Modeling Spatial Data in the MADS Conceptual Model

Christine PARENT

University of Lausanne
HEC Inforge, 1015 Lausanne, Switzerland, Christine.Parent@hec.unil.ch

Stefano SPACCAPIETRA, Esteban ZIMANYI,
Pier DONINI, Corinne PLAZANET, Christelle VANGENOT

Laboratoire de Base de Données
Ecole Polytechnique Fédérale de Lausanne 1015 Lausanne, Switzerland
email: {Firstname.Lastname}@epfl.ch http://lbdwww.epfl.ch
telephone: +41 21 693 52 11, fax: +41 21 693 51 95

ABSTRACT: Despite the well-established benefits of conceptual modeling for application design, current spatio-temporal conceptual models do not cope satisfactorily with designers' requirements. In this paper we first identify the goals of a spatio-temporal conceptual model and then we describe the MADS model along its structural and spatial dimensions. As the modeling concepts are orthogonal, the proposed model achieves both simplicity (as concepts are independent from each other) and expressive power (as concepts may be freely associated). The model, which includes the features of the ODMG standard model for object-oriented systems, has been implemented by translating it into operational models of existing products. Experience gained by modeling a number of applications enabled us to assess the advantages of MADS with respect to traditional entity-relationship modeling. A visual schema editor and a visual query editor are being developed.

KEYWORDS: database design, conceptual modeling, data model, spatial information system, geographic information system.

1 Introduction

For applications manipulating spatio-temporal data, conceptual modeling offers important advantages with respect to modeling approaches favoring a logical design directly related to the particularities of the GIS (Geographical Information Systems) software being used. First, users may express their knowledge about the application using concepts that are close to them, independently of computer concepts. As conceptual modeling is independent from the software tool on which the information system is implemented, the resulting design remains valid in case of technological change (only the translation of the conceptual schema to the logical schema is affected). At the current pace of technological evolution, in particular for GIS tools, this is a key factor for flexibility since it preserves past investments while reducing the cost and increasing the chances of success of a transfer to a new technology. Finally, conceptual modeling, by its readability, facilitates the exchange of information between partners of different organizations. At a time when Internet access to external information sources becomes a banality and a necessity, the capability to understand the semantics of this information is determining for its correct use.

Conceptual modeling has been a very active research domain during the last decades. Still, classical conceptual models do not integrate the specificity of spatio-temporal data. On the other hand, data models of major GIS tools reflect internal implementation techniques and call for such technical knowledge that they are mostly usable by tool experts.

Applications manipulating geodata are difficult to model due to the particularity and complexity of the spatial and temporal components. More facets of real-world entities have to be considered (location, form, and size), more links are relevant (spatial and temporal links), several spatial abstraction levels often need to be represented, and temporal characteristics of data have to be included. Thus, geodata modeling requires advanced facilities, such as:

- objects with complex structure (e.g., non-first-normal-form, composition/aggregation links, generalization links), at least equivalent to those supported by current object-oriented models,
- alternative geometry features, to support both discrete and continuous views of space, representations at different scale/precision, multiple viewpoints from different users.
- explicit relationships to describe structural links as well as spatial (such as adjacency, inclusion, spatial aggregation) and temporal links (such as before, during),
- diverse temporal features, including support of multiple granularities.

This paper discusses the issues in designing a conceptual model for spatial data and proposes such a model, called MADS (for Modeling of Application Data with Spatio-temporal features). MADS has been developed in close cooperation with users from our regional administration and therefore takes into account their priorities. In particular, it is limited to two-dimensional data and favors the discrete view of space. MADS also includes temporal features, but these are not discussed here as they are described in a companion paper [Spaccapietra 98]. Section 2 surveys related work in spatial conceptual modeling. Section 3 briefly presents MADS concepts for modeling traditional data. Section 4 covers the modeling of spatial objects and attributes. Section 5 deals with spatial relationships. Section 6 is devoted to the representation of the continuous view of space. Finally, Section 7 presents perspectives and future work.

2 Related work

Although several conceptual models with spatial features have been proposed, they do not provide satisfactory answers to users' requirements. The proposed models relate to either the entity-relationship (ER) approach [Chen 76], or the object-oriented (OO) approach. The first group includes MODUL-R [Bédard 96, Caron 93] and GéO2 [David 93]. The second group includes GeoOM [Tryfona 97], POLLEN [Gayte 96], and CONGOO [Pantazis 96].

Another classification of these models is according to the solution they chose for integrating **spatiality**. A first solution consists in explicitly representing space by a set of geometrical entity (or object) types, e.g., Point, Line, or Area. Application objects are described in a traditional way as semantic entity (or object) types and their spatiality is defined by a relationship link with one of the geometrical types. GeoOM and POLLEN adopt this solution. This approach has major drawbacks. First, it ends up overcharging schemas, which become soon unreadable due to the proliferation of links. Second, it only allows to associate spatiality to entity types (a relationship may only link entity types), and is therefore restrictive with respect to applications' requirements. As we shall see later, space may be associated with objects, relations, or attributes. Finally, imposing geometrical entity types contradicts the very first principle of conceptual modeling, that only objects of interest to the application should be described. Most applications will not be interested in a point, line, or area object per se.

A second solution allows attaching a spatial type to an application entity type. Thus MODUL-R offers a rich set of spatial types: point, line, area, as well as complex (e.g., point AND line), alternative (e.g., line OR area) and multiple types (two or several geometries non-deducible from one another). CONGOO proposes a very large variety of spatial object types, simple or complex, resulting in a very complete but also very complex model. This approach generates compact schemas: spatial types are represented by an icon associated to entity types. However, as the previous solution, it is restrictive with respect to modeling needs (only entity types may bear spatiality).

The third solution just makes a different choice with respect to the second one: it associates spatiality to attributes rather than to entity types. GéO2 is a representative example: spatial objects are described by entities having one (or several) attribute(s) of domain Geometry. Geometry is an abstract data type describing a set of primitive structures (point, line, and area) with the associated geometrical functions (e.g., intersection, union). The set structure enables users to represent entities of the same type having geometries of different dimensions (e.g., line and area).

The knowledge of the **topological links** between real-world entities is another essential requirement for applications. Current GISs partly answer to this need by offering an automatic computation of the topological links based on the physical coordinates of objects. Three important needs remain unsatisfied, since it is not possible to:

- constrain spatial objects to obey a given topological link (a priori statement of the existence of the link rather than a posteriori observation),
- define application-specific semantics for these links (e.g., adjacency could be defined as "being located at a reciprocal distance of less than 500 meters"), and
- explicitly describe a topological link important to the application, so that it can be seen in the schema, named, and can possibly bear attributes. For example, a road management application would need to materialize road crossings for describing the roadsign equipment.

Most models allow an explicit description of topological links only as classical relationships, without any particular semantics and without adequate integrity control. CONGOO offers the possibility to describe authorized, mandatory, or forbidden topological links between real-world entities, based on the topological relationships of neighborhood and superposition. However, these "topological relationships" only exist as spatial integrity constraints: they are never instantiated. Since they do not link two occurrences of entities, they are not relationships in the ER sense.

Other important concepts for modeling spatial data include: **spatial aggregation**, **generalization/specialization**, and **continuous fields**. These concepts are proposed in some of the models, but their definitions differ among them. Spatial aggregation remains classical and governed by rigid rules that may not be appropriate. Only GéO2 proposes the propagation of attribute values and methods as suggested in [Egenhofer 92b]. As for generalization/specialization, the associated inheritance mechanisms are often badly defined, if at all. Finally, the possibility to describe a continuous view of space is only supported by POLLEN and GeoOM. In POLLEN, the continuous view of an object is expressed by a multivalued link to another object representing the spatial continuous field, itself linked to a geometrical object of type Point, thus inducing the definition of many artificial object types. GeoOM associates this concept to the space and not to the instance, which raises problems with overlapping entities, i.e., entities sharing the same space.

Regarding **notations**, MODUL-R uses graphical icons to visually apprehend the different spatial characteristics. In GéO2, POLLEN and GeoOM, spatiality is described in a textual way. In POLLEN and GeoOM, it is also necessary to follow the relationships to find this information. In CONGOO, special signs are associated to the different spatial object types; spatiality of objects and topological integrity constraints are described textually, which is too tedious for occasional or non-specialist users.

In summary, there still is a need for a spatial conceptual model, rich enough in expressive power for supporting the diversity of application's requirements. The model must also allow defining schemas that are readable and easy to understand. A key element for achieving this double objective is the orthogonality of the structural and spatial dimensions of the model (and more generally of the concepts of the model). Thus, whatever the element in the model (e.g., object, relationship, attribute, aggregation), the spatial dimension may be associated to it. This orthogonality, which is the base of our proposal, lacks in the models mentioned.

Beyond orthogonality, other important characteristics for a spatial conceptual model include: 1) the provision of generic spatial types in addition to the base types, 2) intuitive visual notations, 3) the possibility to explicitly describe topological relationships between entities, 4) a formal definition of all concepts, including aggregation, generalization/specialization and the associated inheritance mechanisms, and 5) the possibility to describe and interrelate the discrete and continuous views of space. These characteristics are discussed in the following sections.

3 Modeling Classical Characteristics

The objectives stated in the previous section guided our design of the MADS conceptual model. An associated visual syntax has been defined to achieve simplicity and readability.

MADS adopts the object+relationship paradigm, which includes the features of the ODMG (Object Database Management Group) standard model for object-oriented systems [Cattell 97]. A set of well-known concepts are supported as follows:

- An **object** represents a real-world entity. An **object type** describes a set of objects with similar structure and behavior.
- A **relationship** is a link between two or more objects, where each object plays a given role. A **relationship type** describes a set of links with similar characteristics (linking objects of the same types, with the same roles, and similar properties).
- An **attribute** represents a real-world property; both object types and relationship types may have attributes. Attributes can be:
  - simple (with atomic values) or complex (i.e., composed of simple or complex attributes);
  - monovalued (with a single value) or multivalued (with a multiset value);
  - mandatory (with a value in every instance) or optional (with a value in some instances and no value in others).

  The mono/multi-valued and optional/mandatory characteristics are integrity constraints expressed via the minimum/maximum **cardinality** concept. Cardinalities also apply to roles in relationship types. They consist in two numbers, min and max, specifying how many (at least, at most) instances of a relationship type may, at a given instant, link an object of the associated type.

  The value of an attribute may be computable, i.e., automatically inferred by the system, from values of other attribute(s) of the same or other objects or relationships. Such an attribute is called a **derived attribute**. The derivation function may invoke both computations and navigations through the database.
- A **method** is an operation that may be realized on the occurrences of an object type. Its declaration is composed of (1) an interface, which specifies the name of the method, the parameters with their type, and eventually the result type; and of (2) one (or several) implementations, i.e., the code of the method.
- A **generalization** link relates two object types, a supertype and a subtype. A generalization link between two objects expresses that both objects represent the same real-world entity at two different abstraction levels. Thus, an object can exist in the population of the subtype only if it also belongs to the supertype.

  An object type may be a subtype of several supertypes, whose populations intersect: e.g., Castle may be a specialization of both Building and Curiosity as in Figure 7. A supertype may be specialized into subtypes according to several criteria: e.g., Building specialized according to its function (Office, Housing) and according to its material (Concrete, Wood, etc.).

  Inherited properties (attributes, methods, relationships, spatial characteristics) can be refined or redefined. Refinement is used to constrain the domain in the subtype to be a subdomain of the corresponding property in the supertype. Redefinition associates an additional value to be stored in objects of the subtype (for methods, a new signature and/or implementation are stored). This allows multiple representations of a given feature. Examples are shown in Section 5.3.

- An **aggregation** link is a special directed binary relationship whose semantics expresses that objects of a type, called composite objects, represent aggregates of objects of another type, called component objects. As for generic relationships, cardinalities are attached to the composite and component roles.

Figure 1 shows a simple example of traditional data modeling using the MADS visual notation.

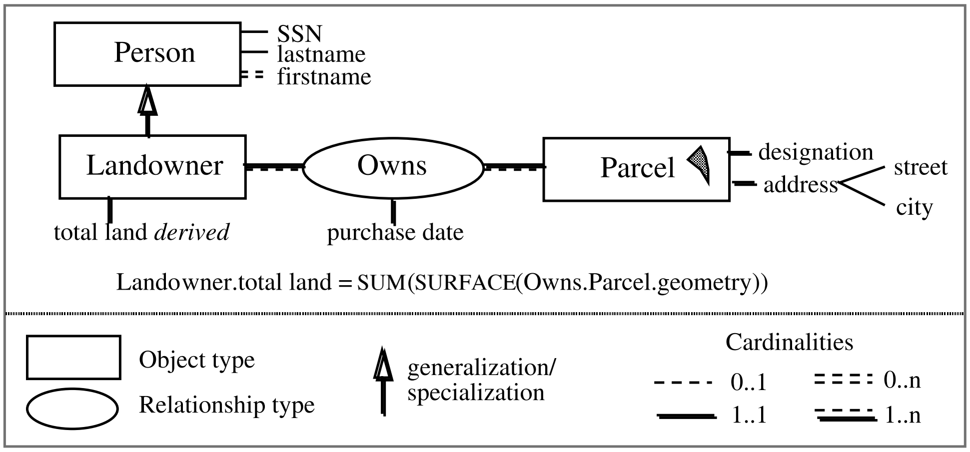

Figure 1: A sample MADS schema with one derived attribute.

4 Modeling Spatial Entities

4.1 Spatial Abstract Types

Current GISs offer two ways to describe relationships between objects and space: a **discrete** view (also called vector view), where the database consists of objects whose location in space may be defined, and the **continuous** view (also called field view) where the database consists of space regions over which properties are defined as continuous field of values. The MADS model favors the discrete view, since it corresponds to the requirements of our users (e.g., for land, road, and water resources management). However, MADS also provides mechanisms for representing a continuous view. For purely pragmatic reasons, the model is currently limited to the representation of two-dimensional data.

Spatiality may be defined more or less precisely. For example, a schematic map of urban transports may need to define only the topological relationships between stops and lines (this stop is on that line). Metrical relationships (this stop is 500 meters from that other) and orientation relationships (this stop is 40° degrees from that other) are secondary and do not need to be faithfully represented. On the contrary, a survey will be exact with respect to all these three types of relationships. Also, the coordinates of objects (e.g., buildings, streets) will also be exact, according to the precision of the map.

In MADS, spatiality may be associated to object types, attributes, relationships, and aggregation links. Spatiality is visualized through icons expressing the information in an unambiguous, visual, and synthetic way. The model allows choosing the level of preciseness for describing spatiality. The spatiality of an object type can be either defined precisely (e.g., point, oriented line), or left undetermined (e.g., point OR line OR area). The location may also be defined either precisely, i.e., in an absolute way by stating the coordinates of the objects, or in a relative way, i.e., by defining the position with respect to other known locations (e.g., Lausanne is 60 km from Geneva on road N1).

The model provides the usual basic spatial types: point, line (whether straight, arc, polyline, closed or not, oriented or not), oriented line, and simple area (possibly with holes but without islands). It also includes spatial types describing spatially homogeneous sets of objects: point set (e.g., for representing the houses of a town), line set (e.g., for representing a road network), oriented line set (e.g., for representing a river with its tributaries), and complex area (i.e. a set of simple areas).

Finally, MADS includes some generic types (of dimension 0, 1 or 2) as supertypes of the previous ones. The simple geo type represents any simple spatial type. The complex geo type allows to describe every composition of simple spatial types, e.g., a fluvial network defined by a set of rivers (oriented lines) and lakes (areas). The geo type, the most generic one, means "this type is spatial" without any commitment to a specific spatial abstract type. For object types associated to a generic spatial type, the spatial type of each instance is specified at creation time.

To each spatial type is associated a set of methods for defining and manipulating instances of this type, e.g. dimension for geo and length for line. According to the applicability of the methods over different spatial types, spatial types are organized in a generalization hierarchy, as shown in Figure 2. Icons associated to each type are also shown. The hierarchy may be easily extended, to suit application needs, by creating new subtypes or by grouping several types in a supertype: e.g., grouping point and area for describing the spatiality of towns, whether small or large.

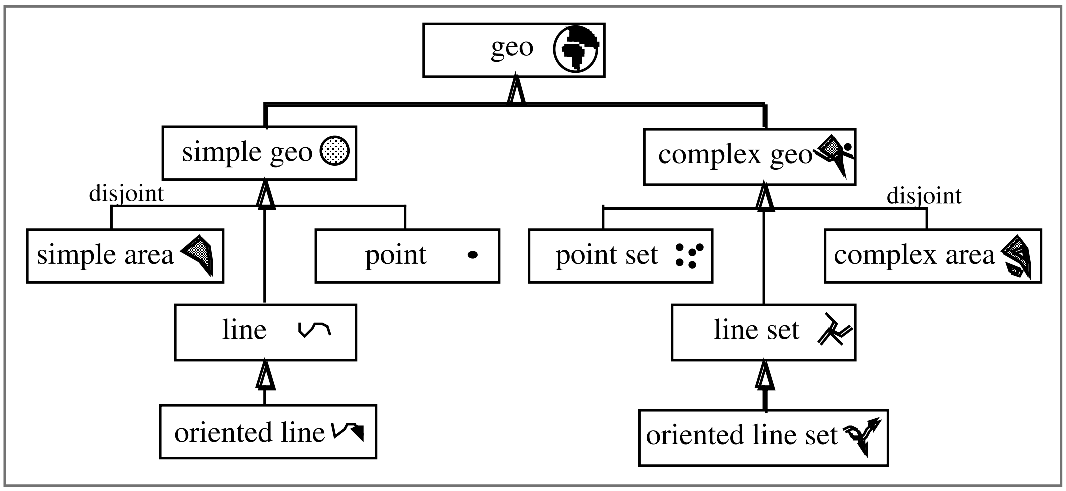

Figure 2: Basic hierarchy of spatial abstract types.

4.2 Spatial Attributes and Object Types

A **spatial object** represents an entity whose spatial reference must be recorded, i.e. that needs to be represented in space by defining its shape (e.g., point, line, or area) and its location. Consistently with the discussion in section 2, the spatiality of an object, grouping the notions of shape and location, is described by a predefined attribute, geometry, whose domain of values is a spatial abstract type. A **spatial object type** represents a set of spatial objects having the same characteristics.

A **spatial attribute** is a simple attribute, monovalued or multivalued, whose domain is a spatial abstract type.

Being a flexible model, MADS allows to describe a spatial feature either as an object or as an attribute according to the abstraction level considered by the application. For example, reservoirs located on rivers can be perceived either as objects on their own or as mere properties of rivers, i.e., as attributes of the River object type (Figure 3). Notice that spatiality can be attached to reservoirs whatever the perception of reservoirs is.

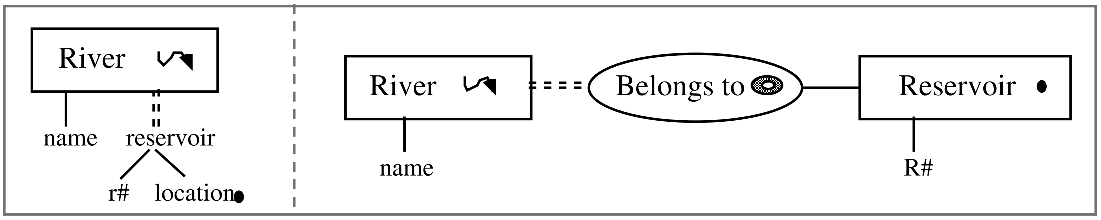

Figure 3: Alternative schemas for rivers and reservoirs.

Several spatial constraints may be associated to spatial attributes and object types. For example, the spatiality of the occurrences of an object type may be constrained: e.g., instances of County cannot overlap, are connected, etc. The spatiality of an attribute may be constrained with respect to the spatiality of its associated object: e.g., in Figure 3 the point attribute reservoir.location must be on the line describing the geometry of its river.

The notions of **envelope** and of **interior** are very useful for describing the contour of spatial objects, such as the shore of a lake or the border of a country. They are implemented as methods on the corresponding spatial type. They are also essential for defining topological relationships (see Section 5.1). For example, the envelope of Italy (of type complex area) is composed of the contour of the boot, of that of its islands (e.g., Sicily, Sardinia) and that of the envelopes of Vatican and San Marino, two enclaves inside Italy.

Much research has been done on the concepts of envelope, interior, and topological relationships for simple objects (e.g., [Egenhofer 92a]). On the contrary, few results are available for composite objects, such as the complex geo type. We extended these concepts for complex areas [Nguyen 97], and we are currently studying the case of general composite objects.

5 Spatial Relationship Types

The spatiality of an application is reflected by the existence of spatial entities, but also by the existence of spatial relationships between these entities. These relationships may be of different types, e.g., topological, orientation, metrical, spatial aggregation.

Spatial relationships can be deduced from the spatiality of objects, provided that it is directly defined by their coordinates. Thus, these relationships implicitly exist and are accessible through the GIS functions. However, it is important to be able to explicitly describe spatial relationships in conceptual schemas. This enriches the schema, allows these relationships to be named, to attach to them attributes and methods, and to give them additional semantics than that obtained through the GIS functions.

In MADS, a **spatial relationship type** is a relationship type linking at least two spatial object types (and possibly other non-spatial object types), and whose semantics is explicitly defined by a spatial integrity constraint. For example, a spatial relationship RoadCrossing linking objects of type Road may refer to the crossing topological relationship as the defining integrity constraint (Section 5.1). That will force the instances of RoadCrossing to link only roads whose lines are effectively crossing each other.

MADS provides two predefined categories of spatial relationships, topological relationships and spatial aggregation, corresponding to the most usual requirements of applications. In addition, any other spatial relationship type may be explicitly declared with the methods attached to the spatial abstract types. For example, a spatial relationship of proximity Along may be defined between the object types City and Lake, with the spatial integrity constraint distance(City, Lake) < 5 km.

5.1 Topological Relationships Types

There exist a great number of topological relationships. They were abundantly studied as spatial operators for manipulating spatial objects. Several classifications have been proposed [Champoux

92, Egenhofer 91, Egenhofer 92a]. However, these theoretical classifications result in too many different spatial operators, which is not useful for a conceptual model serving as a communication tool between humans, not necessarily specialists in topology. We rather adopted the classification proposed in [Clementini 93], which is more user oriented: it groups all elementary topological relationships in a few classes. We added the equality relationship, which is useful when a spatial object plays several roles or when two spatial objects share the same space. The list of pre-defined topological relationship types is given in Table 1.

spatial type	**icon**	**definition**
disjunction		the linked objects have spatially disjoint geometries
adjacency		geometry sharing without common interior
crossing		sharing of some part of the interior, such that the dimension of the shared part is strictly inferior to the higher dimension of the linked objects
overlapping		sharing of some part of the interior, such that the dimension of the shared part is equal to the dimension of the linked objects
inclusion		the whole interior of one object is part of the interior of the other object
equality		sharing of the whole interior and of the whole envelope (valid for spatial object types of the same dimension)

Table 1: Topological relationship types.

Every topological relationship type is characterized by its spatial type, which is visually represented by an icon. Although these icons represent surface objects, these symbols are valid for every spatial object type.

The example of Figure 4 shows a schema with three spatial object types, County (complex area), City (point) and Lake (simple area), and two topological relationship types, Contains (inclusion) and Along (adjacency).

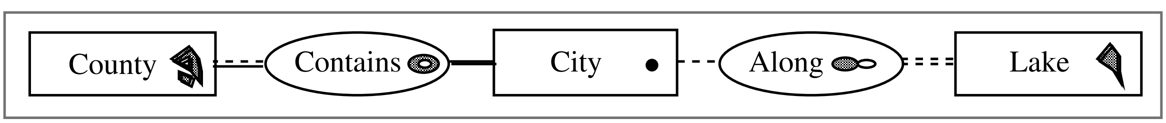

Figure 4: Example of topological relationship types.

5.2 Spatial Aggregation

Aggregation is very common in spatio-temporal applications. For example, Figure 5 represents a county as an aggregation of 10 to 1000 districts, where a district is component of exactly one county. The attribute joiningDate records the date at which the district joined the county.

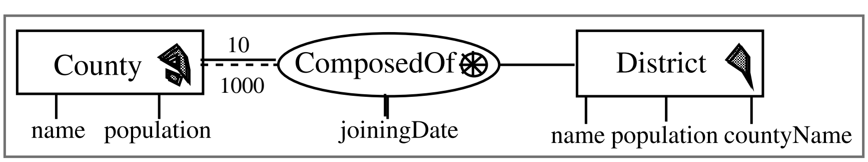

Figure 5: Example of spatial aggregation.

In MADS, aggregation is a binary link directed from the composite to the component object. An object type composed of several object types is represented using several aggregation links, one for each component type. This allows, in particular, to represent cardinalities of roles and attributes of the link for each component type.

It is common that some attributes of the composite and component objects are related. These dependencies are represented either by derived attributes, or by integrity constraints. Some examples of spatial integrity constraints related to aggregation are:

- the areas of the districts of a county are connected;
- the area of a house must be included in that of its lot;
- no house in a lot can be more than 25 meters apart.

Otherwise, the dependencies between geometries and between attributes may be expressed using derived attributes. For example, in Figure 5 the following derivation formulas may be defined:

County.geometry = SPATIALUNION (District.geometry)

County.population = SUM (District.population)

District.countyName = County.name

Formulas may be as complex as needed. For example, in a hydrographic network composed of rivers and lakes, the global water volume is computed from the volume of rivers and lakes.

5.3 Generalization links

Generalization links may relate spatial and non-spatial object types. A non-spatial object type can have a spatial subtype: e.g., only cars equipped with a GPS system have a known location (Figure 6(a)). Conversely, any subtype of a spatial object type inherits the geometry of its supertype, and hence is by definition a spatial object type: e.g., streams are represented by oriented lines, as any river is (Figure 6(b)).

As discussed in Section 3, inheritance can be adjusted to application's need using either refinement or redefinition.

Refining is useful each time a property (attribute or geometry) of an object type has a smaller domain in a subtype than in the supertype. For example, Figure 6(c) shows a frequent situation in spatial databases, where the geometry of objects in subtypes differs according to some known criterion. The generic concept town is used to describe both cities and villages, but cities are to be represented as areas and villages are to be represented as points. In such a case, Town is associated with the generic geo spatial type (alternatively one could define a new spatial type: area OR point), and two subtypes are defined, City and Village. In City the geometry is refined to area, while in Village the geometry is refined to point. These declarations will act as integrity constraints, enforcing cities to be areas and villages to be points. Moreover processes dealing with cities only (resp. villages) will use the methods associated to area (resp. point), and processes dealing with a mixed set of towns (cities and villages) the generic methods associated to geo.

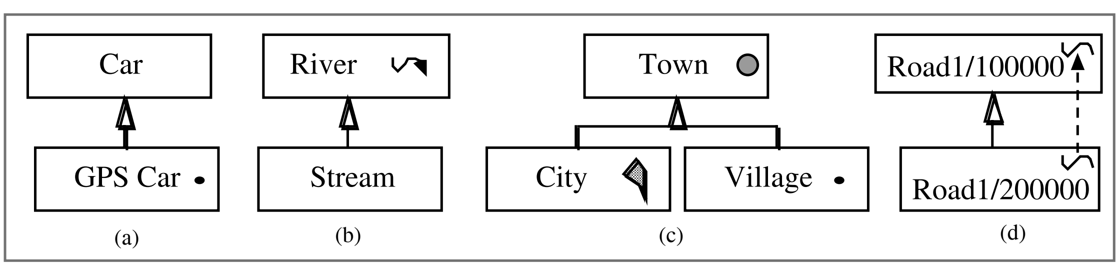

Figure 6: Inheritance, refinement and redefinition of the geometry.

Redefining an inherited attribute aims at a different objective. Redefinition creates a new attribute in the subtype, with the same name. Thus, redefinition of the geometry makes it possible to associate several geometries to the same object. This mechanism can be used to model different representations of the same real world entity at different scales as in Figure 6(d). Objects of the subtype Road1/200000 have two geometries: 1) the one inherited from the supertype, i.e., the line representing the road at the 1/100000 scale, and 2) the locally redefined one, i.e., the line representing the road at the 1/200000 scale. According to the needs of the application, processes will either work on the Road1/100000 object type and use the associated 1/100000 lines, or work on the Road1/200000 object type and be able to access the two geometries: geometry will refer to the simplified lines while Road1/100000.geometry to the inherited 1/100000 lines.

In a MADS diagram, if a subtype definition shows a geometry different from the one of the supertype, it is assumed by default that a refinement is defined. A dashed arrow, as in Figure 6(d) visually represents the redefinition of the geometry [1].

Another kind of design problem solved by generalization links is the sharing of objects by several object types. In Figure 7, the two spatial object types, Curiosity (point) and Building (area) describe different but overlapping sets of real world entities: a castle is a curiosity and a building. Therefore Castle is a subtype of both Curiosity and Building. This situation, called **multiple inheritance**, can lead to an ambiguity when two properties having the same name exist in several supertypes, as geometry. MADS supports multiple inheritance with inherited multiple geometries: each Castle object has two geometries, a point and an area. Access to the multiple inherited property is qualified by the name of the supertype: the two geometries of Castle are named Curiosity.geometry and Building.geometry.

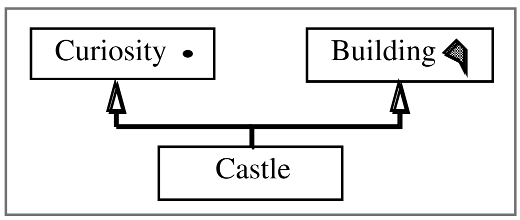

Figure 7: Multiple inheritance inducing several geometries for an object.

6 Space-varying Attributes

Although land management applications apprehend space mainly in a discrete way, often they also need to describe continuous fields, such as elevation or land occupation. A continuous field may be perceived in two different ways:

- as describing the values of a variable over the whole space. This is the approach chosen by [Tryfona 97] and by all GISs offering a continuous view of the space (raster GIS).
- as describing the values of an attribute over the geometry of an object. This is the approach chosen in MADS, both for spatial objects and for spatial attributes with non-zero dimension. The rationale is twofold: 1) it gives more flexibility than the previous solution as it allows the description of overlapping continuous fields, and 2) it provides for seamless integration of both views, discrete and continuous.

An attribute is said **space-varying** if its value is defined by a function whose domain is the set of geometrical elements (points, lines, or areas) in which the geometry of the corresponding object (or attribute) is decomposed. For example, the relief of a county may be represented as a set of elevation values measured on a set of points distributed (regularly or not) over the area of the county. Similarly, the type of crop of an agricultural region may be represented by the value of the dominating crop on sample areas of 100 square meters. Space-varying attributes are represented using the icon ⌘.

---

[1] Notice that refinement or redefinition of an attribute in a subtype implies integrity rules on the restrictions of attribute domains as well as on cardinalities, in order to benefit from the advantages of polymorphism and for ensuring the substitution principle.

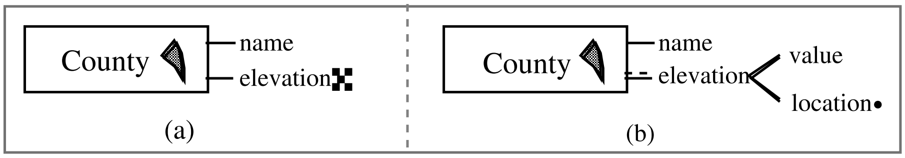

Figure 8: Elevation represented with and without space-varying attributes.

Making space-varying attributes a first-class concept enables designers to directly represent continuous fields without the need of complex constructions. For example, to represent the space-varying attribute elevation of Figure 8(a), a complex multivalued attribute is needed in Figure 8(b). The latter solution makes schemas more complex to understand and requires the addition of an artificial spatial attribute (location) and of a spatial integrity constraint stating that location must be included in the geometry of the county.

Assigning values to a space-varying attribute is made by entering (point, value) couples. The values of a space-varying attribute can be consulted either globally (obtaining a list of couples) or locally (asking the value at a given point). In the latter case, either the value is known at the point, or it is approximated by applying an interpolation method defined by the user.

7 Conclusions and Perspectives

This paper presented the MADS spatial conceptual model. While MADS was designed in an application context of land management and planning, a large range of applications are likely to be able to benefit from this work.

The objective which most influenced the design of MADS is that of orthogonality between the structural and the spatial dimensions. This achieves both simplicity (the concepts are independents) and expressive power (the concepts may be freely combined). In addition, MADS allows to explicitly describe topological relationships between objects. The semantics of these relationships may be customized to match specific needs of the application. The possibility to describe spatial continuous fields is another important functionality supported by MADS. Finally, the use of visual intuitive notations ensures the comfort of the designer and eases the development of visual interfaces.

MADS has been implemented by translating its specifications into specifications acceptable by existing tools: e.g., the IEF design tool [IEF 88], the Swiss INTERLIS exchange standard for geodata [Keller 97], and a relational DBMS. A prototype of a MADS visual schema editor has been developed in Java for ensuring its portability over different platforms. It is based on the principles of direct manipulation and of flexibility [Dennebouy 95]. A schema is easily built using mainly the mouse and dialogue boxes. A multi-paradigm visual query editor will be developed next. A query can, for example, be sketched, then refined in a textual or graphical way (interacting with the query editor) and then its results shown on a map. These and other editors will form a front-end software layer (or a CASE tool) acting as a mediator between designers/users and the underlying system (DBMS or GIS). Our aim is to provide an integrated conceptual solution in which the user's specifications are automatically translated and sent to the underlying GIS or DBMS. The results extracted from the software are then formatted according to the conceptual specifications before being presented to the user.

MADS has been used for modeling several real-world applications: oil management in Colombia, management of the urban networks of clear and used waters in Geneva city, study of the evolution of the watershed of the upper part of the Sarine river, and the management of the water resources of the Vaud county. The latter application, called GESREAU [Crausaz 98], allowed us to measure the benefits of using MADS with respect to a traditional ER model. MADS reduced by 22% the number of object and relationship types with respect to the ER schema (with no semantic loss). Explicit description of topological relationships was used for 2/3 of the relationships in the original

ER schema. Space-varying attributes also proved to be frequently needed. The need for temporal specifications was also highlighted in the re-design process.

Some research axes in spatial databases that we are investigating include:

- other concepts allowing multiple representations of the same object, in particular for modeling changes in the geometry of objects and in their topological links, due to cartographic resolution/scale;
- a visual language for data manipulation;
- formalization of topological relationships for complex objects;
- integration of heterogeneous spatial databases.

References

[Bédard 96] Y. Bédard, C. Caron, Z. Maamar, B. Moulin, D. Valière, Adapting Data Models for the Design of Spatio-Temporal Databases, Computer Environment and Urban Systems, 20 (1), 1996.

[Caron 93] C. Caron, Y. Bédard, P. Gagnon, MODUL-R, Un formalisme individuel adapté pour les SIRS, Revue de géomatique, 3 (3), September 1993.

[Cattell 97] R.G.G. Cattel, D.K. Barry (Eds.), The Object Database Standard: ODMG 2.0, Morgan Kaufmann, 1997.

[Champoux 92] P. Champoux, Notions fondamentales d'analyse spatiale et d'opérateurs spatiaux, Revue des sciences de l´Information Géographique et de l'Analyse Spatiale, 2 (2), 1992.

[Chen 76] P.P. Chen, The Entity Relationship Model - Towards a Unified View of Data, ACM Transactions on Database Systems, 1 (1), pp. 9-36, 1976.

[Clementini 93] E. Clementini, P. Di Felice, P. Van Oosterom, A Small Set of Formal Topological Relationships Suitable for End-User Interaction, Proceedings of the Third International Symposium on Advances in Spatial Databases, SSD'93, pp. 277-295. Springer Verlag, Lecture Notes in Computer Science # 692, 1993.

[Crausaz 98] P.A. Crausaz, A. Musy, GESREAU, un outil de gestion des eaux par une modélisation du territoire, Revue Internationale de Géomatique, Vol.7, N°2, June 1997, pp. 127-139

[David 93] B. David, L. Raynal, G. Schorter, GeO2: Why objects in a geographical DBMS ?, Proceedings of the Third International Symposium on Advances in Spatial Databases, SSD'93, pp. 264-276. Springer-Verlag, Lecture Notes in Computer Science # 692, 1993.

[Dennebouy 95] Y. Dennebouy et al, SUPER: Visual interfaces for object + relationship data models, Journal of Visual Languages and Computing, 6 (1), pp. 73-99, 1995.

[Egenhofer 91] M.J. Egenhofer, J. Herring, High-Level Spatial Data Structures for GIS, Geographical Information Systems, D. Maguire, M. Goodchild, D. Rhind (Eds.), Longman, London, 1991.

[Egenhofer 92a] M.J. Egenhofer, J. Herring, Categorizing Binary Topological Relationships between Regions, Lines and Points in Geographic Databases, Department of Survey Engineering, University of Maine, 1992.

[Egenhofer 92b] M.J. Egenhofer, A.U. Franck, Object-Oriented Modeling for GIS, URISA Journal, 4 (2), 1992.

[Gayte 96] O. Gayte, T. Libourel, J.P. Cheylan, S. Lardon, POLLEN, méthode de conception des systèmes d'information sur l'environnement, IARE, Montpellier, France, 1996.

[IEF 88] IEF Information Engineering Facility, A Guide to Information Engineering Using the IEF, Texas Instruments, 1988.

[Keller 97] S.F. Keller, A Presentation Model and a Mapping Language for INTERLIS, Document RFC-1011e, Federal Directorate of Cadastral Surveying, Berne, Switzerland, September 12, 1997

[Nguyen 97] V.H. Nguyen, C. Parent, S. