PERGAMON

Computers, Environment and Urban Systems
27 (2003) 455–479

Computers, Environment and Urban Systems

www.elsevier.com/locate/compenvurbsys

Derivation and implementation of a semantic GIS data model informed by principles of cognition

Jeremy L. Mennis*

Department of Geography, UCB 260, University of Colorado, Boulder, CO 80309, USA

Received 2 October 2001; received in revised form 15 January 2002; accepted 22 January 2002

Abstract

The purpose of this research is to develop a new kind of semantic GIS data model that is better able to represent users' conceptual models of geographic domains than the conventional vector and raster data models. To this end, I look to the principles of cognition, how humans represent geographic information in their minds, to inform the development of this semantic data model. A three-stage methodology for the derivation and implementation of the semantic data model is presented. First, a conceptual framework of geographic cognition is developed. This framework incorporates principles of cognitive categorization and 'top-down' and 'bottom-up' information processing. Second, by replacing those elements of cognition with their database representation counterparts, often drawing from the object-oriented modeling paradigm, a semantic data model is derived. Finally, a prototype implementation of the semantic data model is presented using the Java programming language and the object-oriented database *Poet* as a development platform. This implementation utilizes a rule-based approach for representing categorical information and for the extraction of semantic entities from observational data.

Keywords: Semantic representation; Cognition; Data modelling; Knowledge discovery; Object-oriented GIS

* Tel.: + 1-303-492-4794; fax: + 1-303-492-7501.
E-mail address: jeremy@colorado.edu (J.L. Mennis).

PII: S0198-9715(02)00019-4

1. Introduction

A number of authors have noted that the conventional vector and raster data models for geographic information systems (GIS) are inadequate for the representation of many complex geographic phenomena (Burrough & Frank, 1995; Peuquet, 1994; Worboys, Hearnshaw, & Maguire, 1990). These inadequacies stem from the legacy of the development of both data models as implementation-driven digital storage solutions for geographic data previously stored using paper maps. As a result, the representations offered by these data models are limited to the familiar point, line, polygon, and grid cell geometric primitives.

The consequences of these representational limitations have become more problematic as the use of GIS has evolved from a tool for data management (for, say, retrieving parcel information) to a tool that is used for more sophisticated applications such as environmental modeling, decision support, and knowledge discovery. These types of applications are often concerned with geographic entities that are dynamic and complex, properties that cannot be represented adequately using the conventional vector and raster data models. Thus, there has arisen a *semantic gap* in GIS database representation—a gap between the database user's conceptual model of a given application domain and the representation of that conceptual model within the GIS.

A number of authors have looked to object-oriented modeling to address this semantic gap and other related issues of representation in GIS databases (e.g. Worboys et al., 1990). Many of these efforts, however, have focused on the representation of complex geometry and geometric relationships as extensions to the vector data model (e.g. Milne, Milton, & Smith, 1993; Tang, Adams, & Usery, 1996), and thus have not focused explicitly on conceptual representation. As another approach to the issue of GIS database representation, some researchers have drawn from the field of ontology. In the context of information science, ontology addresses how the salient features of an application domain may be symbolized in a formal manner (Guarino, 1995). An ontology of a particular domain specifies those entities and their properties that are deemed important for the purpose of representation in a computational environment. Research in ontology in the context of GIS therefore addresses how GIS users' conceptual models of geographic domains may be elicited, formalized, and represented within a GIS context (Mark, Egenhofer, Hirtle, & Smith, 2000). However, much of this work in ontology-driven GIS has focused on developing ontologies of geographic domains for purposes of data sharing and standardization (e.g. Fonseca, Egenhofer, Davis, & Borges, 2000), and has not been extended within the context of advanced GIS functions such as knowledge discovery.

Like previous research in object-oriented GIS and ontology-driven GIS, the purpose of this research is to address the semantic gap in GIS database representation. My objective here is to develop a new kind of GIS data model that is better able to represent users' conceptual models of geographic domains than the conventional vector and raster data models. However, unlike previous researchers I also aim to place this data model within the context of an overall database structure that supports sophisticated tasks such as knowledge discovery. To this end, I look to the principles of cognition, how humans acquire and represent geographic information in their minds, to inform the development of this semantic data model. While a number of researchers have investigated the relationship between select aspects of cognition and GIS (c.f. Egenhofer, Glasgow, Gunther, Herring, & Peuguet, 1999; Nyerges, Mark, Laurini, & Egenhofer, 1995), what has yet to be developed is a semantic GIS data model that is: (1) informed by principles of cognition, (2) generic to a variety of geographic domains, and (3) extended to a working implementation.

In previous research, I have outlined those principles of cognition that are most relevant to GIS database representation and provided a conceptual framework for incorporating those principles within a GIS data model (Mennis, Peuquet, & Qian, 2000). In the present paper, I present a prototype implementation of such a data model (hereafter the 'semantic data model') as well as a methodology for its derivation. This implementation is not intended to be comprehensive in its treatment of cognition, but rather serves as a proof-of-concept for how cognitive principles may be integrated and implemented within a GIS data model. Note also that it is not the purpose of this research to develop a GIS database that 'thinks' or 'behaves' like a person. I do not aim to remove the human from GIS by replacing the role of the human in knowledge discovery and analysis with the actions of the machine. Rather, the purpose of incorporating cognitive principles into GIS database representation is to allow for a more robust interaction between the human and computer by reducing the semantic gap between them. This is accomplished by developing database representations that make it easier for a person to project their own conceptualization of the geographic domain under investigation 'into' the database. In this way users may interact with the database via the way that they conceptualize the application domain, and do not have to conform to the relatively unintuitive data storage schemes used by the computer.

In the following section I review related research concerning how principles of cognition have been applied to GIS data modeling. The third section describes the derivation of the semantic data model as a process of formalizing cognitive principles within a computational context. The fourth section describes the implementation of the data model within a programming environment using an object-oriented database development platform. The final section, the conclusion, offers directions for future research.

2. Cognitive principles in GIS database design

The literature concerning cognition is clearly too vast to be reviewed in great detail here. Rather, in this section I focus on two principles of cognition that deserve particular attention in the development of a semantic data model: categorization and the distinction between percept and concept. I do not offer a thorough review of the fundamentals of these principles, as they have been reviewed extensively elsewhere in the cognitive science and (to some extent) GIS literature (Mark & Frank, 1996; Mennis et al., 2000; Pinker, 1984). Instead, I focus on how these principles have been addressed by GIS database researchers.

2.1. Categorization in GIS

Nyerges (1991) states that the entities that are represented in a GIS database are dependent upon the nature of the conceptual (i.e. cognitive) categories to which those entities are assumed to belong. He goes on to describe a number of different methods for defining a category, including set-based, probabilistic, and prototype approaches. Prototype categorization is dependent upon comparing objects in the environment to an ideal exemplar, the prototype, of that category (Rosch, 1978). While the prototype approach is most closely aligned with cognitive categorization, it is also more difficult to implement in a typical database environment because of the set-based nature of both the relational and object-oriented database models (Nyerges, 1991).

Usery (1993), however, contends that prototype categorization can provide the basis for the development of a 'feature-based GIS', a GIS that is founded upon the representation of real-world entities or features of application domains as described by database users' conceptual models. Usery (1993) suggests that the definition of these features be defined not by geometry, as in conventional GIS data models, but by the basic-level geographic categories found in the application domain. Usery (1993) notes that basic-level geographic categories may be identified by the fact that more attributes may be named for the basic-level than for superordinate and subordinate levels.

