Modeling Behavior of Geographic Objects:
An Experience with the Object Modeling Technique

Nectaria Tryfona
National Center for Geographic Information and Analysis
Department of Spatial Information Science and Engineering
Boardman Hall, University of Maine, Orono, ME 04469-5711, U.S.A.
nectaria@spatial.maine.edu

Dieter Pfoser
Department of Spatial Information Science and Engineering
Boardman Hall, University of Maine, Orono, ME 04469-5711, U.S.A.
dieter@spatial.maine.edu

Thanasis Hadzilacos
Computer Technology Institute, University of Patras
Box 1122, GR-26110, Patras, Greece
thh@cti.gr

Abstract

Behavior of geographic objects holds a critical role in spatial databases. This, along with objects' *position* and *space-varying attributes*, form a minimal set of concepts sufficient to capture spatial peculiarities in terms of the object-oriented rational. We present the semantics and the graphical notation of a prototypical object-oriented model for the conceptual design of spatial databases: by extending the Object Model of the Object Modeling Technique to the Geographic Object Model, we show how the above three concepts fit naturally into any object-oriented tool. We augment this model with the constructs of *spatial aggregation* and *spatial grouping* to express the critical aspects of space-varying attributes, object boundary fuzziness and uncertainty, spatial relationships, and attribute generalization. Our proposal integrates the field- and object-based geographic views in one model. The principal idea behind this effort is the incorporation of a set of concepts into any semantic or object-oriented model, to make them communicate at the conceptual level (semantic interoperability).

1. Introduction

Geographic Information Systems (GIS) is but one of a large number of data-intensive application areas -often referred to as "non-standard"- including mechanical, architectural and VLSI design, robotics, image and voice processing, artificial intelligence and knowledge-based systems. They are characterized by a wide variety of new requirements on design environments, transaction mechanisms and data types.

Spatial[1] databases is an indispensable part of GIS. In order to enjoy the benefits of *maintenance*, *portability*, *reusability* and sometimes even *correctness*, their design must follow the cycle of a structured application development [Brodie, 1982], including the phases of conceptual, logical and physical representation. But, this is not really the case for GIS: usually, they are built without any consideration to any particular methodology. This situation seems to be due to the *lack of* suitable concepts, models and tools that would provide proper extensions for spatial applications, while conforming to some standard systems.

Our position is that on one hand new mechanisms and constructs have to be incorporated into well-known models and tools to capture the particularities of spatial information, while on the other, the maintenance of the philosophy of each of these models is a central and crucial issue. Particularities of GIS steam from the main difference between geographic and "classical" objects: their "position" in space. Therefore, the ability to capture objects' "position" as well as the semantics surrounding it seems to be a critical matter . The latter can be achieved -mainly- by modeling operations on geographic objects, which is translated into understanding objects' *behavior*.

In this paper, we apply an object-oriented structured methodology to the design of geographic applications. Our scope is to model the behavior of geographic objects at the conceptual level by using object-oriented mechanisms. Capturing behavior of objects means representing *static* and *dynamic* concepts of them. We propose to augment object-oriented models with the minimal set of concepts necessary for spatial information systems, while maintaining compatibility with the particular philosophy of each model. The main advantage of our approach is that since developers are not invited to switch away from their favorite models and systems, they are likely to adopt this proposal.

For the object-oriented modeling of geographic applications specifically, good but rather sparse work is available [Scholl and Voisard, 1991], [Tang, et. al., 1996]. Models for spatial data handling fall into two hitherto disjoint classes [Worboys, 1994]: (a) field-based ones (such as the grid model) suitable for representing phenomena, like "temperature" and "vegetation" (see [Scholl and Voisard, 1991] for the definition of the Map Model on the top of O_2), and (b) object-

---

[1] In this paper, terms "spatial" and "geographic" are used interchangeably.

based models which view spatial information in terms of identifiable entities [Pelagatti, et. al., 1991], [Milne, et. al., 1993].

Our approach builds on classic and recent research on object-oriented modeling [Kim, 1990], [Rumbaugh, 1991], [Bancilhon, et. al., 1992]. It combines objects and fields in *one* object-oriented model and draws from previous theoretical and applied research on the subject [Tsironis, 1992], [Worboys, 1994], [Tryfona, 1994]. We are not aware of any other in-depth research towards this direction. The main contributions of this work are:

(a) the integration of the *object* and *field* approaches in one object model, and

(b) showing how the basic characteristics of spatial information -namely objects' *position*, *space-varying attributes* and *operations on them*- fit naturally into standard object-oriented models. Augmented by two new constructs, *spatial aggregation* and *spatial grouping*, an object-oriented model can readily express geographic knowledge, such are spatial relationships, space-varying objects' properties, generalization, and others. This proposal comes to fill the gap between conceptual modeling of GIS (where only the static properties matter) and implementation (where operations (dynamic properties) of geographic data is the central spring).

The rest of the paper is organized as follows: in Section 2 we explain the reason we chose object-oriented modeling techniques to capture the spatial information and we give a short exposition of the proposed concepts needed for spatial data handling, namely the *position* of objects, *space-varying attributes* of objects (such as "vegetation") and *operations* on geographic objects and their properties (such as union, interpolation or spatial computation). Based on these needs, we present a geographic database model governed by the object-oriented rationale. In Section 3 we present the Geographic Object Model, a prototypical model for the object-oriented design of spatial databases, by using the Object Model of the Object Modeling Technique [Rumbaugh, 1991] as a basis to be extended and specialized. We discuss the way *position* and *space-varying attributes* provide an elegant way to handle important issues of spatial applications. The applicability of our approach is presented in Section 4 by an example based on a Network Utility Management Information System [UtilNets, 1994]. In Section 5, we conclude with the results and the future research plans.

