Semantic Granularity in Ontology-Driven Geographic Information Systems

Frederico Fonseca[1] **Max Egenhofer**[1] **Clodoveu Davis**[2] **Gilberto Câmara**[3]

[1]National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469-5711, USA

[2]Prodabel - Empresa de Informática e Informação do Município de Belo Horizonte
Belo Horizonte MG Brazil

[3]National Institute for Space Research (INPE)
Image Processing Division – Brazil
{fred,max}@spatial.maine.edu clodoveu@pbh.gov.br gilberto@dpi.inpe.br

Abstract
The integration of information of different kinds, such as spatial and alphanumeric at different levels of detail, is a challenge. While a solution is not reached, it is widely recognized that the need to integrate information is so pressing that it does not matter if detail is lost, as long as integration is achieved. This paper shows the potential for information retrieval at different levels of granularity inside the framework of information systems based on ontologies. Ontologies are theories that use a specific vocabulary to describe entities, classes, properties and functions related to a certain view of the world. The use of an ontology, translated into an active information system component, leads to Ontology-Driven Information Systems and, in the specific case of GIS, leads to what we call Ontology-Driven Geographic Information Systems.

1. Introduction

The availability of information about the Earth has been increasing steadily through the last years. Contemporary information systems are distributed and heterogeneous, which leads to a number of interesting research challenges. One of them is about how to integrate information of different kinds, such as spatial and alphanumeric, at different levels of detail. Events that happen over a large area, such as the wild fires in and around Los Alamos, New Mexico, in 2000, require a dynamic integration of geographic information. Many times these requirements are so demanding that it does not matter if detail is lost, as long as integration is achieved. Frequently, the information exists, but integration is very difficult to achieve in a meaningful way because the available information was collected by different agents and also with diverse purposes.

The effective integration of multiple resources and domains is known as *interoperation*. Efforts towards geographic information systems (GIS) [1] interoperation are well documented [2-5]. In the past, exchanging geographic information was as simple as sending paper maps or raw data tapes through the mail. Today, computers throughout the world are connected and the use of GIS has become widespread. The scope of interoperability has changed from static data exchange using flat files to global systems, interconnected using sophisticated protocols to exchange information on-line. In the future, computers are expected to be able to share not only information but also knowledge [6]. Although spatial information systems have been characterized as an integration tool, GIS interoperability is far from being fully operational [7].

In this paper we are address the semantic aspects of geographic information integration. In this context, these semantic aspects are related to the meaning of the entities that compose the ontologies. These ontologies represent concepts of the real world or, more specifically, on the geographic world. Our concern is with semantic granularity, rather than spatial granularity. Semantic granularity addresses the different levels of specification of an entity in the real world, while spatial granularity deals with the different levels of spatial resolution or representation at different scales. For instance, inside a community of biology scholars, a specific body of water in the state of New Mexico can be a lake that serves as the habitat for a specific species and, therefore, there can be a special concept or name to be referred to. Nevertheless, it is still a body of water, and when a biologist is working at a more general level it is considered as a body of water and not as a lake. At this higher level it is more likely that the concepts biologists have about this real world entity "body of water" will match concepts held by another community. Therefore, in this more general level of detail, the biologists and the members of another community can exchange information about bodies of water. The information will be more general than when the body of water is seen as the habitat of a specific fish species.

In GIS, the focus is changing from format integration to semantic interoperability. The first attempts to obtain GIS interoperability involved the direct translation of geographic data from one vendor format into another. A variation of this practice is the use of a standard file format. These formats can lead to information loss, as is often the case with the popular CAD-based format DXF. Alternatives that avoid this problem are usually more complex, such as the Spatial Data Transfer Standard (SDTS) [8] and the Spatial Archive and Interchange Format (SAIF) [9]. An argument contending that a common format was not enough to transfer data along with semantics was first brought forth in Mark [10]. Since then, semantics has been treated as more and more important in geographic information integration [6, 11-17]. This paper focuses on finding innovative ways to integrate geographic information. The starting point of the integration of geographic information is the physical universe. This approach differs from usual ones, that start from the implementation and representation points of view. Our approach enables the integration of information based on its semantic content instead of dealing primarily with data formats and geometric representations. The next generation of information systems should be able to solve semantic heterogeneity to make use of the amount of information available with the arrival of the Internet and distributed computing. An information system that intends to solve semantic interoperability should be able to understand the user model of the world and its meanings, to understand the semantics of the information sources, and to use mediation to satisfy the information request regarding the above mentioned sources and users [6].

Ontologies play a key role in enabling semantic interoperability [18]. Ontology for a philosopher is the science of beings, of what is, i.e., a particular system of categories that reflects a specific view of the world. For the Artificial Intelligence (AI) community, ontology is an engineering artifact that describes a certain reality with a certain vocabulary, using a set of assumptions regarding the intended meaning of the vocabulary words. Gruber [19] defines an ontology as an explicit specification of a conceptualization, from which Guarino [20] makes a refined distinction between an ontology and a conceptualization: an ontology is a logical theory accounting for the intended meaning of a formal vocabulary (i.e., its ontological commitment to a particular conceptualization of the world), whereas a conceptualization is the formal structure of reality as perceived and organized by an agent, independently of the vocabulary used or the actual occurrence of a specific situation. The intended models of a logical language that use such a vocabulary are constrained by its ontological commitment. This commitment and the underlying conceptualization are reflected in the ontology by the approximation of these intended models.

Research in the next generation of information systems should focus on a specific kind, such as GIS, before more general architectures can be developed [6]. This new generation of systems is characterized by the use of multiple ontologies and contexts to achieve semantic interoperability. Since Aristotle's theory of substances (objects, things, and persons) and accidents (qualities, events, and processes), ontology has been used as the foundation for theories and models of the world. Since Hayes [21] introduced the use of ontology in AI, current research on ontology use can be found throughout the computer science community in areas such as computational linguistics and database theory. The areas that are being researched range from knowledge engineering, information integration, and object-oriented analysis to applications in medicine, mechanical engineering, and geographic information systems. Ontology has been proposed to play a central role in driving all aspects and components of an information system, leading to ontology-driven information systems [20], and in the specific case of GIS, leads to what we call Ontology-Driven Geographic Information Systems (ODGIS). The use of explicit ontologies will contribute to improve information systems. Since every information system is based on an implicit ontology, when we make the ontology explicit we avoid conflicts between the common-sense ontology of the user and the mathematical concepts in the software, and conflicts between the ontological concepts and the implementation [22].

This paper describes a framework for integrating geographic information based on ontologies. The use of different levels of ontologies leads to the integration of different levels of geographic information from the semantic point of view. The remainder of this paper is organized as follows. Section 2 introduces the abstract paradigm to understand ODGIS, discusses the use of object orientation in ontology representation, and show the different levels of ontologies. Section 3 introduces the basic framework for ontology-driven geographic information systems. Section 4 shows the semantic perspective of information granularity in the ODGIS framework. In section 5 we present the guidelines for implementation. Section 6 presents conclusions and future work.

2. A Foundation for Ontology-Driven GIS

In order to understand how people see the world and how ultimately the mental conceptualizations of the apprehended geographic features are represented in a computer system, we must develop abstraction paradigms. The result of the abstraction process is a general view of the process that goes from the real object to its computer representation. The use of different levels of abstraction allows the development of specific tools for the different types of problems at each level. For instance, Frank [23] considers that an ontology constructed from tiers can integrate different ontological approaches in a unified system. He suggests five tiers: *human-independent reality*, *observation of physical world*, *objects with properties*, *social reality*, and *subjective knowledge*. We introduce the *five-universes paradigm*, which builds on the four-universes paradigm [24, 25], by adding new components and explaining some of the concepts from the point of view of the geographic world.

The development of computational representations of the geographic world has been the subject of much study in the last decade [26]. In assembling our view of the world we build on previous explanations on how people see and mentally represent the world [24, 27-29]. Each of the five levels in our abstraction model deals with conceptual characteristics of the geographic phenomena of the real world. The first two levels, the physical level and the cognitive level, are only briefly described here. This work is concerned mainly with the last three levels, the logical level, the representation level, and the implementation level. Once a level is understood, we are able to face the problems of the next level.

The five universes are the *physical universe*, the *cognitive universe*, the *logical universe*, the *representation universe*, and the *implementation universe* (Figure 2-1). A geographic phenomenon in the physical world is captured by the cognitive system of a person and is classified and stored in the human mind. The representation of the real world object in the human cognitive system is done within the cognitive universe. The formalization of the conceptualizations of the world in the human mind gives us explicit formal structures, the ontologies that are part of the logical universe. When we take into account the peculiarities of the spatial world–for instance, reference systems and conceptualizations such as fields and objects–we are dealing with the representation universe. The shift to the implementation universe is made through the translation of the components of the representation universe into computer language constructions and data structures.

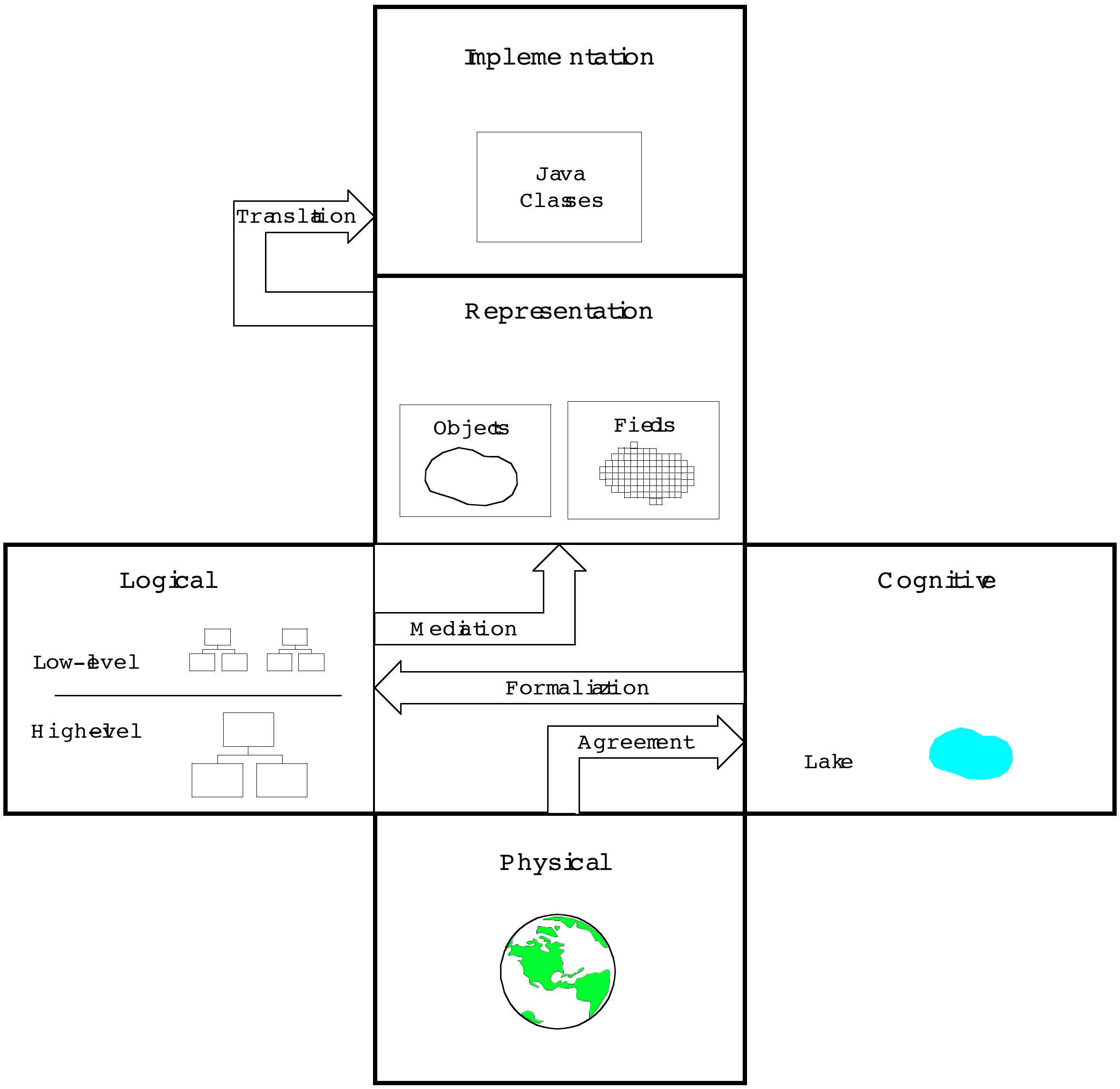

Figure 2-1 The five-universe-paradigm.