Usery (1993) demonstrates that a set-based categorization scheme for geographic data that exhaustively spatially partitions a region cannot be maintained because certain features may be considered members of two or more categories. For example, a river may be considered both a type of hydrography and a type of transportation. Usery (1993) proposes a method to solve this problem by separating the spatial representation of the feature from the categorical representation using object-oriented modeling, an approach which others in object-oriented GIS modeling have also advocated (e.g. Worboys et al., 1990). He notes that a feature may be recognized and instantiated as a member of a category by comparing the combination of attributes associated with the feature to the expected attributes of a category prototype, although the methodology for this approach is not specified.

2.2. Percept and concept in GIS

A distinction can be made between that that which is perceived directly by the senses, referred to here as *percept*, and that which is conceptualized in the mind, referred to here as *concept* (Neisser, 1987). The process of vision and visual cognition provides an excellent example of this distinction. Certainly, there is a difference between the physical process by which the human eye senses patterns of light and the cognitive aspect of how humans are able to identify objects in the environment using that visual input (Kosslyn, 1994). Of particular relevance here is the influential work of Marr (1982), who was interested in machine vision and the processing of visual input into the recognition of objects in the environment.

Of course, the process of cognition does not only proceed in a 'bottom-up' fashion—from percept to concept. Humans typically use prior knowledge about objects to interpret new visual input as well as to guide to what they give their attention (Jackendoff, 1992; Neisser, 1976). This process of using prior categorical knowledge to interpret the visual scene, going from concept to percept as opposed to vice-versa, can be referred to as a 'top-down' approach to visual information processing.

In the context of data exploration in GIS, Freksa and Barkowsky (1996: 118) note that the attempt to incorporate conceptual models within a database representation results in two potentially opposing representational goals for generic (i.e. not application specific) GISs: (1) the representation of knowledge that pertains to any potential application domain for which the GIS may be used, and (2) the representation of raw, uninterpreted data and the ability to 'construct' relevant knowledge from these data on demand. In other words, there is a distinction between the storage of raw, uninterpreted data and the knowledge that may be derived from these data.

The data/knowledge dichotomy may be understood as analogous to the cognitive percept/concept. In fact, similar to the cognitive knowledge acquisition process, knowledge may be related to data in a GIS context using both a 'bottom-up' and 'top-down' processing model—data may be inductively analyzed to 'construct' knowledge in the form of identified objects and categories or a priori categorical knowledge may be used to analyze the data and extract objects (Freksa & Barkowsky, 1996).

This combination of 'top-down' and 'bottom-up' information processing in GIS was also addressed by Peuquet (1994), who describes a conceptual framework for GIS database representation that integrates location-based (raster), object-based (vector), and time-based representations. Both the location-based and time-based representations are considered a 'rawer' form of information and may be interpreted using 'bottom-up' processing to inform the development of the object-based representation. Location- and time-based data may also be interpreted in a 'top-down' processing fashion using a priori knowledge stored in the object-based representation.

The data/knowledge dichotomy and the process of abstraction between them have also been suggested as the basis for integrating remote sensing systems and GIS technology (McKeown, 1987). Gahegan and Flack (1999) develop this idea to propose an approach to the integration of remote sensing systems and GIS. They present a model for this integration that is based on the process of abstraction from different data representation models, beginning with remotely sensed image data (low-level of abstraction) to the representation of geographic entities and their categorical properties (high-level of abstraction). The transformation of data between these representational models is defined by transformations of the data's individual properties as defined within each model, including the data values, spatial and temporal extent, uncertainty, and lineage.

3. Derivation of the semantic data model

This section and the following section describe the process through which the semantic data model is derived and implemented, respectively. This process is structured according to a methodology adapted from Howard and MacEachren's (1997) approach to interface and system design. In this approach there are three levels of abstraction through which the design process proceeds: conceptual, operational, and implementation (Fig. 1). The conceptual level addresses the overall goals and critical issues that the system is intended to address. The operational level is concerned with specifying the methodology for how these goals are to be met in a computational context. At the implementation level, the methods specified at the operational level are implemented in a software and hardware environment. Because my goal is to use the principles of cognition to inform knowledge representation, I begin at the conceptual level by describing a conceptual framework of geographic cognition. This conceptual framework embodies the overall representational goals that are subsequently addressed at the operational and implementation levels.

The operational level concerns the derivation of the semantic data model from the conceptual framework of cognition through the replacement of principles of cognitive representation with analogous principles of computational representation. While the semantic data model provides a relatively detailed view of the computational aspects of geographic representation, it is intended to be independent of any particular implementation environment. It is at the implementation level, presented in the following section, that this semantic data model is actually implemented via a software environment.

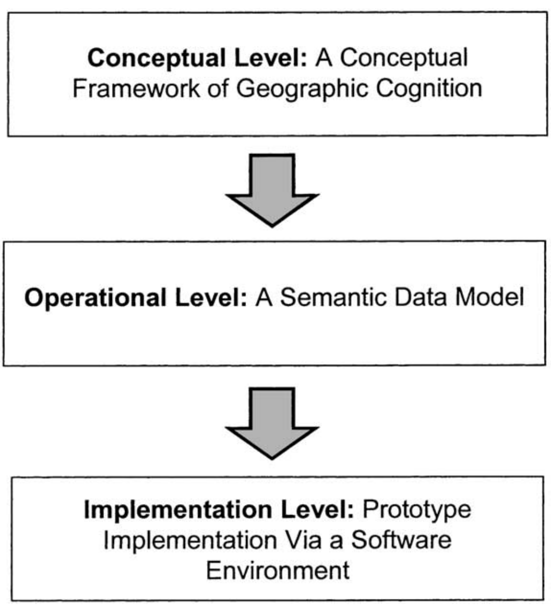

Fig. 1. Levels of abstraction in the methodology for the derivation and implementation of the semantic data model (adapted from Howard & MacEachren, 1997).

3.1. The conceptual level

Fig. 2 presents a conceptual framework of geographic cognition that incorporates those elements of cognition that I believe are most relevant to GIS database representation. In describing Fig. 2, I begin with the idea of the cognitive category, symbolized by the term 'Category' at the top of the diagram. Research in cognition indicates that categories are arranged in taxonomic hierarchies involving superordinate categories and more specific subordinate categories that inherit their superordinate's properties (Rosch, 1978). Thus, in Fig. 2 a category is associated with, and has a place within, a taxonomy of categories.

The cognitive science literature also indicates that each category has a schema, a set of expected properties associated with a category that allow a person to recognize an instance of that category in the environment (Neisser, 1976). Schemata therefore act as the categorical criteria for membership within a category, whether they take an image-like form (Kosslyn, 1994), a propositional form (Thorndyke, 1984), or some combination thereof (Tversky, 1993). While the specific expected properties vary from category to category, of course, there is evidence to suggest that certain properties are particularly important in geographic categorization, such as size, shape, and part-whole relationships (Smith & Mark 1998; Tversky & Hemenway, 1983).

As a simple example, consider the geographic category 'lake'. A lake fits within a taxonomic hierarchy—a lake may be considered within the superordinate category 'water body' and may have subordinate categories 'fresh water lake' and 'salt water lake'. A lake also has a set of expected properties associated with it. For instance, a lake can be considered to be 'composed of' a spatially contiguous region of water that covers the surface of the earth. This contiguous region of water must also be within a certain size range. A contiguous region of water that is only 20 feet in diameter would

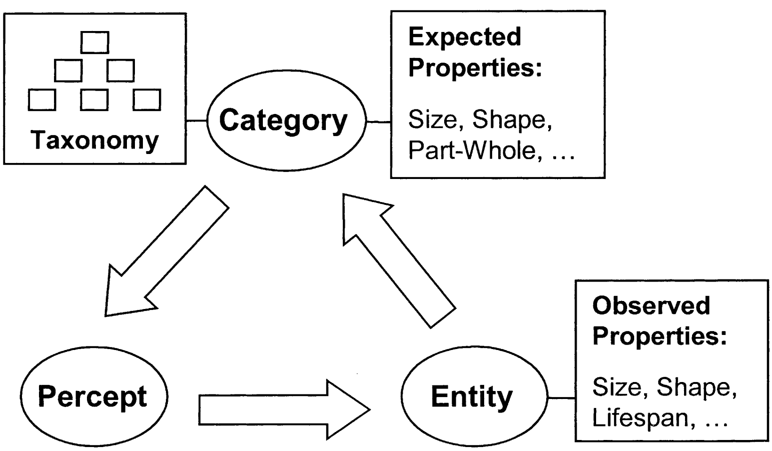

Fig. 2. A conceptual framework of geographic cognition.

probably not be considered a lake but a pond, while a water body 500 miles across may be considered a sea (example adapted from Smith & Mark, 1998).