2. Object-Oriented Design for Geographic Information Systems

What distinguishes geographic objects from all the others (i.e., the classical ones, such as a "car") is their position in space (actually the fact that their position in space *matters*). By adding this special characteristic to our database we are led to add also operations on it. Operations on objects' position play an important role and are a vital part of GIS (what's the difference between a GIS and a Computer Aided Design system?). From the perspective of conceptual modeling, object-oriented design of spatial applications allows the representation of dynamic properties independently of implementation issues. Dealing with classical conceptual models, like the Entity-Relationship [Chen, 1976] and its extensions, or IFO [Abiteboul and Hull, 1987], or the Semantic Data Model [Hull and King, 1987], we lack to capture the most important part of geographic applications: objects' *behavior* in space.

2.1 Spatial Needs at the Level of Conceptual Modeling

In this section we record spatial concepts of geographic applications that distinguish them from the classical ones in terms of semantics and modeling needs:

(a) In the real world, most objects have a **position** which is the object's link with space. In information systems we are only interested in position for *some* objects: those are the *geographic* objects of the application. The position of an object is a function on all and only on geographic objects and returns for each geographic object *a part of space*.

(b) Next, there is the need to model **space** in order to locate objects in it. Space is a set. The elements of space are called *points*. Any set will do for space. A very important intuition and interesting theories come up from non-standard spaces, even non-numeric ones; however for practical purposes of current spatial applications, space is modeled as a subset of R^3. All specific discussions and examples in this paper use R^2 as space.

(c) A fundamental peculiarity of spatial information systems is that some properties of interest do not properly belong to any particular object. For example, "soil_type" in a cadastral application. Although one application view may regard the "soil_type" of the land parcels as an attribute of the parcel, it is clear that: (i) it is defined whether or not a parcel exists at that position in space, and (ii) when a parcel is moved, it will not keep the value of "its" attribute; rather it inherits new value from the new position. These attributes are called **spatial** or **space-varying** attributes. Informally, space-varying attributes are properties of space which indirectly become properties of objects situated at some position in space. Overlapping objects share the same values for these attributes. *The value of a space-varying attribute depends on position only*, and not on the object itself. Formally, a *space-varying attribute* is a function whose domain is space and range is any set. Under this perspective, "soil_type" needs to be modeled as a function from space to the set {*sand*, *clay*, ...}.

(d) Additionally, geographic objects are related to each other in space. Relationships among geographic objects are actually conditions on object's position and are called **spatial relationships**. For example, the spatial relationship MEET between two land parcels shows that they share common borders.

2.2 Geographic Object Databases

At this point we describe a model to handle the above presented needs. The following model is based on object-oriented terminology [Kim, 1990], and uses concepts from [Hadzilacos and Tryfona, 1992], [Delis, et. al., 1994] [Tryfona, 1994].

Objects

A database is a *set of objects* which represent part of the real world; each object belongs to an *object class* (we ignore multiple inheritance [Nierstrasz, 1989] in order to make the representation more clear, but our model does not depend on this). An object class is characterized by a set of *static properties*, or *attributes* and a set of *dynamic properties* or *methods*; each attribute is associated with a *domain*, which is an unrestricted set of values. Methods are the only means to access the attributes. Static and dynamic properties (attributes and methods) construct the *behavior* of an object. So, each object in a database instance is represented by a set of values each belonging to the domain of the corresponding attribute of the object class.

Domains are implemented by data types and methods are implemented by procedures. In order to serve the needs of spatial information handling applications, new data types, called *geometric*, are added to the classical database data types (real, integer, string, date, etc.). There are several such data types, of which we will use POINT, LINE, and REGION. In geographic databases, objects have one more characteristic: their *dimension* in space. Its domain is $\{\text{point}, \text{line}, \text{region}, \text{none}\}$, or equivalently $\{0,1,2,\text{none}\}$. Objects with dimension 0,1,or 2 are called *spatial* or *geographic*. They have the special property *position*, whose domain is a finite subset of sets of geometric figures of dimension not exceeding the dimension of the object. For example, a spatial object with dimension 1 (i.e., linear) can have as position lines and points. Positions of geographic objects, or in other words subsets of space, are also called *geometric figures*.

Layers

Layers are used to represent spatial attributes (Section 2.2). Informally speaking, a *layer*, this widely used spatial data organizing concept, is “a logical separation of map information according to a theme” [Burrough, 1986]. Examples are a vegetation map, or a road map. Formally, a layer is a set of geometric figures with associated values, so it is defined as a function from geometric figures to attributes or as a relation with the geometric figure as the key attribute [Delis, et. al., 1994]. In manipulating layers it is sufficient to be able to change the geometric figures (i.e. the domain of the function) or the attributes (i.e. the range of the function) and to combine such changes through function composition.

Geographic methods

There are three types of methods on geographic databases:

(a) those which manipulate objects and act only on descriptive attributes (e.g., retrieve the name of a river).

(b) those which manipulate objects and act on objects' position. They are categorized according to:

- the type of the returned value (real vs. geographic). Examples are DISTANCE, AREA, UNION, NODES, etc. (e.g., find the distance between two building blocks)
- whether they are primary (i.e., defined along the domain of the attribute) or derived; examples are PERIMETER, LENGTH, INTERSECTION, etc. (e.g., find the perimeter of a building block), and

(c) those which manipulate layers; there are four types of this category [Delis, et. al., 1994]: ATTRIBUTE DERIVATION, SPATIAL COMPUTATION, OVERLAY, and RECLASSIFICATION, (e.g., overlay the building block and the road map).

Methods of types (b) and (c) are called *geometric* or *geographic methods* or *operations*.

Spatial Relationships

A relationship or association is a condition on a tuple of values of attributes, possibly from different objects. Relationships which include positions are called *spatial*. Spatial relationships are translated into spatial integrity constraints of the database. The definition of a "square", in a cadastral application, as a "land parcel which is not contained in any building block" is an example of using spatial integrity constraints. Conceptual modeling should lead to straightforward solutions for explicitly storing topology in the logical and physical levels -a common practice despite topology being derivable from objects' positions [Hadzilacos and Tryfona, 1992].

Based on the above retionale, object class "land_parcel" has a position in space which is a REGION and "land_parcel_id" and "land_parcel_owner" are its static properties, whereas AREA, and PERIMETER are its geographic methods. Two land parcels MEET each other. In order to connect two land parcels we apply the geometric method UNION .

