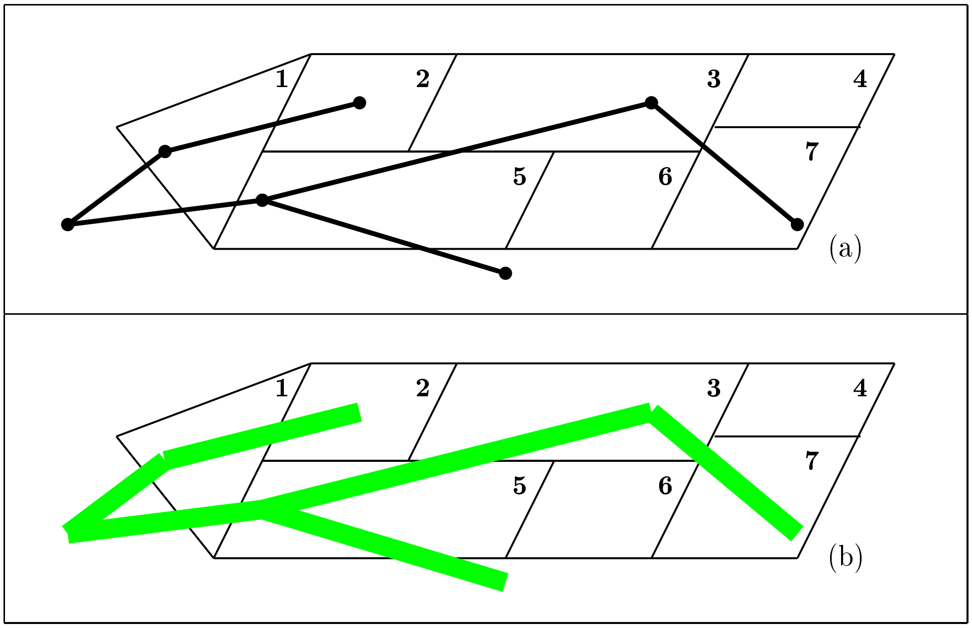

Fig. 13. Inter-ontology relationships for water pipes and cadastral parcels

6 Conclusion

In this paper, we have focused on a fundamental issue in the design of interoperable GIS for urban applications, the development and use of ontologies to support semantic interoperability. The extensive ongoing research on interoperation of information has demonstrated the importance of allowing multiple information systems to share and exchange data across systems boundaries. This is even more crucial in spatial information systems in which data acquisition and manipulation incur high costs. We have argued that sharing and exchanging data requires that the data providers and receivers must agree on a common reference context by which they can resolve discrepancies in their views and understanding of the shared data.

To achieve this goal, we have stated how ontologies can be used to provide formal support and tools for designing urban management applications in which the decision making process involves combining information from different heterogeneous information sources. Ontology-based interoperation and applications exhibit several advantages including precise description of queries and systems information content, dynamic support for integration and query dependent interoperation. The main contribution of the paper is a methodology to allow the definition of multi-layered ontologies for urban management applications. The solution consists in describing an application domain by abstraction layers and defining inter-relationships among the layers. For each layer, we show how to construct ontologies by first defining a generic functional model described by abstract data types, then domain ontologies are derived from the functional model by specializing its components and properties. We have presented several examples to illustrate how the ontologies can be used in application domains such as urban (traffic, electric, water...) networks.

The development of ontology is still hampered by the complexity of abstracting a reduced number of inherent properties from a large number of terms. Our future work will focus on 1) a formal definition of the concepts used to create a multi-layered ontology, using different inter-related layers allows to reduce the number of terms that must be considered at each level and 2) the design of tools to allow users to collaborate in the ontology generation process.

References

Bishr, Y. (1998). Overcoming the semantics and other barriers to gis interoperability, *International Journal of Geographical Information Science* **12**(4): 299–314.

Borgida, A., Brachman, R., McGuinness, D. & Resnick, L. (1989). Classic: A structural data model for objects, *ACM SIGMOD International Conference on Management of Data*, Portland, Oregon, pp. 58–67.

Burstall, R. & Goguen, J. (1977). Putting theories together to make specifications, *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1045–1058.

Coenen, F., Beattie, B., Bench-Capon, T. J. M., Shave, M. J. R. & Diaz, B. M. (1996). An ontology for linear spatial reasoning., *International Conference on Database and Expert Systems Applications (DEXA-96)*, Lecture Notes in Computer Science, Vol. 1134, Springer, pp. 718–727.

Devogele, T., Parent, C. & Spaccapietra, S. (1998). On spatial database integration, *International Journal of Geographical Information Science* **12**(4): 335–352.

Farquhar, A., Fikes, R. & Rice, J. (1996). The ontolingua server: A tool for collaborative ontology construction, *Technical Report KSL-96-26*, Computer Science Department, Stanford University, Knowledge System Laboratory.