Our abstraction levels are based on the realistic view of the world, which considers that the physical world exists independently of our perception. Vegetation, rivers, and mountains are part of the real-world phenomena in which we are interested. In the realist perspective, the process of representation of geographical reality involves the assignment of concepts to elements of the physical world, by virtue of collective agreement of a community that shares common perceptions [30]. This process of collective agreement enables the connection between the physical universe and the cognitive universe. Through this process, concepts that correspond to real world objects are formed within a community of experts. But these concepts are not merely stored in the mind in a haphazard way; they are organized in a logical framework [31]. When this framework is made explicit using logical methods, we obtain ontologies [32], which are the formal representations of the logical schemes of the human mind and belong to the logical universe.

The logical universe contains two types of ontologies. High-level ontologies contain the more general theories of the world, such as the general concepts of a theory of natural geography. Low-level ontologies are specializations of more general ontologies. They can be detailed descriptions of specific domains and of the tasks that deal with these domains. The logical universe is connected to the representation universe by *semantic mediators*.

The representation universe is where a finite symbolic description of the elements in the logical universe is made so that we can apply operations on them. Here the ontologies of objects and fields are defined as the basic conceptualizations of the geographic world. Also here is the place to deal with all the concerns related to how these concepts are captured from the real world and how they are measured. The ontologies present at the representation level and at the logical level can be translated into computer languages, generating classes that belong to the implementation universe.

The implementation universe includes computational elements, such as algorithms, vector and raster data structures, and classes in object-oriented languages. In this work we deal only with classes that are derived from entities in the ontologies.

ODGIS are built using software components derived from various ontologies. The ontologies and the software components are based on object orientation techniques. In the next section we describe some basic concepts in object orientation and their relationship to the basic components of the ODGIS framework.

2.1 Object Orientation inside the ODGIS Framework

The use of the object data model as the basic conceptualization of space has been discussed before in the literature. The issue of defining the geographic space is actually the issue of defining and studying the geographic objects, their attributes, and relationships [33]. The object view of the spatial world [34] avoids problems such as the horizontal and vertical partitioning of data [35], although objects can provide both, if necessary. Furthermore, an object representation of the geographic world offers many views of a geographic entity. Objects are also useful in zooming operations, because when we get closer to a scene, instead of seeing enlarged objects we see different kinds of objects [36-38]. These operations are performed through aggregation as in the case of a house constituted by walls and a roof, or a block formed by land parcels [35].

We model geographic phenomena using an object-oriented approach. This approach should not be mistaken with the conceptualization for the representation of the geographic world. The most accepted models for this representation are the *object* and *field* models [27, 28]. The *object model* represents the world as a surface occupied by discrete, identifiable entities with a geometrical representation and descriptive attributes. These objects are not necessarily related to a specific geographic phenomenon and they usually correspond to constructed features, such as roads and buildings. The *field model* views geographic reality as a set of spatial distributions over geographic space. Climate and vegetation cover are typical examples of geographic phenomena modeled as fields. Although this dichotomy has been subject to criticism [39], it has proven to be a useful frame of reference and has been adopted, with some variations, in the design of the current generation of GIS technology [40]. We accept this model and use it for the representation of geographic entities.

A *class* is an extension of the concept of an abstract type, a structure that represents a single entity, describing both its information content and its behavior. A class defines the structure and the set of operations that are common to a group of objects [41]. An *instance,* or *object*, represents an individual occurrence of a certain class. While the class is the type definition, an instance is the data structure represented in the memory of a computer and manipulated by a software system. In this work, the terms *object* and *instance* are used interchangeably. An object functions as a complex data structure that is capable of storing all of its data, along with information about the necessary procedures to create, destroy, and manipulate itself. In an object-oriented GIS, for instance, the separation of spatial and non-spatial attributes is avoided because everything is stored in the same structure.

The ability to hide from the user the internal structure of an object is called *encapsulation*. With encapsulation it is possible to manipulate the object's data only by using a set of predefined functions. This approach ensures data independence: the internal data structures used by the object can change without influencing the user's perception of them.

Inheritance is a classification mechanism in which a class can be the subclass of another (i.e., it incorporates the other's features in addition to its own). Features can be attributes, functions or rules. A subclass is called a descendant. A superclass is any class that is higher than the given class in the hierarchy. When a given class descends directly from only one superclass, it is called *single inheritance*; when a class descends from more than one immediate superclass, it is called *multiple inheritance* [42]. Multiple inheritance has benefits and drawbacks. For instance, any system that uses multiple inheritance must provide an adequate solution to problems such as name clashes (i.e., when features inherited from different classes have the same name). Although the implementation and use of multiple inheritance is non-trivial [43], its use in geographic data modeling is essential [34]. In order to avoid the problems of multiple inheritance and at the same time represent the diverse character of the geographic entities, we introduce the concept of *roles*.

An object is something–it has an identity [44]–but it can play different roles. Usually the notion of role is linked with change in time. An object is only one thing but it can play different roles during its lifetime. The use of roles in object orientation is reviewed in detail by Pernici [45], Albano *et al.* [46], Wong *et al.* [47], and Steimann [48]. The use of roles in the specification of ontologies is discussed in Guarino [32]. The concept of role as interfaces as we use in the implementation of this work is reviewed in Steimann [49].

One of the most common uses of roles is to represent changes in an object during its lifetime. The typical example is of a person that plays the roles of a student, a parent, and a member of a club. In this work roles help to express different points of view of the same phenomenon. One community may see a certain phenomenon `X` and consider that `X` is a occurrence of an entity `A`. Another community may classify the same phenomenon `X` as being `B`. For this second community, `B` may also play a role of `A`.

The main objective of using roles in this work is to employ them as a tool to connect different ontologies. Therefore we use here a more unrestrained definition of roles than other authors [32] who argue that roles should have their own hierarchy and can only subsume or be subsumed by another role. Some authors consider that an object can play a role only if the role is a subtype [50] or a supertype [51] of the object. This point of view is not adopted here, because for us a role is an entity in another ontology. Each community has a right to its own point of view and information must be integrated on that basis, hence the use of a flexible specification of role. A more rigid specification would require, for instance, a habitat to be a subclass of a geographical region. As a consequence, in a biologist's ontology, a habitat would not be an entity but only a role. Using a more flexible specification of role we can allow a habitat to be an entity. In this specific point of view, a habitat has an identity and all the attributes that characterize an entity as being distinct from other entities. In our framework every role is an entity. An entity plays roles that are entities in other ontologies. We consider roles and the entities that play them to be parts of separate and independent hierarchies.

As an example, for a biologist a habitat can play the role of a lake or the role of woods near the lake. Some authors would argue that habitat is only a role and should be always played by a geographic location. We do not agree with this argument. In our framework a habitat is an entity in a biologist's ontology. The biologists work with the entity habitat having all the characteristics of a lake. They can also use a role of lake. The lake entity can be reused while avoiding the redefinition of all its properties. Using lake as a role instead of as a superclass gives the biologist more flexibility. They can also define that habitat inherits from a more related entity in the biological sense, thus avoiding a too strong dependence on the geographic point of view. Another reason for using lake as a role is to make it easier to obtain metadata and data from other sources.

A role can be viewed in different ways [48]. First, a role is viewed as a named relationship. This point of view stresses that roles exist only within some particular context. Second, a role can be viewed as a specialization or a generalization. The problem with this point of view is that it contradicts Guarino's [52] and mixes the dynamic nature of the role concept with the rigid properties of a type hierarchy. Finally, roles can be represented as adjunct instances. According to this point of view, roles are considered to be totally dependent on the instances that play them and do not carry their own identity. The object and its roles form an aggregate.

The extraction of roles and the resulting generation of a new instance of a class can be classified by what is called in the literature as *object migration* or *dynamic reclassification* [53, 54]. The term migration is used to model the change from one role to another in systems in which class membership is the main mechanism for assigning roles. Dynamic reclassification by role-based systems enable objects to dynamically change their membership in types and classes. This concept can be extended into *multiple classification*, (allowing an object to be an instance of multiple classes), *dynamic reclassification*, (allowing an object to gain and lose class memberships throughout the object's lifetime), and *dynamic restructuring*, (allowing an object's structure to change dynamically throughout the object's lifetime) [55].

2.2 Ontology Levels

There is a distinction between coarse and fine-grained ontologies. A coarse ontology consists of a minimal number of axioms and is intended to be shared by users that already agree on a conceptualization of the world. A fine-grained ontology needs a very expressive language and has a large number of axioms. Coarse ontologies are more likely to be shareable and should be used on-line to support the system's functionality. On the other hand, fine-grained ontologies should be used off-line, because they are accessed eventually for reference purposes [20]. Our solution allows the user to incrementally go from coarse to fine-grained ontologies on-line, thus eliminating the division between on-line and off-line ontologies.

In this work we use the term *low-level* ontologies for fine ontologies that represent very detailed information and *high-level* ontologies for coarse ontologies that represent more general information. Thus, if a user is browsing high-level ontologies he or she should expect to find less detailed information. We propose that the creation of more detailed ontologies should be based on high-level ontologies, so that each new ontology level incorporates the knowledge present in the immediate higher level. These new ontologies are more detailed, because they refine general descriptions of the level from which they inherit.

Ontologies are classified in four groups, according to their dependence on a specific task or point of view [13].

- *Top-level ontologies* describe very general concepts. In ODGIS a top-level ontology describes a general concept of space. For instance, a theory describing parts and wholes [56], and their relation to topology, called mereotopology [57], is at this level.
- *Domain ontologies* describe the vocabulary related to a generic domain, which in ODGIS can be remote sensing or the urban environment.
- *Task ontologies* describe a task or activity, such as image interpretation or noise pollution assessment in ODGIS.
- *Application ontologies* describe concepts that depend on both a particular domain and a task, and are usually a specialization of them. In ODGIS these ontologies are created from the combination of high-level ontologies. They represent the user needs regarding a specific application, such as an assessment of lobster abundance in the Gulf of Maine.

Representing geographic entities–either constructed features or natural differentiations on the surface of the earth–is a complex task. They are not merely located in space, they are intrinsically tied to space [58]. For instance, boundaries that seem simple can in fact be very complex. An example is the contrast between soil boundaries, which are fuzzy, and land parcels, whose boundaries are crisp. Users who are developing an application can use the accumulated knowledge of experts that have specified an ontology of boundaries instead of dealing with these complex issues by themselves. The same is true for ontologies that deal with geometric representations, land parcels, and environmental studies. Users should be able to create new ontologies building on existing ontologies whenever possible. An example of a *backbone taxonomy*, which represents the most important properties in a high-level ontology is given in Figure 2-2 [59].