In cognition, categorical information is used to interpret sensory perceptions (e.g. sights, sounds, smells, etc.), symbolized by the term 'Percept' in Fig. 2. The result of this interpretation is the recognition of individual entities that are observed in the environment, symbolized by the term 'Entity' in the figure. Each entity has its own observed set of properties of shape, size, etc. and is recognized as a member of a particular category. For instance, if we again consider the example of the category 'lake', an example of a percept would be a visual scene that a person sees when viewing a landscape. They may identify within this visual image an individual water body on the surface of the earth and visually estimate its apparent properties, such as its size. By comparing the observed size of the water body to the expected size for the category 'lake', the water body may be recognized as an individual lake, e.g. Lake Erie.

As Neisser (1976) notes, however, the expected properties of a category are not static but may evolve over time. Therefore, Fig. 2 shows another arrow pointing from 'Entity' back to the 'Category', indicating that the recognition of entities in the environment may modify the definition of a category. This new categorical definition may then be used to reinterpret sensory perceptions, resulting in a new set of observed entities, and so on in a continuous process of learning and interacting with the environment.

The use of categorical information to interpret sensory perceptions implies a 'top-down' perspective on acquiring geographic knowledge in which a priori knowledge is used to identify observed instances of a category in the environment. The conceptual framework of cognition shown in Fig. 2 also supports a 'bottom-up' view of learning, however, by simply beginning the knowledge acquisition process with sensory perception (Percept) and from the patterns embedded within the perception, extracting individual entities from the environmental scene. These individual entities that are recognized in the environment may then be grouped together based on similarities in properties to form individual categories. The expected properties of these categories would therefore be derived from the distribution of observed properties among the observed entities that are members of that category, as opposed to originating via some a priori knowledge source.

Clearly, geographic cognition as it actually occurs incorporates a combination of 'top-down' and 'bottom-up' processing. As humans, we are constantly processing our sensory perceptions to extract meaningful entities from the environment. As a first step in this process, we must use 'bottom-up' processing simply to distinguish the existence of individual entities within an environmental scene. We may then recognize these entities as instances of a particular category and use our lifetime of stored experiences as a priori categorical knowledge to further interpret the entities within a scene by defining what we *expect* to recognize in the environment and by *directing* our future interactions with our environment (Neisser, 1976). For instance, in the above example concerning the category 'lake', the feature recognition process requires the viewer to distinguish the presence of water as an entity distinct from other types of entities in the landscape, such as the land surface, before it can be recognized as a lake. This process can be considered three-fold: first, we distinguish via patterns of light and dark that there is a separate entity distinct from its surroundings, and second, this entity is identified as a water body. Third, by observing the water body's size and other properties (such as whether it is completely enclosed by land), we may decide that this water body is indeed a lake. Of course, our prior knowledge about the existence of water bodies directs our behavior and thus what we perceive through our senses.

While by no means does the conceptual framework presented in Fig. 2 provide an exhaustive depiction of geographic cognition, I believe it captures many of the principles of knowledge representation and acquisition that people use in their everyday interaction with the geographic environment. Notably, the conceptual framework captures certain key elements of cognition, including sensory perception and the role of schemata in the categorization process. This framework also recognizes that cognition is a continuous process, that categories are modifiable, and that the categorization process may proceed in an inductive ('bottom-up') or deductive ('top-down') manner. In addition, this framework addresses the distinction between percept and concept, with percept identified explicitly, and concept being composed of categories and entities.

3.2. The operational level

The task in progressing from the conceptual to the operational level in the context of this research is to develop a data model that is derived from the principles of cognition outlined at the conceptual level. I describe this procedure as a two-fold process. First, I demonstrate how the conceptual framework of geographic cognition described at the conceptual level may be translated into a database representation context by replacing those cognitive representation elements with their computational representation counterparts. Second, I describe how these computational representation elements may be specified in greater detail and integrated within a semantic data model.

3.3. Computational analogs to elements of cognitive representation

Fig. 3 describes a framework of representation similar to that presented in Fig. 2, but applied to GIS database representation as opposed to cognition. In this figure, the elements of cognition that are illustrated in Fig. 2 are replaced with their database representation counterparts, often drawing from the object-oriented modeling paradigm. First, note that the idea of category is replaced in Fig. 3 with the term 'Class'. A class in object-oriented modeling stands for a 'type-of' something, generally (though certainly not exactly) serving the same primary function as that of category in cognition. Classes in object-oriented modeling are arranged in generalization and aggregation hierarchies just as categories are analogously stored in taxonomic and partonomic hierarchies. And just as categories have expected properties via the use of schemata, a class may also store a set of attributes that describe that class. Note also that sensory perception, 'Percept' in Fig. 2, is replaced by

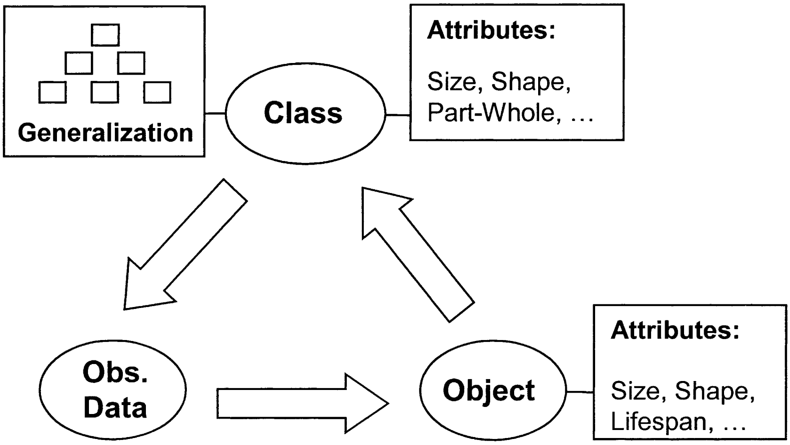

Fig. 3. Computational analogs to the cognitive representation elements described in Fig. 2.

observational data, symbolized by the term 'Obs. Data', in Fig. 3. In a geographic context, these observational data have a spatial and temporal reference. The idea of observational data being equated with percept extends from the work of Peuquet (1994) and Gahegan and Flack (1999) who suggest that a location-based array of observational data values, is a 'lower' form of geographic information abstraction compared to the representation of geographic objects. In this context, I consider observational data to be un-interpreted (as much as this is possible) measurements or observations taken from the environment, such as a data set of temperature readings or a remote sensing image of reflectance values.

The term 'Entity' in Fig. 2 is replaced by the term 'Object' in Fig. 3. An object in Fig. 3 refers to the computational representation of a real-world entity. Just as each entity in cognition is considered a member of a particular category, each object is assigned a membership to a particular class. By replacing each cognitive element of Fig. 2 with its computational analog, we may support a process of knowledge discovery similar in structure to that of geographic cognition. A class with attributes may be used to interpret observational data, the result of which is the recognition of individual objects. These objects, as members of a given class, may be used to refine the definition of the class, as outlined in the class attributes.