3. The Formal Geographic Object Model

Next step is to show the way to capture the above described needs by using object-oriented techniques. We use the well-known Object Oriented Modeling Technique (OMT) [Rumbaugh, 1991]; however our approach is not depended on the particular model. We chose OMT as a pilot example for reasons of popularity and completeness in terms of methodologies. This technique consists of building a model of an application domain and then adding implementation details to it during the system design. We present a way the graphical notation capturing object-oriented concepts can be used to handle spatial needs. New constructs are added to encompass spatial particularities.

3.1 The Object Model

In this section we present the Geographic Object Model to capture spatial needs. It is based on extensions and specializations of the Object Model supported by the Object Modeling Technique (OMT) [Rumbaugh, et. al., 1991].

Object modeling has been widely discussed in the literature. Its main advantage is the high level of representation abstraction that it provides by dealing with concepts and not with implementation issues.

The OMT methodology [Rumbaugh, et. al, 1991] uses three kinds of models to describe a system: the *Object Model*, describing objects and relationships, the *Dynamic Model*, describing interactions among objects, and the *Functional Model*, describing the data transformations of the system. In this work we deal with the Object Model as it appears to be the most important part of the methodology.

The Object Model contains object diagrams. An *object diagram* is a graph whose nodes are object classes and whose arcs are relationships among classes. Its basic elements are the:

(a) *object classes*, which represent a set of autonomous ontologies (objects) and show their internal structure. An object that is created according to a certain object class is an *instance* of that class. The symbol representing a class has three areas: The upper area contains the class name, the middle area its attributes, and the lower area its operations (methods).

(b) *attributes* of object classes, which capture their properties; properties associate a value from a domain of values for that attribute with each object in an object class.

(c) *associations* among object classes, which are used to model relationships. Each association represents a set of similar relationships. It is often necessary to clarify associations according to how many instances from one object class can be associated with how many instances from another (*cardinality*). Associations may often have attributes (*link attributes*) as a result of the combined object classes through specific relationships.

(d) *generalization* hierarchy, which is a “is-a” or “is-a-kind-of” association. The general object class represents the *supertype* and the special classes the *subtypes*. Attributes and associations of the supertype are inherited by the subtypes.

(e) *aggregation*, which is a “part-of” association and is related to the construction of complex object classes.

Figure 3.1 represents the basic elements of the Object Model.

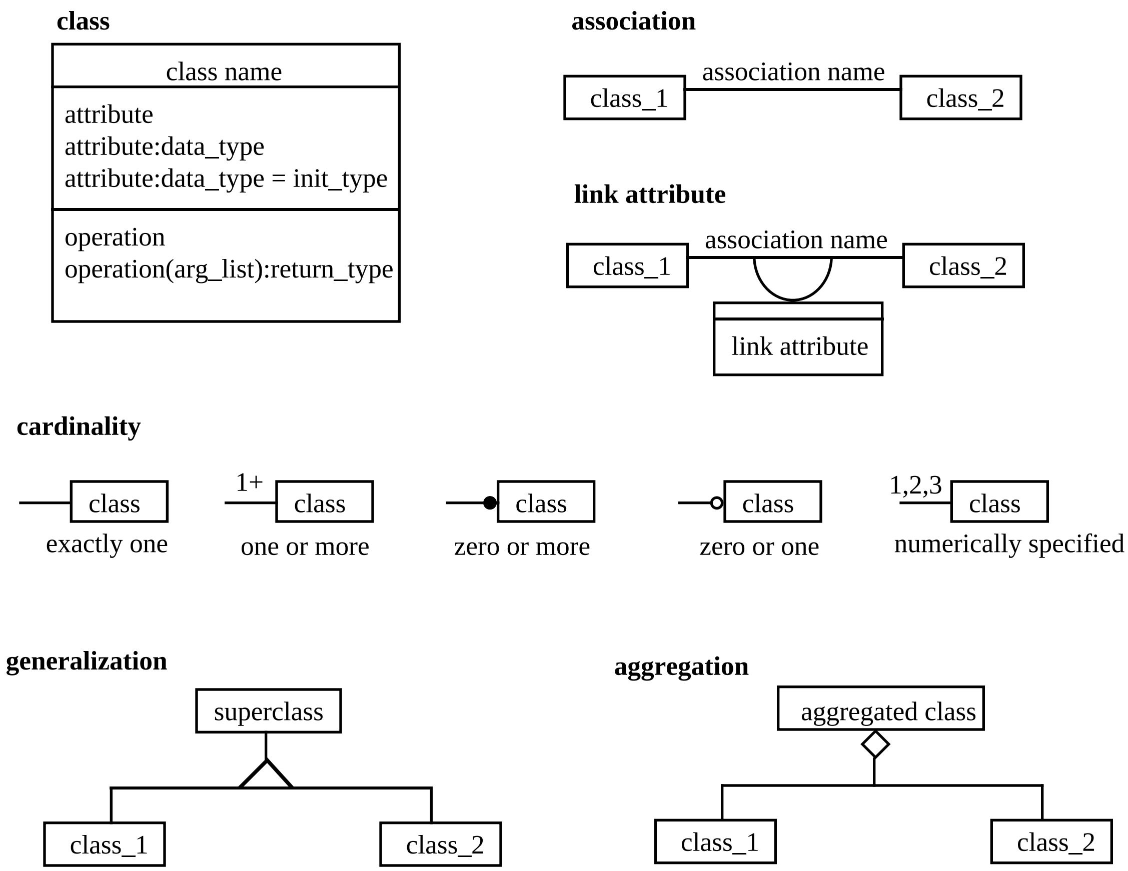


Figure 3.1: Basic elements of the Object Model [Rumbaugh, et. al., 1991].