Fikes, R., Farquhar, A. & Rice, J. (1997). Tools for assembling modular ontologies in ontolingua, *Technical Report KSL-97-03*, Computer Science Department, Stanford University, Knowledge Systems Laboratory.

Genesereth, M. & Fikes, R. (1992). Knowledge interchange format version 3.0 reference manual, *Technical Report 92-1*, Logic Group, Computer Science Department Stanford University CA.

Gennari, J. H., Oliver, D. E., Pratt, W., Rice, J. & Musen, M. A. (1995). A web-based architecture for a medical vocabulary server, *Annual Symposium on Computer Applications in Medical Care*, pp. 275–279.

Gruber, T. (1993). A translation approach to portable ontology specifications, *International Journal of knowledge acquisition for knowledge-based systems* **2**(5): 199–220.

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation, *International Journal of Human and Computer Studies, special issue on The Role of Formal Ontology in the Information Technology.* **43**(5/6).

Guarino, N. (1997). Understanding, building and using ontologies, *International Journal of Human and Computer Studies* **46**(2/3): 293–310.

Harel, D. (1988). On visual formalisms, *Communications of ACM (CACM)* **31**(5): 514–530.

Harel, D. & Naamad, A. (1996). The statemate semantics of statecharts, *ACM Transactions on Software Engineering and Methodology (TOSEM)* **5**(4): 293–333.

Jones, D. (1998). Developing shared ontologies in multi-agent systems, *International Workshop on Intelligent Information Integration (ECAI-98)*, Brighton, UK.

Laurini, R. (1998). Spatial multi-database topological continuity and indexing: A step towards seamless gis data interoperability., *International Journal of Geographical Information Science* **12**(4): 373–402.

Leclercq, E., Benslimane, D. & Yétongnon, K. (1999). Semantic mediation between cooperative spatial information systems: The Amun data model, *International Conference on Advances in Digital Libraries (ADL'99)*, IEEE Press, Baltimore, MD, pp. 16–27.

Litwin, W., Mark, L. & Roussopoulos, N. (1990). Interoperability of multiple autonomous databases, *ACM Computing Surveys* **22**(3): 265–293.

Loeckx, J. (1987). Algorithmic specifications: A constructive specification method for abstract data types, *ACM Transactions on Programming Languages and Systems (TOPLAS)* **9**(4): 646–685.

MacGregor, R. (1988). A deductive pattern matcher, *National Conference on Artificial Intelligence (AAAIPress / The MIT Press)*, St. Paul, MN, pp. 403–408.

MAR (n.d.). *content designation in the USMARC formats, American Library Association's Machine-Readable Bibliographic Information Committee (MARBI).* http://www.loc.gov/marc/.

Meyer, B. (1987). Eifel: Object-oriented design for software engineering., *European Software Engineering Conference (ESEC)*, Strasbourg, France, pp. 221–229.

Open-GIS (1996). The open gis guide part i: Introduction to interoperable geoprocessing, *Technical report.*

RDF (1998). *Resource Description Framework (RDF).* http://www.w3.org/RDF/.

Sheth, A. (1999). *Interoperating Geographic Information Systems*, Kluwer, chapter 1 Changing Focus on Interoperability In Information Systems: From System, Syntax, Structure to Semantics.

Sheth, A. & Larson, J. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases, *ACM Computing Surveys* **22**(3): 183–236.

Tomasic, A., Raschid, L. & Valduriez, P. (1998). Scaling access to heterogeneous data sources with disco, *IEEE Transactions on Knowledge and Data Engineering* **10**(5): 807–823.

Včkovski, A. (1998). *Interoperable and Distributed Processing in GIS*, PhD thesis, Université de Zurich. Taylor et Francis, ISBN 0-7484-0792-8.

Voisard, A. & Schweppe, H. (1994). A multilayer approach to the open gis design problem, *Second International ACM GIS Workshop*, pp. 23–29.

Voisard, A. & Schweppe, H. (1998). Abstraction and decomposition in interoperable gis, *International Journal of Geographical Information Science* **12**(4): 315–333.

Wariyapola, P., Patrikalakis, N., Abrams, S., Elisseeff, P., Robinson, A., Schmidt, H. & Streitlien, K. (1999). Ontology and metadata creation for the poseidon distributed coastal zone management system, *IEEE Forum on Research and Technology Advances in Digital Libraries*, pp. 180–189.

Weinstein, P. (1998). Ontology-based metadata: Transforming the marc legacy, *ACM Digital Libraries*, Pittsburgh, PA, USA, pp. 254–263.

Wirsing, M. (1990). *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, Elsevier Science Publishers, chapter Algebraic Specification, pp. 677–788.

XML (1998). *Extensible Markup Language (XML).* http://www.w3.org/XML.