As an example, consider again the category 'lake'. In the context of applying the representational structure described in Fig. 3, there may be a class LAKE placed within a generalization hierarchy and containing a series of expected properties as attributes. These attributes may be that a lake is composed of a contiguous region of water covering the surface of the earth and that this region is within a certain range in size. If we consider an observational data set that indicates the presence of water on the surface of the earth, the attributes of the class may be used to interpret those observational data to extract contiguous regions of water, measure the size of those regions, and then recognize those regions as individual lake entities if they meet the size criteria of the LAKE class attribute.

This 'top-down' approach to knowledge discovery, in which a class is defined a priori to the exploration of the observational data, may be complemented by a 'bottom-up' approach. The 'bottom-up' approach to knowledge discovery corresponds to 'bottom-up' learning as described in the conceptual framework of cognition. In the 'bottom-up' approach to knowledge discovery, patterns embedded in the observational data may be used to drive the recognition of individual objects, which may then be grouped into classes.

Note that Fig. 3 also provides for the separation of observational data and 'higher-level' information stored in objects and classes, just as cognition distinguishes between percepts and concepts. This approach follows the suggestions of Freksa and Barkowsky (1996), who maintain that knowledge discovery demands that the user be able to 'construct' different knowledge representations from the observational data. In other words, as (assumedly) indisputable facts the observational data are not subject to alteration, but the user may wish to develop different interpretations of what those observational data represent.

There are, of course, significant differences between the nature of geographic cognition and the computational analog shown in Fig. 3. For instance, consider the differences between a cognitive category and a class, given the digital nature of computational representation. While in a cognitive category these expected properties may take a number of different forms (e.g. imagery, prototypes, etc.), the nature of a class with attributes conjures a scenario where each expected property is equated with an attribute that lists the expected range of values acceptable for that property. For instance, one attribute of a class may list the maximum and minimum size acceptable for membership in that class, as in the lake example described above. In this sense, a class with attributes is much more akin to the idea of a rule-based approach to categorization and knowledge representation, and therefore relates more closely to the propositional theory of schema as described by Thorndyke (1984). Further, cognitive categories are not binary classifiers but may be graded in nature. With the exception of some of the more experimental approaches that integrate fuzzy set theory into GIS and image analysis (c.f. Burrough, 1996) database classification is inherently set based.

That said, given recent advances in object-oriented and multi-media databases, there is no computational reason that a variety of different types of representations for the expected properties of a class need be stored as set-based rules. For example, a graphic image may be stored as an attribute of a class, analogous to the use of mental imagery as cognitive schema. Prototypes may be stored as attributes and may take the form of a graphic image or a set of explicit rule-like properties, as suggested by Jackendoff (1992). The computational implication of having a variety of different forms for digitally encoding the expected properties of a category is that there must be different algorithms for using those expected properties to extract individual instances of those categories from the observational data. For example, if the expected properties are stored as a set of rules (e.g. size, shape, etc.), then those rules may be encoded in a set of queries on the observational data to extract instances of that category. On the other hand, if the expected properties are stored as a graphic image that stands for a prototype, then an algorithm that measures similarity to the graphic image would have to be applied to the observational data.

While a variety of approaches to representing the expected properties of a category are indeed possible, I focus on the rule-based approach in this research. My primary reason for this is that there is a clear similarity between a category with rule-like expected properties and the established database representation structures of class and attributes. In addition, the computational analog to the rule-base form of expected properties may be found in the well-established frame structure (Minsky, 1975). Finally, the algorithms for using a rule-based approach to categorization may be developed using standard query languages while other approaches, such as those based on similarity to a graphic image, are much more computationally complex. For these reasons, as well as in the interest of developing a working prototype implementation within the scope of this research, the rule-based approach to representing the expected properties of a category is used.

While I acknowledge that the use of a rule-based approach to representing the expected properties of a category does not necessarily 'mimic' cognitive categorization in all its subtleties, my aim here is not necessarily to implement the most sophisticated aspects of cognition. Rather, my aim is to develop a prototype implementation that draws from the overall process of cognition to improve GIS database representation and thus better support advanced tasks such as knowledge discovery. Note that the overall structure of the semantic data model is intended to support the representation of more sophisticated notions of categorization through, say, the use of fuzzy set theory to represent graded categories. For the purpose of developing this prototype implementation, however, the rule-based approach was used. In the concluding discussion I discuss how the semantic data model may incorporate other, non-rule-based, approaches to implementing the principles of cognitive categorization.

3.4. The semantic data model

One of the primary themes of the conceptual framework of cognition (Fig. 2) specified at the conceptual level is the distinction between percept and concept. In the context of database representation, percept was equated with observational data and concept with the categories and entities that may be extracted from those data. Based on this distinction, the semantic data model is composed of two distinct, yet interrelated, components: the Data Component and the Knowledge Component (Fig. 4). The Data Component represents 'un-interpreted' observational data that are spatially and temporally referenced. One can consider the Data Component a multi-dimensional data space with three spatial dimensions, one time dimension, and any number of thematic dimensions that record an observed value at a given coordinate time and location. While I acknowledge that people do not cognitively store a 'multi-dimensional data space' as a means to encode perceptions, the purpose of the semantic data model is to develop a relatively formal structure that can ultimately be used to derive a working implementation in a computational environment. Because computers are digital machines, the representation of observational data must conform to a discrete storage structure. The 'multi-dimensional data

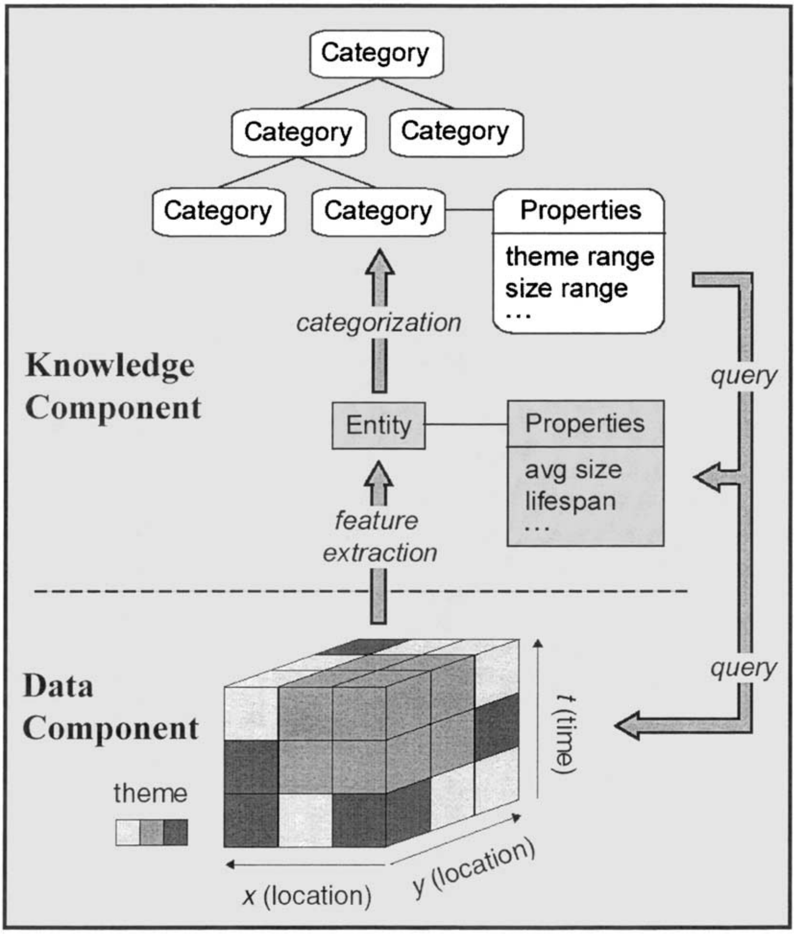

Fig. 4. Diagram of the semantic data model.

space' provides such a structure while still maintaining the ability to support knowledge representation within the Knowledge Component.