3.2 The Geographic Object Model

Based on the study of spatial aspects that call for special modeling techniques and constructs at the conceptual design of geographic applications, we present the Geographic Object Model (Geo-OM) as a part of a methodology to build geographic databases. Geo-OM includes special object classes and associations to express semantics of space, geographic objects' position, objects' space-varying attributes, spatial relationships and operations on objects. Two new constructs are added to express the spatial dimension of complex geographic object classes: *spatial aggregation* and *spatial grouping.*

A. **Specializations of OMT**

I. A special object class is introduced: $SPACE$.

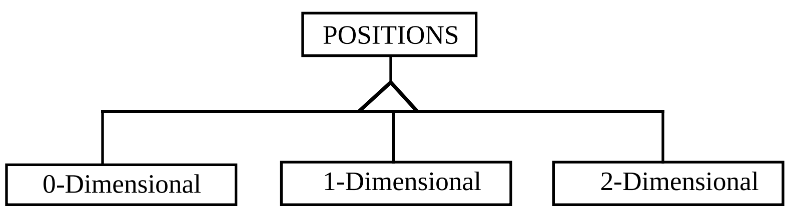
II. A second special class is introduced: $POSITIONS$ is used to represent geographic classes' positions in space. $SPACE$ is related to $POSITIONS$ by:

$$\forall p(p \in POSITIONS \Leftrightarrow p \subseteq SPACE) \qquad \textbf{(1)}$$

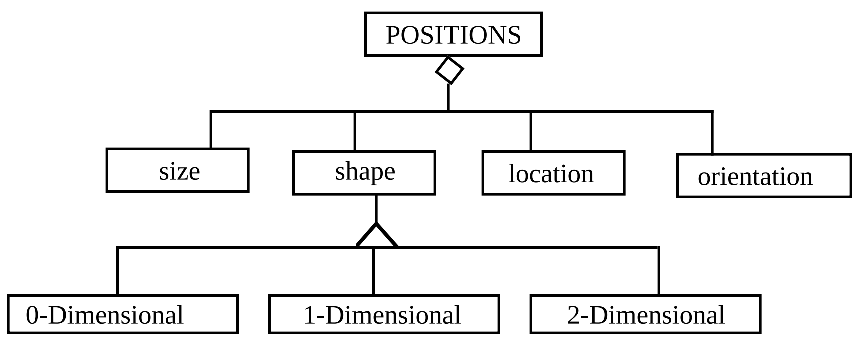
The domain of $POSITIONS$ is a finite subset of sets of *geometric figures*, i.e. points, lines and regions. A position is fully and non-redundantly determined by four elements [Tryfona and Hadzilacos, 1995a]: *shape*, *size*, *location* (centroid) and *orientation*; these are *part_of* $POSITIONS$.

In order to represent *shape*, the special classes $0-Dimensional$, $1-Dimensional$, and $2-Dimensional$ are introduced and related to $POSITIONS$ by a generalization hierarchy. $0-Dimensional$, $1-Dimensional$, and $2-Dimensional$ are (ISA) *shapes* of an object. As the position of a complex geographic object can be any combination of points, lines and regions in space, *shape* (as well as position itself) is determined by the higher dimension of geometric figures that constitute object's position (*dimension*). For example, a "high-tension tower" of an electricity company in a 2 dimensional map is represented by a set of discrete points (usually 3 or 4) and its shape is $0-Dimensional$.

Figures 3.2a and 3.2b represent object's position in space in two different ways. The first one (Figure 3.2a) is used when only the *shape* the geographic class matters.



(a)



(b)

Figure 3.2: Modeling objects' position in space.

III. A special “many-to-one” association $is\_located\_at$ relates each geographic class to its -not necessarily unique- position in space. There are several reasons why the position of a geographic object might be represented in more than one ways in the same application (most likely belonging to different application views). For example, a “city” may be represented as a point or a region depending on the scale of the map; going from one representation to the other is not always automatic and we may need to store both in the database. This is why we cannot restrict $is\_located\_at$ to be an “exactly one” association.

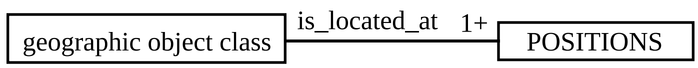
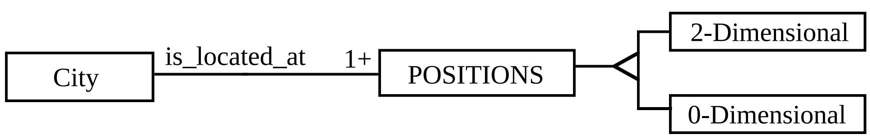
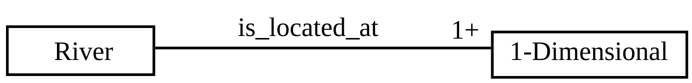


Figure 3.3: $is\_located\_at : geographic\_object\_class \rightarrow POSITIONS$.

Additionally, when a geographic object is represented uniquely then the class $POSITIONS$ is omitted. For example, Figure 3.4a illustrates a “city” as a $2-Dimensional$, and/or $0-Dimensional$ object class while Figure 3.4b represents a “river” as a $1-Dimensional$.



(a)



(b)

Figure 3.4: Modeling different views of (a) a “city”, (b) a “river”.

IV. Object classes $0-Dimensional$, $1-Dimensional$, and $2-Dimensional$ (which are connected to geographic classes via the $is\_located\_at$ association) have:

(a) in the middle area, the attribute GEOMETRIC_TYPE, whose data type is (respectively) one of the POINT, LINE and REGION types defined in Section 2.2 (this is the actual “position” of the object) and,

(b) in the lower area, the set of pre-defined methods (such as DISTANCE, LENGTH described in Section 2.2, geographic methods (b)) and others (like “move”) on objects’ position. They all represent the dynamic properties of the position of the object class. For example, to show that geographic objects may change their position in space, class *POSITIONS* “participates” at the operation $move(p_1) = p_2, \quad p_1, p_2 \in POSITIONS$.

In that way, objects are connected to their position in space which is represented by a geometric data type. The only way to access and manipulate position is via methods. All the descriptive attributes (e.g., (“river-id”) and non-geographic operations (e.g., “retrieve” name) of the geographic object class are presented in its area. Figure 3.5 depicts this approach.

