A model to cultivate objects and manipulate fields

G. Camara, U.M. Freitas, R.C.M. Souza
Image Processing Division - INPE
Av. dos Astronautas, 1758
12.225, Sao Jose dos Campos SP - Brazil

M.A. Casanova, A.S. Hemerly
Rio Scientific Center - IBM Brasil
Av. Presidente Vargas, 824/844
20071-001 Rio de Janeiro RJ - Brazil

Claudia B. Medeiros
DCC - IMECC - UNICAMP - CP 6065
13081-970 Campinas, SP - Brazil

Abstract

Most approaches for data modelling in GIS are hampered by having to account for structural differences in collected data and by the need to accommodate different views of the same geographic reality. In this paper we propose a new data model for GIS which solves this problem by introducing a multi-level data abstraction philosophy that clearly distinguishes between conceptual construction and geometrical and topological representation. It supports multiple representations for geographic objects and features a concept similar to hypermedia links to account for the relationship between objects and their representations. The model is being used as a basis for the development of SPRING, a geographic data management system that includes functions for image processing, geographic analysis and digital terrain modelling.

1 Introduction

1.1 Motivation

Initial work on data modelling for geographic applications dealt primarily with geometrical and spatial data structures and their organization (including the fundamental notion of topology). The early data models directly reflected the underlying geometries and were closely linked to basic data structures. This situation is very clear in a large number of systems available today, where the user refers directly to arc-node structures, in the case of vector-oriented systems, or to grids or quadtrees, for raster-bases ones. Such systems are inherently limited in their extent, since changing the underlying data model usually means a complete software rewrite and, moreover, since they have limited abilities in combining raster and vector data. Thus, they reflect, at a lower level, the problems that arise at the modelling stage.

As Goodchild [Goo92] stated, the GIS industry has matured to a point where questions of data structures, algorithms and functionality are becoming standardized. Data modelling at a higher level is seen as playing a critical role in determining the usability and adequacy of a system. This concern has led to a number of conceptual formulations for geographic data models, and to a growing interest on object-oriented concepts.

The geographic data model we introduce in this paper reflects these remarks in so far as it follows the multi-level data abstraction philosophy introduced in [GV93] that clearly distinguishes between conceptual construction and geometrical and topological representation. In more detail, from the structural point of view, the data model:

- permits a geographic entity to have one or more representations, stored separately from the entity;
- uses hypermedia links to account for the relationship between geographic entities and their representations.

Now, from the behavioral point of view, the data model:

- permits defining very high-level operations, hiding from the user the lower-level operations on representations and their implementations,
- supports the simultaneous manipulation of different data structures representing the same geographic entity, thus equating the raster versus vector debate.

The model serves as a basis for the definition of the classes and operations of an object-oriented geographic data management system, called SPRING, that runs on UNIX workstations, under the X window system, and that was coded using the C^{++} language. SPRING is being developed at the Image Processing Division of the Brazilian National Space Institute - INPE - in cooperation with the Brazilian Agricultural Research Organization - EMBRAPA - and the IBM Brasil Rio Scientific Center.

Finally, this paper is organized as follows. Section 2 contains a preliminary discussion about geographic data modelling. Section 3 introduces the data model, while Section 4 analyzes the impact of the model on the design of GIS operations. Finally, Section 5 contains the conclusions.

1.2 Related Work

The title of this paper originates from a study presented in [Cou92], that discusses the need for GIS that allow combination and interaction of raster-based and vector-based data structures. The issues raised in the paper also point out that, while some geographic entities are better represented by the field model (e.g., human migrations) other are more adequately handled by using the object model (e.g., human-made artifacts such as roads).