The Knowledge Component represents knowledge in the form of categories and the individual entities that may be observed to exist within the observational data. The Knowledge Component also incorporates the taxonomic relationships associated with the structure of categories. Both categories and entities have a set of properties. In the case of categories, these properties are the rules that form the categorical criteria for membership within the category. For entities, the properties are values that are observed or measured for a given individual entity.

The semantic data model supports the extraction of entities from the observational data by generating complex queries on the observational data from the expected properties associated with a category. Note also that the process whereby categorical knowledge is used to extract entities from the data space is intended to be an iterative and ongoing process, not a ‘one-time’ operation. Thus the semantic data model supports computational knowledge discovery in a manner analogous to how humans interpret sensory input using stored cognitive categories to identify individual entities in the environment. Observational data about the environment are stored in the Data Component, then interpreted and categorized to derive meaningful information that is stored in the Knowledge Component. This knowledge may then be used to further interpret the observational data.

3.5. The extraction of entities from the data space

The semantic data model supports the extraction of entities from the observational data in a manner similar to the cycle of entity extraction presented in Fig. 3. The expected properties of a category are used to interpret the observational data and extract instances of that category. This interpretation takes place by generating a complex query on the observational data using the expected properties associated with a category. This complex query first extracts entities that meet the thematic property of a category to extract a set of ‘candidate’ entities. Once the properties of these ‘candidate’ entities are calculated, they may be compared to the other properties of the category to see whether an entity meets the categorical criteria.

For example, consider again the category ‘lake’ that is being used to identify individual lake entities from a data space containing observations of the presence of surface water. Consider that the two expected properties associated with the category ‘lake’ are that it is composed of a spatially contiguous region of surface water and it is within a certain range in size. I consider these two properties as different types: *first-order* properties and *second-order* properties. First-order properties describe ‘composition’—what an entity of this type is ‘made of’. For instance, in this example we define a lake as being ‘made of’ a contiguous body of water on the surface of the earth. Second-order properties are those that describe an entity of this type as a whole, for instance its size.

The reason for making this distinction between types of properties is that the complex query that performs the feature extraction process must query first on the first-order properties before querying on the second-order properties. For instance, in the lake example, a contiguous region of water must be identified prior to its size being measured. Therefore, the initial part of the complex query to extract lake entities would extract spatially contiguous regions of surface water from the data space. These spatially contiguous regions would be recognized as ‘candidate’ entities for membership within the category lake. The size of each candidate entity would then be measured and, if it met the size criteria listed in the expected properties for the category ‘lake’, it would be recognized as a lake entity.

4. Implementation of the semantic data model

The implementation level is concerned with implementing the semantic data model described at the operational level within a database programming environment.

A number of academic experiments in implementing non-conventional GIS data models have chosen to develop using an object-oriented database package as a development platform (e.g. Milne et al., 1993; Scholl and Voisard, 1992). Note that the object-oriented modeling approach is intended to capture the semantics of an application domain. Object-oriented databases therefore provide a development environment with a significantly lesser degree of representational constraint than that offered by relational (and object-relational) database packages, currently the commercially dominant data model for spatial and non-spatial database applications.

Because one of the primary motivations for this research is to mitigate the representational limitations associated with conventional GIS and relational database models, I chose to implement the semantic data model using an object-oriented database platform. I compared the features of a number of different object-oriented databases, considering issues such as scalability, indexing schemes, support for inheritance and other object-oriented modeling features, and cost. Ultimately, the object-oriented DBMS *Poet* (Poet Software Corporation, San Mateo, California) was chosen for implementation because it supports the relationship structures and query capabilities that the implementation of the semantic data model demands, has an intuitive graphic user interface, and supports customization through a variety of programming languages. Java served as the development programming language for customizing the database representation within *Poet*.

4.1. Implementation of the data component

Fig. 5 provides an overview of the implementation of the Data Component and Knowledge Component in the Unified Modeling Language (UML), an object-oriented visual modeling language (Booch, Rumbaugh, & Jacobson, 1999). Each rectangle in the diagram represents a Java class and each rectangle is divided into two smaller rectangles. The top rectangle contains the name of the class and the bottom rectangle contains the attributes (properties) of the class. Methods associated with the classes are not shown in the diagram for simplicity; however, certain methods that are deemed important to the discussion are mentioned in the text. Each line between the rectangles (with a diamond) represents an aggregation relationship between the classes. The numbers associated with the lines (e.g. 1..*) indicate the cardinality of the relationship. The multi-dimensional data space of the Data Component is represented by a set of ATTVALUE objects. The ATTVALUE (i.e. attribute value) class represents a particular observation or measurement of a variable, i.e. a temperature value of 55° F. This class has the attribute *name*, the name of the variable (e.g. temperature), and the attribute *value*, the observed value (e.g. 55° F). Each ATTVALUE object is also composed of one LOCATION object and one TIME object. The LOCATION class identifies a location in three-dimensional space, having attributes x, y, and z. The attributes x and y describe a planimetric position on the earth's surface, and the attribute z gives the altitude. The TIME class contains the attributes *year*, *month*, and *day*.

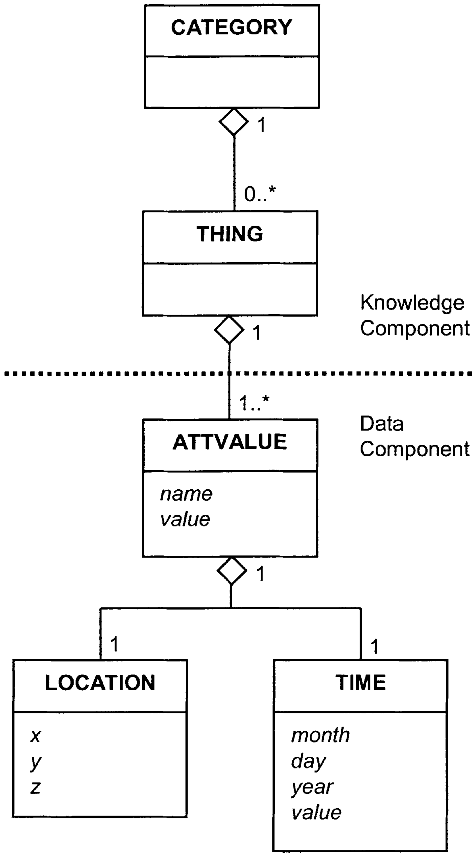

Fig. 5. Unified modeling language (UML) diagram of the implementation of the semantic data model in Java.

4.2. Implementation of the knowledge component

The implementation of the Knowledge Component is also presented in Fig. 5. The THING class is intended to represent one individual, semantic entity. The THING class is therefore analogous to the 'Entity' symbol shown in Fig. 4 which describes the Knowledge Component of the semantic data model. A THING object is composed of a collection of one or more ATTVALUE objects. Fig. 4 shows the analogous relationship in the semantic data model whereby an entity is composed of a set of elements in the multi-dimensional data space. A THING object can therefore be understood as a 'portion' of the data space—a collection of spatio-temporally referenced observations—with explicit storage of location, time, and theme value attributes.

The CATEGORY class is intended to represent a category, or type of entity, just as it is in the semantic data model (Fig. 4). A CATEGORY object is composed of a collection of one or more THING objects. A geographic domain is represented by extending these classes to represent actual entities and entity types (categories) as conceptualized by the user. In these extended classes, attributes may be added that describe the properties of that particular entity or category, just as entities and categories have expected properties in the semantic data model shown in Fig. 4. I have developed two libraries of properties, one for properties of THING objects (the TProp package) and one for properties of CATEGORY objects (the CProp package).