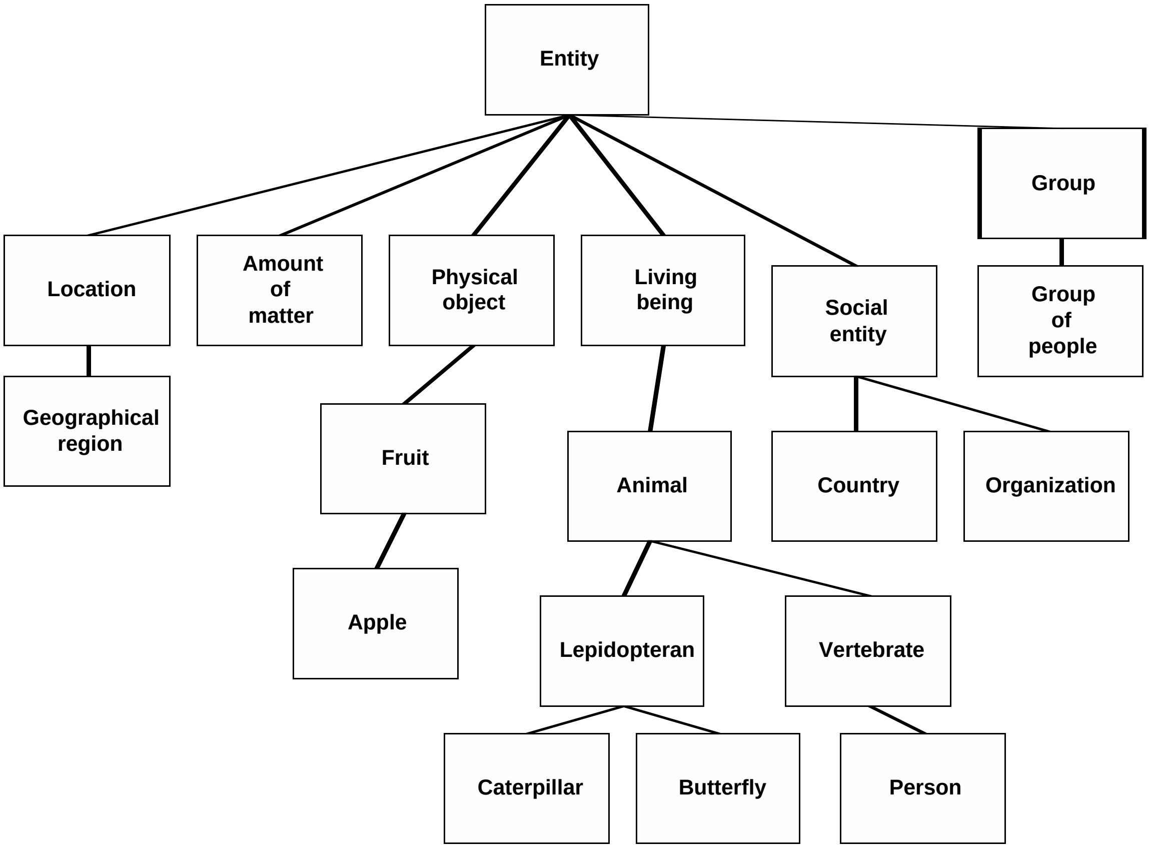

Figure 2-2 A basic taxonomy, from Guarino and Welty [59].

If a local government is starting a GIS project based on ontologies, we can use a basic urban ontology such as [60]:

- The geographic coverage of the local government area;
- The people within the area;
- The buildings and facilities;
- The business activities;
- The land itself.

Instead of defining these five main branches in detail, the users could use the backbone taxonomy introduced above and, from it, start their own ontology. A sample result can be seen in Figure 2-3 where the class *People* is derived from the class *Person*, *Business* is derived from *Organization*, and *Land* is derived from *Geographical region.* At the same time, if the urban ontology is general enough, it can be used as the foundation for other local government projects.

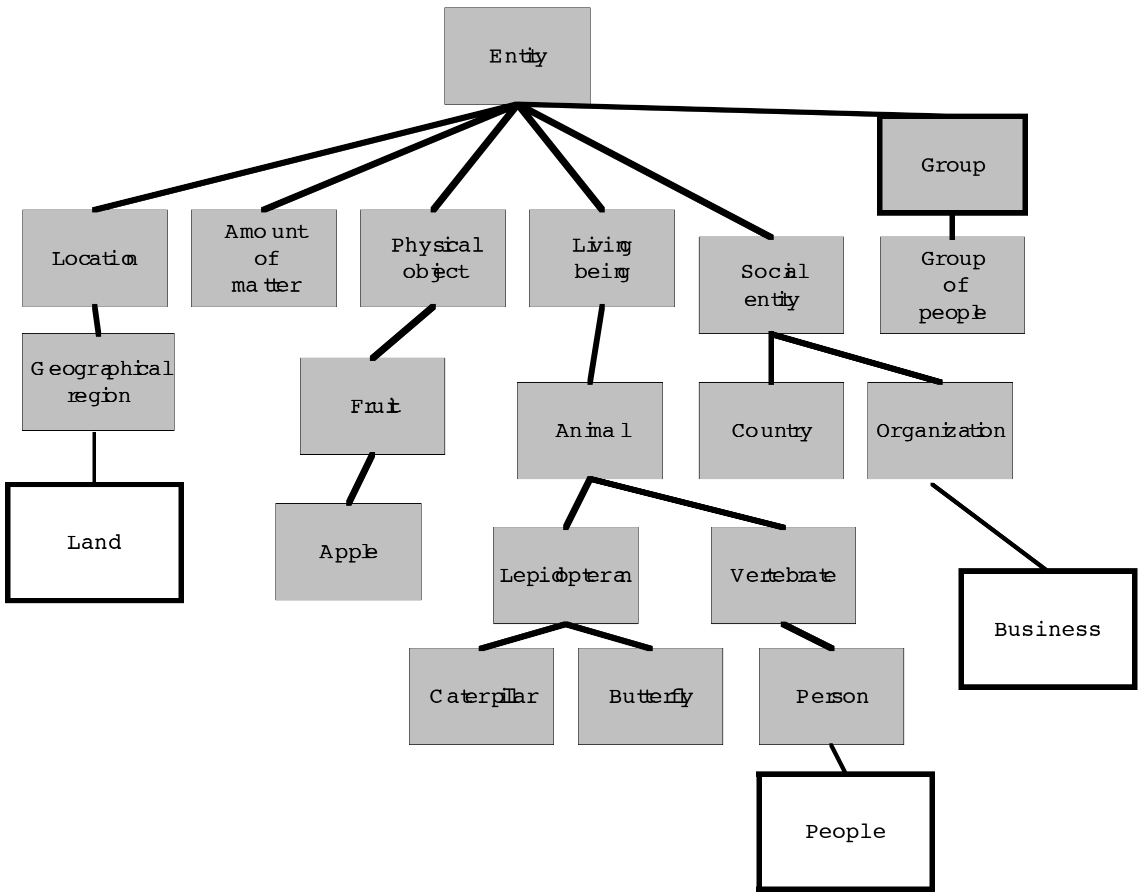

Figure 2-3 Deriving new classes from a high-level ontology.

An application developer can combine classes from diverse ontologies and create new classes that represent user needs. In this way, a class that represents *Building* in the urban ontology can be built from *Physical object* in the basic taxonomy. At the same time, *Building* can be seen as a location and can also hold a social entity or an organization. Thus, *Building* can play the roles of *Location* and *Organization* extracted from the urban ontology. Thus the real class is *Building*, but it plays many roles (Figure 2-4) that, together, give the class its unique characteristics.

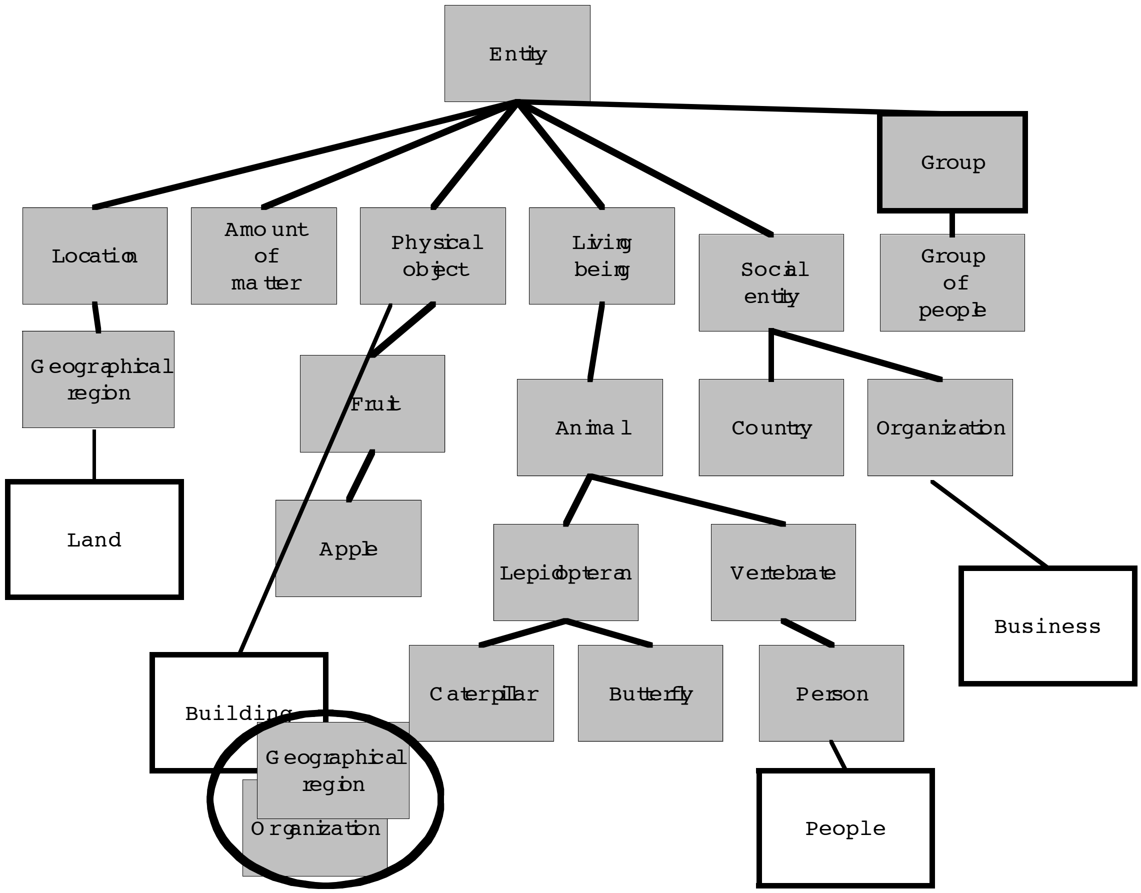

Figure 2-4 A class can play many roles.

3. Ontology-Driven Geographic Information Systems

In this section we introduce the multiple-ontology approach for ontology-driven geographic information systems. This approach enables the reuse of knowledge and a better understanding of the geographic phenomena. Two kinds of ontologies for the geographic world are introduced. One is called Phenomenological Domain Ontology and aims at capturing the different dimensions and internal properties of the geographic phenomena. The other type is concerned with the description of specific subjects and tasks and is called the Application Domain Ontology. The multi-ontology approach leads to bi-directional integration of geographic information.

3.1 A Multiple-Ontology Approach

Ontologies for the geographic world, or geo-ontologies, can be divided in two types. One type is the *Phenomenological Domain Ontology* (PDO). This ontology captures the different dimensions and internal properties of the geographic phenomena. This specific ontology is distinct and independent from the other type, the *Application Domain Ontology* (ADO). This ontology is concerned with description of specific subjects and tasks that the GI scientists use as a source of information.

Since the PDO is concerned with how the geographic phenomenon can be captured and represented by computer systems, it is located in the representation universe. The ADO is part of the logical universe because it deals with the description of the phenomenon itself, where it fits in the world, and how it can be best described. The connection between PDO and ADO is made by semantic mediators (Figure 3-1).