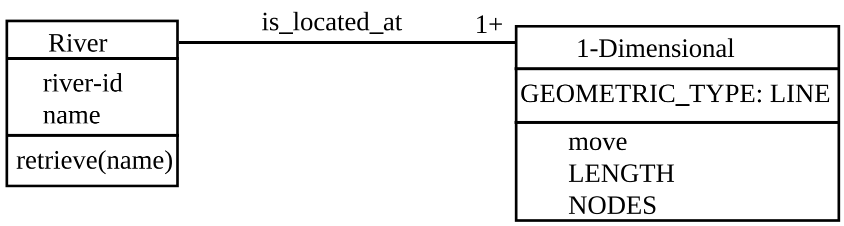


Figure 3.5: Modeling operations on geographic objects.

V. Spatial relationships (such as INSIDE, MEET, OVERLAP and others) among geographic classes are translated into spatial integrity constraints among objects’ positions. Two types of spatial relationships (associations) can occur in an object-oriented geographic database:

(a) relationships among different instances of the same object class; for example, two “land_parcels” have a common border (MEET); in that case, they are represented as operations of the object class, and

(b) associations between different object classes; for example a “network_hub” is always INSIDE the area it serves. Figures 3.6a and b depict this approach:

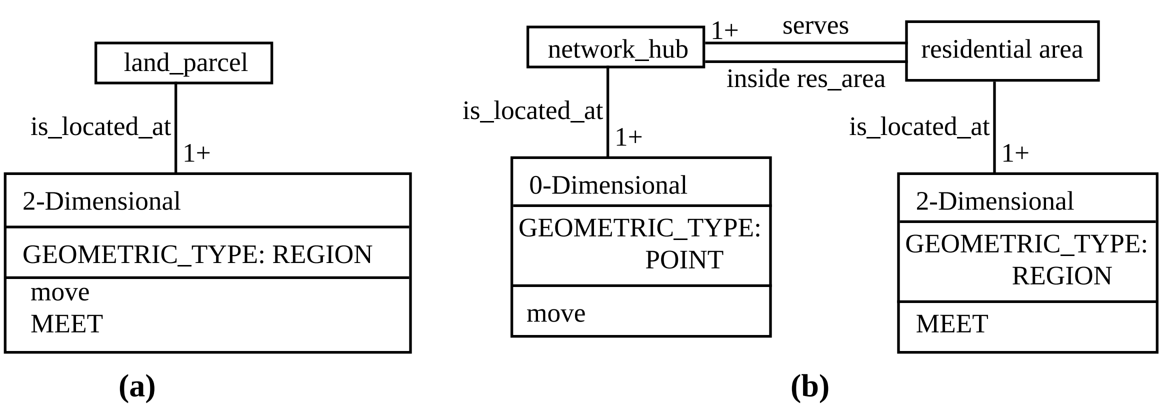


Figure 3.6: Expressing spatial relationships: (a) as operations, (b) as associations.

B. Extensions of OMT

VI. That "a network is an (ordered) set of network segments" differs from "a country being a set of people" in that the former grouping has a spatial dimension as well: the position of the network is the geometric union of the positions of its constituent segments -whereas nothing of the sort holds in the second case. To capture this additional semantic two new constructs are introduced: *spatial aggregation* (*spatial_part_of*) and *spatial grouping* (*spatial_member_of*).

- Let the aggregation (Spatial Part) $SP = C_1 \times C_2 \times \ldots \times C_n$, be a geographic class, and let $C_{i1}, \ldots, C_{ik}$ be its geographic parts. We say that SP is a *spatial* aggregation, if and only if the position of sp is the geometric union of the positions of its geographic parts:

$$\forall sp \in SP \; \forall c_i \; (c_i \in C_i) \; \left(sp = \langle c_1, \ldots, c_n \rangle \Rightarrow p(c_{i1}) \cup \ldots \cup p(c_{ik}) = p(sp)\right) \qquad \textbf{(2)}$$

where $\cup$, $\langle c_1, \ldots, c_n \rangle$ and p are the operators for geometric union, aggregation and the function $is\_located\_at$ respectively.

Figure 3.7a shows the graphic representation of spatial aggregation and Figure 3.7b the example of a "city" entity set composed by "name", "state", "country" (non-geographic classes) and "residential area", "park", "industrial area":

spatial_part_of

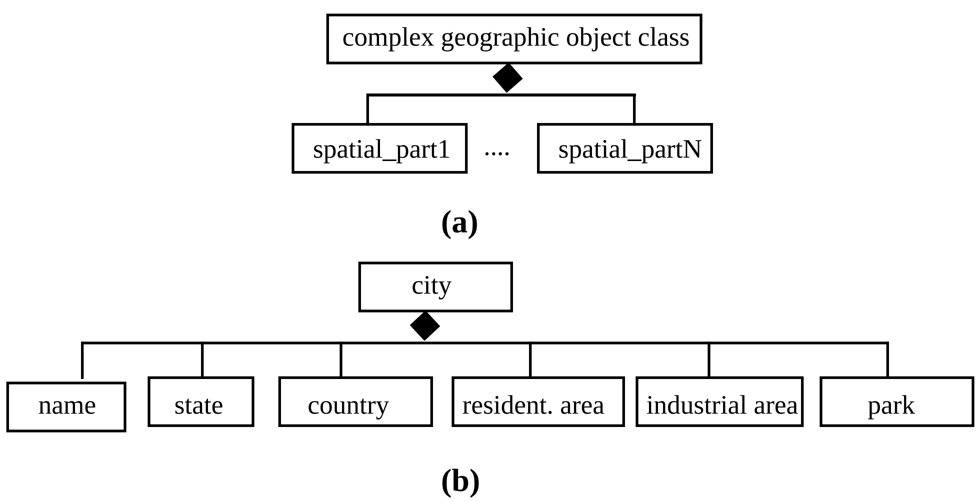


Figure 3.7: (a) spatial aggregation, (b) an example.