As stressed in [RM92], "data modelling within a GIS is typically governed by the structures available to represent spatially related phenomena". As a consequence, GIS may constrain the types of geographic entities which can be manipulated, by restricting the allowable structures. Most recent results point out the importance of using the object-oriented paradigm to solve this problem (e.g., [ZM92, SV92, WHM90, MMS93, SA93, DRS93, AA93, PMP93]). Other authors propose different flavors of entity-relationship models, where the emphasis lies in the application, rather than in a general model (e.g., [FR93]).

Most of the above discussions present the advantages of encapsulation and reusability, but do not directly attack the problems of alternative representation and contextual changes. The advantages of dissociating an entity's representation from its description are discussed in different contexts (e.g [RM92, GR93]). In particular, [GR93] discusses briefly an example of separating graphic and geometric properties from objects themselves. However, the issues of multiple operations are not discussed, neither the distinctions among different kinds of vector and raster data.

Our model is an improvement over the above proposals in the sense that it considers the different levels of geographic entity definition. It thus eliminates the problem of hampering the modelling of a reality with low level concerns.

2 Multi-level Data Modelling for GIS

When adapted to geographic databases, the multi-level data abstraction philosophy introduced in [GV93] distinguishes between the following levels of abstraction:

- The *real world level* comprises the elements of the geographic reality that will be modelled, such as *crops*, *terrain*, *rivers* and *telephone networks.*
- The *conceptual level* introduces tools for formally modelling geographic elements at a high level of abstraction. The conceptual level should be formally defined, facilitate database design, act as a blueprint for the implementation of geographic data management systems and guide the design of data definition/manipulation languages and user interfaces in general.
- The *representation level* introduces the details pertaining to the geometrical and topological representation of the spatial properties of the geographic elements defined at the conceptual level.
- The *implementation (or physical or internal)* level covers the data structures and operations used to implement the representations, including spatial access methods, such as R-trees and quadtrees. A good model must never confuse conceptual structures with storage structures used in systems' implementations, since the latter are designed with other issues in mind, such as performance and storage utilization.

This multi-level approach to geographic data modelling facilitates the discussion about the dichotomies of field versus object and raster versus vector, as we shall see in Section 3. It also clearly indicates that the user interface of a geographic information system should reflect entities at the conceptual level, hiding as much as possible the details of the representation and implementation levels. Indeed, by working at the conceptual level, the user deals with abstract concepts that are closer to his daily reality, rather than having to understand the intricacies of representations and their implementation. At the same time, it allows modelling complementary operations for a given geographic entity, which may be influenced by its underlying representation.

The real world, the conceptual and the implementation levels have exact matches in traditional database design, whereas the representation level does not. Indeed, in traditional database design, the question of deciding how to represent the properties of an element defined at the conceptual level is too simple to deserve separate discussion. Moreover, properties have just one representation. For example, the database designer will decide the representation of employee names and salaries right when he defines employees; furthermore, it hardly makes sense to store employee salaries, say, in two or more currencies since the conversion routine is trivial most of the time.

By contrast, the representation of the spatial properties of geographic elements involves questions of scale, precision, cartographic projection, etc... that deserve careful attention. As discussed in detail in section 3, the database designer may decide to store spatial properties using alternative representations. Therefore, it is a reasonable design strategy to concentrate the discussion about representation in a separate level.

3 Structural Aspects of the Model

In this section, we discuss how to model geographic data at the conceptual and representation levels, using an object-oriented framework. Figures 1 and 2 at the end of this paper summarize the object class hierarchies of the model at these levels. whereas Figure 3 schematically exemplifies the concepts introduced. We leave all aspects related to the definition of methods to Section 4.

We follow [KL89]'s class-based framework, since there is no standard definition for object-oriented models. Very briefly, we assume that an object is an instance of a class and is characterized by its *state*, or set of attribute values, and *behavior*, or set of operations or methods that can be applied to the object. An object o can be constructed out of other objects $o_1,...,o_n$, in which case o is called *complex* and $o_1,...,o_n$ are called the *components* of c. If an object is not complex, then it is called *simple.* Classes can be structured into hierarchies; the ancestors of a class C in the hierarchy are sometimes called the *superclasses* of C.