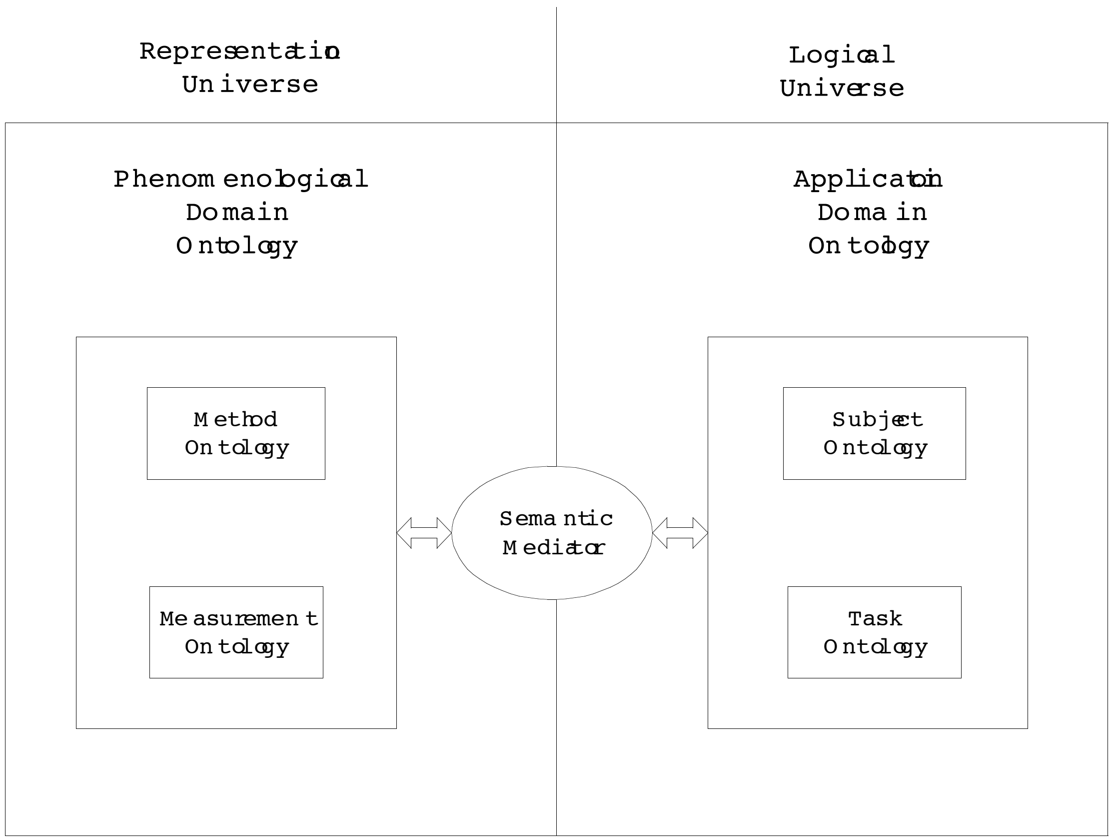

Figure 3-1 Phenomenological and application ontologies

One of the objectives of separating geo-ontologies in PDO and ADO is to emphasize the detection of *spatio-temporal configurations* of geographic phenomena. In a single time instance, the set of matchings of a concept from the application domain ontology to an instance of a concept on the phenomenological ontology is called a *spatial configuration*. Given a temporal sequence of geographic phenomena, the set of spatial configurations is called a spatio-temporal configuration. This idea is consistent with the identity-based modeling of change [44], where object identity is proposed as a central notion for modeling spatial-temporal change. The framework allows an object, identified as part of the user ontology, to be related to different descriptions in the PDO, because of changes in the object during a time series. Consider, for example, mapping urban sprawl for a city by analyzing a 20-year time series of LANDSAT images. The geometries that describe the evolution of the urban boundaries of the city change continually, are recorded annually, and yet the identity of the object remains the same.

Another objective is to be able to reuse elements of the same ontology in different applications. With this separation we make clear what are the specific methods and what are the more general ones. The specific methods can be reused for similar phenomena, while the general ones have a broader use. A simple example is the case of detecting or extracting line segments from a series of images. *Line segment* is a concept that is part of the structural ontology of the image. It has clearly defined geometric properties. These lines can take different roles in domain ontologies of different user communities. Another example is that all the methods for spatial analysis over polygons available on the PDO side can be reused for every application on the ADO side.

Each geographic object is unique as a concept in the logical universe and above. Although we choose different conceptualizations to represent it–*objects* and *fields*–its nature does not change. For instance, a reservoir is a reservoir, regardless of whether it is represented by an aerial photograph, a polygon, or a digital terrain model. Figure 3-2 shows a reservoir represented in three different ways.

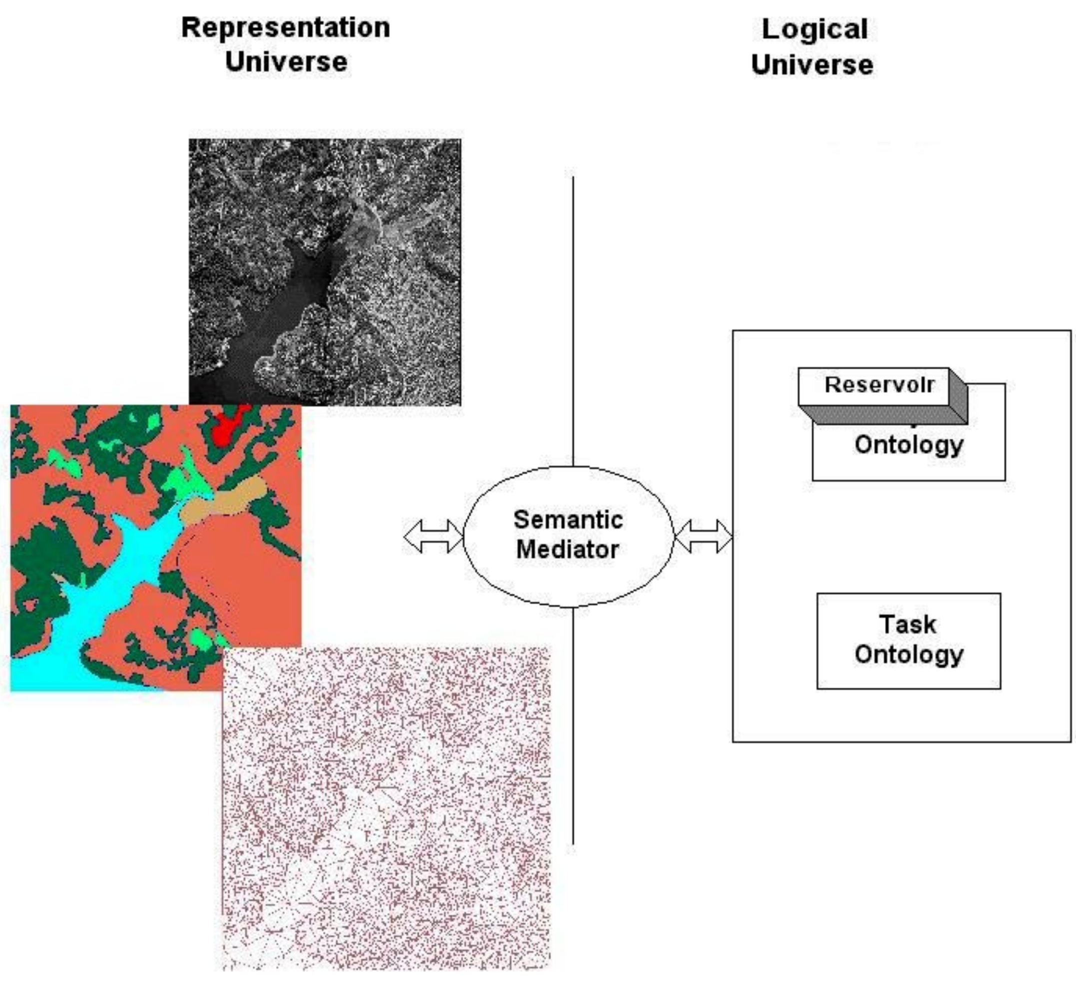

Figure 3-2 Three different representations of reservoir.

The representations are located in the representation universe, while the concept and its formal description are located in the logical universe. The concept reservoir is described only once in a high-level ontology, such as a natural-geography ontology, but it can be linked to more than one element in the PDO, (i.e., one for each of the different representations mentioned above).

3.2 Bi-Directional Integration

One of the main objectives of this work is to integrate geographic information from different sources. The various geospatial information communities have different views of the world. These views can be formalized in different ontologies. Therefore, it is necessary to accommodate multiple ontologies, which in our model lie inside both the logical universe and inside the representation universe.

We introduce here two different ways to integrate ontologies. The first is the integration inside a subject, and is called *vertical integration*. The other kind of integration is called *horizontal integration*, and involves integrating ontologies of different subjects (Figure 3-3).

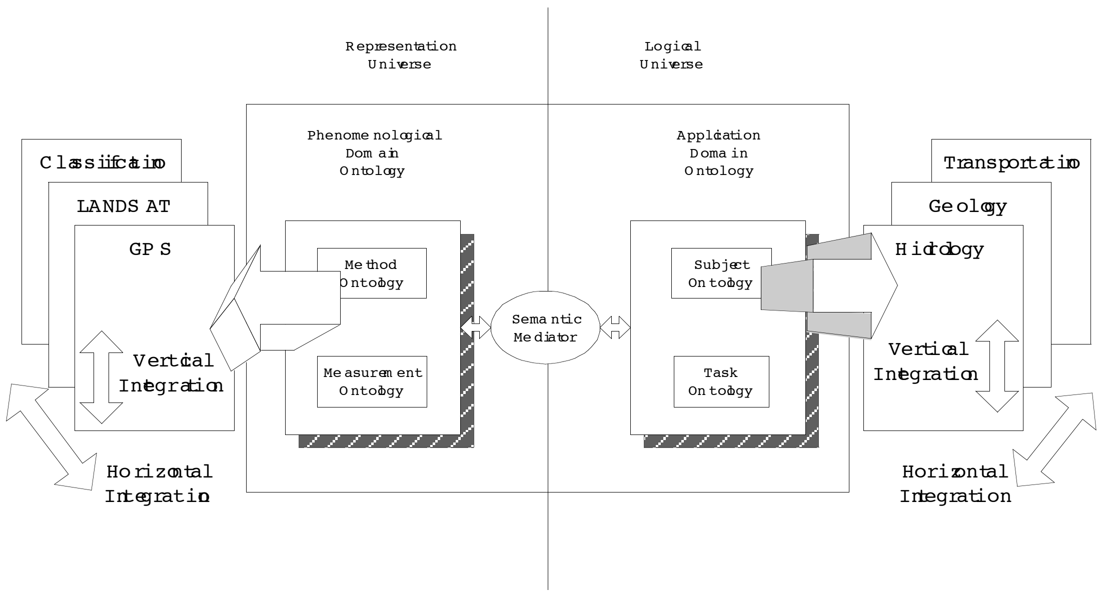

Figure 3-3 Horizontal and vertical integration.

When a new ontology is specified, it is necessary to have a set of operations that allow the reuse of previous ontologies or parts of them. In an ODGIS environment three operations are available: *inheritance*, *inclusion*, and *roles*. Inheritance is used for vertical integration and roles are used for horizontal integration. Inclusion can be used for both integrations.

Classes in ODGIS are defined hierarchically, taking advantage of *inheritance*. It is possible to define more general classes, containing the structure of a generic type of object, and then specialize these classes by creating subclasses. The subclasses inherit all properties of the parent class and add some more of their own. *Roles* are used to get around problems with multiple inheritance. In multiple inheritance for instance, a geographic feature can be at the same time a lake and a tourist attraction. In ODGIS we represent this entity as a lake that plays the role of a tourist attraction. Later on, the lake can be perceived as an environmentally protected area, that is, it can take on yet another role. Thus, in ODGIS an entity can have many roles.

Inclusion is an operation in which an entity of an ontology is used to specify any part of an entity in a new ontology. For instance, an ontology that deals with representations of spatial objects will include many parts from a geometry ontology.

The integration operations are used in different stages of the ontology specification process. This separation happens because the levels of detail are different at the many stages of ontology specification. We suggest the use of inheritance in the high-level ontology integration and inheritance and roles at the low-level integration. Inclusion is used in every level of integration.

The multi-level ontology approach generates a very flexible model. In order to exploit this flexibility, we need a specific model for navigation among the diverse entities. We choose to develop the navigation model in the implementation universe. Since the classes extracted from the ontologies are in this level, the navigation model is based on the change of classes.

4. Change of Granularity in ODGIS

The ODGIS framework can be presented based on two main aspects: *knowledge generation* and *knowledge use*. The knowledge-generation phase comprises the specification of the ontologies using an ontology editor, the generation of new ontologies from existing ones, and the translation of the ontologies into software components.

The knowledge-use phase of an ODGIS relies on products developed in the previous phase: a set of ontologies specified in a formal language and a set of classes. The ontologies are available to be browsed by the end user, and they provide metadata on the available information. A set of classes that contains data and operations constitutes the system's functionality. These classes are linked to geographic information sources through the use of mediators. In this section we will discuss the operations of generalization and specification over the instances of the classes. The operations described here are applied over instances of the classes, the real objects with data and operations.

Information in ODGIS is treated as instances of classes. The classes are modeled from ontologies. The instances are created by mediators that extract data from databases and shape it according to the model of classes defined in the ontologies. In ODGIS, we observe changes of granularity in the model in which the ontologies are represented, and in the use of the system for information retrieval when we are dealing with the instances of the classes. In this section we review in detail what was suggested for change of granularity of instances in Fonseca *et al.* [61].

There are two types of changes of classes in ODGIS. The first type occurs when an instance of a class immediately above or immediately below is generated from a given class. We call this transformation *vertical navigation*. The second type occurs when one of the roles played by the object is extracted from one instance. This way a new instance is generated producing a new object that belongs to the class of the role. We call this transformation *horizontal navigation* (Figure 4-1).