- Let SM (Spatial Member) be a grouping of geographic class C; we say it is a *spatial* grouping if and only if for every instance of SM, say sm, the instances of C which form sm are topologically inside it (topological relationship: COVERS):

$$\forall sm \in SM \; \forall c_i \in sm \left(p(sm) \text{ COVERS } p(c_i) \right) \qquad \textbf{(3)}$$

Figures 3.8a and b depict spatial grouping and an example of a network consisting of segments:

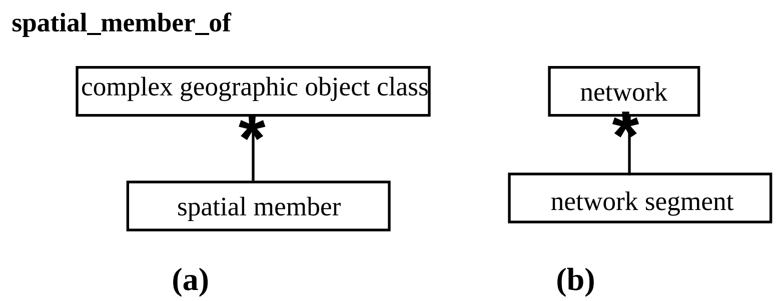


Figure 3.8: (a) spatial grouping, (b) an example of spatial grouping.

VII. Spatial or space-varying attributes are modeled as special object classes. The upper area of the object class figure contains the *name* of the spatial attribute, while the operation (lower) area shows the operations in which the spatial attribute participates. As a spatial attribute represents a domain of values for the whole space (field view), it is a *layer*. So, all operations on layer described in Section 2.2 can be applied.

Additionally, we need (based on its definition) to describe the layer as a set of geometric figures with associated values. For that purpose we use the *spatial_member_of* construct between the layer and its members. The object class describing the members has in its upper area the member name, and in the middle the attribute value and the GEOMETRY which is of geometric data type, i.e., can be POINT, LINE, REGION or any combination thereof, and shows the current geometry of the member. In each GEOMETRY an attribute value is connected. Members are distinguished from each other by their geometry[2].

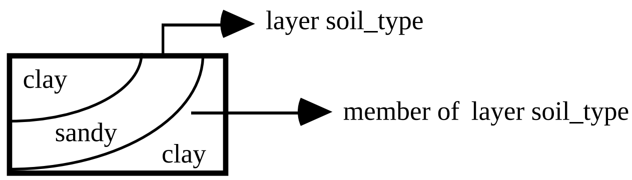
Furthermore, an association is needed to connect space-varying attributes to space:
Let SPA be the association that connects space-varying attributes to $SPACE$. It is:
Let $SPA: SPACE \rightarrow spatial\_attribute$, $is\_in: POSITIONS \rightarrow SPACE$ and $g \in GO$ a geographic object belonging to object class GO with position $p \in POSITIONS$.

$$SPA(g) = SPA\ (is\_in(is\_located\_at_{GO}(g))) \qquad \textbf{(4)}$$

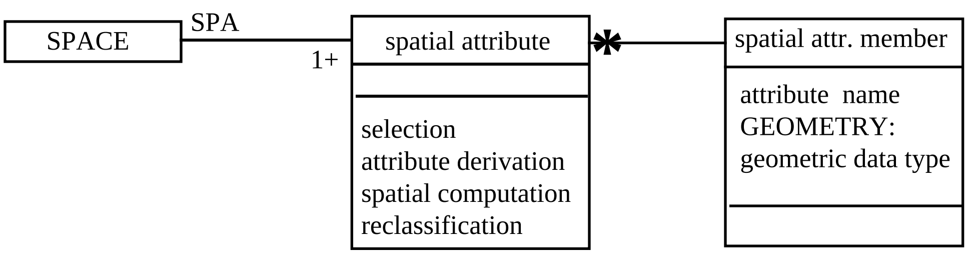
Figure 3.9a shows graphically a "soil_type" layer and its members. Figure 3.9b illustrates the connection between space and space-varying attributes. The idea is to keep separated the descriptive from the geographic characteristics (operations and attributes) of the geographic

[2] The reason we introduce the attribute GEOMETRY and not use the already existing GEOMETRIC_TYPE, which is also of geometric data type, is that we want to distiguish between parts of space created by the presence of different spatial attribute values and positions of objects in space (respectively).

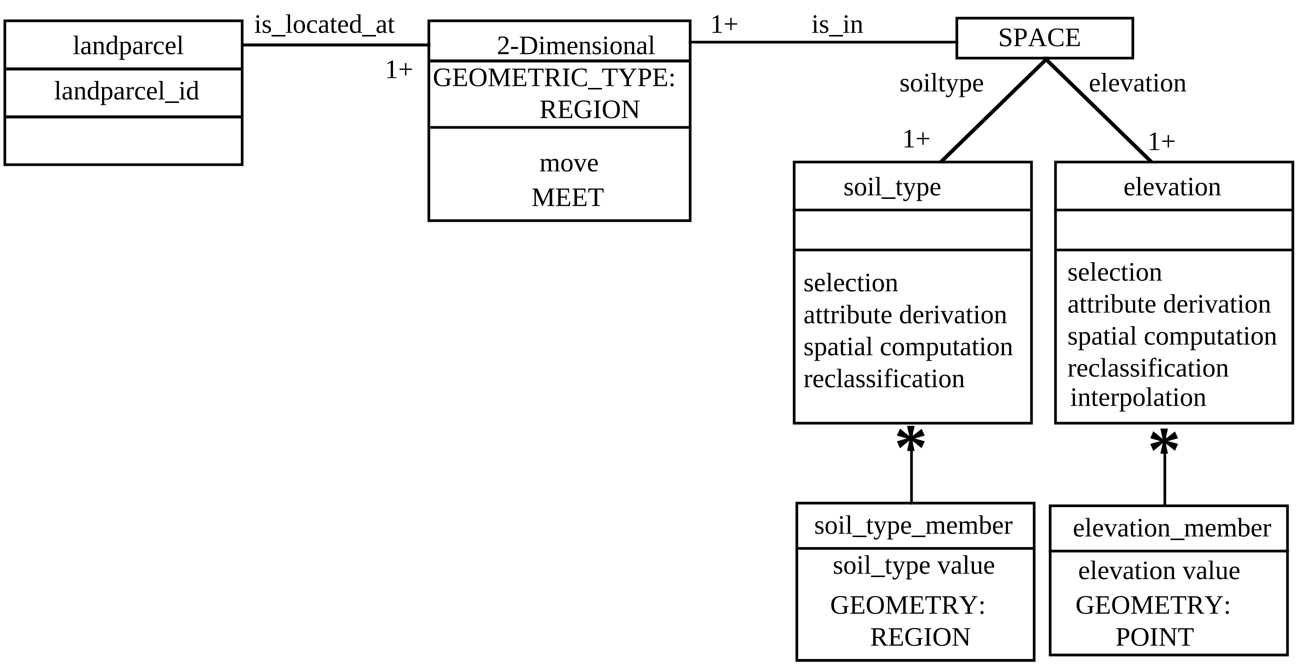
objects. Figure 3.9c shows the land parcel with its position in space. “soil_type” and “elevation” are spatial attributes and they are connected to $SPACE$. Object class “land_parcel” has only the descriptive properties as data, where the position ($2-Dimensional$) has the spatial properties: in practice, the designer relates the space-varying attribute to space and not to object and then restricts it (the attribute) to object’s position .