3.1 Conceptual Level

We assume that we are given a collection of classes, called *geo-referencing classes*, whose objects describe regions of the Earth's surface (including degenerated regions: points and polylines) and are called *geo-regions.* As the name implies, these classes abstract geo-referencing schemes, hiding them from the conceptual level design of databases. We also assume that geo-regions may be simple or complex objects, in the latter case constructed out of components from the same geo-referencing class.

At the *conceptual* level, our model distinguishes two basic classes of objects: geographic fields and geographic objects. A database is therefore a collection of objects of these classes, subjected to certain restrictions described in this section. Also, geographic fields and geographic objects have attributes whose values are geo-regions. However, we do not require that all geo-regions in a database come from the same geo-referencing class, which means that two geo-regions may not be directly comparable.

Following [Goo92], a *geographic field* or *geo-field* defines a mapping that abstracts the spatial distribution of a geographic variable over some region of the Earth's surface. More precisely, a geo-field is an instance of the class GeoField and always has a *domain attribute*, whose value is a geo-region A, a *range attribute*, whose value is a set V, and a *geo-field mapping attribute*, whose value is a mapping from A into V. (We will also refer to A as the domain of the geo-field).

Motivated by the experience obtained during the design and implementation of the SPRING system, we introduce the following sub-classes of GeoField:

ThematicField - an instance of this class, called a *thematic field*, defines a (partial) geo-mapping $f : A \mapsto V$ such that V is a finite set. The elements of V are called *geo-classes* and, intuitively, define the "themes" of the thematic field. Assume that $V = \{v_1, ..., v_n\}$. Then, since f is a function, $\{f^{-1}(v_1), ..., f^{-1}(v_n)\}$ is a finite set of disjoint sub-regions of A. That is, a thematic field satisfies by definition the so-called "planar enforcement" rule.

DTM - an instance of this class, called a *digital terrain model* or simply a *DTM*, defines a geo-mapping $f : A \mapsto V$ such that f is defined only for a finite subset of A. Intuitively, a DTM specifies an approximation of the spatial distribution of a geographic variable (such as topography or geophysical data).

RemoteSensingField - an instance of this class, called an *remote sensing field*, defines a geo-mapping $f : A \mapsto V$ such that f is total, V is a finite set, A is divided into a finite set of cells of uniform size and, given any cell c in this set, f has the same value for all points in c. Intuitively, a remote sensing field defines the spatial distribution of a variable that represents data collected by remote sensing.

This basic class hierarchy of geo-fields can be further extended without hampering the fundamental concepts of the model.

We now turn to *geographic objects* or *geo-objects*, which are instances of the class GeoObject. Geo-objects can be either elementary, compound or weak.

An *elementary geographic object*, or briefly an *elementary geo-object*, corresponds to an individualizable entity of the geographic realm. It no components which are also geo-objects (but it may have components of other classes) and always has an explicit *location attribute* whose value is a simple or complex geo-region (again we will also refer to this geo-region as the location of the geo-object).

A *compound geo-object* is a geo-object constructed out of other geo-objects, whose locations must belong to the same geo-referencing class. The specific object constructors allowed are not relevant for the discussion that follows, except that we assume that they are a subset of the constructors used to build complex geo-regions. The location L of a compound geo-object O is not explicitly given by an attribute value, but it is rather computed from those of the component geo-objects. Roughly speaking, L is defined as a complex object constructed as O, except that each elementary geo-object that is a component of O is replaced by its location. Note that, by definition and previous assumptions, the location of a compound geo-object is also a geo-region in the same geo-referencing class as the locations of the component geo-objects.

A *weak geo-object* is a geo-object that contains only the location attribute and exists as long as it is part of a (unique) compound geo-object.