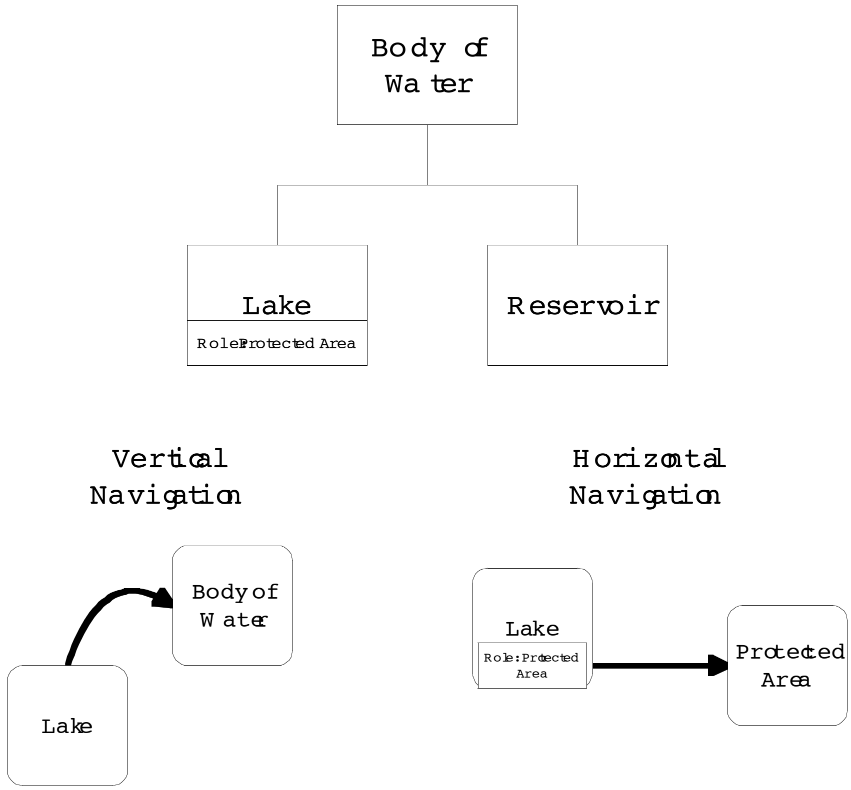

Figure 4-1 Vertical and horizontal navigation in an ontology of bodies of water.

Vertical navigation implies a change of level of detail, because it produces a new instance with more detail or one with less detail than the original instance. Horizontal navigation does not imply a change of the level of detail. The new class generated by horizontal navigation can be at any level in the hierarchy of classes.

4.1.1 Semantic Granularity in ODGIS

The abstraction of concepts and notions about real-world objects is an important part of the creation of information systems. In the abstraction process, certain characteristics of the objects are identified and coded in a model in such a way that the set of characteristics is representative of the much more complex real-world objects. Depending on the user's interest, however, this set of characteristics can be defined to be more or less detailed.

Some authors consider *granularity* in a spatial database to be the same as resolution, thus implying that granularity is related to the level of distinction between elements of a phenomenon that is represented by the dataset [62]. Hornsby [44] points out the difference between resolution and granularity. Resolution refers to the amount of detail in a representation, while granularity refers to the cognitive aspects involved in selection of features. This kind of granularity is called *semantic granularity*. The notion of granularity applied to GIS leads to studies of the variation in the representation of geographic objects and phenomena across a wide range of scales. Certain phenomena are scale-dependent, (i.e., their representation varies across the scales). For instance, if an urban settlement is perceived at a small scale, the level of detail is usually small enough for an entire city with all its complex internal structure to be represented as a point or as a simple polygon on a map. If the same city is perceived at a larger scale it becomes necessary to represent its internal structure with more detail, for instance depicting blocks, squares, major streets, and buildings. Considering a geographic database where two representations of the same phenomenon have to coexist, Beard [63] shows how it could be possible to maintain and update only the most detailed version of the objects and then to filter out unwanted detail to produce the less-detailed version. Here we work with a higher level of abstraction dealing with information systems instead of databases. In an ODGIS, a concept can have more than one representation. For instance, the usual concepts about a river are independent of how it is represented, whether as a network for transportation or as an important element of the environment of a region. In an ontology, a river is defined first by its general meaning. More specialized ontologies deal with representation issues later.

In the ODGIS framework there are different levels of ontologies. Accordingly, there are also different levels of information detail. Low-level ontologies correspond to very detailed information, and high-level ontologies correspond to more general information. Thus, if a user is browsing high-level ontologies he or she should expect to find less detailed information. We propose that the creation of more detailed ontologies should be based on the high-level ontologies, such that each new ontology level incorporates the knowledge present in the previous level. These new ontologies are more detailed, because they refine general descriptions of the level from which they inherit. We follow Hornsby's [44] approach because we consider that the level of semantic granularity is related to the level of ontology used. Ontologies can be used to specify how high-level abstractions relate to concepts in a lower level by establishing methods that help to implement rules and constraints.

4.1.2 The Mechanism for Changes of Granularity

There are two operations for changes in the level of detail: generalization and specialization. In generalization a class with a certain level of detail generates a new class with less detail. For instance, using Guarino's ontology (Figure 2-2), a `Geographical Region` can be generalized into a `Location`. Specialization is the inverse operation, in which a more general class is converted into a more specific class.

In ODGIS every class inherits from a basic class called `Object`. This specific class has two basic methods to be used in changes of granularity. One method is used to generalize new classes and it is called `Up`, and the other is used to specialize classes and it is called `Create_From`.

For example, if a user is dealing with instances of the class `lake` and of the class `reservoir`, the user can see and manipulate the instances of those objects as instances of `body of water`. This way the user is able to obtain better results in queries, retrieving more objects than if he had used only `lake` or `reservoir`.

In specialization we can consider the same example but in a different order. The user has an instance of `lake` but he/she is interested in using some methods only available for the class `reservoir` or the user wants to combine in a detailed fashion the data available about the class `lake` with the data available about the class `reservoir`. The solution presented here allows the user to generalize first the instances of `lake` into `body of water`, and then from this new set of instances, specialize them into `reservoir`.

4.1.3 Generalization and Specialization

The generalization operation implies generating a new instance of a class with less detail and less knowledge than the original instance. To perform this operation it is necessary to have knowledge about which data are going to be discarded and which data are going to be kept, or transformed. The best place to do this is inside the instance that has all the data of the object that is going to be generalized. The operation that performs the generalization is called `Up` and it implies changes not only to non-graphic data but also changes in representation formats. Generalizations of representation formats have been discussed elsewhere [64-66]. What is presented here is the framework in which this kind of operation can happen. ODGIS is a framework that enables the integration of existing knowledge, either at the logical level or at the representation level.

The specialization operation implies generating a new instance of a class with more detail and, therefore, with more knowledge embedded in it. In order to accomplish specialization we choose to place the method for specialization in the class that will receive the result of the operation. This choice was made because the know-how to perform this operation resides in the new class. Therefore, the class provides the methods and the rules for creating a new instance of itself from a more generic instance.

For example, if an instance of `reservoir` is going to be created, only the `reservoir` class knows all the details necessary to create an instance of itself. To create a class of `reservoir` from `lake` it is necessary that (1) an instance of `lake` creates an instance of `body of water; (2)` an empty instance of `reservoir` is created; and (3) the instance of `reservoir` populates itself with data from the instance of `body of water.` The result is an incomplete but working version of an instance of `reservoir`.

To make the instance of `reservoir` complete, the mediators have to look into the source of `reservoir` and then use similarity matching techniques [67] to try to match the new instance with available data. The result of this operation is a more complete instance. From the point of view of `lake`, this new instance is richer, because it has all the information that it had before as `lake`, plus the information retrieved by the mediator from the source of `reservoir`.

4.1.4 Role Extraction

In an ODGIS, an object can play many roles. The object cannot change its own class without losing its identity, but it can play different roles depending on the context. In order to provide the user with the ability to work with these different roles we introduced the concept of *horizontal navigation*. The result is the creation of a new instance of the class of the role played by an object. One of the roles played by the object is extracted, i.e., one new instance is available for the user.

This kind of operation is not a specialization or generalization, since a role can be seen as being at the same level of the classes that contain it, instead of being at the level of their superclasses or subclasses. For instance, a `lake` can play the role of a `link` in a transportation network. The ontology of bodies of water and the ontology of transportation can be at the same level.

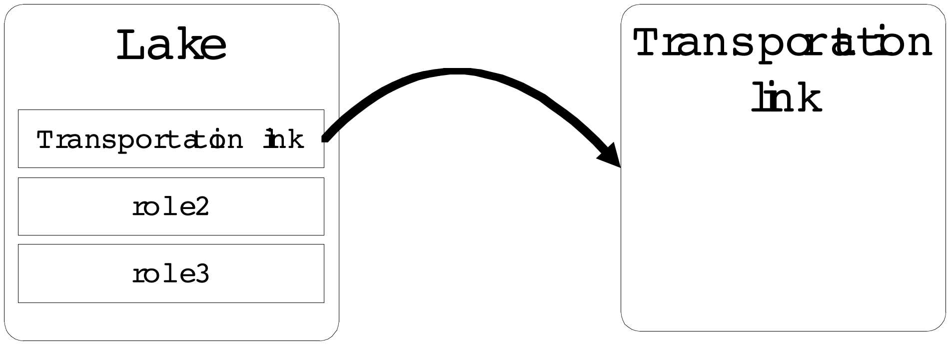

Figure 4-2 Role extraction.

The slots for roles are defined in the general class `object`. The rules and methods for generating an instance of a role should be provided in this class. The method for extracting a role is called `extract`. For instance, the syntax to extract the role `link` from `lake` is: `new object link = lake.extract(link)`.

5. Guidelines for Implementation

In this section we analyze the options for implementation of the main components of an ODGIS. We are suggesting here specific tools for implementation. We know that these tools are not the only solutions, but the evolution of ontology-driven information systems will lead to the use of similar tools, or to an evolution of these tools.

An ontology-driven information system deals with instances of classes called objects. These objects are extracted from geographic databases and carry data and operations. One of the most suitable options for implementing interoperable objects [68] or components that need to share both code and data across a heterogeneous network is the use the programming language Java [69, 70], because compiled Java code (bytecode) can be executed by Java interpreters available on most computers. Furthermore, the object-oriented structure of Java offers many features for the implementation of distributed objects.

5.1 The Ontology Editor

The ontology editor allows users to work on the specification of ontologies. After the ontology is specified, the user may query and update the ontologies using remote applications on the Internet.

The set of ontologies is represented in a hierarchy. The components of the hierarchy are classes modeled by their distinguishing features–*parts*, *functions*, and *attributes* (Figure 5-1). This structure for representing ontologies is extended from Rodríguez [67] with the addition of *roles*. Roles allow for a richer representation of geographic entities and avoid the problems of multiple inheritance.

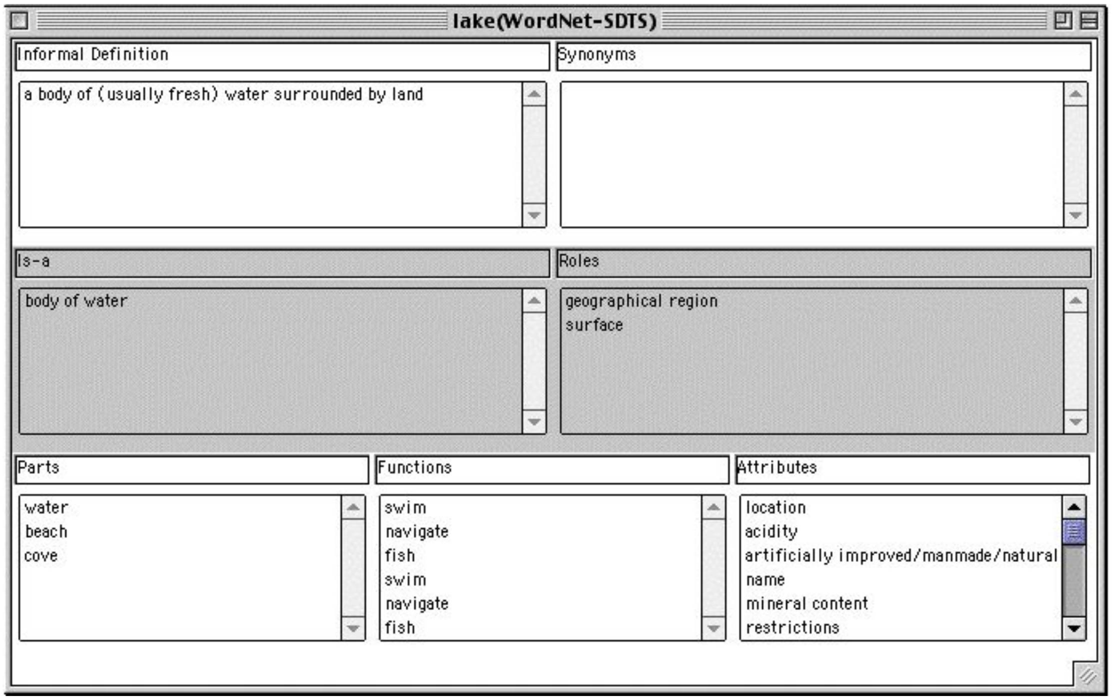

Figure 5-1 Basic structure on an ontology class.