The motivation for developing these two libraries of properties stems from the fact that there are very few, if any, 'generic' properties that apply to all geographic entities. While there are certainly properties that are commonly used in geographic categorization, such as size, location, and part-whole relationships, categorical identity is not homogeneously determined in an absolutely logical and predictable fashion. Rather, categorization varies—some categories are defined by function (e.g. navigable versus non-navigable waterways), some by size and shape (e.g. a pond versus a lake), and some via a more rigid scientific classification (e.g. rivers may be classified by order). The approach of having libraries of properties that may be added to the CATEGORY and THING classes allows the user to specify what properties are particularly important for a given category or entity within the context of their conceptual model of the application domain. Note that these properties are not intended to be exhaustive toward listing all possible properties of geographic entities and categories. Rather, they represent the beginnings of a larger library that is intended to be added to as more specific modeling situations arise. These properties may then be reused over time to describe different semantic entities in various application domains.

Each property in each of the libraries (the TProp and CProp packages) is defined as a separate class. Each class in the Tprop package has an attribute *value*, that describes the value of that property. Initial classes implemented for the TProp package include the following:

BIRTH:	when a THING object began its existence
DEATH:	when a THING object ends its existence
LIFESPAN:	the duration of a THING object's existence
SIZEAVG:	the average size of a THING object over its lifespan
SIZEMIN:	the minimum size of THING object over its lifespan
SIZEMAX:	the maximum size of THING object over its lifespan
EVOLVESINTO:	indicates a different THING object that the present THING object becomes later in its existence, if the properties of this entity change over time and thus change its categorization
EVOLVESFROM:	indicates a different THING object that the present THING object was previously in its existence, if the properties of this entity change over time and thus change its categorization

There are also a number of classes contained in the TProp package that may be extended to represent properties that concern the thematic attributes of a THING object. The following property classes meet this purpose:

ATTVALAVG:	the average value of a theme over the lifespan of a THING object
ATTVALMIN:	the minimum value of a theme over the lifespan of a THING object
ATTVALMAX:	the maximum value of a theme over the lifespan of a THING object

The THING class also contains methods for calculating the values for the properties found in the TProp library, such as the method *calcBirth( )*, which when called calculates the value for the BIRTH property for itself. Other methods in the THING class include *calcDeath( )*, *calcLifespan( )*, *calcSizeMin( )*, *calcSizeMax( )*, and others. These methods allow for the properties of a THING object to be compared to the properties of a CATEGORY object so that it may be categorized correctly, as is explained below.

The CProp package contains property classes that describe properties similar to the TProp package. However, while the TProp classes are intended to describe the properties of an actual entity (e.g. the actual lifespan of a THING object), the CProp classes are intended to describe the range of acceptable values of a particular property that a THING object must have to be a 'member' of a particular CATEGORY. For instance, while a THING object may have the property LIFESPAN, a CATEGORY object may have the property LIFESPANRANGE that describes a range in lifespan. The LIFESPANRANGE property specifies the maximum and minimum lifespan that a THING object can have to be considered a member of a particular CATEGORY. Each class in the CProp package has attributes of *high* and *low* to specify the maximum and minimum value of the range. By adding properties to a CATEGORY object for domain-specific representation, the CATEGORY class serves a role similar to that of propositional cognitive schemata (as described by Thorndyke, 1984). The CProp package contains the following property classes:

LIFESPANRANGE:	minimum and maximum lifespan of a THING object
SIZERANGE:	minimum and maximum size of a THING object
ATTVALRANGE:	minimum and maximum theme value (of an ATTVALUE object) contained in a THING object
ATTVALAVGRANGE:	minimum and maximum average theme value over the course of a THING object's lifespan

4.3. A rule-based approach to entity extraction

THING objects are extracted from the data space, and assigned a 'membership' to an appropriate CATEGORY object, through a rule-based approach to categorization. A CATEGORY object is initially created with a set of CProp classes to define the categorical criteria, but with an 'empty' collection of THING objects, i.e. it has no 'observed' entities as 'members'. The CATEGORY object's properties are then used to generate and execute a complex query on the data space using *Poet*'s implementation of Object Query Language (OQL). Note that this is analogous to the complex query process illustrated in Fig. 4, whereby the expected properties of a category are used to recognize instances of that category in the data space. The result of this complex query is a set of entities that may be considered members of a particular category.

The basic process by which THING objects are identified and assigned to a particular CATEGORY object follows three steps:

- *Step 1 queries on the first-order properties of a CATEGORY object*. A CATEGORY object's ATTVALRANGE object is used to execute a query on the data space that returns a collection of those ATTVALUE objects that have a *value* within the range specified in the CATEGORY object's ATTVALRANGE object.
- *Step 2 identifies candidate THING objects for membership within a CATEGORY object*. The collection of ATTVALUE objects generated by step one are organized into temporary, candidate THING objects by grouping spatially and temporally contiguous ATTVALUE objects.
- *Step 3 queries on the second-order properties of a CATEGORY object*. The properties of those candidate THING objects, such as SIZE and LIFESPAN, are measured and compared to the appropriate properties in the CATEGORY object, such as SIZERANGE and LIFESPANRANGE. Those candidate THING objects that meet the categorical criteria of the CATEGORY object are added to the CATEGORY object's collection of THING objects and stored persistently in the database. The THING objects that do not meet the categorical criteria are deleted.

4.4. An example using the category 'lake'

As a description of how the semantic data model and rule-based approach to entity extraction works, consider again the example of representing lakes using a data set that describes the variation of the presence of water on the surface of the earth over time. Here, we will define a lake to be a body of water that maintains a particular range in size over its lifetime. In this case, the multi-dimensional data space would be composed of a set of ATTVALUE objects that indicate the presence or absence of water on the surface of the earth at a set of locations and times (Fig. 6). For instance, the attribute *value* in the ATTVALUE class can be assigned a value of one to indicate the presence of water and zero to indicate the absence. We may then define a type of water body, a lake, by extending the CATEGORY class to create the new class LAKECAT. Properties contained in the CProp library of category properties may be added to the category LAKECAT, such as the properties ATTVALRANGE and SIZERANGE. The ATTVALRANGE property may be made to indicate that the presence of water is required for membership in the category LAKECAT while the SIZERANGE property indicates that a water body must be within a range of sizes to be considered a member of the category LAKECAT. We may then instantiate the class LAKECAT to assign values to these properties.

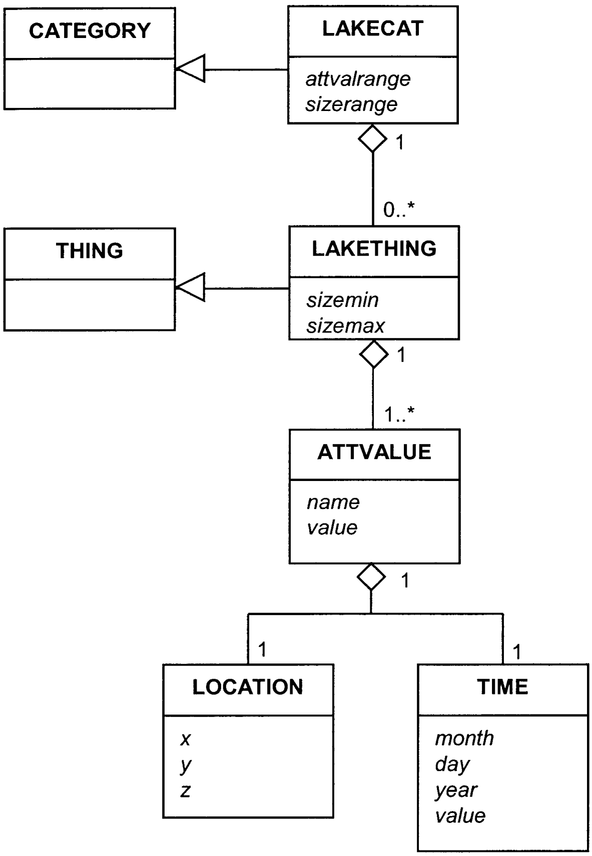

Fig. 6. UML diagram of the implementation of the semantic data model, applied to the representation of lakes using a data set that describes the spatial and temporal variation of the presence of water on the surface of the earth.