For example, we may model roads as geo-objects and the entire road network as a compound geo-object whose components geo-objects are the individual roads; if the road network contains certain local roads that are not relevant enough to be given explicit names and treated as (first-class) geo-objects, they may be modelled as weak geo-objects with respect to the road network.

Alternatively, if it is not necessary to individualize the roads, the entire road network may be modelled as a single elementary geo-object whose location is a complex geo-region (the road locations).

3.2 Representation Level

At the representation level, the design of a database becomes more flexible in the following ways:

- geo-fields lose the region and mapping attributes, which are now represented by one or more separate objects, called (geo-field) mapping representations. We stress that the mapping of a geo-field may be represented more than once in different ways.
- At the designer's discretion, the geo-objects of a given class may retain their location attribute, or they may have their location represented separately and collected together in the so-called geo-objects maps, again more than once.

From another perspective, the design of a database at the representation level can be viewed as a refinement of that at the conceptual level. The database designer starts with the geographic fields and geographic objects identified at the conceptual level and decides, for each class of geo-objects, whether their locations should be represented separately from the objects, and, for each class of geo-fields, how their mappings should be represented.

A database at the representation level is therefore a collection of geo-objects and geo-fields, together with representations for their locations and mappings, as described in this section.

3.2.1 Representation of Geo-Field Mappings

At the representation level, geo-fields lose the domain and mapping attributes, which are represented by one or more separate objects of the classes - ThematicMap, ThematicImage, Image, GRID, TIN and Pts, whose instances have familiar definitions. Objects of these classes are collectively called (geo-field) *mapping representations*. In detail, these classes are defined as follows:

ThematicMap - an instance of this class specifies a geo-field mapping $f : A \mapsto V$, where V is finite, by describing a finite set of disjoint sub-regions of A, using points, lines and polygons, and assigning each sub-region to an element of V (the regions may degenerate into points or lines). Therefore, $f(p) = v$ iff there is a sub-region r such that $p \in r$ and r is assigned to v (if there is no sub-region r such that $p \in r$, $f(p)$ is undefined). Since the sub-regions are disjoint and each one is assigned to a single element of V, f is well-defined. Note that we distinguish between a thematic map and a cadastral map, as the latter does not obey the planar enforcement rule and usually cannot be reduced to a single "theme".

THEMATICIMAGE - an instance of this class specifies a geo-field mapping $f : A \mapsto V$, where V is finite, by describing a matrix of non-overlapping cells of uniform size that completely cover A and assigning each cell to an element of $V \cup \{\lambda\}$. Therefore, $f(p) = v$ iff the cell c such that $p \in c$ is assigned to v and $v \neq \lambda$; if the cell is assigned to λ, $f(p)$ is undefined.

IMAGE - an instance of this class specifies a geo-field mapping $f : A \mapsto V$, where f is total and V is finite, by describing a matrix of non-overlapping cells of uniform size that completely cover A and assigning each cell to an element of V. (The argument about how f is defined is identical to the previous one).

GRID - an instance of this class specifies a geo-field mapping $f : A \mapsto V$ by describing a matrix (or grid) of uniformly spaced points of A and assigning each point to a value in V.

TIN - an instance of this class specifies a geo-field mapping $f : A \mapsto V$ by describing a triangular interpolation network.

PTS - an instance of this class specifies a geo-field mapping $f : A \mapsto V$ by describing a finite set of ordered pairs from $A \times V$.

To relate geo-fields (in this new sense, without the domain and mapping attributes) and mapping representations, we also consider as part of the data model, at the representation level, a IS_REPRESENTED_IN many-to-many relationship containing pairs of the form (f, r), where f is a geo-field and r is a representation for f, subjected to the restrictions: (1) if f is a thematic field then r is a thematic map or a thematic image; (2) if f is a remote sensing field then r is an image; and (3) if f is a DTM then r is an image a set of points, a grid or a TIN.

Since the role of a pair (f, r) is to navigate from f to r and vice-versa, we call (f, r) a *hyperlink*, by analogy with hypermedia systems [HS92].