Once the ontology is specified, the ontology editor has facilities for translating ontologies from repositories into application environments. We use Java as the implementation language. The basic mechanism for inheritance in Java is through the use of the keyword extends. This mechanism allows a new class to inherit from only one parent class. The entities in the ontologies are translated into Java interfaces. A Java interface describes the set of public methods that a class that implements the interface must support, and also their calling conventions. But a Java interface does not implement those methods. Each descendant class has to provide the code for each existing interface method (Figure 5-2).

```
Public interface lake
{
Vector roles;
//parts
Object water;
Object cove;
//functions
public void swim();
public void navigate();
public void fish();
//attributes
public String location;
public String acidity;
public String artificially_improved;
public String name;
public String mineral_content;
public String restrictions;
public String temperature;
public String charted_depth;
}
```

Figure 5-2 A Java interface for lake.

5.2 The Ontology Browser

In the ODGIS approach, the application program relies on classes derived from ontologies. These classes can be as simple as one entity or as complex as a part of an ontology. The application developer is able to browse the ontology that is the origin of these classes. The ontology browser has two important functions. First, it can be used during ontology specification by users who wish to collaborate in composing a shared ontology. Second, once the ontology has been specified, the browser is used to show the available geographic entities to the users. Mediators connect entities in ontologies to features in spatial databases.

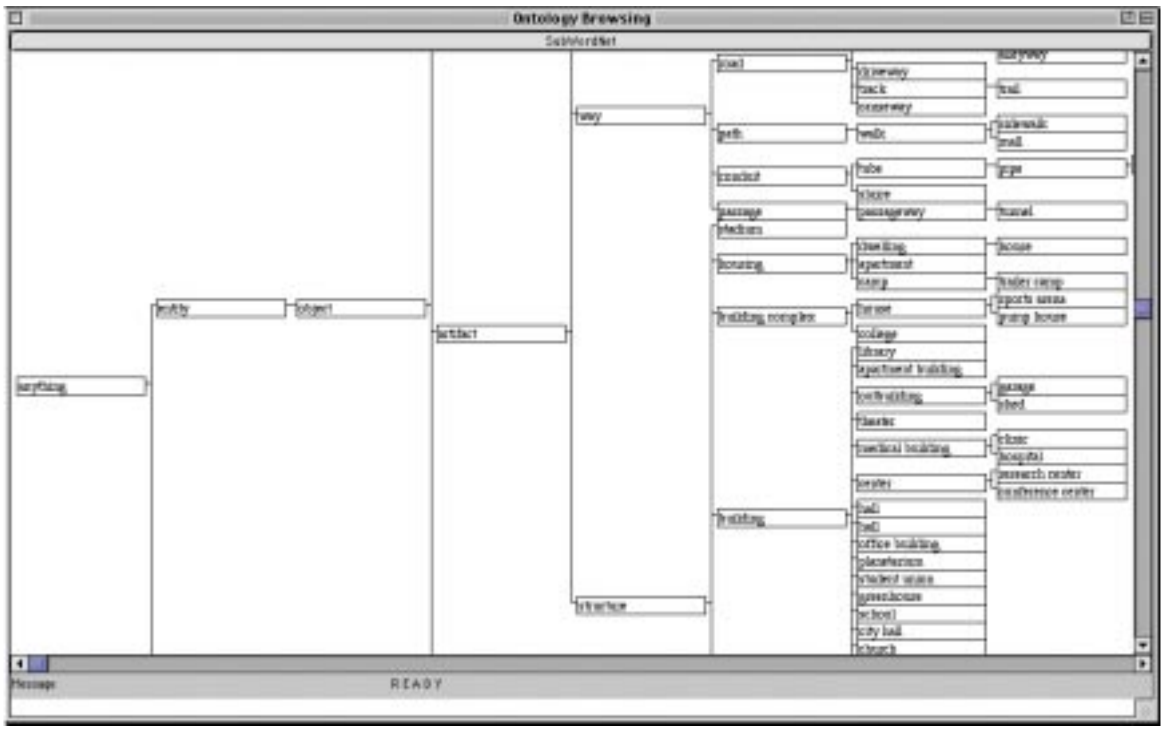

Figure 5-3 Browsing a top-level ontology.

For instance, a user wants to retrieve information about bodies of water of a determined region. First, the user browses the ontology server looking for the related classes. After that, the ontology server starts the mediators that look for the information and return a set of objects of the specified class. The results can be displayed (Figure 5-4) or can undergo any valid operation, such as statistical analysis.

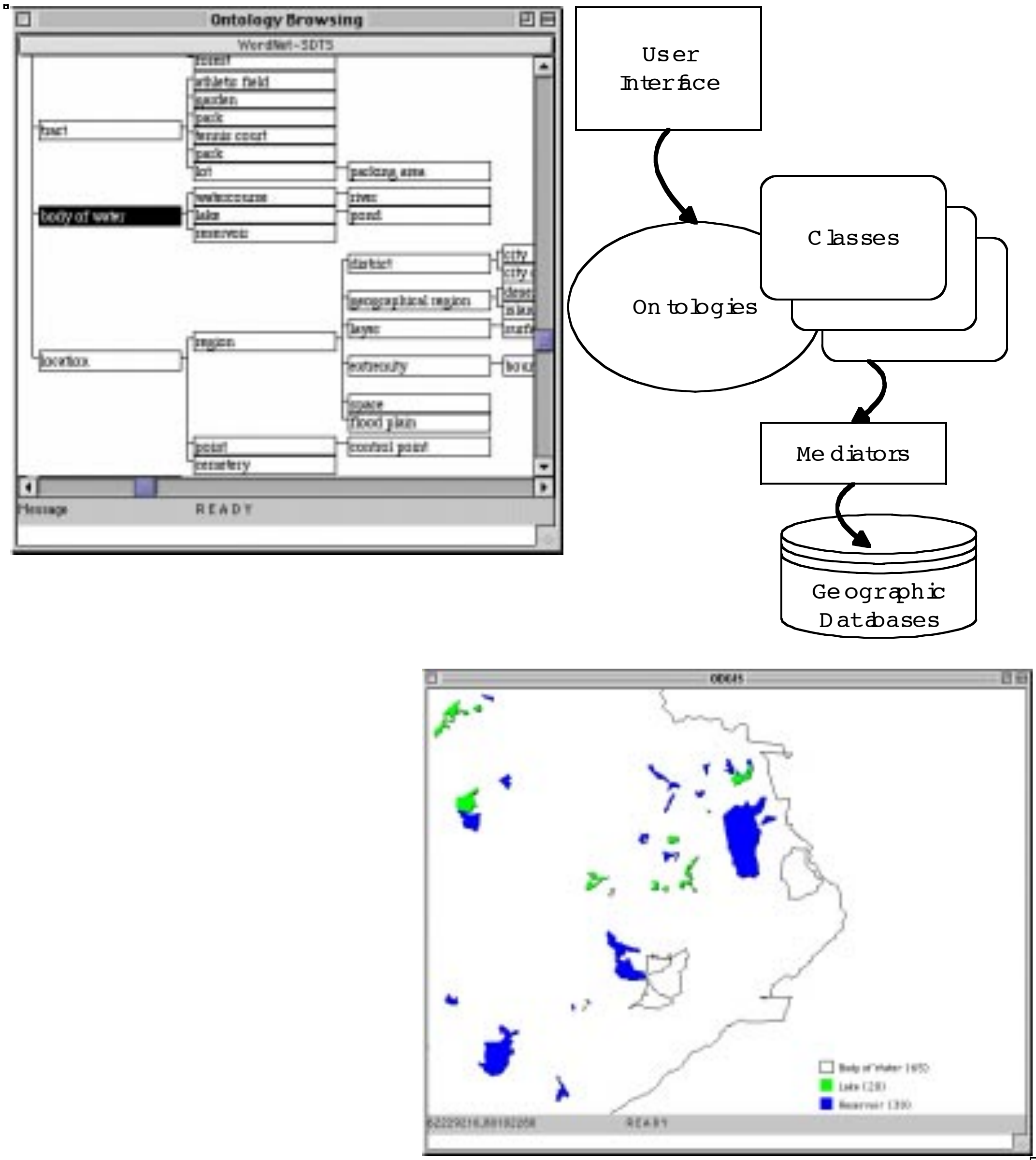

Figure 5-4 Schema for a query processing with an ODGIS.

5.3 Querying the System

The framework allows the user to browse at different levels of information. Ontologies are structured in a hierarchical way. This kind of organization leads to queries by level.

The entities chosen to be queried are `body of water`, `lake`, and `reservoir` (Figure 5-5).

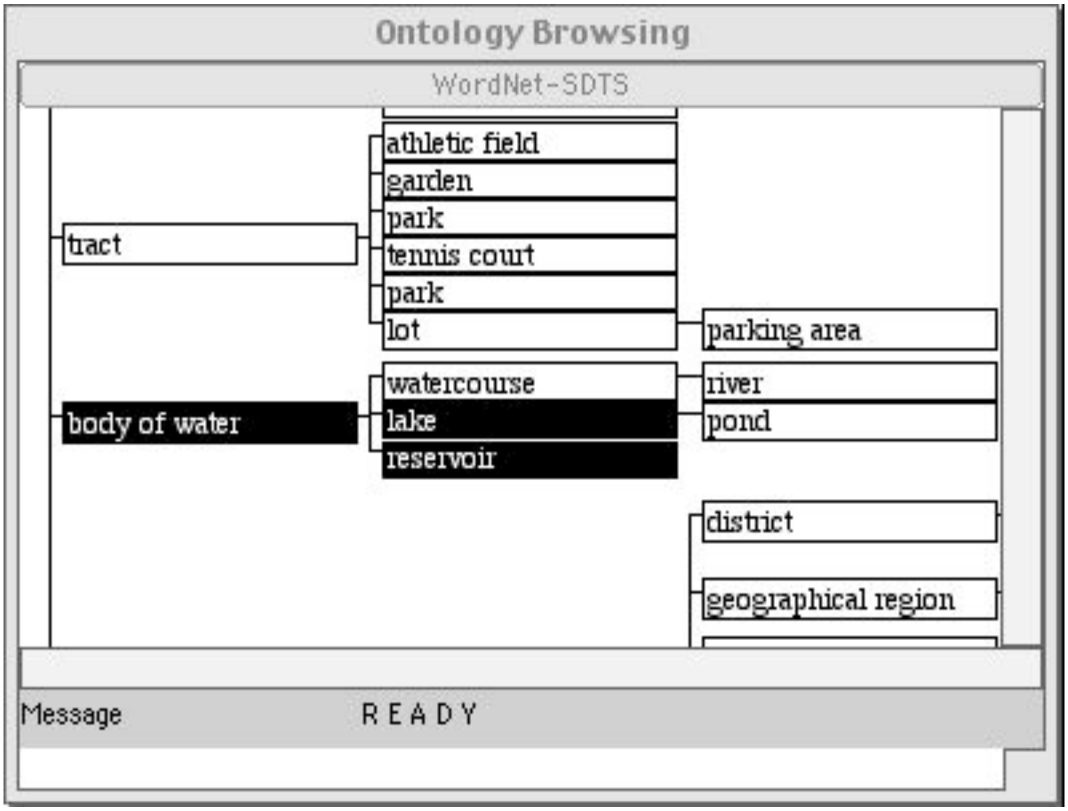

Figure 5-5 Query by level.