(a)



(b)



(c)

Figure 3.9: Modeling spatial attributes of geographic objects.

3.3 Discussion

The concepts of positions and space-varying attributes provide an elegant way to handle important issues of spatial applications (details can be found in [Tryfona and Hadzilacos, 1995b]):

- The field vs. object dichotomy corresponds to space-varying vs. geographic object classes.
- The raster vs. vector choice corresponds to modeling space as a set homomorphic to Z^2 or R^2.
- Fields correspond to space-varying attributes which correspond to layers, for which a complete set of operations defined.
- Spatial relationships are reduced to algebraic or geometric conditions (integrity constraints) on (or among) the object class *POSITIONS* [Hadzilacos and Tryfona, 1992].
- Attribute generalization -an important aspect of scaling and map generalization- is achieved through algebraic transformations of the functions which represent space-varying attributes.
- Fuzzy points specified using pairs of probability distributions suffice for modeling uncertainty and fuzziness and fuzzy geometric figures in the object class *POSITIONS* [Hadzilacos, 1995].

4. Example of Usage

In this section we present an excerpt from a real application dealing with a Network Utility Management System [UtilNets, 1994]. We show the conceptual level of modeling by using the graphical notation of the proposed Geographic Object Model. Let’s consider the following description:

“...A utility (underground) network is composed by links (line segments) and service reservoirs (points or regions). Its whole structure can be seen either as a linear or as a regional object. It is important to record the type of soil underground the network... Additionally, the network is related to the city it supplies with water...”

Based on the above, the *position* of the “network” is represented either by a line (1-Dimensional), or by a region (2-Dimensional). *Parts* of the network are the “reservoir” and the “link”.

The “soil_type_of_the_underground_network” (depicted by the dotted line) is a restriction of “soil_type” of the *space* to the position where the network exists. So, it is modeled as a space-varying attribute.

Figure 4.1 illustrates this description in the Geographic Object Model. (For reasons of simplicity we duplicated the figures depicting $0-Dimensional$, $1-Dimensional$, and $2-Dimensional$ object classes.)

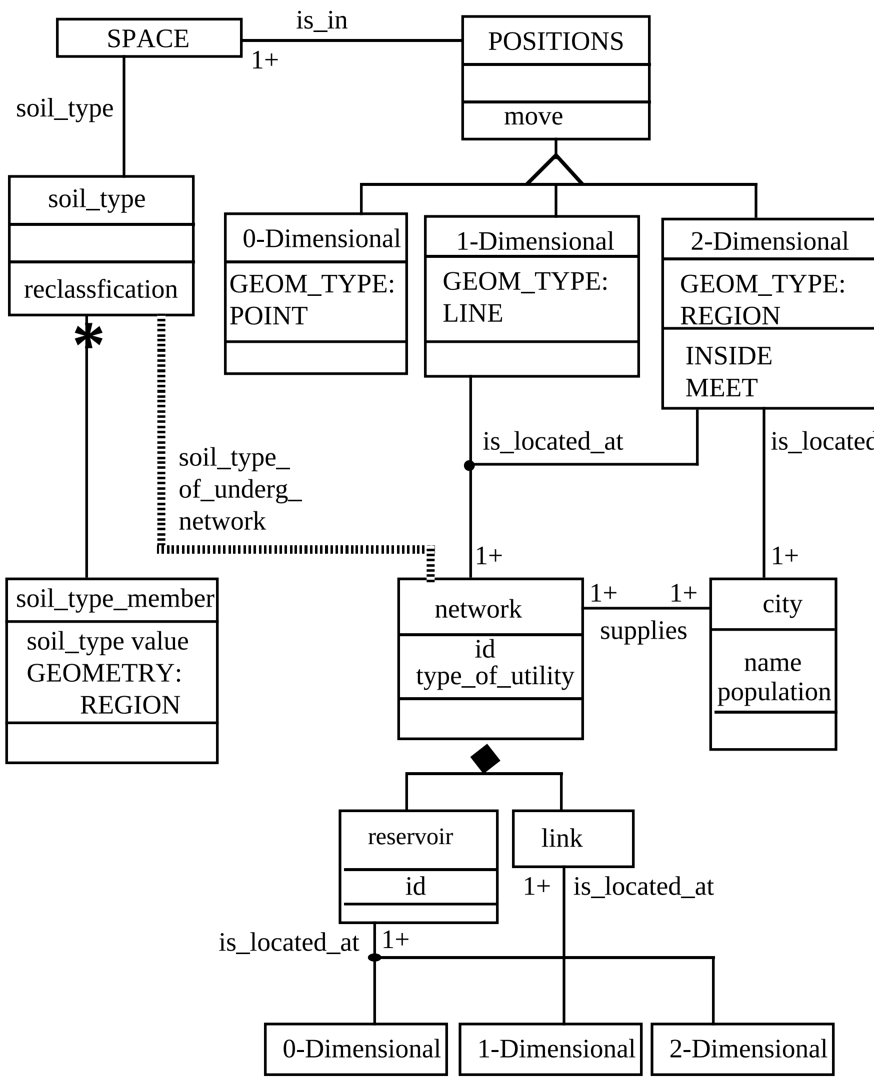


Figure 4.1: An excerpt from a Network Utility Management System by using the Geo-OM.