The advantages and disadvantages of adopting various classes of mapping representations have been extensively discussed in the literature over the last decade. Most studies have come to the conclusion that all of them are useful alternatives and a general GIS should not strongly favor any of them.

The mapping representation classes considerably help organizing GIS operations. Indeed, many raster-based systems fail to distinguish between, for example, operations on thematic fields and digital terrain models and express operations in terms of "integer grids" and "floating-point grids", which should be seen as lower level operations on representations of the class GRID. Actually, Boolean operations on the so-called "integer grids" should be seen, at a higher level, as operations over thematic fields, whereas arithmetic operations on "floating-point grids", as operations over digital terrain models.

We stress that a geo-field may have more than one mapping representations. For example, an altimetry geo-field A_1 may have as mapping representations: the original restitution samples, G_1; a regular grid G_2 calculated by the "inverse-square to samples" method; a triangular grid G_3; the output contour lines G_4, with text associated; or a false-color image G_5.

3.2.2 Representation of Geo-Object Locations

At the representation level, the designer may decide that the geo-objects of a given class should retain their location attribute, or that they may have their location represented separately and collected together in a single object.

In the first case, the value of the location attribute of the geo-object must explicitly indicate a representation of a geo-region and a geo-referencing scheme. For example, the designer may decide that the location of an oil well or a soil sample may be represented by a pair of real numbers indicating latitude/longitude.

To account for the second option, we introduce a *geo-objects map*, or simply a *map* if the context allows, as an instance of the class GEOOBJECTSMAP with the following attributes:

- a description of the geo-referencing scheme used (e.g., cartographic projection and scale);
- a description of the geo-region A covered (in the geo-referencing scheme adopted);
- a RANGE attribute, whose value is a set R of geo-regions, in the geo-referencing scheme adopted, wholly contained in A.

We also define two specializations of GEOOBJECTSMAP:

CADASTRALMAP - whose instances, called *cadastral maps*, are such that the value of RANGE is a collection of sets of points, lines and polygons contained in the area defined.

NETWORKMAP - whose instances, called *network maps*, are such that the value of RANGE is a collection of sets of points and lines.

Therefore, in the case of a cadastral map, a geo-region is a set consisting of points, lines and polygons in a given coordinate system, while, in the case of a network map, it is a set consisting of points and lines.

To relate geo-objects and their locations in maps, we extend the IS_REPRESENTED_IN many-to-many relationship to contain pairs of the forms:

- (G, M), where M is a compound geo-object and M is a map or a geo-field mapping representation;
- $(G, (M, r))$, where G is an elementary geo-object, M is a map and r is a geo-region belonging to the range R of M.

Again, we call these pairs *hyperlinks* and, in the second case we say that the hyperlink is *anchored* in r.

For added flexibility, in the first case, we also allow any geo-field mapping representation to be used instead of a geo-objects map. Thus, a database may have a hyperlink from a compound geo-object representing a road network and a raster image, say.

Note that a geo-object may have several location representations for many reasons, such as: (1) the representations belong to distinct maps that differ in terms of the geo-referencing scheme used, the precision adopted, the level of detail captured, the theme covered or that simply come from different sources; (2) they describe different historical versions of the same location; (3) they partially represent the geo-object location in different regions (covered by different maps).

For example, a street, defined as a geo-object, may be represented by a set of polygons on a cadastral map, by a line on a small-scale emergency planning city map, etc... The cadastral map may have been obtained from a 1910 land survey and the small scale city map from a more recent aerial photogrammetric survey. The street may cross several city maps (sheets) and, hence, it may be represented by different lines, one in each map (sheet). Note that all these representations are associated to a single geo-object.

The added flexibility introduced by multiple location representations has its price, though. Indeed, the location representation of a geo-object remains undefined until the user selects a map where location of the object is represented. We will discuss the full impact of this point in Section 4.

