GeoInformatica 5:3, 221–260, 2001
© 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.

OMT-G: An Object-Oriented Data Model for Geographic Applications

KARLA A.V. BORGES,[1,2] CLODOVEU A. DAVIS JR.[1] AND ALBERTO H.F. LAENDER[2]
[1] PRODABEL - Empresa de Informática e Informação do Município de Belo Horizonte, Av. Presidente Carlos Luz, 1275 31230-000, Belo Horizonte, MG, Brazil
E-mail: {karla, clodoveu}@pbh.gov.br
[2] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Av. Presidente Antônio Carlos, 6627 31270-010, Belo Horizonte, MG, Brazil
E-mail: {kavb, laender}@dcc.ufmg.br

Received March 15, 1999; Revised June 1, 2000; Accepted March 29, 2001

Abstract

Semantic and object-oriented data models, such as ER, OMT, IFO, and others, have been extensively used for modeling geographic applications. Despite their semantic expressiveness, such models present limitations to adequately model those applications, since they do not provide appropriate primitives for representing spatial data. This paper presents OMT-G, an object oriented data model for geographic applications. OMT-G provides primitives for modeling the geometry and the topology of spatial data, supporting different topological structures, multiple views of objects, and spatial relationships. OMT-G also includes tools to specify transformation processes and presentation alternatives, that allow, among many other possibilities, modeling for multiple representations and multiple presentations. In this way, it overcomes the main limitations of the existing models, thus providing more adequate tools for modeling geographic applications. A comparison with other data models is also presented in order to stress the main advantages of OMT-G.

Keywords: Geographic Information Systems, geographic data modeling, geographic software design, database modeling

1. Introduction

The first data models developed for geographic applications were guided by existing GIS internal structures, forcing the user to adjust his/her interpretation of spatial phenomena to whatever structures were available. As a consequence, the modeling process did not offer mechanisms that would allow for the representation of the reality according to the user's mental model. Even well-known semantic and object-oriented data models, such as the Entity-Relationship (ER) model [11], the Object Modeling Technique (OMT) model [39], and the IFO model [1], do not offer adequate facilities to represent geographic applications. Even though these models are highly expressive, they present limitations to the adequate modeling of such applications, since they do not include geographic primitives that would allow for a satisfactory representation of spatial data.

The difficulties in using such models are countless, among which the fact that many geographic applications need to deal with details such as location constraints, time of observation, and accuracy [34]. Furthermore, in conventional models it is impossible to distinguish between object classes that have a geographic reference and purely alphanumeric classes. It is also difficult to represent the geometric nature of objects and the spatial relations between them. Spatial relations are abstractions that help us to understand how, in the real world, objects relate to each other [31]. Many spatial relations need to be explicitly represented in the application's schema, in order to make it more understandable. Topologic relations are fundamentally important to the definition of spatial integrity rules [5], which in turn determine the geometric behavior of objects.

There are particular characteristics of geographic data that make modeling more complex than in the case of conventional applications. Modeling the spatial aspects is fundamentally important in the creation of a geographic database, mainly because it deals with an abstraction of geographic reality where the user's view of the real world varies, depending on what he/she needs to represent and what he/she expects to gain from this representation. It can be perceived that modeling geographic data requires models which are more specific and capable of capturing the semantics of geographic data, offering higher abstraction mechanisms and implementation independence. Within this geographic context, concepts such as geometry and topology are important in the determination of spatial relationships between objects. These concepts are also decisive in the data entry process, and in spatial analysis.

In addition to geometry, spatial location, associated information, and temporal characteristics, geographic data have diverse origins. Spatial environmental data are an example of such diversity, since they encompass available data on topography, weather, soil properties, geology, vegetation, land use, hydrography, and water quality. Some of these phenomena, such as elevation and soil properties, vary continuously in space. Others, such as geological fault lines and river networks, can be discretized, while some can belong to both categories, depending on the level of detail considered [25].

This paper discusses peculiarities of geographic data, describes the requirements of a geographic data model and, as a response to the observed deficiencies, proposes a data model for geographic applications, named OMT-G (Object Modeling Technique for Geographic Applications), which was initially based on the classic OMT class diagram notation [6], and later adapted to approach the concepts and notation of the Unified Modeling Language (UML) [37], [39]. This course of action has been taken because the Object Management Group (OMG) has adopted UML, a natural evolutionary step from OMT, as its standard modeling language.

OMT-G offers primitives that provide the means for modeling the geometry and topology of geographic data, making the modeling of geographic applications easier. Modeling using OMT-G reduces the gap between the conceptual design and the implementation of geographic applications, by allowing a more precise definition of the required objects, operations, and visualization parameters. This is achieved by using three different diagrams to (1) specify object classes and their relationships, (2) specify transformation operations between classes, and (3) specify the various visual aspects each object class may assume as required by the application. In this process, some geographic application design issues that are seldom investigated in the spatial data modeling literature are considered, such as spatial integrity constraints [5] and multiple representations [15], [16].

The paper is organized as follows. The remainder of this section presents the main factors involved in the space discretization process and the basic requirements for a geographic data model. Section 2 describes the OMT-G data model, including primitives for the class, transformation, and presentation diagrams. Section 3 discusses an example of use of OMT-G. Section 4 compares OMT-G with other data models for geographic applications. Finally, Section 5 presents our conclusions.

1.1. Geographic data abstraction levels

Data models can vary according to the level of abstraction they provide. For geographic applications three levels of abstraction can be envisaged (figure 1):

- *Conceptual representation level*—Provides a set of formal concepts with which geographic entities, such as rivers, buildings, streets and vegetation, can be modeled as perceived by the user, in a high abstraction level. Basic classes to be created in the database, continuous or discrete, are defined at this level. These classes are associated with spatial representation classes which vary according to the user's degree of perception. This level has no direct correspondent in the design of traditional database applications, since such applications seldom deal with representation (or multiple representation) issues. As an example, consider an application involving schools. In a conventional system, the `School` class would include identification attributes, such as the school's name and number, and location attributes, such as the school's address. In a GIS, the location attribute could also be represented alphanumerically, as a street address, but it can be better represented geographically, as a set of coordinates. The alphanumeric encoding of the street address in a conventional system usually does not vary; on the other hand, the school can be represented in a GIS by a symbol, or by the limits of its building, or by the parcel in which the school has been built, or even by all of these representations combined.
- *Presentation level*—Provides tools with which to specify the various different visual aspects that the geographic entities will have to take when used as part of an application. Classes are defined at the conceptual representation level considering all the necessary representation alternatives for each phenomenon. This notion is further refined at the presentation level, where every representation alternative is associated to one or more presentations. These include the selection of simple graphic attributes for screen visualization, as well as sophisticated classification schemes used in thematic mapping and complex map generalization operations such as displacement of features to enhance readability in a printed map.
- *Implementation level*—Defines standards, storage mechanisms, data structures, and standard functions to implement physically each representation, as defined at the conceptual representation level, and each required presentation, as defined at the presentation level.

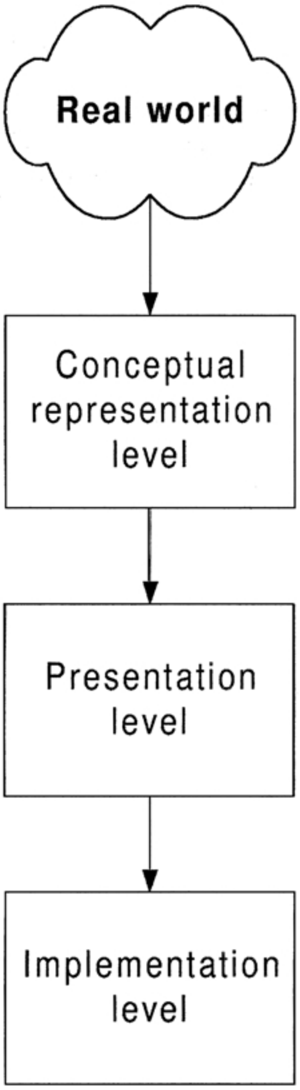


Figure 1. Geographic applications specification levels.

1.2. Requirements for a geographic data model

Considering the peculiarities of geographic data, and based on our experience on modeling geographic applications for the city of Belo Horizonte (Minas Gerais, Brazil) and on previous works in the literature [3], [4], [8], [10], [26], [28], [34], [41], a set of requirements deemed necessary to a data model targeted at geographic applications is listed hereafter.

A data model for geographic applications must: (a) provide a high abstraction level, allowing for the representation of the fields and objects views introduced in [22], [24]; (b) represent and differentiate among the numerous types of data involved in geographic applications, such as point, line, area, image, TIN, and so on, using appropriate primitives and constructs; (c) represent different types of spatial relations, from simple associations to complex networks; (d) be able to specify spatial integrity constraints; (e) support georeferenced classes and conventional classes, as well as the relationships among them; (f) support spatial aggregation relationships; (g) represent multiple views of a given geographic object; (h) be able to express versions and temporal series, as well as temporal relationships; (i) be implementation independent; and (j) provide an easy and clear visualization and understanding of the data structure.

2. The OMT-G data model

2.1. Model overview

Starting from the primitives of the UML class diagram, geographic primitives were introduced with the objective of increasing its semantic capabilities, thereby reducing the distance between the mental model of the space to be modeled and the usual representation model. Therefore, OMT-G provides primitives to model the geometry and topology of geographic data, providing support for ''whole-part'' topologic structures, network structures, multiple views of objects, and spatial relationships. Besides, the model allows for the specification of alphanumeric attributes and associated methods for each class. The main strong points of the model are its graphic expressiveness and its representation capabilities, since textual annotations are replaced by the drawing of explicit relationships, representing the dynamics of the interaction between the various spatial or non-spatial objects.

The OMT-G model is based on three main concepts: *classes*, *relationships*, and *spatial integrity constraints*. Classes and relationships are the basic primitives that are used to create application schemas with OMT-G. For that purpose, OMT-G proposes the use of three different diagrams in the process of designing a geographic application. The first, and more usual one, is the *class diagram*, in which all classes are specified, along with their representations and relationships. From this diagram, it is possible to derive a set of spatial integrity constraints that must be observed in the implementation. When the class diagram indicates the need for multiple representations of any class, or when the application involves the derivation of some class from others, a *transformation diagram* must be built. In it, all transformation processes can be specified, allowing for the identification of any required methods for the implementation. Finally, a *presentation diagram* must be built in order to provide guidelines for the visual aspect of objects in the implementation. There can be several visual aspects for any given class, which allows for the definition of a view or set of views for each application or group of users. The primitives for each of these diagrams will be covered in the next sections.

The identification of spatial integrity constraints is an important activity in the design of a schema for a particular database application and involves the identification of the integrity constraints that must hold on the database. The main types of integrity constraint that occur frequently in database modeling are domain constraints, key and relationship structural constraints, and general semantic integrity constraints [19]. Cockcroft [14] extends that classification in order to encompass the peculiarities of spatial data. This classification is based on the distinction between topological, semantic, and user rules, as follows.

- Topological integrity constraints. Topology is the study of geometrical properties and spatial relations. There has been some theoretical research into the principles of formally defining spatial relationships [17]. These principles can be applied to application-specific entities and relationships to provide a basis for integrity control. Area subdivision is an example of this constraint. One city's administrative regions must be contained within the city limits, and there must not have any spot in the municipal territory that belongs to more than one administrative region or to none.
- Semantic integrity constraints. These constraints are concerned with the meaning of geographic features. Semantic integrity constraints apply to database states that are valid by virtue of the properties of the objects that need to be stored. An example of this constraint is the rule that does not allow a building to be intercepted by a street segment.
- User defined integrity constraints. User defined integrity constraints allow database consistency to be maintained as defined by the equivalent of ''business rules'' in non-spatial database management systems (DBMS). This type of constraint acts, for instance, on the location of a gas station, which, for legal reasons, must lie farther than 200 meters from any existing school. The municipal permitting process must consider this limitation in its analysis. User-defined rules may be stored and enforced by an active repository.

In the OMT-G model, topological integrity constraints are achieved through spatial aggregation, spatial relationship, connectivity, and geo-field integrity rules. Likewise, semantic integrity constraints are achieved through spatial relationship integrity rules. User-defined integrity constraints are in turn obtained from methods that can be associated to the classes. These rules are described in the next sections.

Starting from the above described principles, OMT-G fulfills all the requirements for a geographic data model presented in Section 1.2, except for the temporal characteristics. In addition, it provides the following features: (a) follows the object-oriented paradigm, supporting the concepts of class, inheritance, complex object, and method; (b) represents and distinguishes the several types of data involved in geographic applications, using a symbolic depiction that allows the immediate understanding of the nature of the data, thereby eliminating the extensive hierarchy of classes usually employed to symbolize the geometry and the topology of spatial objects; (c) represents the interaction between the objects, making both spatial relations and simple associations explicit; (d) represents ''whole-part'' topological structures and networks through spatial aggregation; (e) formalizes the possible spatial relations, taking under consideration the geometric shape of the class; and (f) translates topological and spatial relationships into spatial integrity constraints. OMT-G primitives lead to three diagrams: *class*, *transformation* and *presentation*.

2.2. *Class diagram*

In OMT-G, the *class diagram* is used to describe the structure and contents of a geographic database. It contains specific elements of the structure of the database, in special object classes and their relationships, and no transformations or other dynamic processes are considered. The class diagram only contains fixed rules and descriptions that define, conceptually, how the data are to be structured, including information on the representation that is to be adopted for each class. For that reason, the class diagram is the most fundamental product of the conceptual representation level, as described in Section 1.1. In the following sections, the OMT-G primitives that are used to create the class diagram for a geographic application are described.

2.2.1. Class Structure. The classes defined by the OMT-G model represent the three main groups of data (continuous, discrete, and non-spatial) that can be found in geographic applications, thereby allowing for an integrated view of the modeled space. The classes can be *georeferenced* or *conventional*.

The distinction between conventional and georeferenced classes allows different applications to share non-spatial data, therefore making it easier to develop integrated applications and to reuse data [34]. A *georeferenced class* describes a set of objects that have spatial representation and are associated to features on Earth [9], assuming the fields and objects view as proposed in [22], [24]. A *conventional class* describes a set of objects with similar properties, behavior, relationships, and semantics, and which can have some sort of relationship with spatial objects, but which do not have geometric or geographic properties.

Georeferenced classes are specialized into *geo-field* and *geo-object* classes. Geo-field classes represent objects and phenomena that are continuously distributed over the space, corresponding to variables such as soil type, relief, and mineral contents [9]. Geo-object classes represent individual, particular geographic objects, which can be traced back to real world elements, such as buildings, rivers, and trees. A georeferenced class is symbolized by a rectangle, subdivided in three parts (figure 2a). The top left-hand rectangle is used to indicate the geometry of the representation. The notation used for conventional classes corresponds to the notation used in the UML [37]. A simplified symbolization can be used in both cases (figure 2b). Objects may or may not have non-spatial attributes, listed in the middle section of the complete representation. Associated methods or operations are specified in the lower section.

OMT-G presents a fixed set of geometric types, using a symbolic representation that distinguishes geo-object and geo-field classes within a georeferenced class (figures 3 and 4). Adding pictograms to the primitive element used to portray geographic classes (instead

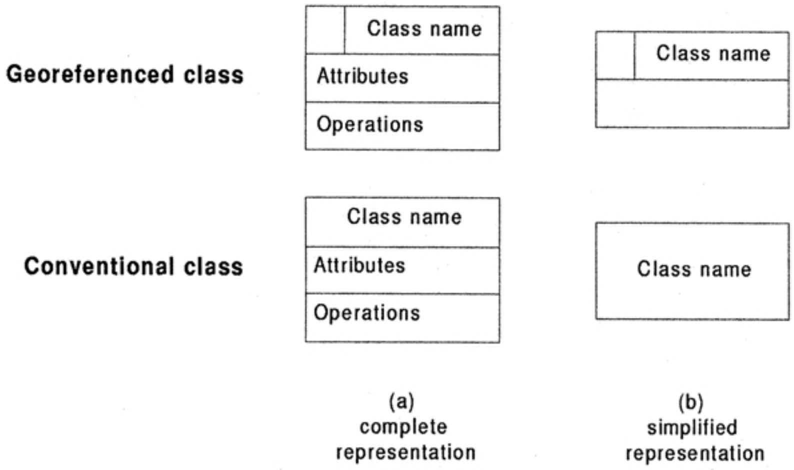


Figure 2. Graphic notation for the basic classes.

of using relationships to describe the geometry of the object) significantly simplifies the final diagram [29].

OMT-G has five geo-field descendant classes: *isoline*, *planar subdivision*, *tesselation*, *sampling*, and *triangular irregular network* (figure 3), and two geo-object descendant classes: *geo-object with geometry* and *geo-object with geometry and topology* (figure 4). From these specializations, and from the creation of a spatial aggregation primitive (''whole-part'' primitive), as well as from standardized spatial relationships, some spatial integrity rules can be deduced. These rules constitute a set of constraints that must be observed in the operations that update the geographic database. The GIS can include features that enforce the fulfillment of some spatial integrity rules. However, most of them require the definition of integrity control operations, to be associated with the classes, but such operations must be implemented by the application's developer. Controlling the integrity constraints must be considered one of the main implementation activities. It is convenient to have the geographic application schema to reinforce at least the situations where this control cannot be disregarded. Many mistakes in the data entry process can be avoided if digitizing procedures based on these constrains are implemented.

A *geo-object with geometry* class represents objects which have only geometric properties (points, lines, and polygons), and is specialized precisely in classes named *Point*, *Line*, and *Polygon*. Examples include, respectively, bus stop, curb line, and

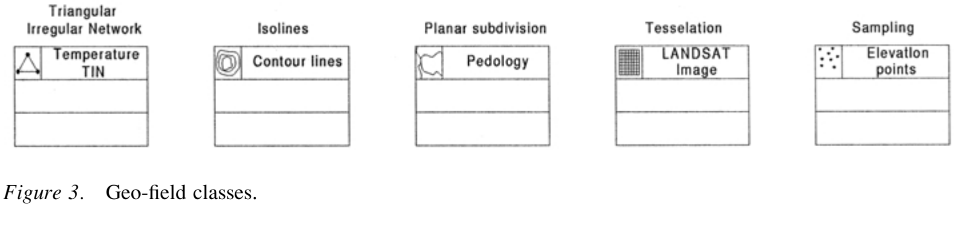


Figure 3. Geo-field classes.

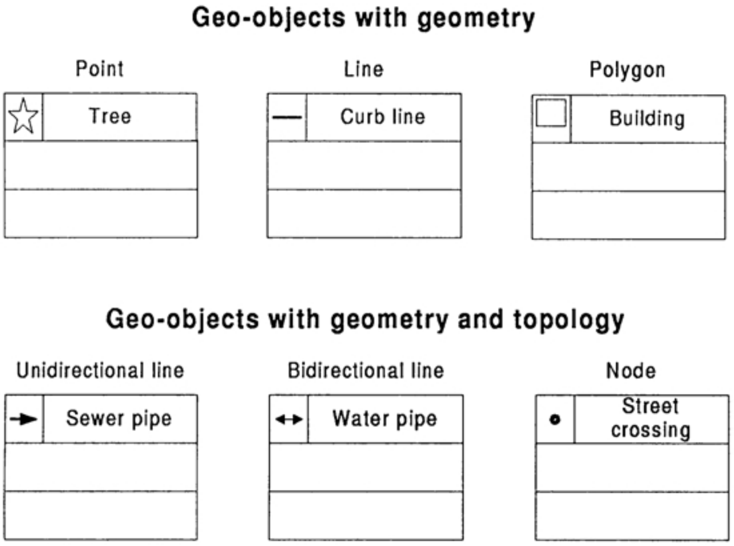


Figure 4. Geo-object classes.

municipal limits. A *geo-object with geometry and topology* represents objects which have, in addition to geometric properties, topological connectivity properties, and are specifically suited to the representation of spatial network structures, such as water supply systems, electrical distribution systems, or road networks. These properties are present in objects that are either nodes or arcs, in a graph-theoretic approach. Unidirectional lines indicate that the network has a definite flow direction, such as in sewage systems. Bidirectional lines indicate that there is a flow and a connection. The direction of the flow, in this case, is deemed irrelevant, since it can occur in any direction, as in water or electrical networks. The focus here is not on the implementation of the relationship, but rather on the semantics of the connection among network elements, which is a relevant element for spatial integrity assurance procedures. The implementation will depend on specific characteristics of the underlying GIS. This class specializes into subclasses *Node*, *Unidirectional Line*, and *Bidirectional Line*.

From the usage of geo-field primitives, the spatial integrity rules listed in table 1 can be derived.

2.2.2. ***Relationships.*** An existing problem in most data models is that the possibility of modeling the relationships between real world phenomena is often neglected [34]. Considering the importance of spatial and non-spatial relations in the understanding of the modeled space, OMT-G represents the three types of relationship that can occur between its classes: simple associations, topological network relations, and spatial relations. The discrimination of such relations has the objective of defining explicitly the type of interaction that occurs between classes. There are some applications that do not make use of spatial relations, but nevertheless there are applications on which spatial relations have

Table 1. Geo-field integrity rules.

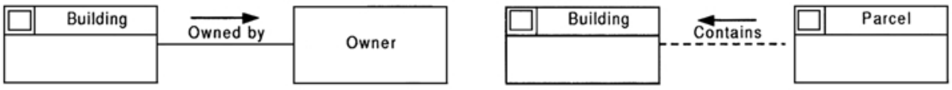
Planar Enforcement rule	1. Let F be a geo-field and let P be a point such that $P \subset F$. Then a value $V(P) = f(P, F)$, i.e., the value of F at P, can be univocally determined.
Isoline	2. Let F be a geo-field. Let $\nu_0, \nu_1, \ldots, \nu_n$ be $n + 1$ points in the plane. Let $a_0 = \overline{\nu_0\nu_1}, a_1 = \overline{\nu_1\nu_2}, \ldots, a_{n-1} = \overline{\nu_{n-1}\nu_n}$ be n segments, connecting the points. These segments form an *isoline* L if, and only if, (1) the intersection of adjacent segments in L is only the extreme point shared by the segments (i.e., $a_i \cap a_{i+1} = \nu_{i+1}$), (2) non-adjacent segments do not intercept (that is, $a_i \cap a_j =$ for all i, j such that $j \neq i + 1$), and (3) the value of F at every point P such that $P \in a_i, 0 \leq i \leq n - 1$, is constant.
Tesselation	3. Let F be a geo-field. Let $C = \{c_0, c_1, c_2, \ldots, c_n\}$ be a set of regularly-shaped cells covering F. C is a *tesselation* of F if and only if for any point $P \subset F$, there is exactly one corresponding cell $c_i \in C$ and, for each cell c_i, the value of F is given.
Planar subdivision	4. Let F be a geo-field. Let $A = \{A_0, A_1, A_2, \ldots, A_n\}$ be a set of polygons such that $A_i \subset F$ for all i such that $0 \leq i \leq n - 1$. A forms a planar subdivision representing F if and only if for any point $P \subset F$, there is exactly one corresponding polygon $A_i \in A$, for which a value of F is given.
Triangular Irregular network	5. Let F be a geo-field. Let $T = \{T_0, T_1, T_2, \ldots, T_n\}$ be a set of triangles such that $T_i \subset F$ for all i such that $0 \leq i \leq n - 1$. T forms a *triangular irregular network* representing F if and only if for any point $P \subset F$, there is exactly one corresponding triangle $T_i \in T$, and the value of F is known at all vertices of T_i.

a very relevant meaning, and therefore should be explicitly included in the application schema.

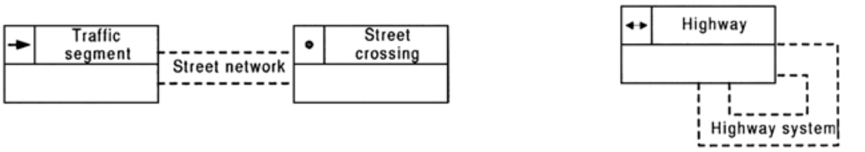
2.2.2.1. Simple associations, spatial relations, and network relations. *Simple associations* represent structural relationships between objects of different classes, conventional as well as georeferenced. *Spatial relations* represent the topologic, metric, ordinal, and fuzzy relationships. Some relations can be derived automatically, from the geometry of each object, during the execution of data entry or spatial analysis operations. Topologic relations are an example of this. Others need to be specified by the user, in order to allow the system to store and maintain that information. The latter are called *explicit relations* [36].

In OMT-G, simple associations are indicated by continuous lines, whereas spatial relations are indicated by dashed lines (figure 5). Therefore, it is simple to distinguish between simple associations (alphanumeric relationships) and spatial relations.

Based on previous works [9], [12], [17], [18], [35], OMT-G considers a set of nine different spatial relations between georeferenced classes. In [12] a minimum set of spatial relation operators is identified, comprising only five spatial relations, from which all others can be specified: *touch*, *in*, *cross*, *overlap*, and *disjoint*. However, we consider that sometimes a larger set is required due to cultural or semantic concepts that are familiar to the users. These include relations such as *adjacent to*, *coincide*, *contain*, and *near*, which are in fact special cases of one of the five basic relations, but deserve special treatment because of their common use in practice. Spatial integrity constraints for these relations are listed in table 2, but additional constraints can be formulated in case some additional



(a) Simple association (b) Spatial relationship



(c) Arc-node network relationship (d) Arc-arc network relationship

Figure 5. Relationships.

Table 2. Spatial relationship integrity rules.

	Basic Relations
Touch	1. Let A, B be two geo-objects, where neither A nor B are members of the **Point** class. Then $(A \textit{ touch } B) = \text{TRUE} \Leftrightarrow (A^o \cap B^o = 0) \wedge (A \cap B \neq \varnothing)$.
In	2. Let A, B be two geo-objects. Then $(A \textit{ in } B) = \text{TRUE} \Leftrightarrow (A \cap B = A) \wedge (A^o \cap B^o \neq \varnothing)$.
Cross	3. Let A be a geo-object of the **Line** class, and let B be a geo-object of either the **Line** or the **Polygon** class. Then $(A \textit{ cross } B) = \text{TRUE} \Leftrightarrow$ $\dim(A^o \cap B^o) = ((\max(\dim(A^o), \dim(B^o)) - 1) \wedge (A \cap B \neq A) \wedge (A \cap B \neq B)$.
Overlap	4. Let A, B be two geo-objects, both members of the **Line** or of the **Polygon** class. Then $(A \textit{ overlap } B) = \text{TRUE} \Leftrightarrow$ $\dim(A^o) = \dim(B^o) = \dim(A^o \cap B^o) \wedge (A \cap B \neq A) \wedge (A \cap B \neq B)$.
Disjoint	5. Let A, B be two geo-objects. Then $(A \textit{ disjoint } B) = \text{TRUE} \Leftrightarrow A \cap B = \varnothing$
	Special Cases
Adjacent to	6. Let A be a geo-object of the **Polygon** class and let B be a geo-object of either the **Line** or the **Polygon** class. Then $(A \textit{ adjacent to } B) = \text{TRUE} \Leftrightarrow (A \textit{ touch } B) \wedge \dim(A \cap B) = 1$.
Coincide	7. Let A, B be two geo-objects. Then $(A \textit{ coincide } B) = \text{TRUE} \Leftrightarrow A \cap B = A = B$.
Contain	8. Let A, B be two geo-objects, where A is a member of the **Polygon** class. Then $(A \textit{ contain } B) = \text{TRUE} \Leftrightarrow ((B \textit{ in } A) = \text{TRUE}) \wedge ((A \textit{ coincide } B) = \text{FALSE})$.
Near(*dist*)	9. Let A, B be two geo-objects. Let C be a buffer, created at a distance *dist* around A. Then $(A \textit{ near}(\textit{dist}) B) = \text{TRUE} \Leftrightarrow (B \textit{ disjoint } C) = \text{FALSE}$

relation is required by the application. These include any kind of directional or relative spatial relations, such as *north of*, *left of*, *in front of*, or *above*.

Some relationships are only allowed between specific classes, because they depend on the geometric representation. For instance, the existence of a *contain* relationship assumes that one of the classes involved is a polygon. In this aspect, the traditional applications differ from geographic ones, where associations between conventional classes can be freely built, being independent from factors such as geometric behavior. The set of concepts the user has about each real world object strongly suggests a particular representation, because there is an interdependence between the representation, the type of interpretation, and the usage given to each object class. In OMT-G this is considered in order to allow the placement of relations involving georeferenced classes.

Considering the previously listed spatial relationship types, some spatial integrity rules can be established (table 2). These rules are formulated using a notation commonly found in computational geometry, in which objects are indicated by upper-case italic letters (e.g. A,B), their boundaries are denoted as ∂A, and their interiors as A^o (note that $A^o = A - \partial A$). The boundary of a point object is considered to be always empty (therefore the point is equivalent to its interior), and the boundary of a line is comprised of its two endpoints. A function, called *dim*, is used to return the dimension of an object, and returns 0 if the object is a point, 1 if it is a line, or 2 if it is a polygon.

The *disjoint* rule is very important to maintain the integrity of the data stored in the database, and it must be used in order to check input data. For instance, if the classes `Street Segment` and `Building` are disjoint, it means that there can never be any street segment overlapping a building. If it becomes necessary to draw a street segment over a building, the building must first be deleted. The street segment and building creation routines can enforce this rule.

The *near* rule is the only one described in table 2 that requires a parameter. Since the notion of proximity varies according to the situation, a precise distance must be supplied in order to allow for the correct evaluation of the relationship. As an example, consider the classes `Address` and `Bus Stop`. In order to establish the relationship between instances of these classes, the maximum distance at which the bus stop is still considered to be near some address must be defined, for instance 500 meters.

In OMT-G, *network relations* are relationships among objects that are connected with each other. As previously mentioned, a network relationship only shows the need for a logical connection, not a requirement for the implementation of a particular structure. Network relations are indicated by two parallel dashed lines, linking a node class to an arc class. Network structures can be built without nodes, requiring a recursive relationship on the class which represents graph segments. The name given to the network is annotated between the two dashed lines (figure 5c). The *connectivity rules*, which apply to network relationship primitives, are listed in table 3.

As an example of the usage of these rules, consider a sewage network which is an arc-node logical structure. Nodes are used to represent network elements such as manhole, sewage treatment station, and discharge, and arcs are used to symbolize piping segments. The system is required to ensure the connection between all types of nodes and segments. Network relations can be maintained by the GIS using special data structures, and are

Table 3. Connectivity rules.

Arc-node structure	Let $G = \{N, A\}$ be a network structure, composed of a set of nodes $N = \{n_0, n_1, \ldots, n_p\}$ and a set of arcs $A = \{a_0, a_1, \ldots, a_q\}$. Members of N and members of A are related according to the following constraints: 1. For every node $n_i \in N$ there must be at least one arc $a_k \in A$. 2. For every arc $a_k \in A$ there must be exactly two nodes $n_i, n_j \in N$.
Arc-arc structure	Let $G = \{A\}$ be a network structure, composed of a set of arcs $A = \{a_0, a_1, \ldots, a_q\}$. Then the following constraint applies: 1. Every arc $a_k \in A$ must be related to at least one other arc $a_i \in A$, where $k \neq i$.

represented by connecting arcs and nodes. Connectivity rules are usually enforced by the GIS itself.

2.2.2.2. Cardinality. Relationships are characterized by their cardinality. The notation for cardinality adopted by OMT-G (figure 6) is the same used by UML [37].

2.2.3. Generalization and specialization. *Generalization* is the process of defining classes that are more general (superclasses) than classes with similar characteristics (subclasses) [19], [27]. *Specialization* is the inverse process, in which more specific classes are detailed from generic ones, adding new properties in the form of attributes. Each subclass inherits attributes, operations, and associations from the superclass.

In the OMT-G model, the generalization and specialization abstractions apply both to georeferenced classes and conventional classes, following the definitions and notation proposed for UML, where a triangle connects a superclass to its subclasses (figure 7a, b). Each generalization can have an associated *discriminator*, indicating which property is being abstracted by the generalization relationship.

Generalizations (spatial or not) can be specified as *total* or *partial* [27], [37]. A generalization is total when the union of all instances of the subclasses is equivalent to the complete set of instances of the superclass. UML represents the totality constraint by using the predefined constraint elements *complete* and *incomplete*, but in OMT-G we have

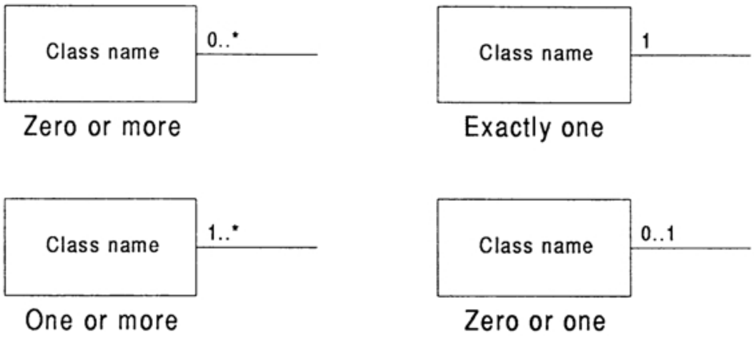


Figure 6. Cardinality.

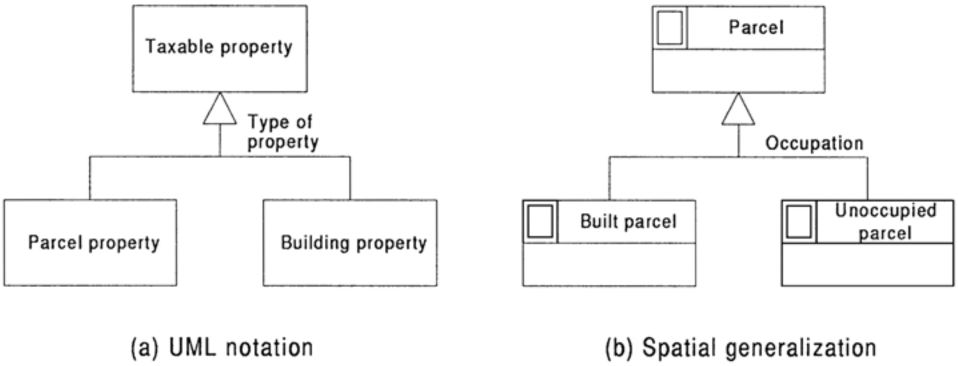


Figure 7. Generalization.

adopted the notation presented in [27], in which a dot is placed in the upper vertex of the triangle that denotes generalization (figure 8). Additionally, OMT-G also adopts the original OMT notation [39] for the UML predefined constraint elements *disjoint* and *overlapping*, that is, in a disjoint relation the triangle is left blank and in a overlapping relation the triangle is filled. Therefore, the combination of the disjunction and totality

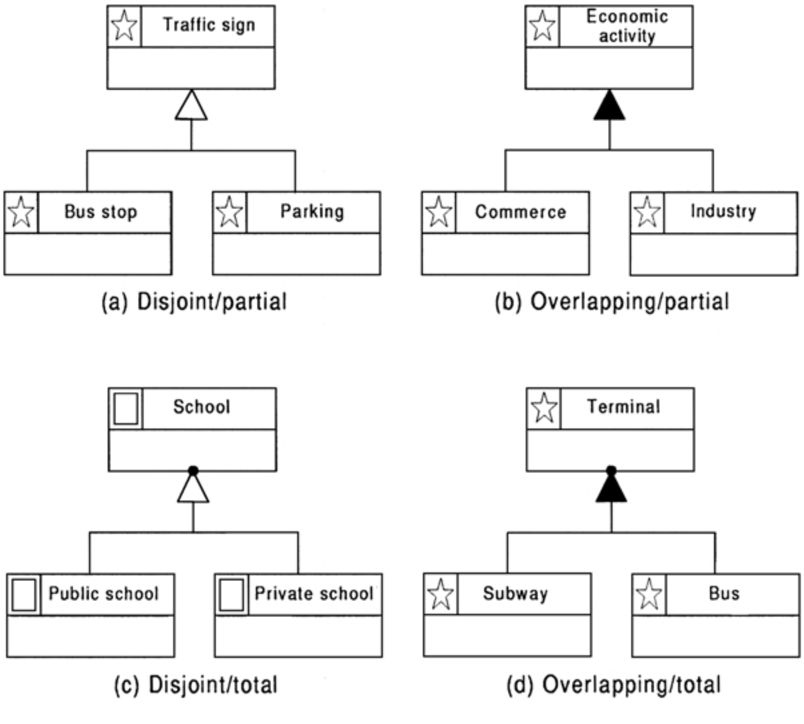


Figure 8. Spatial generalization examples.

aspects of generalization generates four types of constraints that apply to generalization/specialization. Figure 8 shows examples of each combination.

2.2.4. Aggregation. Aggregation is a special form of association between objects, where one of them is considered to be assembled from others. The graphic notation used in OMT-G follows the one used by UML (figure 9). An aggregation can occur between conventional classes, between georeferenced classes, and also between georeferenced and conventional classes (figure 10). When the aggregation is between georeferenced classes, *spatial aggregation* must be used.

Spatial aggregation is a special case of aggregation in which topological ''whole-part'' relationships are made explicit [2], [26]. The usage of this kind of aggregation imposes spatial integrity constraints regarding the existence of the aggregated object and the corresponding sub-objects. Beyond providing more clarity and expressiveness to the model, the observation of these rules contributes to the maintenance of the semantic integrity of the geographic database. In spatial aggregation, also called topological ''whole-part'', the geometry of each part is entirely contained within the geometry of the whole. Also, no overlapping among the parts is allowed and the geometry of the whole is fully covered by the geometry of the parts. The notation for this structure is presented in figure 11, where it is specified that blocks are composed of parcels, that is, blocks are geometrically equivalent to the union of adjacent parcels. This implies that (1) no area belonging to the block can exist outside of a parcel, (2) no overlapping can occur among parcels that belong to a block, and (3) no area belonging to a parcel can exist outside of a block. These three principles are stated in table 4 and correspond to the spatial integrity constraints associated with the spatial aggregation primitive.

Notice that the class diagram does not specify whether the whole can be assembled from individual parts in an automatic fashion, nor does it specify whether the parts can be obtained automatically from the whole. If such automatic generation of instances can be specified, then it is done in the transformation diagram (see Section 2.3), by specifying exactly which transformation operation should be used. This transformation must ensure the application of the integrity constraints for spatial aggregation, as stated in Table 4.

2.2.5. Conceptual generalization. Generalization, in the cartographic sense[1], can be seen as a series of transformations that are performed over the representation of spatial information, geared towards improving readability and understanding of data. For instance, a real world object can have several different spatial representations, according to the current viewing scale. A city can be represented in a small-scale map as a point, and as a polygon in a large-scale map. In this sense, this paper uses the term *representation* in the sense of a *coding of the geometry* of geographic objects (involving aspects such as

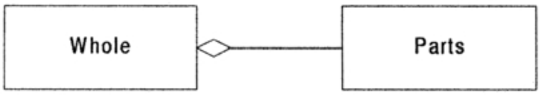


Figure 9. UML aggregation.

Figure 10. Aggregation between conventional and georeferenced classes.

resolution, spatial dimension, precision, level of detail, and geometric/topologic behavior) [15], [16].

Defining how simple or elaborate a representation needs to be is dependent on how the user perceives the real world object, and how this representation affects the spatial relationships it can establish with other modeled objects. Considering the need for such relationships, there can be demand for more than one representation for a given object. This is often the case when geographic information needs to be shared among various applications in an enterprise-wide database.

Therefore, in the development of geographic applications, there are situations in which two or more representations for a real-world object need to coexist. This means that, depending on the user's view, it is necessary to have distinct geometric shapes to represent the same geographic object, at the same scale and at the same time. Additionally, there is often the need to represent the same object with varying levels of resolution or detail, configuring adequate representations for various ranges of scales.

In opposition to the concept of representation, this paper uses the term *presentation* in the sense of *visualization* or *graphical aspect* (involving parameters such as color, line type, line thickness, and fill pattern) of the geo-objects and geo-fields on paper or on the computer's screen [15], [16]. Notice that the discussion that follows does not involve presentation aspects of the application, such as decisions regarding line types, colors, and other visual parameters. These decisions are made at the presentation level (see Sections 1.1 and 2.4). The focus of the conceptual representation level is on the representation of the geometric shape of the real world objects, considering that, in order to correctly specify spatial relationships, the geometric aspect must be clearly defined beforehand.

The spatial primitive *conceptual generalization* is used to record different user views. Since it can be perceived in various ways, the superclass does not have a specific representation. However, its subclasses are represented by distinct geometric shapes, being allowed to inherit the superclass' alphanumeric attributes and to include specific attributes of their own. The objective of the use of this primitive is to allow relationships involving each representation style to be made explicit. As previously shown, the way a class is represented influences the spatial relationship types that can occur. The same representation alternative is allowed in more than one subclass, because in each one the level of detail or resolution can vary.

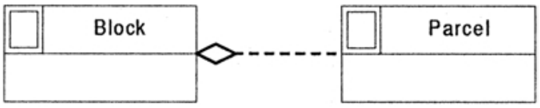


Figure 11. Spatial aggregation (''whole-part'').

Table 4. Spatial aggregation integrity rules.

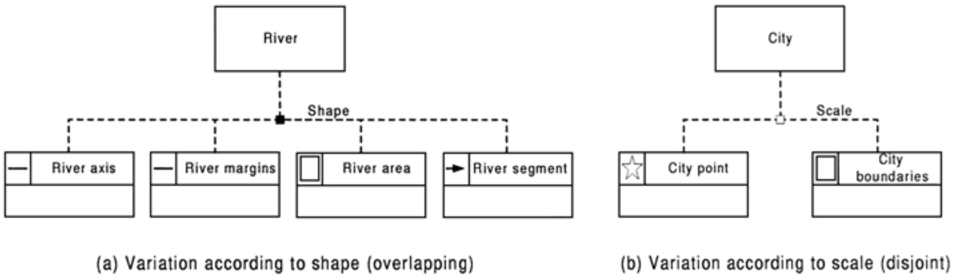
Spatial aggregation	Let $P = \{P_0, P_1, \ldots, P_n\}$ be a set of geo-objects. Then P forms another object W by spatial aggregation if and only if 1. $P_i \cap W = P_i$ for all i such that $0 \le i \le n$, and 2. $\left(W \cap \bigcup_{i=0}^{n} P_i\right) = W$, and 3. $((P_i\, touch\, P_j) \vee (P_i\, disjoint\, P_j)) =$ TRUE for all i, j such that $i \neq j$.

Conceptual generalization can occur in two representation variations: *according to geometric shape* and *according to scale*. The variation according to geometric shape is used to record the simultaneous existence of multiple scale-independent representations for a class. For instance, a river can be represented by its axis, as a single line, as the space between its margins, as a polygon covered by water, or as a set of flows (directed arcs) within river sections, forming a hydrographic network (figure 12a). Variation according to scale is used in the representation of different geometric aspects of a given class, each corresponding to a range of scales. A city can be represented by its political borders (a polygon) in a larger scale, and by a symbol (a point) in a smaller scale (figure 12b).

The notation used for conceptual generalization uses a square to connect the superclass to its subclasses. The subclass is connected to the square by a dashed line. As a discriminator, the word *Scale* is used to mean variation according to scale, and the word *Shape* is used to determine variation according to geometric shape. The square is blank when subclasses are disjoint and filled if subclass overlapping is allowed (figure 12).

The variation according to geometric shape can also be used in the representation of classes which simultaneously have georeferenced and conventional instances. For instance, a traffic sign can exist in the database as a non-georeferenced object, such as a warehouse item, but it becomes georeferenced when installed at a particular location (figure 13)

In many situations, it is possible to derive one or more representations from a primary one. This can be achieved using transformation operations, based on the use of geometric, spatial analysis, and map generalization operators [15], [16]. These operations can be specified in the OMT-G model also at the conceptual representation level, using transformation diagrams, covered in Section 2.3.



(a) Variation according to shape (overlapping)

(b) Variation according to scale (disjoint)

Figure 12. Cartographic generalization.

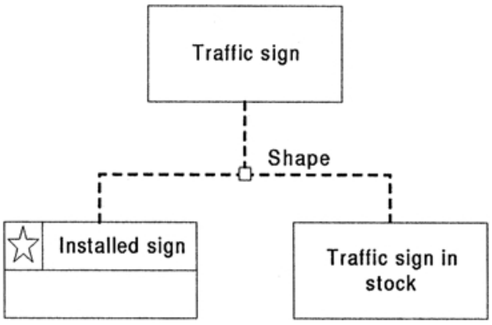


Figure 13. Cartographic generalization with a conventional class.

Conceptual generalization must not be mistaken with operations involving simply the variation of a symbol's size or line thickness. Whenever there is no change in the representation alternative nor in the level of detail, the OMT-G model sees the operation as a change in the visualization parameters, and therefore it should be specified at the presentation level.

2.3. Transformation diagram

The transformation diagram proposed for OMT-G follows the UML notation for the state and activity diagrams and is used to specify transformations between classes. Even though it is used to specify transformation operations, the transformation diagram still operates at the conceptual representation level. This is because both the source and the results of the transformation are representations, considering the concept of representation presented in Section 2.2.5.

Transformation diagrams are based on the class primitives, as defined for the class diagrams. Classes involved in some kind of transformation are connected with continuous lines, with arrows indicating the source and the result of the transformation. The transformation operators involved, and their parameters, are indicated as text over the connecting lines.

In the transformation diagram, it can be indicated whether or not the result of the transformation is to be materialized. Very simple results, or results that are intermediate steps in a more complex transformation often do not need to be materialized, and can remain stored only temporarily. Such temporary classes are depicted using dashed lines. Classes that are the result of some transformation, and that need to be materialized (due to the complexity of the transformation process or due to specific needs of the application) are indicated with continuous lines, exactly like in the class diagram.

The transformations specified in the transformation diagram can relate any number of source classes, and any number of resulting classes, depending only on the nature of the transformation operation. Chains of transformations can also be used, therefore allowing the specification of complex processes of spatial analysis. Figure 14 presents an example of such an operation. In it, the `Relief` class, represented by a TIN, is used along with the `Street crossing` class, to produce another class, `Crossing level`, represented as a set of samples. In the resulting class, the geometry is obtained from the `Street crossing` class, and the value of the geo-field at each crossing point location is calculated by interpolation on the TIN geo-field (`Interpolate` method), thus filling out the `Level` attribute for the samples.

A transformation operator adequate for the transformation diagram can basically be any algorithm that manipulates and modifies existing data on the representation of an object. This is often necessary in the execution of complex spatial analysis procedures, in which a given class or set of classes need to be transformed so that they can be more easily compared. Figure 15 shows an example of a set of operations required to perform an analysis about the risk of erosion at a certain region, given information on soils, vegetation, and relief. Initially, the vegetation (represented by a planar subdivision) and the digital elevation model (represented by a TIN) are transformed into tesselations with an appropriate resolution, using a rasterization procedure in the first case and an interpolation procedure in the second case. As a result, both geo-fields become compatible with the soils tesselation. Then, the analysis operation can be carried out cell by cell, determining how high is the erosion risk at any of them, and producing a new tesselation with the results.

Some operations can be better characterized as *transformation to representation* (TR) operations when there is only one source and only one resulting classes, and the resulting class is either (1) of a different nature than the source class (i.e., it must belong to a different georeferenced class), or (2) less detailed than the source class, while maintaining its representation nature [15], [16]. The specification of these transformations is usually required in the transformation diagram if either the cartographic generalization or the aggregation primitives are used in the class diagram.

A study on the possible variety of operators for TR, considering all possible combinations of the nature of source and result classes, has been presented in previous

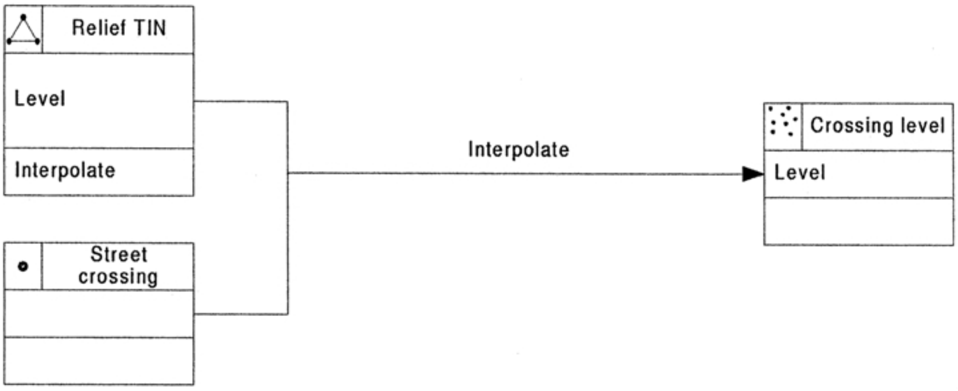


Figure 14. Operation example.

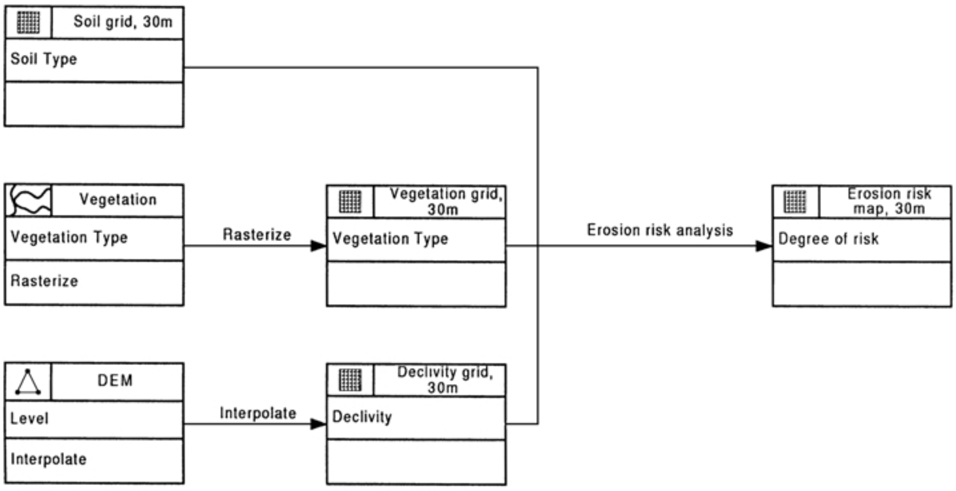


Figure 15. Spatial analysis example.

works [15], [16] as part of a framework on multiple representations. The operators employed for such transformations are based on well-established algorithms, defined in the fields of computational geometry, map generalization, and spatial analysis, such as the ones listed in table 5. Observe that this listing is not exhaustive.

When the class diagram contains an aggregation primitive, there is often the need to specify the operation that will actually build the whole from the parts. Figure 16 shows an example of such an operation, corresponding to the class diagram fragment presented in Figure 11. Instances of `Parcel` are to be assembled, using the cartographic generalization operator called `Amalgamation` [32], to create `Block` instances. This operation qualifies as a TR operation, since the resulting class (`Block`) is less detailed than the original one (`Parcel`), and the transformation involves just two classes. Sometimes, however, it is not possible to specify such an automatic transformation, specially in the case where the whole is obtained first, and the parts have to be cut out of it. Notice that the transformation also specifies how a certain `Block` attribute, `Block_value`, is to be filled out with the sum of individual `Parcel` values.

As an example in multiple representations, consider the class diagram fragment in figure

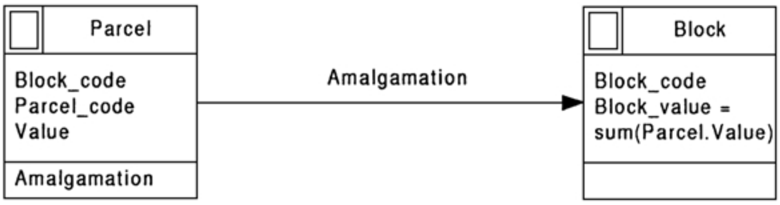


Figure 16. Aggregation in the transformation diagram.

Table 5. Transformation to representation operators.

Geometric Operators

Centroid determination: Select a point that is internal to a given polygon, usually its center of gravity.

Convex hull: Define the boundaries of the smallest convex polygon that contains a given point set.

Delaunay triangulation: Given a point set, define of a set of non-overlapping triangles in which the vertices are the points of the set.

Isoline generation: Build a set of lines and polygons that describe the intersection between a given 3-D surface and a horizontal plane.

Polygon triangulation: Divide a polygon into non-overlapping neighboring triangles.

Skeletonization: Build a 1-D version of a polygonal object, through an approximation of its medial axis.

Voronoi diagram: Given a set of sites (points), divide the plane in polygons so that each polygon is the locus of the points closer to one of the sites than to any other site.

Map Generalization Operators

Aggregation: Join point elements which are very close to each other, representing the result with the limits of the area occupied by the point set.

Amalgamation: Join nearly contiguous and similar areas, by eliminating borders between them.

Collapse: Reduce the dimension of the representation of an object, caused by its representation's size reduction. An area element (2-D) that becomes too small due, for instance, to scale reduction, would be represented as a line (1-D) or point (0-D).

Merging: Join two or more parallel lines that are too close to each other into a single line.

Refinement: Discard less significant elements, which are close to more important ones, in order to preserve the visual characteristics of the overall representation but with less information density. In the opposite sense, this operator is often named **Selection**.

Simplification: Reduce the number of vertices employed to represent the element, in order to produce an appearance that is similar to the original, though simpler.

Smoothing: Displace the vertices used in the representation, in order to eliminate small disturbances and to capture the main tendencies as to the graphical shape.

Spatial Analysis Operators

Buffer construction: Create a polygon that contains all points of the plane closer than a given distance to an object.

Classification: Separate objects in groups, according to a set of criteria.

Grid analysis: Manipulate information contained in tesselations (mostly in the form of digital images), including **vectorization** (extract points, lines and polygons from an image), **rasterization** (transform points, lines, and polygons into an image), **image classification** (group cells according to their value), **resampling** (change the dimensions of the image by means of interpolation on the original cells), and others.

Polygon overlay: Determine the intersection between two sets of polygons.

Selection: Retrieve objects from an object set, based on spatial or alphanumeric criteria.

Spatial interpolation: Determine the value of a geo-field at a given point, based on information from other points.

Surface analysis: Extract information from a three-dimensional surface model, such as declivity, flood plains, and drainage profiles.

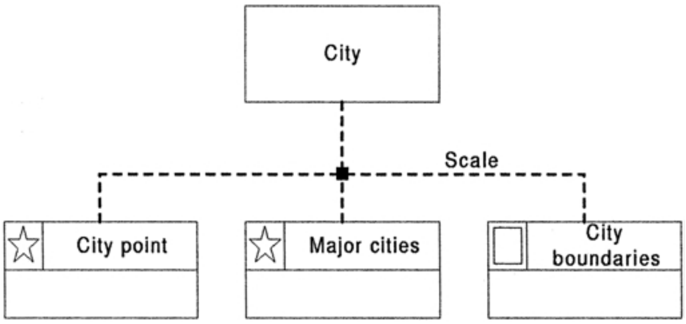


Figure 17. Alternative representations for the City class (class diagram).

17. While reading the class diagram, it does not become clear if and how the three alternative representations for the `City` class will coexist in the application. Even though some relationship among them is indicated by the cartographic generalization primitive, the class diagram does not explore the semantics that leads to the possibility of generating some representations from others. The transformation diagram (figure 18) solves this problem by recognizing and specifying that the `City Boundaries` class contains enough information to generate the geometry of both the `City Point` and the `Major Cities` classes. Both transformations are indeed TR transformations because the representation alternatives are different (polygon and point).

2.4. Presentation diagram

The presentation model for OMT-G assembles the requirements posed by the user in terms of output alternatives for each geographic object. These alternatives may include

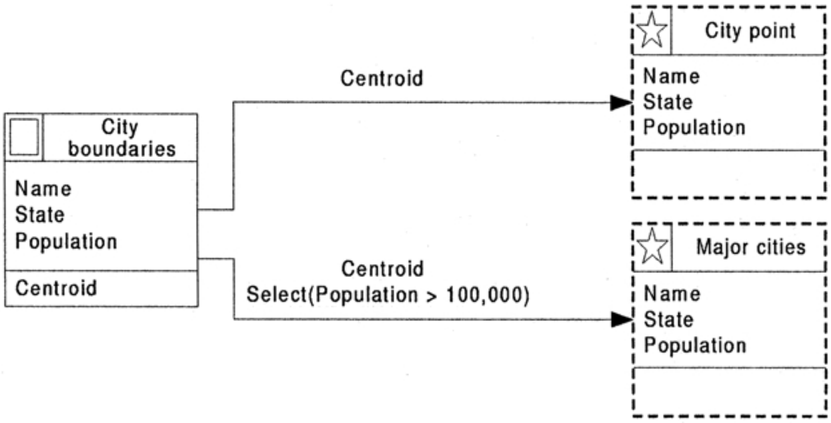


Figure 18. Transformations between representations of the City class (transformation diagram).

presentations defined for viewing on the screen, for printout as maps or charts, or both.

Presentations are defined starting from a representation that has been defined at the conceptual representation level. *Transformation to presentation* (TP) operations are then specified in order to achieve the visual aspect desired from the simple geometric shape defined for the representation. Observe that a TP operation does not modify the representation alternative that has been defined previously, nor does it change the level of detail defined at the conceptual representation level. If that is necessary, then a new representation must be created from an existing one, and this is done at the conceptual representation level, using the multiple representation tools (such as the cartographic generalization primitive) and the specification of the transformation operation that have been described for the class and the transformation diagrams.

For the presentation diagram, OMT-G provides three primitives. The first is the class primitive, which is the same one that has been defined for the class and transformation diagrams. The second is the TP operation, similar to the one used in the transformation diagram. It is denoted by a simple dashed line, with an arrow indicating the direction of the transformation, over which an expression, based on some convenient operator, is specified. In the process of defining this transformation expression, any geometric characteristics or alphanumeric attributes that have been defined at the conceptual representation level for the object can be used as parameters. The TP operation primitives are drawn with dashed lines in order to establish a clear visual distinction from the TR operations, drawn in the transformation diagram with continuous lines. The third is a presentation specification primitive, which is denoted by a box divided into two sections. The top section indicates the name of the class, the name of the presentation, and the application in which it is used. The second is divided in two: to the left there is a pictogram indicating the visual aspect the objects will have after the transformation, and to the right there are more precise specifications of the graphic attributes, including information on color, line type, line thickness, fill pattern, fill color, background color, and symbol name. There can be any number of pictograms on the left part of the second section of the presentation specification primitive, each associated with a value or a range of values. In this case, the right section must detail all symbols that will be used. Graphic attributes that are common to all values or ranges can be specified only once, while variable graphic attributes are specified as lists of individual values. As in the case of the transformation diagram, the results of the transformations (i.e., the presentations) are indicated with dashed lines if they are not to be materialized in the database, and with continuous lines otherwise.

Every georeferenced class specified in the class diagram needs to have at least one corresponding presentation indicated in the presentation diagram. In case there is more than one presentation for a given representation, one of them must be identified as the default. Alternatively, each user or user group can specify their own default presentation.

The most common TP operations [15], [16] involve the simple specification of graphic attributes. However, other more sophisticated operations can be employed, including typical spatial analysis operators such as *classification* (used to produce choropleth maps)

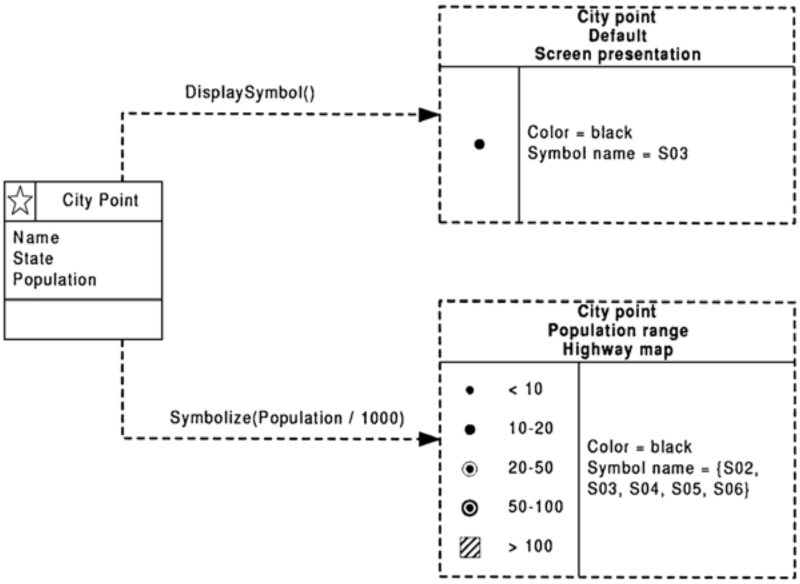


Figure 19. Presentation model for the `City Point` class.

Table 6. Transformation to presentation operators.

Output Operators
Graphic attributes specification: Allow the specification of visual parameters for screen or paper presentation, for each representation type, defining the required set of graphic characteristics: color, line type, line thickness, fill type, fill color, symbol type, pseudocoloring look-up table, and so on.
Map Generalization Operators
Exaggeration: Increase the dimensions of elements considered important for the map but, if represented in their real dimensions, would be too small to be perceived visually.
Enhancement: Modify the characteristics of a symbol, in order to make it more adequate to visualize in smaller scales.
Displacement: Intentionally shift the position of a feature, in order to make it distinct from other, which is too close or superimposed with it.
Spatial Analysis Operators
Classification: Group objects into categories which share identical or similar characteristics.
Symbolization: Adopt a visual appearance for an object based on its essential characteristics, specially after the results of a classification.

and *symbolization*, as well as map generalization operators such as *exaggeration*, *displacement*, and *enhancement* [32]. Table 6 presents a brief and non-exhausting listing of such operators.

Notice that the definition of presentations from a representation can be done even in the case of conventional objects, i.e., objects that do not have geometric or geographic features. This means that alternative visual aspects for tables and fields can also be specified at this level, simplifying the final implementation work. This feature is similar to user views.

As an example, consider the definition of a presentation for the `City Point` class presented earlier. First, a default screen presentation for the symbolic objects is defined. Then, a different presentation is specified, in which different symbols are assigned to city instances depending on their population attribute, to be used in a highway map (figure 19).

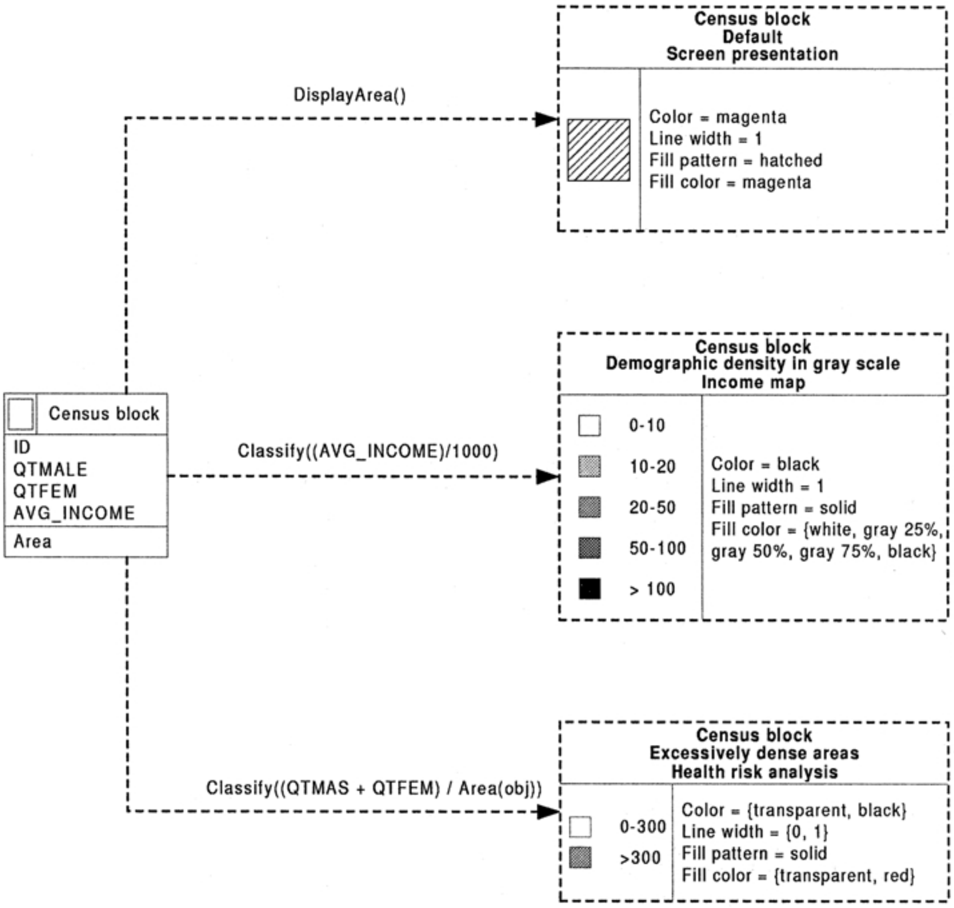


Figure 20. Presentation model for the `Census Block` class.

These are TP operations because the results, even though visually different from the basic `City Point` class, have the same geometric detail level as the original.

Figure 20 shows another presentation diagram example. The basic class, in this case, represents census blocks as polygons. Three different presentations are derived from it: the first is the default screen presentation, with a hatched pattern; the second shows the results of a classification over the average income attribute; the third displays only the census blocks in which there is a high demographic density, as part of a health risk analysis application. The demographic density is calculated using the attributes for number of males (`QTMALE`) and females (`QTFEM`), along with the area of the polygon. All three transformations qualify as TP operations, since neither changes the level of detail of the original census blocks, and neither changes the nature of the representation (polygon).

3. Discussion of an example

In order to illustrate the main features of OMT-G, a modeling example is presented in this section, corresponding to part of an urban cadastral database system, and privileging aspects related to property cadastre for taxing purposes. Its class diagram (figure 21) includes most of the primitives defined in OMT-G, and is described next.

The geographic space corresponds to a municipality, in practice Belo Horizonte's territory. The city can be represented as a point (`City point`) or as a polygon (`City boundary`), at its boundaries. These boundaries contain a number of blocks (`Block`), the fundamental cadastral unit used in the application. Blocks are in turn subdivided into parcels (`Parcel`). However, parcels can be represented either by a symbol (`Parcel symbol`) or by their area (`Parcel boundary`), in which case blocks can be obtained from the spatial aggregation of parcels. On the other hand, blocks also contain the symbolic parcels. Parcel symbols and parcel boundaries are two shape variations of the same geographic object, and therefore constitute a case of cartographic generalization. The polygonal type of parcels can be specialized in two other classes: `Built parcel` and `Unoccupied parcel`. In the first case, it is related to a territorial property record (`Parcel property`). In the second, the parcel boundaries must contain at least one building, also represented as a polygon. Each building is then related to a corresponding property record (`Building property`). A generalization of the property types is considered, as the `Taxable property` class. These conventional classes are managed by a conventional legacy system.

The `Address` class is defined as a point geo-object, and is represented by a symbol. Addresses are related to `Segment` instances, indicating to which street segment does any of them belong. Street segments are related to the `Crossing` class through a network relationship, indicating that these classes must form a graph-type structure in the GIS, thereby allowing for connectivity and minimum path analyses. Notice that the street network is represented by the topologic network relationship primitive, and therefore the cardinality and the spatial constraint *connect to* are implicit. This is an important feature of OMT-G, since it allows for a better representation of the application's semantics without overloading the schema. The `Thoroughfare` conventional class is modeled as an

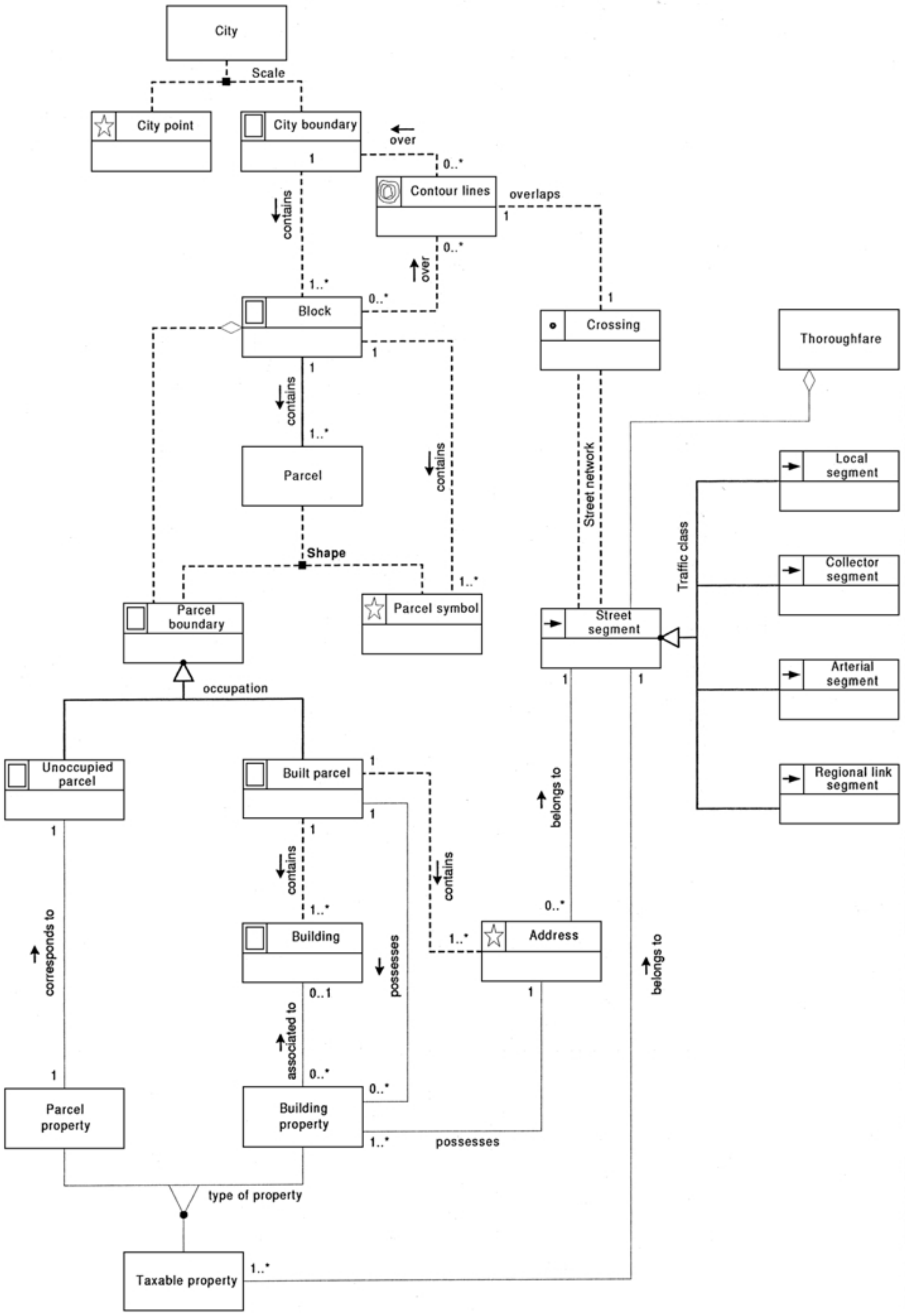


Figure 21. Partial static schema of the urban cadastral database.

aggregation of segments, and therefore it can only be visually perceived by selecting which segments belong to a given thoroughfare. Segments can be further specialized, according to their traffic class (`Local segment`, `Collector segment`, `Arterial segment`, and `Regional link segment`, in increasing order of traffic volume).

Relief is represented by Contour lines, a geo-field that covers the whole municipal territory. Observe that, since the geo-fields by definition cover the whole modeled space (the municipal territory), there is no need to explicitly relate any of them to other geo-object classes. The definition of geo-fields ensures that the value of the field at any given point can be determined, and this information can be used by any application and any other geo-object or geo-field.

Figure 22 shows the transformation diagram for the example in figure 21. According to the class diagram, there are three different situations in which a transformation can be specified. The first one is related to the cartographic generalization primitive, from which the first transformation can be specified. It is a TR operation, since there is a change in the representation's dimension (from polygon to point), using a centroid determination method. The resulting representation does not need to be materialized.

The second situation specifies the generation of a `Crossing levels` class from the interpolation of the `Contour lines` geo-field at the location of every `Crossing` instance. `Crossing levels` is a set of sample points, which happens to coincide with the `Crossing` nodes, therefore defining a set of level points at every street crossing.

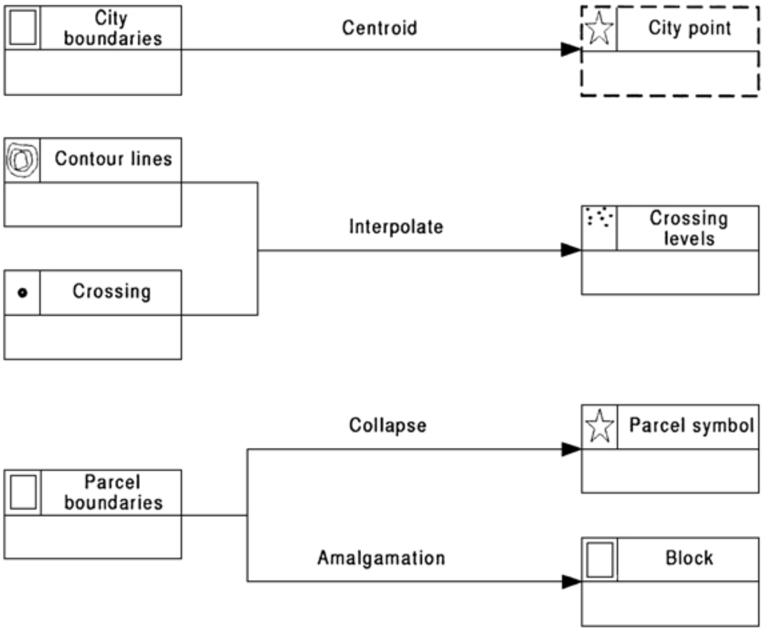


Figure 22. Transformation diagram for the example.

The third set of transformations is based on the `Parcel boundaries` class. First, it is possible to obtain parcel symbols from parcel boundaries, using a `Collapse` method. Contrary to the `City point` case, the `Parcel symbol` class will be materialized, mainly considering its importance for the application. Another transformation is specified from parcel boundaries, in which the geometry of the `Block` class is generated by amalgamation (union of adjacent polygons) and materialized.

Figure 23 shows part of the presentation diagram for the application example. The complete diagram would have to include at least one presentation definition for each georeferenced class included in the class diagram. In this fragment, two different presentations are defined for each city representation class (`City point` and `City boundary`), one of them being a default screen presentation. The other presentations correspond respectively to a symbolization and a classification of the city according to its population. In each case, graphic attributes are both illustrated on the left section and specified on the right section of the primitive.

A second group of presentations is included, specifying the graphic attributes for each of the street segment specializations. Each of them uses a different color, allowing for an easy visual distinction on the screen, and each of them uses a repeating arrow pattern along the line, from which the traffic flow direction can be perceived. The local segments constitute an exception, because since most of the local streets are actually rather narrow, the arrow pattern is dismissed.

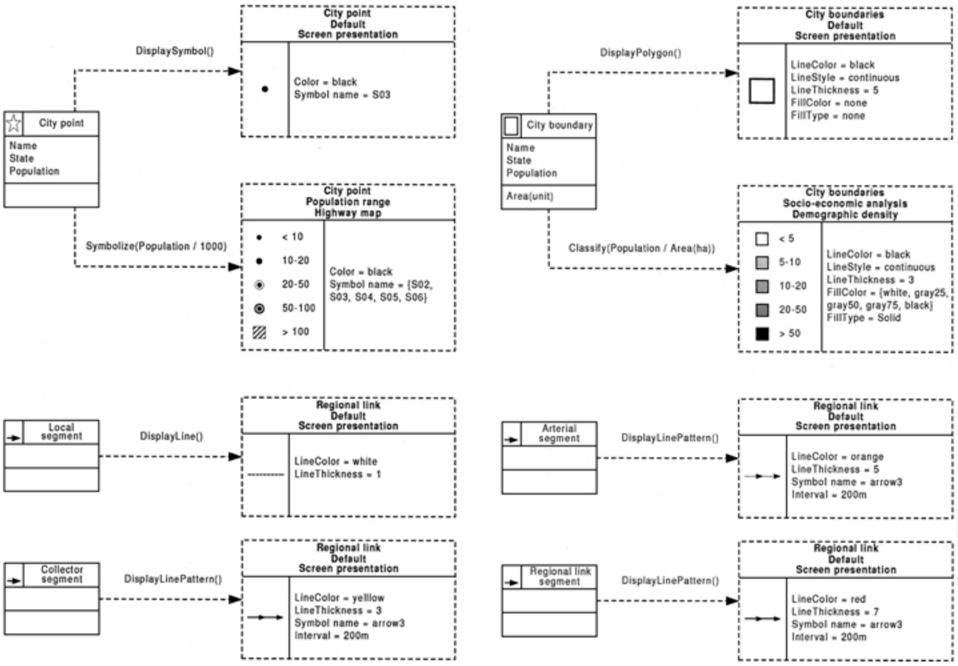


Figure 23. Presentation diagram fragment for the example.

4. Comparison with other data models

Several other data models have been proposed in the literature for modeling geographic applications [2], [3], [26], [34], [41], [44]. However, it is interesting to highlight that some of these models do not constitute a proper data model, since they do not define modeling-specific primitives. Instead, they only provide standards to be followed by geographic application modelers. The models GISER [41] and GMOD [34] fall within this category. Furthermore, the comparison will be developed on the basis of the class diagram and its primitives alone, since none of the aforementioned data models includes tools for the creation of transformation or presentation diagrams.

A summary of the comparison among all analyzed data models, including OMT-G, is shown in table 7.

As presented in table 7 and explained in the remainder of this section, OMT-G is the only model that:

- includes class, transformation, and presentation diagrams, allowing for the modeling of geographic applications;
- includes topological, semantic, and user-defined integrity constraints;
- includes primitives for the representation of multiple views;
- differentiates between spatial relationships and simple associations.

Table 7. Comparison among geographic data models.

	OMT-G	*EXT. IFO*	*OMT EXT.*	*GISER*	*GEO -OOA*	*GMOD*	*MODULO-R*
1. Represents the concept of field	✓			✓		✓	
2. Represents the concept of object	✓	✓	✓	✓	✓	✓	✓
3. Includes primitives to represent georeferenced and conventional classes	✓				✓		✓
4. Differentiates between spatial relationships and simple associations	✓						
5. Represents topological connectivity relationships	✓			✓	✓		
6. Includes spatial integrity rules	✓						
7. Supports aggregation relationships	✓	✓	✓	✓	✓	✓	✓
8. Includes primitives for spatial aggregation	✓		✓		✓		
9. Represents multiple views of the same entity	✓			✓		✓	✓
10. Includes primitives for the representation of multiple views	✓						
11. Models temporal aspects of geographic information				✓	✓	✓	✓
12. Includes primitives for dynamic modeling	✓						
13. Includes primitives for presentation modeling	✓						

Moreover, OMT-G diagrams tend to become smaller than the others, because of the higher semantic content of its primitives.

In EXT.IFO [44], the IFO model is adapted to geographic applications, representing the basic spatial object types: point, line, and polygon. However, it does not represent fields, nor does it represent spatial aggregations, multiple views or other fundamental geographic modeling constructs.

In [2], an extension to the OMT object model is proposed (OMT EXT), which includes primitives for modeling topological relationships, namely *partition*, *covering*, and *disconnected class*. The *partition* and *covering* primitives are similar to the ones presented in OMT-G. The concept of *disconnected class* varies between the OMT EXT and OMT-G. In OMT EXT [2], this concept is associated with the subclasses derived from *partitions* or *coverings*. In OMT-G, disconnected classes are represented through the disjoint spatial relationship, where the disjunction rule is associated with the classes, as a way to ensure the integrity of the non-relationship.

In [41], a model called *Geographic Information System Entity Relational Model* (GISER) is proposed. It is an extension of the ER model for geographic applications, using the ER model notation, according to extensions in [19]. The GISER data model integrates the field-based and object-based models of geographic data by using the *discretized-by* relationship between feature fields and coverage entities. This model differs from OMT-G mainly because it does not introduce specific primitives for modeling spatial applications. GISER has predefined entities and relationships which represent the fields and objects view (figures 24 and 25), the network relationships (figure 27), and the multiple visualization forms of an entity (figure 28). Since it does not have specific primitives, sometimes it turns out to be difficult to represent two simultaneous conditions for a given entity, as in the case of a river, for instance, which belongs to a network relationship and has more than one graphic form. The resulting diagram tends to get larger, making it harder to grasp. Figure 27 presents the modeling of a river network where it can be perceived that OMT-G provides a simpler and more elegant solution.

The GeoOOA model [26] is an extension to Coad/Yourdon's Object-Oriented Analysis (OOA) model [13], where primitives supporting the following abstractions were introduced: spatial class types, temporal class, topological ''whole-part'' structures (*coverage*, *containment*, and *partition*) (figure 26), and network structures (figure 27). GeoOOA distinguishes between object classes with or without spatial representation, and supports a fixed set of geometric types through the use of pictograms which distinguish, within the georeferenced class (geoclass), classes like point, line, polygon, and raster (figures 24 and 25). This model presents characteristics which are similar to OMT-G ones. However, it does not adequately represent fields (just raster images) and the multiple ways to visualize an object. It also lacks spatial integrity constraints. Moreover, in the representation of the network structure (figure 27) it can be perceived that OMT-G offers greater representation ease.

GMOD [34] is an object-oriented model proposed for the geographic applications development environment called UAPÉ (*geo-User Analysis and Project Environment*). It is an extension to the model described in [8], allowing for the definition of georeferenced phenomena according to both views, fields and objects, through predefined classes. It also

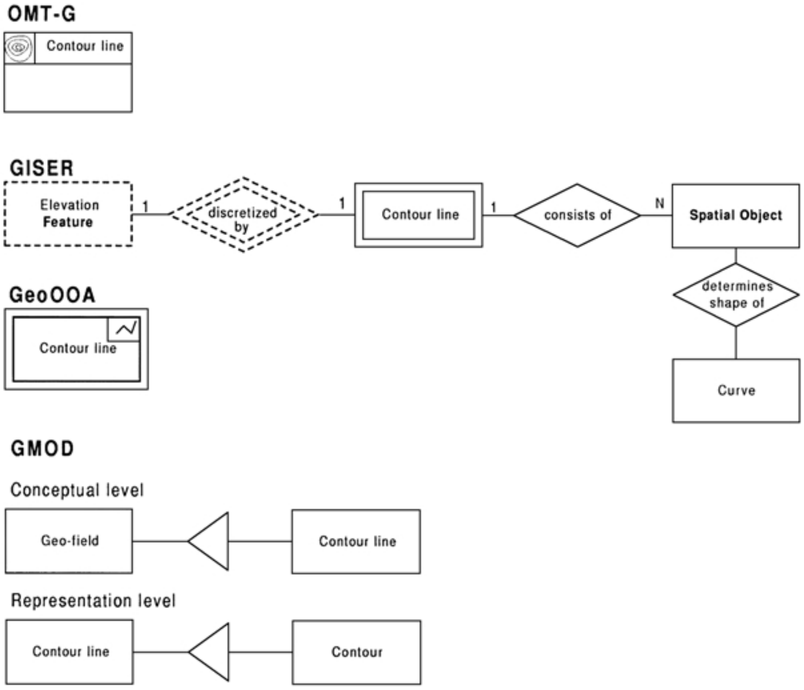


Figure 24. Geo-field schema (contour lines).

has predefined classes in order to model the geometry of spatial entities, as well as the temporal dimension, and introduces new relationships (*causal*, *version*) between entities (figures 24 and 25).

The MODUL-R model [3] is an extension to the model used by the MERISE method [42], targeted at urban geographic applications. It presents a fixed set of geometric types through the use of spatial pictograms that represent the geometric shapes of entities (figure 25). Besides, the combination of these pictograms represent multiple views of the same entity. The OMT-G model show similarities in the use of pictograms, when distinguishing between georeferenced and conventional classes and for the representation of multiple visualization forms for a geographic entity. However, the representation of multiple geometric natures by OMT-G allows relationships deriving from each nature to be made explicit, since they are represented by distinct subclasses in the model. MODUL-R also does not distinguish between fields and objects, and it does not include primitives to represent neither topologic connectivity nor spatial aggregation (figures 26 and 27). Moreover, only OMT-G supplies integrity constraints associated with the primitives and spatial relations.

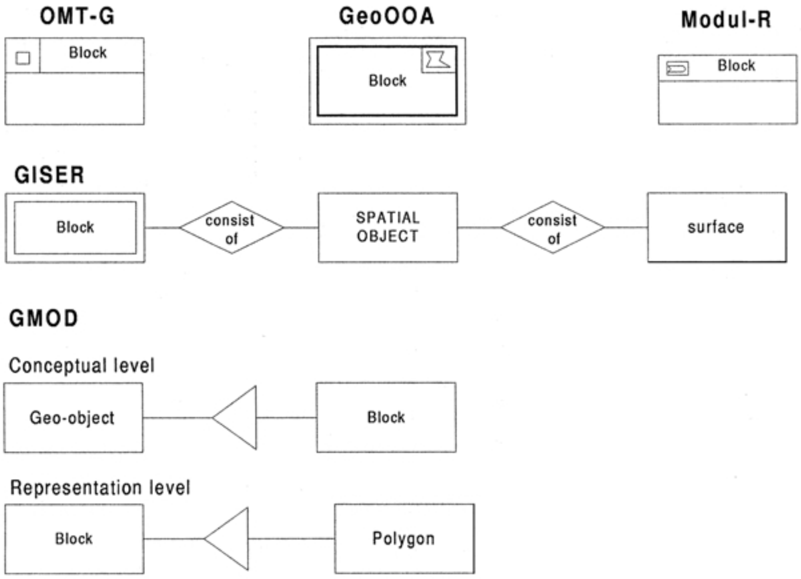


Figure 25. Geo-object schema (polygon-block).

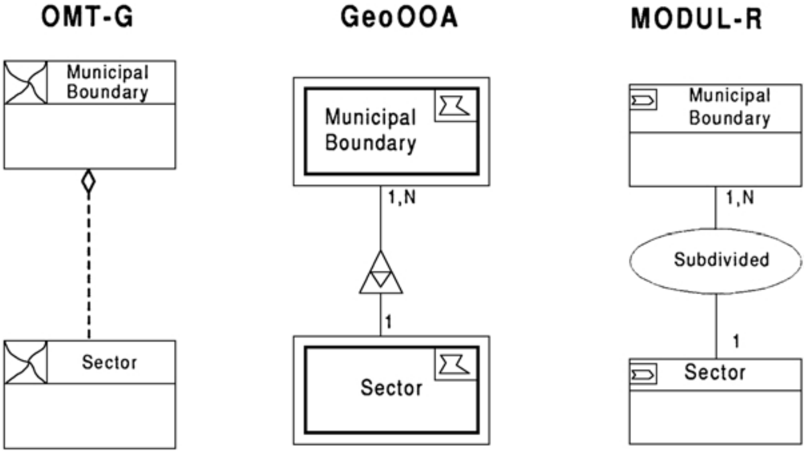


Figure 26. Spatial aggregation.

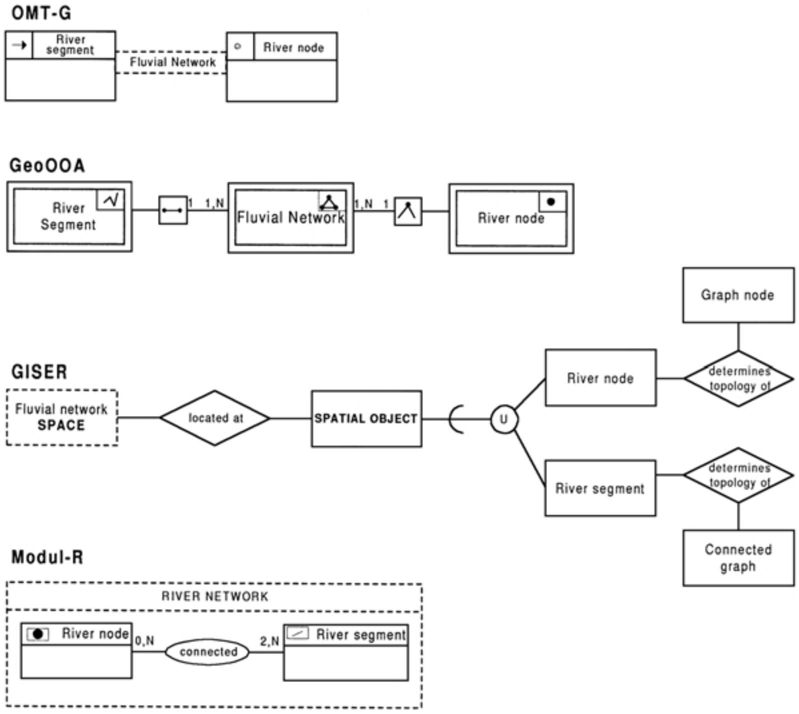


Figure 27. Network structure (fluvial network).

Among the presented models, only GISER can represent the modification of visual attributes of a class through a visual restriction on the *display* relation. However, the modified visual representation is not specified. OMT-G is clearer, since the same notation is used in generalization, only switching the connection line type to dashed (figures 28 and 29), along with an associated discriminator, indicating which property is being abstracted by the spatial generalization relationship.

Only the OMT-G model, through the use of a specific primitive, differentiates between the existence of multiple representation forms of the same class, indicating also whether these forms occur simultaneously or not (figure 29).

5. Conclusions

In this paper, OMT-G, an object modeling technique for geographic applications, has been presented. OMT-G offers primitives that provide the means for modeling the geometry and topology of geographic data, making the modeling of geographic applications easier. Because it uses pictograms in the representation of the geometry of georeferenced object

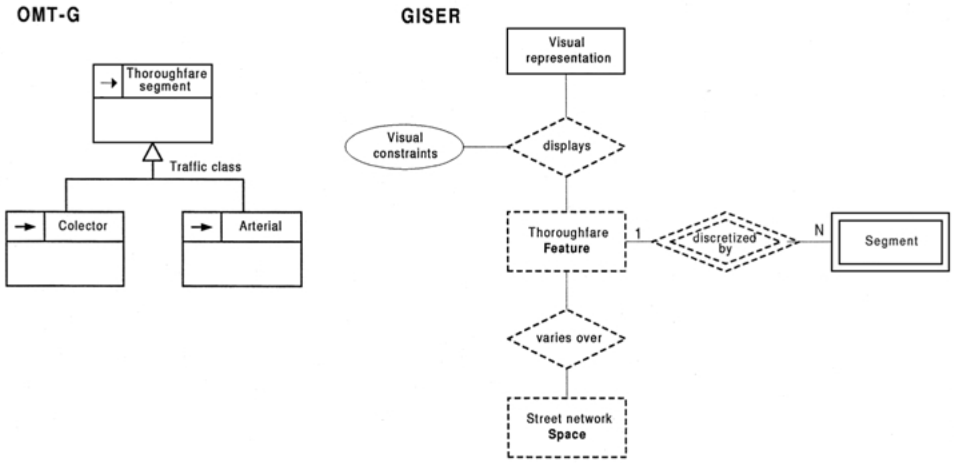


Figure 28. Spatial generalization.

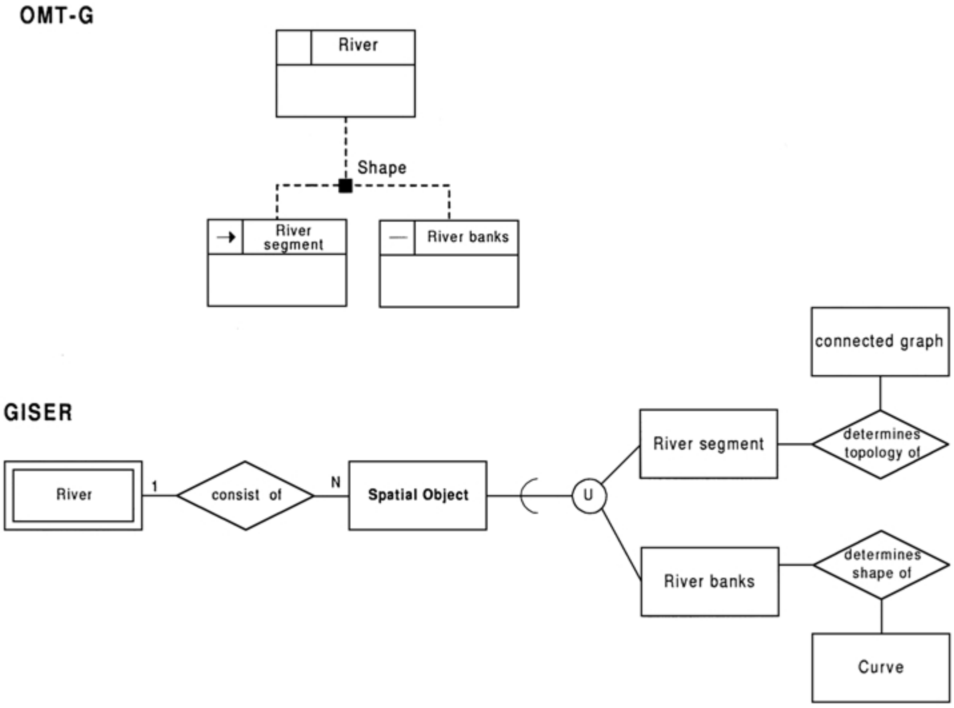


Figure 29. Multiple representations (cartographic generalization).

classes, the schema built by using OMT-G is more compact, more intuitive and more understandable than those derived from models which describe geometric types through relationships. Furthermore, the gap between the conceptual design and the implementation of geographic applications is reduced, by allowing a more precise definition of the required objects, operations, and visualization parameters using the combination of class, transformation, and presentation diagrams.

The richness of expression of the model has not diminished its capacity of being easily understood. In fact, OMT-G presents clear advantages for several important classes of applications, such as those in the urban, environmental, and automated cartography fields. Urban applications tend to be very complex, involving a large number of geographic entities and relationships, and so their static modeling benefits from the concise set of primitives offered by OMT-G. Having less dense class diagrams is an advantage that should not be understated, especially in urban applications, and the separate transformation and presentation diagrams also help to achieve that. Environmental applications, on the other hand, tend to have simple class diagrams, but demand a widely varied and complex set of spatial analysis operations that can be specified using the transformation diagram. The focus of automated cartography applications differs from urban and environmental applications due to their tendency to emphasize the visual aspects of data, and therefore the features of the presentation diagram become more important. In our experience with the OMT-G model, it has proved being capable of representing the particular aspects of geographic data, regardless of the application field, while preserving clarity and representation ease. OMT-G is currently being used by GIS professionals in several Brazilian governmental units and utilities companies, and is being disseminated to a wider range of organizations.

In our comparison evaluation, the spatio-temporal aspect [7], [38], [43] mentioned in Section 1.2 has not been considered, since OMT-G still does not offer temporal representation mechanisms. However, studies are being developed in order to extend OMT-G so that spatio-temporal aspects of geographic data can be represented. In addition, a graphical tool to support the use of OMT-G for modeling geographic applications is under consideration. A prototype of such a tool has been implemented as a stencil for the diagramming toolbox software Microsoft Visio, and is available both at http://www.pbh.gov.br/prodabel/cde/omt-g.html and http://www.omtg.hpg.com.br.

Acknowledgments

Karla A. V. Borges and Clodoveu A. Davis Jr. wish to thank Prodabel for its continued strategic support of research and development activities, specially in the GIS and public administration fields. Alberto H. F. Laender is partially supported by CNPq under individual research grant number 351036/95-4.

The authors would also like to thank the anonymous reviewers, whose comments have been greatly helpful to improve the quality of this paper.

Notes

1. An effort must be made to avoid confusion in the meaning of the word *generalization*. In cartography, it is about generating a less detailed representation from a more detailed one, in order to reduce the scale of a map [33]. In data modeling, on the other hand, it refers to a type of abstraction used in semantic and object-oriented data models [19].

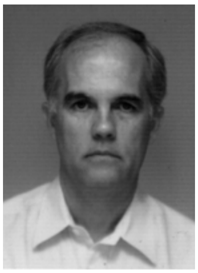
References

1. S. Abiteboul and R.H. Richard. ''IFO: a formal semantic database model,'' *ACM Transactions on Database Systems*, Vol. 12(4):525–565, 1987.
2. G. Abrantes and R. Carapuça. ''Explicit representation of data that depend on topological relationships and control over data consistency,'' in *Proc. Fifth European Conference and Exhibition on Geographical Information Systems—EGIS/MARI'94*, Vol. 1:869–877, 1994. (wwwsgi.ursus.maine.edu/gisweb/egis/eg94100.html)
3. Y. Bédard, C. Caron, Z. Maamar, B. Moulin, and D. Vallière. ''Adapting data models for the design of spatio-temporal databases,'' *Computers, Environment and Urban Systems*, Vol. 20(1):19–41, 1996.
4. K.A.V. Borges and F.T. Fonseca. ''Geographic data modeling in discussion,'' in *Proc. GIS Brasil'96*, 525–532, 1996. In Portuguese.
5. K.A.V. Borges, A.H.F. Laender, and C.A. Davis Jr. ''Spatial data integrity constraints in object oriented geographic data modeling,'' in *Proceedings of the 7th International Symposium on Advances in Geographic Information Systems (ACM GIS'99)*, 1–6, 1999.
6. K.A.V. Borges. ''Geographic data modeling—an extension of the OMT model for geographic applications,'' Master's thesis, João Pinheiro Foundation, Minas Gerais Government School, 1997. In Portuguese.
7. M.A. Botelho ''Incorporating spatio-temporal facilities to object-oriented database management systems,'' Master's thesis, DCC-UNICAMP, 1995. In Portuguese.
8. G. Cămara, U. Freitas, R. Souza, M. Casanova, A. Hemerly, and C. Medeiros. ''A model to cultivate objects and manipulate fields,'' in *Proc. 2nd ACM Workshop on Advances in GIS*, 20–28, 1994.
9. G. Cămara. ''Models, languages, and architectures for geographic databases,'' Ph.D. Thesis, INPE, 1995. In Portuguese.
10. C. Caron and Y. Bédard. ''Extending the individual formalism for a more complete modeling of urban spatially referenced data.'' *Computers, Environment and Urban Systems*, Vol. 17:337–346, 1993.
11. P. Chen. ''The entity-relationship model—toward a unified view of data,'' *ACM Transactions on Database Systems*, Vol. 1(1):9–36, 1976.
12. E. Clementini, P. Felice, and P. Oosterom. ''A small set of formal topological relationships suitable for end-user interaction,'' in *Proc. 3rd Symposium on Spatial Database Systems*, 277–295, 1993.
13. P. Coad and E. Yourdon. Object-Oriented Analysis, 2nd edition. Prentice Hall, 1991.
14. S. Cockcroft. ''A taxonomy of spatial data integrity constraints,'' *GeoInformatica*, Vol. 1(4):327–343, 1997.
15. C.A. Davis Jr. ''Multiple representations in geographic information systems,'' Ph.D. Thesis, Universidade Federal de Minas Gerais, Belo Horizonte, 2000. In Portuguese.
16. C.A. Davis Jr. and A.H.F. Laender. ''Multiple representations in GIS: materialization through geometric, map generalization, and spatial analysis operations,'' in *Proceedings of the 7th International Symposium on Advances in Geographic Information Systems* (ACM GIS'99), 60–65, 1999.
17. M.J. Egenhofer and R.D. Franzosa. ''Point-set topological spatial relations,'' *International Journal of Geographical Information Systems*, Vol. 5(2):161–174, 1991.
18. M.J. Egenhofer and J. Herring. ''A mathematical framework for the definition of topological relationships,'' in *Proc. 4th International Symposium on Spatial Data Handling*, 803–813, 1990.
19. R. Elmasri and S. Navathe. *Fundamentals of database systems.* 2nd Edition. Addison-Wesley, 1994.

20. M. Feutchwanger. ''Towards a geographic semantic data model,'' Ph.D. thesis, Simon Fraser University, 1993.
21. A.U. Frank. ''Qualitative spatial reasoning: cardinal directions as an example,'' *International Journal of Geographical Information Systems*, Vol. 10(3):269–290, 1996.
22. A.U. Frank and M.F. Goodchild. ''Two perspectives on geographical data modeling,'' National Center for Geographic Information and Analysis (NCGIA). Technical Report 90-11, 1990.
23. J. Freeman. ''The modelling of spatial relations,'' *Computer Graphics and Image Processing*, Vol. 4:156–171, 1975.
24. M.F. Goodchild. ''Geographical data modeling,'' *Computers & Geosciences*, 18(4):401-408, 1992.
25. K.K. Kemp. ''Environmental modeling with GIS: A strategy for dealing with spatial continuity,'' Ph.D. thesis, University of California at Santa Barbara, 1992.
26. G. Kösters, B. Pagel, and H. Six. ''GIS-application development with GeoOOA,'' *International Journal of Geographical Information Science*, Vol. 11(4):307–335, 1997.
27. A.H.F. Laender and D.J. Flynn. ''A semantic comparison of modelling capabilities of the ER and NIAM models,'' in, R. Elmasri, V. Kouramajian, and B. Thalheim (Eds.) *Entity-Relationship Approach—ER'93*, 242–256, Springer-Verlag, 1994.
28. J. Lisboa Filho. ''Conceptual data models for geographic information systems,'' Technical Report EQ-12, UFRGS, 1997. In Portuguese.
29. J. Lisboa Filho and C. Iochpe. ''Comparative analysis of conceptual data models for geographic information systems,'' Technical Report RP-266, UFRGS, 1996. In Portuguese.
30. D.M. Mark, M.J. Egenhofer, and A.R.M. Shariff. ''Towards a standard for spatial relations in SDTS and geographic information systems,'' in *Proc. GIS/LIS'95*, 686–695, 1995.
31. D.M. Mark and A.U. Frank. ''Language issues for geographical information systems,'' National Center for Geographic Information and Analysis (NCGIA), Technical Report 90-10, 1990.
32. R.B. McMaster and K.S. Shea. Generalization in Digital Cartography, Association of American Geographers, 1992.
33. J.C. Müller, R. Weibel, J.P. Lagrange, and F. Salgé. ''Generalization: state of art and issues,'' in J.C. Müller, J.P. Lagrange, and R. Weibel (Eds.) *GIS and generalization: methodology and practice*, 3–17, Taylor & Francis, 1995.
34. J.L. Oliveira, F. Pires, and C.M.B. Medeiros. ''An environment for modeling and design of geographic applications,'' *GeoInformatica*, Vol. 1(1):29–58, 1997.
35. D. Papadias and Y. Theodoridis. ''Spatial relations, minimum bounding rectangles, and spatial data structures,'' *International Journal of Geographical Information Science*, Vol. 11(2):111–138, 1997.
36. D.J. Peuquet. ''A conceptual framework and comparison of spatial data models,'' *Cartographica*, Vol. 21:666–113, 1984.
37. ''Rational Software Corporation,'' *The Unified Language*. Notation guide, version 1.1 July 1997. (http://www.rational.com).
38. A. Renolen. ''Conceptual modelling and spatiotemporal information systems: How to model the real world,'' in *Proc. 6th Scandinavian Research Conference on GIS (SCANGIS'97)*, 1997. (http://www.iko.unit.no/home/agnar)
39. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. Object-Oriented Modeling and Design, Prentice-Hall, 1991.
40. J. Rumbaugh. OMT Insights: Perspectives on Modeling from the Journal of Object-Oriented Programming. SIGS Books, 1996. 390 pp.
41. S. Shekhar, M. Coyle, B. Goyal, D. Liu, and S. Sarkar. ''Data models in geographic information systems,'' in *Communications of the ACM*, Vol. 40(4):103–111, 1997.
42. H. Tardieu, A. Rochfeld, and R. Colletti. La méthode Merise: Principes et Outils, Tome I. Les Éditions d'Organisation, 1986.
43. M.F. Worboys. ''A unified model for spatial and temporal information,'' *The Computer Journal*, Vol. 37(1):26–34, 1994.
44. M.F. Worboys, H.M. Hearnshaw, and D.J. Maguire. ''Object-oriented data modelling for spatial databases,'' *International Journal of Geographical Information Systems*, Vol. 4(4):369–383, 1990.

Karla A.V. Borges received her B.S. degree in Civil Engineering in 1982 from PUC-MG, and her M.Sc. in Informatics and Public Administration from João Pinheiro Foundation, in 1997. She is the leader of geographic data modeling efforts for the GIS project of Belo Horizonte, Brazil, and the former head of the Urban Management Applications Department at Prodabel (Belo Horizonte's municipal information technology company). Currently, she is a Ph.D. student at the Computer Science Department of Universidade Federal de Minas Gerais. Her main interests are geographic databases, geographic data modeling, urban GIS, and ontologies.

Clodoveu A. Davis Jr. received a B.S. degree in Civil Engineering in 1985 from Universidade Federal de Minas Gerais. He also has M.Sc. and Ph.D. degrees in Computer Science, also from Universidade Federal de Minas Gerais, in 1992 and 2000, respectively. He led the team at Prodabel that conducted the implementation of GIS technology in the city of Belo Horizonte, Brazil, and coordinated several geographic application development efforts. Currently, he is a researcher at Prodabel's Development and Studies Center, and the editor of *Informatica Publica*, a Brazilian journal on information technology for the public sector. His main interests are urban GIS, geographic databases, map generalization, and multiple representations in GIS.



Alberto H.F. Laender received his B.S. degree in Electrical Engineering and the M.Sc. degree in Computer Science from the Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, in 1974 and 1979, respectively, and the Ph.D. degree in Computing from the University of East Anglia, Norwich, England, in 1984. He joined the Computer Science Department of the Universidade Federal de Minas Gerais in 1975, where he is currently a Full Professor and the head of the Database Research Group. He was also twice the Coordinator of the Computer Science Graduate Program (1987–89 and 1993–96). In 1997, he was a visiting scientist at the Hewlett-Packard Palo Alto Laboratories. He has served as a program committee member for several international conferences on databases and Web-related topics, and was one of the program committee co-chairs of 19th International Conference on Conceptual Modeling held in Salt Lake City, Utah, in October 2000. He is also a founder member of the Brazilian Computer Society and an Editorial Board member of the *Journal of the Brazilian Computer Society* and of the International Systems Review. His research interests include database modeling, database design methods, database user interfaces, semistructured data, and Web data management.