The user has to find the concepts in the ontology tree. The queries for `lake` presented the following result: 79 objects found (Figure 5-6). The query for `reservoir` is similar to the previous query. The result for `reservoir` was: 91 objects found (Figure 5-7).

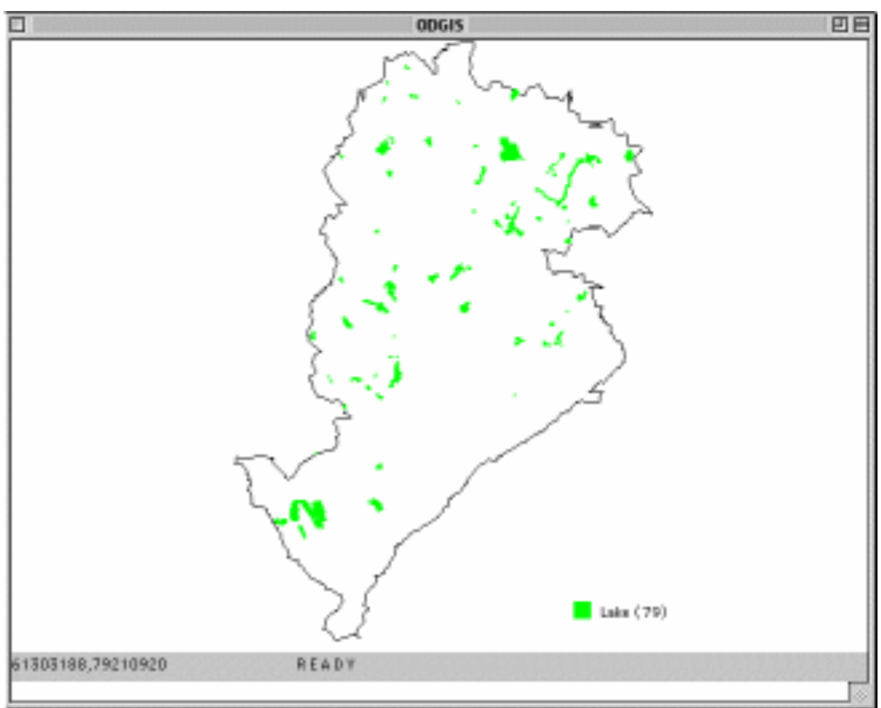

Figure 5-6 Query for `lake`.

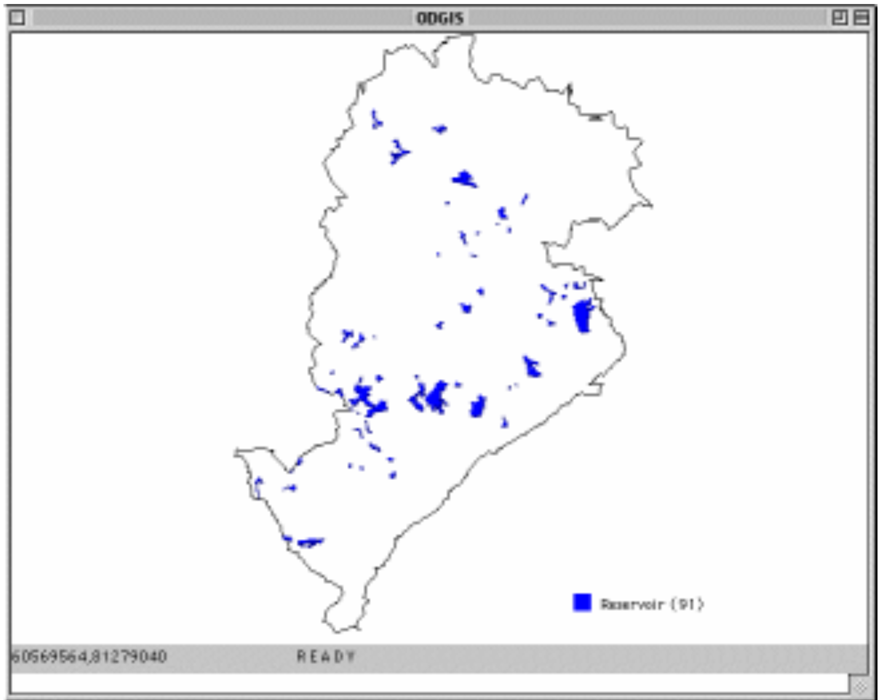

Figure 5-7 Query for `reservoir`.

When browsing the ontology of bodies of water, the user may choose to query for `body of water`. This entity is located one level higher than `lake` and `reservoir`, that is, it is necessary to explore the concept of `body of water`, finding that it includes both the concepts of `lake` and `reservoir`, thereby selecting both during the query. As a result, the query was performed at a high semantic level. The result of the query for `body of water` was: 176 objects found (Figure 5-8).

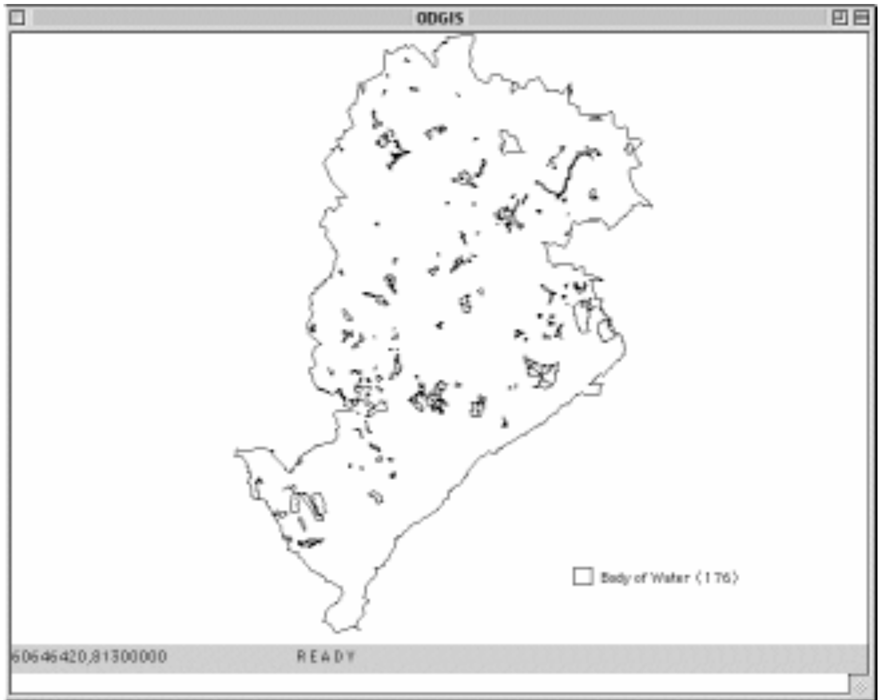

Figure 5-8 Query for `body of water`.

The results of the query using the semantic query for `body of water` can be compared against the results of the first two queries and against the result of the sum of these first two queries.

The query for `body of water` returned more objects (176) than the query for `lake` (79) and more than the query for `reservoir` (91). This result was expected and shows that with the semantic search broadened more adequate results are produced. These results correspond more closely to the user's notions about bodies of water, assuming that the concepts the user works with are adequately laid out as an ontology.

The expected results of a query for `body of water` could be the sum of the `lake` and `reservoir`, but we obtained a higher number (176) than that sum (170). This result has two explanations. Both show the strength of the semantic approach for geographic information integration.

First, one reason for retrieving 176 objects instead of 170 is that since we are in a high level in the hierarchy, other classes beyond `lake` and `reservoir` can be retrieved and classified as `body of water`, thus producing a broader result.

The second reason implies that, among the information systems integrated in this particular scenario of ODGIS, some of them can have information classified only at high conceptual levels, for instance, `body of water`. The are two reasons for this more generic classification to happen.

- Unclassified information collected from other sources.
- The source does not disclose the classification at a high level of detail. It only releases information at the lower semantic levels, because of security reasons or commercial purposes.

6. Conclusions

This work investigated new ways to integrate geographic information at different levels of detail. We chose to use ontologies as the foundation of the integration, because ontologies can represent real world entities using a sophisticated structure with components such as definitions, parts, functions, attributes, and rules of relationship. Furthermore, ontologies capture the semantics of information, can be represented in a formal language, and can be used to store related metadata. Ontologies can be used to establish agreements about diverse views of the world and consequently carry the meaning of the original ideas that are embedded in the representation of geographic phenomena in the human mind. The ontologies are linked to information sources through semantic mediators, therefore, the integration of ontologies leads to integration of information.

Our approach for integration of geographic information started from entities of the physical universe. This approach differs from usual approaches that start from the implementation and representation universes. Our approach enables the integration of information based on its semantics content instead of dealing primarily with data formats and geometric representations. In order to integrate information it was necessary to integrate first ontologies

The problem of the different levels of detail was approached by the introduction of a navigation mechanism that allows an object (i.e., the implementation of an ontology entity) to change its class by generalization or specialization. In a generalization, a more specific object drops some pieces of information and turns itself into a more general instance. In a specialization, a more general object gathers more information and becomes a more specific object. We also introduced the operation called role extraction, in which a role played by an object can be extracted and transformed into a new instance. This new instance acts as an independent object. Therefore, the new instance can be matched with an object associated with another entity in a different ontology.

We proposed the use of a special parent class that allows navigation from application ontologies to top-level ontologies, passing through domain and task ontologies. This navigation capability shortens the gap between generic and specialized ontologies, enabling the sharing of software components and information. ODGIS employs user classes that are derived through inheritance from various ontologies to approach heterogeneity issues.

An ontology editor and an embedded translator from entities to classes were developed to support the knowledge-generation phase of the architecture. For the knowledge-use phase, a user interface to browse ontologies was also developed, and the container of geographic objects was extended from Fonseca and Davis [71].

Acknowledgments

This work was partially supported by the National Science Foundation under NSF grant numbers SBR-9700465 and IIS-9970123, NASA/EPSCoR fellowship grant numbers 99-58 and EP-02-02 # 2001516, and an ESRI graduate fellowship. Max Egenhofer's research is further supported by NSF grants IRI-9613646, IIS-9970123, and EPS-9983432; by the National Imagery and Mapping Agency under grant numbers NMA202-97-1-1023 and NMA201-00-1-2009; by the National Institute of Environmental Health Sciences, NIH, under grant number 1 R 01 ES09816-01, and by a contract with Lockheed Martin. Gilberto Câmara's research has been partially supported by a joint CNPq (Brazil) / NSF (USA) grant on "Computational Issues in Interoperability in GIS" (CNPq process number 480322/99).

References

[1] P. Longley, M. Goodchild, D. Maguire, and D. Rhind, "Geographical Information Systems,". New York: John Wiley & Sons, 1999.

[2] H. Onsrud and G. Rushton, "Sharing Geographic Information,". New Brunswick, NJ: Rutgers: CUPR Press, 1995.

[3] OpenGIS, *The OpenGIS® Guide-Introduction to Interoperable Geoprocessing and the OpenGIS Specification*. Wayland, MA: Open GIS Consortium, Inc, 1996.

[4] A. Vckovski, "International Journal of Geographical Information Science - Special Issue: Interoperability in GIS,", vol. 12, pp. 297-298, 1998.

[5] M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, *Interoperating Geographic Information Systems*. Norwell, MA: Kluwer Academic Publishers, 1999.

[6] A. Sheth, "Changing Focus on Interoperability in Information Systems: from System, Syntax, structure to Semantics," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, pp. 5-29, 1999.

[7] A. Vckovski, "Preface," in *Interoperating Geographic Information Systems—Second International Conference, INTEROP'99*, vol. 1580, *Lecture Notes in Computer Science*, A. Vckovski, K. Brassel, and H.-J. Schek, Eds. Berlin: Springer-Verlag, pp. V-VI, 1999.

[8] USGS,"View of the Spatial Data Transfer Standard (SDTS) Document", 6/12/98, http://mcmcweb.er.usgs.gov/sdts/standard.html, 1998.

[9] M. Sondheim, K. Gardels, and K. Buehler, "GIS Interoperability," in *Geographical Information Systems 1 Principles and Technical Issues*, P. Longley, M. Goodchild, D. Maguire, and D. Rhind, Eds. New York: John Wiley & Sons, pp. 347-358, 1999.

[10] D. Mark, "Toward a Theoretical Framework for Geographic Entity Types," in *Spatial Information Theory*, vol. 716, *Lectures Notes in Computer Science*, A. Frank and I. Campari, Eds. Berlin: Springer-Verlag, pp. 270-283, 1993.