3.2.3 Layers

Since geo-objects maps and geo-field mapping representations have certain similarities, we introduce the notion of *representation layer* or simply *layer* as a generalization of both concepts. A layer thus corresponds to a generalized notion of a map and represents, for a given geographic region, the values of a geo-field mapping or the locations of a set of geo-objects. The notion of layer stresses the integration of different data formats, e.g., allowing combination of functions typically used in conjunction with remote sensing images (in raster format) with those of network analysis (where data is usually represented in vector format).

4 Behavioral Aspects of the Model

4.1 Implications of a Multilevel Model

The consequences of adopting a multilevel, object-oriented model are threefold:

- From the design perspective, it permits the incremental definition of geographic databases, starting with high-level geographic entities and their behavior and moving towards the geometrical and topological representation details.
- From the user interface perspective, it allows users to operate at a higher level of abstraction, hiding much of the so-called "GIS assembler". Naturally, experienced users will still be able to manipulate more specific objects, such as geo-field representations, if so desired.
- From the implementation perspective, it permits defining one or more different implementations for the operations on geographic entities at the conceptual level, using the operations defined for the objects at the representation level.

This section focus on the last point, pointing out the advantages and difficulties of separating geo-fields and geo-objects from the representation of their mappings and locations, respectively, from the point of view of operation definition.

4.2 Defining Behavior at the Conceptual Level

A geo-field should be viewed primarily as a (database) object over which high-level GIS operations can be specified, and a geo-field mapping representation, as a lower-level object (or data structure) over which some of the high-level operations can be implemented. The same reasoning applies to geo-objects and their location representations. The user is then relieved of the duty of dealing directly with lower-level vector or raster operations. For instance, the user would refer to the "combination of the land use map and the "highway map", without being aware that this implies combining a vector (highway) and a raster (land use) layers, which would in turn imply determining a common structure for performing this operation.

However, to fully realize this promise, we must take care of several problems. First, we must equip the basic subclasses of GEOFIELD and GEOOBJECT with enough high-level operations that relieve the user from resorting to the lower-level operations on their representations. Second, we must include in the interface tools that allow the application designer to specify new high-level operations for geo-field and geo-object classes, old or newly defined, and to describe their implementation in terms of the lower-level operations on representations. Preferably, the tools should allow the designer to describe more than one implementation for the same high-level operation, depending on the class of the representation selected. Third, we must include in the system some mechanism to deal with the fact that a high-level operation on a geo-field or geo-object may have multiple execution alternatives, either because there are multiple implementation alternatives or because there are multiple representations for a given geographic entity.

For example, the application designer may specify a DECLIVITY sub-class of DTM with an operation `calcdeclivity` to calculate declivities. He may also specify that objects in DECLIVITY may only have representations from the classes TIN and GRID and describe two implementations for `calcdeclivity`, one for each representation class. At some point in time, the database may then have an object d of the class DECLIVITY that has several distinct representations belonging to the classes TIN and GRID. If the user applies `calcdeclivity` to d, the system must choose which implementation of `calcdeclivity` should be applied to which representation of d to realize the operation.

Similar issues are raised when separating geo-objects from their representations. From the onset, this complicates the use of the various spatial operators proposed in the literature since they directly deal with geo-object representations (assumed to be unique for a given geo-object). For example, consider the query 'Which is the area of County C?' and assume that C has several representations given by maps with distinct scales and cartographic projections. Then, the query is ambiguous since the system would not know which representation to use.

4.3 Dealing with Multiple Execution Alternatives

In general, we detect the following major approaches to solve the problem of multiple execution alternatives. First, when the user invokes a high-level operation on a geo-field, the system may perform all possible alternatives to execute the operation, which is not at all practical.