We may also extend the THING class to create the new class LAKETHING in order to represent an individual lake entity that may be extracted from the data space (Fig. 6). Because we have noted that the size of a lake is an important property that informs whether a particular water body is indeed recognized as a lake, we may add the properties SIZEMIN and SIZEMAX from the TProp library to the LAKETHING class. Note that at this point (prior to the extraction of lake entities from the data space), there are no instantiations of the LAKETHING class—no individual lake entities have as yet been extracted from the data space and added to the instantiated LAKECAT object's collection of THING objects.

The extraction of lake entities from the data space proceeds according to the steps described above. In step 1, the ATTVALRANGE property of the category LAKE-CAT is used to compose a query in OQL that selects only those ATTVALUE objects that indicate the presence of water on the surface of the earth—those with a *value* = 1. In step 2, those selected ATTVALUE objects that are contiguous in space and time are grouped into individual 'candidate' LAKETHING objects. In step 3, each candidate LAKETHING object calls the *calcSizeMin()* and *calcSizeMin()* methods and calculates its own SIZEMIN and SIZEMAX property values. Subsequently, the SIZERANGE property of the category LAKECAT is used to compose another query that compares the SIZEMIN and SIZEMAX values of each candidate LAKETHING object with the SIZERANGE of the LAKECAT object and selects only those candidate LAKETHING objects that have minimum and maximum sizes that are within the size limits specified in the SIZERANGE property.

Using these same methods, much more complex hierarchies may be created for the purpose of representing the conceptual entities that populate a geographic domain and for using this representation to extract individual entities from the data space. For instance, one could construct a much richer hierarchy of types of water bodies, such as small, medium, and large lakes, as well as make distinctions between salt-water versus fresh-water lakes and shallow-water versus deep-water lakes, if the observational data were available. In addition, if other properties such as shape were incorporated into the TProp and CProp libraries of properties, other water body entities such as estuaries and lagoons could also be identified.

In addition, one could capture the process by which water bodies change over time. For instance, by calculating the size of each water body at each time interval of the observational data collection, a single body of water could be categorized as a small lake at one set of times and then become a medium size lake when its size grew above a certain threshold. The TProp properties EVOLVESFROM and EVOLVESINTO could be added to the LAKETHING class in order to explicitly store which lakes have changed from one type to another over time. A researcher may then explore patterns of when and where certain lakes have changed in size over time.

Also, recall that the definition of the expected properties of a category are intended to be flexible and subject to revision, and that the process of category definition and entity extraction is intended to be iterative in nature. Thus, a researcher interested in understanding the spatial and temporal distribution and relationships among various water bodies may wish to initially define a hierarchy of water body types using a priori expert knowledge to assign values to the expected properties of each category. However, subsequent to the first entity extraction operation, the researcher may review the results (e.g. by generating summary statistics for each of the water body categories), decide to revise the expected properties for certain water body categories, and re-run the entity extraction operation. The researcher may then compare the implications of using different categorical definitions on the extraction of entities from the observed data.

5. Concluding discussion

This research has demonstrated that principles of cognition can inform the development and implementation of a semantic GIS data model. Such a model is able to represent a geographic domain as it is conceptualized by the user in a more robust manner than the conventional vector and raster GIS data models. A semantic data model informed by principles of cognition therefore supports complex modeling and data exploration tasks in GIS by allowing the database user to interact with the conceptual entities and relationships of the application domain explicitly within the database.

The semantic data model described here differs from previous research in object-oriented semantic GIS database modeling research in two significant ways. First, I have separated the computational representation of a type of entity (i.e. a category) from that of the entity itself. In previous research, a category was represented by a class and an entity by an instantiation of that class (e.g. Tang et al., 1996; Worboys et al., 1990). However, this does not account for the fact that the properties of a category are different from the properties of an entity, and that the properties of a category are, in fact, the criteria for categorical membership. By representing categories and entities as different classes, each may have their own types of properties (and methods). Second, previous efforts in object-oriented GIS have focused on structural representation, a static representation of an application domain that is determined when the database is built (e.g. Tang et al., 1996; Worboys et al., 1990). The purpose of the semantic data model presented here is not simply to represent a given domain, but to facilitate the analysis of that domain by allowing a researcher to investigate and interpret the observational data in a variety of different ways. This investigation is supported by the establishment of a library of properties for both categories and entities, the interactivity by which a researcher may experiment with different settings for those properties, and the automatic feature recognition though which entities are extracted from the observational data based on user-derived expert knowledge. The semantic data model can therefore be considered a process of learning about an application domain as much as a structural representation of that domain.

One of the key advantages of the semantic data model is its ability to support multiple representations of a single geographic domain. By distinguishing between the observed data, which is taken to be objective, and the knowledge that may derived from those data, which is taken to be subjective, multiple interpretations of the observational data may be developed. For instance, in the lake example given above, two people may disagree on precisely what is a 'small' versus 'medium' size lake. The semantic data model may be used to develop two different hierarchies of types of lakes, one with the size criteria for each type of lake set according to the beliefs of one person, the other set according to the beliefs of the other person. Different types of lakes could then be extracted from the observational data according to the two different categorical hierarchies and the differences between the two hierarchies compared. Note that although the observational data never changes, the semantic data model supports different semantic interpretations of those data based on the subjective knowledge provided by the users. The users have the option of deciding which or both of these interpretations should be made persistent in the database; or perhaps the users may revise their categorical hierarchy and develop yet another semantic interpretation of the observational data.

While demonstrating the overall principles of the semantic data model, the prototype implementation described here incorporates only certain principles of cognition that I thought were most relevant to GIS database representation. However, there are many more sophisticated aspects of cognition that would provide a much richer environment for computational geographic representation than that currently implemented. In terms of categorization, research may focus on the integration of a richer set of category and entity properties, such as those associated with the function of a geographic entity or its spatio-temporal relationships with other categories and entities. The implementation thus far only supports one category property that concerns inter-relationships among entities: the EVOLVESINTO and EVOLVESFROM properties that indicate the mutation of one entity into another. However, it is easy to imagine that other more sophisticated category properties may be developed and added to the TProp and CProp libraries, such as those that indicate spatial and temporal topologic relationships among entities.

In addition, research should focus on developing more sophisticated representations of categories that incorporate principles such as prototype categorization and the use of exemplars, imagery, and other approaches to categorization that do not treat categories as mathematical sets. Certainly, fuzzy set theory could be incorporated into category representations to represent the graded nature of many categorical properties, as has been demonstrated by a number of authors (e.g. Papadias et al., 1999; Wang & Hall, 1996). Future research may also address related issues in object recognition—how the cognitive processes through which instances of categories are identified in the environment can be transformed into a computational context and implemented algorithmically.