5. Conclusions and Future Work

Object-oriented modeling techniques allow for the representation of static and dynamic properties of applications objects. Dynamic properties refer to objects' behavior and is a vital part of the development and use of GIS. So far, valuable works exist in the area of conceptual modeling of geographic applications. These contribute to the efficient handling of static characteristics of geographic data. On the other hand, sparse works dealing with object-oriented

GIS lack the ability to provide a standard background to capture dynamics (i.e., semantics) of geographic objects.

In this paper we apply an object-oriented structured methodology to the design of GIS. By using OMT as a pilot environment, we provide to the designer a way to capture static and dynamic properties of spatial information and still use his/her favorite model: special object classes are introduced to represent space, objects' position and space-varying attributes, while dynamic properties (i.e. operations on geographic objects) are connected to them, in order to complete the picture of spatial information in terms of behavior. Our proposal introduces two new constructs, namely spatial aggregation and spatial grouping, which capture the spatial dimension of constructing complex objects. Additionally, the two widely used object- and field-based views are for the first time integrated in one object-oriented model. The main idea keeps geographic from descriptive data separated: by "removing" the geographic part, the application is still "valid" -and in terms of modeling correct- reflecting part of the real world as a "classical" system.

It is important that our approach does not depend on the Object Modeling Technique we show here; this is rather a prototypical example. Any other object-oriented model having been augmented by the above concepts and mechanisms will have the same ability in representing geographic information.

This work comes to fill the gap between conceptual modeling of GIS (where only the static properties matter) and implementation (where operations (dynamic properties) of geographic data is the central spring).

The long term objective of this research effort is the complete understanding of semantics of geographic objects. Semantics are captured not only by attributes and relationships but also by operations on objects. This will lead us to the definition of a pool of minimal semantics that are necessary to be exchanged between different applications at the conceptual level in order to make them communicate (issue of interoperability of semantics).

Acknowledgments

The authors wish to thank Nikolina Renieri and Andrea Rodriquez for their helpful discussions and comments on previous versions of this work.

6. References

Abiteboul, S., and Hull, R., 1987, *IFO: A Formal Semantic Database Model*, ACM TODS, 4, pp. 525-565.

Bancilhon, F., Delobel, C., and Kanellakis, P., 1992, *Building an Object-Oriented Database System: The Story of O_2.* Morgan Kayfmann Publishers, San Mateo, California.

Burrough, P., 1986, *Principles of Geographical Information Systems for Land Resources Assessment*, Oxford University Press.

Brodie, M., 1982, *On the Development of Data Models*, in "On conceptual Modeling", Brodie, Mylopoulos Schmidt (eds).

Chen, P.S., 1976, *The Entity-Relationship Model: Toward a Unified View of Data*, ACM TODS, 1, pp. 9-36

Delis, V., Hadzilacos, Th., and Tryfona, N., 1994, *An Introduction to Layer Algebra*, Proceedings of the 6th International Symposium on Spatial Data Handling, Edinbrough, UK

Hadzilacos, Th., 1995, *On Layer Based Systems for "Undetermined Boundaries"*, in "Objects with Undetermined Boundaries", Burrough and Frank (eds), (Taylor and Francis).

Hadzilacos, Th., and Tryfona, N., 1992, *A Model for Expressing Topological Integrity Constraints in Geographic Databases*. LNCS, 639, pp. 252-268, (Berlin: Springer-Verlag).

Hull, R., and King, R., 1987, *Semantic Data Modeling: Survey, Applications and Research Issues*, ACM Computing Surveys, Vol. 19, No. 3.

Kim, W., 1990, *Introduction to Object-Oriented Databases*, Cambridge, MA:MIT Press.

Nierstrasz, O., 1989, *A survey of Object-Oriented Concepts*, in "Object-Oriented Concepts, Databases and Applications", W. Kim and F. Lochovsky (eds), ACM Press.

Milne, P., Milton, S., and Smith, J.L., 1993, *Geographic Object-Oriented Databases - A Case Study*, International Journal of Geographic Information Systems, 7, 1.

Pelagatti, G., Belussi, A., Negri, M., Rossi, M., and Sbatella, L., 1991, *Design and Implementation of a Geographic Information System: The case of Regione Lombardia*, Proceedings of the 2nd European Conference on Geographic Information Systems, Brussels, Belgium, 2-5 April, 2, pp. 856-865.

Rumbaugh, J. , Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W., 1991, *Object-Oriented Modeling and Design*. Englewood Clifs, NJ:Prentice-Hall.

Scholl, M., and Voissard, A., 1991, *Object-Oriented Database Systems for Geographic Applications: An Experience with* O_2, International Workshop on Geographical Databases, Capri, Italy, May, 16-17.

Tang, A.Y., Adams, T.M., and Usery, E.L., 1996, *A Spatial Data Model Design for Feature-Based Geographical Information Systems*, International Journal of Geographic Information Systems, 10, 5.

Tryfona N., and Hadzilacos, Th., 1995a, *Geographic Applications Development: Models and Tools at the Conceptual Level*, 3rd ACM Workshop in Geographic Information Systems, Dec. 1-2, Baltimore, MD.

Tryfona N., and Hadzilacos, Th., 1995b, *Conceptual Data Modeling for Geographic Applications: Desiderata and Solutions*. TR-95.02.8, C.T.I., Patras, Greece.

Tryfona, N., 1994, *A Structured Database Design Methodology for Geographic Applications*, Ph.D Thesis, Dept. of Computer Engineering and Informatics, University of Patras.

Tsironis, G., 1992, *A Network Utility Management System*, Diploma Thesis, Dept. of Computer Engineering and Informatics, University of Patras.

UtilNets, 1994, *Work Program of Brite-Euram Project 7120*, DG XII, European Union, Brussels, Belgium.

Worboys, M., 1994, *Object-Oriented Approaches to Geo-referenced Information*, International Journal of Geographic Information Systems, 8,4, pp. 385-399.