Second, the system may restrict the definition of new classes of geo-fields or geo-objects and their operations to avoid ambiguities. An option is to dismiss this alternative since it blocks exactly the added flexibility the model permits. On the other hand, in certain cases it might be feasible to include additional parameters into the spatial operations that explicitly indicate which representation to use. For example, the query 'Which is the area of County C?' would be intuitively rephrased to read 'Which is the area of County C, using map M?'. As another example, the distance operation, typically defined with two operands, would have three operands: the two objects whose distance is requested and a third parameter indicating which map to use; a second variant would have four parameters, indicating two objects and two maps. Again, some ambiguity may arise if the maps differ in terms of the geo-referencing scheme, scale, precision, etc.

Third, the system may be equipped with an optimizer that automatically decides which implementation of the operation is the most convenient to use, thereby choosing the appropriate representation of the geo-field or geo-object. If the representation chosen is not available, the optimizer may in fact decide that the best strategy is to create a temporary representation from those available and then invoke the necessary conversion operations. The optimization rules may consider, in addition to a cost function, information about the current user session, such as the current scale.

Fourth, the system may interact with the user, and ask him to clarify the situation. In this case, the system may filter any implausible alternative before returning to the user, based again on contextual information about the current user session. The solution adopted may not treat all situations uniformly, to avoid inflexible or excessively verbose interfaces.

Notice that the use of contextual information about the current user section assumes greater importance. Indeed, the user may start by selecting a region with the geographic object representations of all geographic entities of interest (both geo-fields and geo-objects). From then on, all spatial operations will implicitly refer to these representations.

5 Conclusions

This paper presented a new model for geographic data which allows separating the location representation from the geo-object, as well as the mapping representation from the geo-field. This separation provides a solution to the raster versus vector debate, which hampers modelling geographic reality in an appropriate way. Indeed, geo-fields and geo-objects become objects over which high-level GIS operations can be specified, while representations are lower-level objects (closer to the data structure concept) over which some of the high-level operations can be implemented. The full implications of this separation were discussed in Section 4.

References

[AA93] G. Alonso and A. El Abbadi. GOOSE: Geographic object oriented support environment. In *Proc. ACM/ISCA Workshop on Advances in Geographic InformationSystems*, pages 38–43, 1993.

[Cou92] H. Couclelis. People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in Computer Science 639, pages 65–77, 1992.

[DRS93] B. David, L. Raynal, and G. Schorter. GeO2: Why objects in a geographical DBMS? In *Proc 3rd International Symposium Large Spatial Databases*, 1993.

[FR93] R. Fernandez and M. Rusinkiewicz. A conceptual design of a soil database for a GIS. *International Journal of Geographical Information Systems*, 7(6):525–540, 1993.

[Goo92] M. Goodchild. Geographical data modelling. *Computers and Geosciences*, 18(4):401–408, 1992.

[GR93] O. Gunther and W-F Riekert. The design of GODOT: an object-oriented geographic information system. *IEEE Data Engineering Bulletin*, pages 4–9, september 1993.

[GV93] J. M. Gomes and L. Velho. Modelling in computer graphics. In *VI Brazilian Symposium on Computer Graphics and Image Processing*, 1993. tutorial in Protuguese.

[HS92] W. Hall and D. Simmons. An open model for hypermedia and its application to geographical information systems. *Computer Graphics Forum*, 2(3), 1992.

[KL89] W. Kim and F.H. Lochovsky, editors. *Object-Oriented Concepts, Databases, and Applications*. ACM Press, New York, 1989.

[MMS93] P. Milne, S. Milton, and J. Smith. Geographical object-oriented databases: a case study. *IGIS*, 7(1):39–56, 1993.

[PMP93] N. Pissinou, K. Makki, and E. Park. Towards the design and development of a new architecture for geographic information systems. In *Proc. 2nd International Conference on Information and Knowledge Management- CIKM*, pages 565–573, 1993.

[RM92] J. F. Raper and D. J. Maguire. Design models and functionality in GIS. *Computers and Geosciences*, 18(4):387–400, 1992.

[SA93] R. Subramanian and N. Adam. The design and implementation of an expert object-oriented geographic information system. In *Proc. 2nd International Conference on Information and Knowledge Management- CIKM*, pages 537–546, 1993.

[SV92] M. Scholl and A. Voisard. *Building an Object-oriented System – the Story of O2*, chapter Geographic Applications – an Experience with O2. Morgan Kaufmann, California, 1992.

[WHM90] M. Worboys, H. Hearnshaw, and D. Maguire. Object-oriented data modelling for spatial databases. *IGIS*, 4(4):369–384, 1990.

[ZM92] F. Zhan and D. Mark. Object-oriented spatial knowledge representation and processing: Formalization of core classes and their relationships. In *Proc 5th International Symposium on Spatial Data Handling*, pages 662–671, 1992. Volume 2.

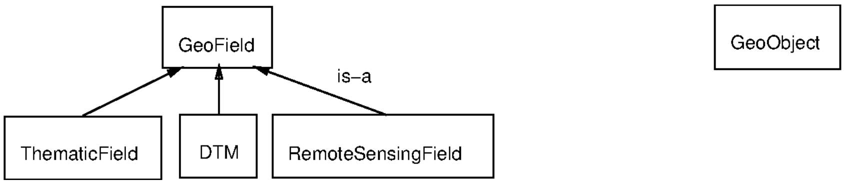

Figure 1: Summary of the Conceptual Level of the Model

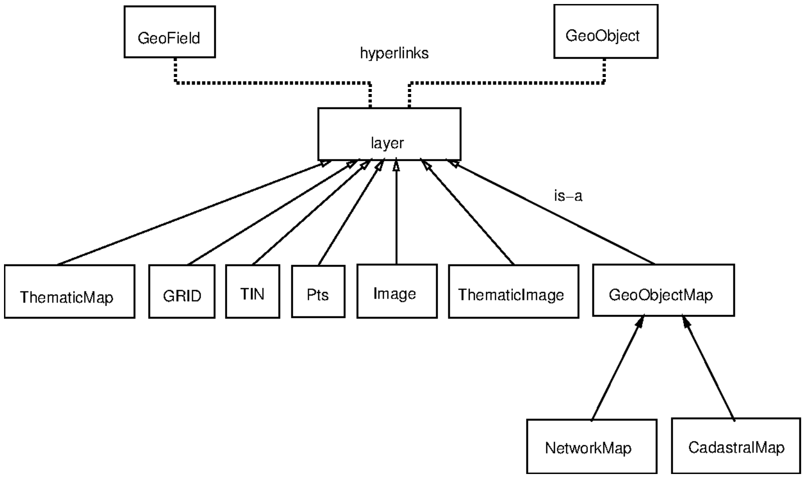

Figure 2: Summary of the Representation Level of the Model

Typical GIS Applications

Application	Typical Scales	Typical Real-World Element	Object Types	Representation Types	Typical Operations
Forestry	1:10.000 to 1:500.000	vegetation cover	thematic field	thematic map thematic image	spatial analysis overlay
			remote sensing field	image	image enhancement/ image classification
Agriculture	1:25.000 to 1:250.000	crop cover	thematic field	thematic map thematic image	spatial analysis overlay
			remote sensing field	image	image enhancement/ image classification
		terrain elevation	DTM	GRID or TIN	slope/aspect
Geology	1:100.000 to 1:2.500.000	terrain elevation	DTM	GRID or TIN	3D visualisation
		gravimetry	remote sensing field	image	image enhancement
Electrical Networks	1:1.000 to 1:10.000	poles, transmission lines	geo-object	network map	spatial query and dedicated calc.
Urban Studies	1:1.000 to 1:25.000	parcels, streets	geo-object	cadastral map	spatial query
		land use	thematic field	thematic map thematic image	spatial analysis

Figure 3: Examples of the Concepts