Incorporating these more sophisticated aspects of cognition into the implementation would provide alternatives to the rule-based approach to representing categorical knowledge and the 'top-down' approach to knowledge discovery (i.e. extracting entities from the data space using a priori categorical knowledge) presented here. While alternative approaches would certainly demand new, and potentially complex, algorithms for entity extraction, the structure of the implementation of the semantic data model is organized to incorporate these other approaches to knowledge representation and discovery. For example, alternative representations of the expected properties of a category, such as a graphic image, could simply be added to the CProp library of classes. Inductive 'bottom-up' approaches to knowledge discovery, such as categorizing entities based on similarities in observed properties, could also be implemented. For example, a set of 'un-categorized' THING objects could be extracted from the data space using criteria that addresses the 'first-order' properties of a CATEGORY object, the properties of those individual THING objects could then be calculated, and cluster analysis could be used to assign those THING objects to a CATEGORY object based on those properties. The expected properties for the newly formed CATEGORY objects may then be derived from the distribution of the properties of the collection of those THING objects assigned to the CATEGORY. For instance, in the water body example given above, a researcher could extract spatio-temporally contiguous regions of the presence of water on the surface of the earth (a first-order property), identify spatial and temporal properties of those regions (e.g. size, duration, and nearness to other water bodies), and inductively categorize those regions into different types of water bodies according to the multi-dimensional clustering of their properties.

Other extensions to the semantic data model may be developed for the representation of dynamic processes. A process may be represented by a set of individual 'states' that are captured over time. In the semantic data model, a process 'category' and an individual process 'entity' may thus be represented by extending the CATEGORY and THING classes, respectively. However, in order to represent the properties of a dynamic process, the static properties implemented thus far, such as the size of an entity, must be extended. For instance, an important property for representing a particular process may be its rate-of-change of some measured attribute over time. The representation of dynamic processes in the semantic data model therefore demands the implementation of algorithms that calculate spatial, temporal, and attribute change since it is these dynamic properties that play a key role in differentiating one type of process from another.

Acknowledgements

The author would like to thank Donna Peuquet for her guidance in support of this research, Diansheng Guo for assistance with the implementation, and the anonymous reviewers for their helpful comments. This research was supported in part by a NASA Graduate Fellowship in Earth System Science and by US EPA grant R825195–01–0.

References

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language user guide*. Massachusetts: Addison Wesley Longman.

Burrough, P. A. (1996). Natural objects with indeterminate boundaries. In P. A. Burrough, & A. U. Frank (Eds.), *Geographic objects with indeterminate boundaries* (pp. 3–28). London: Taylor & Francis.

Burrough, P. A., & Frank, A. U. (1995). Concepts & paradigms in spatial information: are current geographical information systems truly generic? *International Journal of Geographical Information Systems*, *9*(2), 101–116.

Egenhofer, M. J., Glasgow, J., Gunther, O., Herring, J. R., & Peuquet, D. J. (1999). Progress in computational methods for representing geographical concepts. *International Journal of Geographical Information Science*, *13*(8), 775–796.

Fonseca, F. T., Egenhofer, M. J., Davis, C. A. Jr., & Borges, K. A. V. (2000). Ontologies and knowledge sharing in urban GIS. *Computers, Environment, and Urban Systems*, *24*(3), 232–251.

Freksa, C., & Barkowsky, T. (1996). On the relations between spatial concepts and geographic objects. In P. A. Burrough, & A. U. Frank (Eds.), *Geographic objects with indeterminate boundaries* (pp. 109–121). London: Taylor & Francis.

Gahegan, M., & Flack, J. (1999). The integration of scene understanding within a geographic information system: a prototype approach for agricultural applications. *Transactions in GIS*, *3*(1), 31–49.

Guarino, N. (1995). Formal ontology, conceptual analysis, and knowledge representation. *International Journal of Human–Computer Studies*, *43*, 625–640.

Howard, D., & MacEachren, A. M. (1997). Interface design for geographic visualization: tools for representing reliability. *Cartography and Geographic Information Systems*, *23*, 59–77.

Jackendoff, R. (1992). *Languages of the mind: essays on mental representation*. Cambridge, MA: The MIT Press.

Kosslyn, S. M. (1994). *Image and brain: the resolution of the imagery debate*. Cambridge, MA: The MIT Press.

Mark, D.M., Egenhofer, M., Hirtle, S., & Smith, B. (2000). *UCGIS emerging research theme: ontological foundations for geographic information science*. White Paper on Emerging Research Themes, University Consortium for Geographic Information Science. Available at: http://www.ucgis.org.

Mark, D. M., & Frank, A. U. (1996). Experiential and formal models of geographic space. *Environment & Planning B: Planning & Design*, *23*, 3–24.

Marr, D. (1982). *Vision*. W.H. New York: W.H. Freeman & Company.

McKeown, D. M. (1987). The role of artificial intelligence in the integration of remotely sensed data with geographic information systems. *IEEE Transactions on Geoscience and Remote Sensing*, *GE-25*(3), 330–348.

Mennis, J. L., Peuquet, D. J., & Qian, L. (2000). A conceptual framework for incorporating cognitive principles into geographical database representation. *International Journal of Geographical Information Science*, *14*(6), 501–520.

Milne, P., Milton, S., & Smith, J. L. (1993). Geographical object-oriented databases—a case study. *International Journal of Geographical Information Systems*, *7*(1), 39–55.

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision* (pp. 211–277). New York: McGraw-Hill.

Neisser, U. (1976). *Cognition and Reality*. New York: W.H. Freeman & Company.

Neisser, U. (1987). From direct perception to conceptual structure. In U. Neisser (Ed.), *Concepts and conceptual development: ecological and intellectual factors in categorization* (pp. 11–24). Cambridge: Cambridge University Press.

Nyerges, T. L. (1991). Geographic information abstractions: conceptual clarity for geographic modelling. *Environment and Planning A*, *23*, 1483–1499.

Nyerges, T. L., Mark, D., Laurini, R., Egenhofer, M. (Eds.). (1995). *Cognitive aspects of human–computer interaction for geographic information systems*. Dordrecht: Kluwer.

Papadias, D., Karacapilidis, N., & Arkoumanis, D. (1999). Processing fuzzy spatial queries: a configuration similarity approach. *International Journal of Geographical Information Science*, *13*(2), 93–118.

Peuquet, D. J. (1994). It's about time: a conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers*, *84*(3), 441–461.

Pinker, S. (1984). Visual cognition: an introduction. *Cognition*, *18*, 1–63.

Rosch, E. (1978). Principles of categorization. In E. Rosch, & B. Lloyd (Eds.), *Cognition and categorization* (pp. 27–77). Hillsdale, N.J: Lawrence Erlbaum Associates.

Scholl, M., & Voisard, A. (1992). Object-oriented database systems for geographic applications: an experiment with O2. In G. Gambosi, M. Scholl, & H.-W. Six (Eds.), *Geographic database management systems* (pp. 103–137). Berlin: Springer-Verlag.

Smith, B. & Mark, D.M. (1998). Ontology and geographic kinds. *Proceedings of the Seventh Annual Conference on Spatial Data Handling*, Vancouver, BC, Canada.

Tang, A. Y., Adams, T. M., & Usery, E. L. (1996). A spatial data model design for feature-based geographical information systems. *International Journal of Geographical Information Systems*, *10*(5), 643–659.

Thorndyke, P. (1984). Applications of schema theory in cognitive research. In J. Anderson, & S. Kosslyn (Eds.), *Tutorials in learning and memory: essays in honor of Gordon Bower* (pp. 167–192). San Francisco: W.H. Freeman & Company.

Tversky, B. (1993). Cognitive maps, cognitive collages, and spatial mental models. In A. Frank, & I. Campari (Eds.), *Spatial information theory: a theoretical basis for GIS* (pp. 14–24). New York: Springer-Verlag.

Tversky, B., & Hemenway, K. (1983). Categories of environmental scenes. *Cognitive Psychology*, *15*, 121–149.

Usery, L. (1993). Category theory and the structure of features in geographic information systems. *Cartography and Geographic Information Systems*, *20*(1), 5–12.

Wang, F., & Hall, G. B. (1996). Fuzzy representation of geographical boundaries in GIS. *International Journal of Geographical Information Systems*, *10*(5), 573–590.

Worboys, M. F., Hearnshaw, H. M., & Maguire, D. J. (1990). Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, *4*(4), 369–383.