[11] V. Kashyap and A. Sheth, "Semantic Heterogeneity in Global Information System: The Role of Metadata, Context and Ontologies," in *Cooperative Information Systems: Current Trends and Directions*, M. Papazoglou and G. Schlageter, Eds. London: Academic Press, pp. 139-178, 1996.

[12] Y. Bishr, "Semantic Aspects of Interoperable GIS,". The Netherlands: Wageningen Agricultural University, pp. 154, 1997.

[13] N. Guarino, "Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration.," presented at Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97, Frascati, Italy, 1997.

[14] E. Mena, V. Kashyap, A. Illarramendi, and A. Sheth, "Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure," in *Formal*

Ontology in Information Systems, N. Guarino, Ed. Amsterdam: IOS Press, pp. 269-283, 1998.

[15] M. Gahegan, "Characterizing the Semantic Content of Geographic Data, Models, and Systems," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, pp. 71-84, 1999.

[16] F. Harvey, "Designing for Interoperability: Overcoming Semantic Differences," in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, pp. 85-98, 1999.

[17] A. Rodríguez, M. Egenhofer, and R. Rugg, "Assessing Semantic Similarity Among Geospatial Feature Class Definitions," in *Interoperating Geographic Information Systems—Second International Conference, INTEROP'99*, vol. 1580, *Lecture Notes in Computer Science*, A. Vckovski, K. Brassel, and H.-J. Schek, Eds. Berlin: Springer-Verlag, pp. 1-16, 1999.

[18] G. Wiederhold, "Interoperation, Mediation and Ontologies," presented at International Symposium on Fifth Generation Computer Systems (FGCS94), Tokyo, Japan, 1994.

[19] T. Gruber, "A Translation Approach to Portable Ontology Specifications," Knowledge Systems Laboratory, Stanford University, Stanford, CA, Technical Report KSL 92-71, 1992.

[20] N. Guarino, "Formal Ontology and Information Systems," in *Formal Ontology in Information Systems*, N. Guarino, Ed. Amsterdam, Netherlands: IOS Press, pp. 3-15, 1998.

[21] P. Hayes, "The Naive Physics Manifesto," in *Expert Systems in the Microelectronic Age*, D. Michie, Ed. Edinburgh, Scotland: Edinburgh University Press, pp. 242-270, 1978.

[22] A. Frank, "Spatial Ontology," in *Spatial and Temporal Reasoning*, O. Stock, Ed. Dordrecht, The Netherlands: Kluwer Academic Publishers, pp. 135-153, 1997.

[23] A. Frank, "Ontology to Separate Tiers of Agreement," presented at GIScience 2000—First International Conference on Geographic Information Science, Savannah, GA, 2000.

[24] J. Gomes and L. Velho, "Abstraction Paradigms for Computer Graphics," *The Visual Computer*, vol. 11, pp. 227-239, 1995.

[25] G. Câmara, A. Monteiro, J. Paiva, J. Gomes, and L. Velho, "Towards a Unified Framework for Geographical Data Models," presented at Geoinfo 2000—II Workshop Brasileiro de Geoinformatica, São Paulo, Brazil, 2000.

[26] M. Egenhofer, J. Glasgow, O. Günther, J. Herring, and D. Peuquet, "Progress in Computational Methods for Representing Geographical Concepts," *International Journal of Geographical Information Science*, vol. 13, pp. 775-796, 1999.

[27] H. Couclelis, "People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS," in *Theories and Methods of Spatio-Temporal Reasoning in*

Geographic Space, vol. 639, *Lecture Notes in Computer Science*, A. U. Frank, I. Campari, and U. Formentini, Eds. New York: Springer-Verlag, pp. 65-77, 1992.

[28] M. Goodchild, "Geographical Data Modeling," *Computers and Geosciences*, vol. 18, pp. 401-408, 1992.

[29] A. Requicha, "Representations for Rigid Solids: Theory Methods and Systems," *ACM Computing Surveys*, vol. 12, pp. 437-464, 1980.

[30] J. Searle, *The Construction of Social Reality*. New York: The Free Press, 1995.

[31] D. Bryant and B. Tversky, "Internal and External Spatial Frameworks for Representing Described Scenes," *Journal of Memory and Language*, vol. 31, pp. 74-98, 1992.

[32] N. Guarino and C. Welty, "A Formal Ontology of Properties," in *Proceedings of EKAW-2000: The 12th International Conference on Knowledge Engineering and Knowledge Management*, vol. 1920, *Lecture Notes in Computer Science*, R. Dieng and O. Corby, Eds., pp. 97-112, 2000.

[33] J. Nunes, "Geographic Space as a Set of Concrete Geographical Entities," in *Cognitive and Linguistic Aspects of Geographic Space*, D. Mark and A. Frank, Eds. Norwell, MA: Kluwer Academic Publishers, pp. 9-33, 1991.

[34] M. Egenhofer and A. Frank, "Object-Oriented Modeling for GIS," *Journal of the Urban and Regional Information Systems Association*, vol. 4, pp. 3-19, 1992.

[35] W. Kuhn, "Are Displays Maps or Views?," presented at AUTO-CARTO 10, Tenth International Symposium on Computer-Assisted Cartography, Baltimore, MD, 1991.

[36] M. Tanaka and T. Ichikawa, "A Visual User Interface for Map Information Retrieval Based on Semantic Significance," *IEEE Transactions on Software Engineering*, vol. SE 14, pp. 666-670, 1988.

[37] S. Timpf and A. Frank, "Using Hierarchical Spatial Data Structures for Hierarchical Spatial Reasoning," in *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '97*, vol. 1329, *Lecture Notes in Computer Science*, S. Hirtle and A. Frank, Eds. Berlin: Springer-Verlag, pp. 69-83, 1997.

[38] G. Volta and M. Egenhofer, "Interaction with GIS Attribute Data Based on Categorical Coverages," in *Spatial Information Theory, European Conference COSIT '93*, vol. 716, *Lecture Notes in Computer Science*, A. Frank and I. Campari, Eds. New York: Springer-Verlag, pp. 215-233, 1993.

[39] P. Burrough and A. Frank, "Spatial Conceptual Models for Geographic Objects with Undetermined Boundaries,". London: Taylor & Francis, 1996.

[40] G. Câmara, R. C. M. Souza, U. M. Freitas, and J. C. P. Garrido, "SPRING: Integrating Remote Sensing and GIS with Object-Oriented Data Modelling," *Computers and Graphics*, vol. 20, pp. 395-403, 1996.

[41] B. Meyer, *Object-Oriented Software Construction*. New York: Prentice Hall, 1988.

[42] L. Cardelli, "A Semantics of Multiple Inheritance," in *Semantics of Data Types*, vol. 173, *Lecture Notes in Computer Science*, G. Kahn, D. McQueen, and G. Plotkin, Eds. New York: Springer-Verlag, pp. 51-67, 1984.

[43] E. Tempero and R. Biddle, "Simulating Multiple Inheritance in Java," Victoria University of Wellington, School of Mathematical and Computing Sciences, Wellington, New Zealand CS-TR-98/1, May 22, 1998 1998.

[44] K. Hornsby, "Identity-Based Reasoning about Spatio-Temporal Change," in *Spatial Information Science and Engineering*. Orono: University of Maine, pp. 164, 1999.

[45] B. Pernici, "Objects with Roles," presented at IEEE/ACM Conference on Office Information Systems, Cambridge, MA, 1990.

[46] A. Albano, R. Bergamini, G. Ghelli, and R. Orsini, "An Object Data Model with Roles," presented at 19th International Conference on Very Large Data Bases, Dublin, Ireland, 1993.

[47] R. Wong, H. Chau, and F. Lochovsky, "A Data Model and Semantics of Objects with Dynamic Roles," presented at 13th International Conference on Data Engineering, Birmingham, UK, 1997.

[48] F. Steimann, "On the Representation of Roles in Object-Oriented and Conceptual Modelling," *Data & Knowledge Engineering*, vol. 35, pp. 83-106, 2000.

[49] F. Steimann, "Roles = Interfaces: A Merger of Concepts," in *Journal of Object-Oriented Programming*, vol. (in press)., 2001.

[50] C. Bock and J. Odell, "A More Complete Model of Relations and Their Implementation: Roles," *Journal of Object-Oriented Programming*, vol. 11, pp. 51-54, 1998.

[51] D. C. Halbert and P. D. O'Brien, "Using Types and Inheritance in Object-Oriented Languages.," in *Proceedings of ECOOP'87 European Conference on Object-Oriented Programming*, vol. 276, *Lecture Notes in Computer Science*, J. Bézivin, J.-M. Hullot, P. Cointe, and H. Lieberman, Eds., pp. 20-31, 1987.

[52] N. Guarino, "Concepts, Attributes and Arbitrary Relations," *Data and Knowledge Engineering*, vol. 8, pp. 249-261, 1992.

[53] A. O. Mendelzon, T. Milo, and E. Waller, "Object Migration," presented at Proceedings of the 13th ACM SIGACT—SIGMOD—SIGART Symposium on Principles of Database Systems, Minneapolis, MN, 1994.

[54] J. Su, "Dynamic Constraints and Object Migration," presented at 17th International Conference on Very Large Data Bases, Barcelona, Spain, 1991.

[55] H. A. Kuno and E. A. Rundensteiner, "The MultiView OODB View System: Design and Implementation," *TAPOS - Theory and Practice of Object Systems*, vol. 2, pp. 202-225, 1996.

[56] R. Casati and A. Varzi, *Parts and Places*. Cambridge, MA: MIT Press, 1999.

[57] B. Smith, "On Drawing Lines on a Map," in *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '95*, vol. 988, *Lecture Notes in Computer Science*, A. Frank and W. Kuhn, Eds. Berlin: Springer Verlag, pp. 475-484, 1995.

[58] B. Smith and D. Mark, "Ontology and Geographic Kinds," presented at International Symposium on Spatial Data Handling, Vancouver, BC, Canada, 1998.

[59] N. Guarino and C. Welty, “Ontological Analysis of Taxonomic Relationships,” in *Proceedings of ER-2000: The 19th International Conference on Conceptual Modeling*, vol. 1920, *Lecture Notes in Computer Science*, A. Laender and V. Storey, Eds., pp. 210-224, 2000.

[60] W. Huxhold and A. Levinsohn, *Managing Geographic Information System Projects*. New York: Oxford University Press, 1995.

[61] F. Fonseca, M. Egenhofer, C. Davis, and K. Borges, “Ontologies and Knowledge Sharing in Urban GIS,” *Computer, Environment and Urban Systems*, vol. 24, pp. 232-251, 2000.

[62] J. Stell and M. Worboys, “Stratified Map Spaces: A Formal Basis for Multi-resolution Spatial Databases,” presented at International Symposium on Spatial Data Handling, Vancouver, BC, Canada, 1998.

[63] K. Beard, “How To Survive A Single Detailed Database,” presented at AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography, Baltimore, MD, 1987.

[64] M. Bertolotto and M. Egenhofer, “Progressive Vector Transmission,” presented at 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, 1999.

[65] C. Davis and A. Laender, “Multiple Representations in GIS: Materialization Through Map Generalization, Geometric and Spatial Analysis Operations,” presented at 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, 1999.

[66] C. Parent, S. Spaccapietra, and E. Zimanyi, “MurMur: Database Management of Multiple Representations,” presented at The AAAI-2000 Workshop on Spatial and Temporal Granularity, Austin, TX, 2000.

[67] A. Rodríguez, “Assessing Semantic Similarity Among Spatial Entity Classes,” in *Spatial Information Science and Engineering*. Orono, ME: University of Maine, pp. 182, 2000.

[68] M. Betz, “Interoperable Objects,” *Dr. Dobb's Journal*, vol. 4, pp. 22-26, 1994.

[69] P. Clemens, “Coming Attractions in Software Archictecture,” Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU/SEI-96-TR-008, January 1996 1996.

[70] S. M. Lewandowsky, “Frameworks for Component-Based Client/Server Computing,” *ACM Computing Surveys*, vol. 30, pp. 4-27, 1998.

[71] F. Fonseca and C. Davis, “Using the Internet to Access Geographic Information: An OpenGis Prototype,” in *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, Eds. Norwell, MA: Kluwer Academic Publishers, pp. 313-324, 1999.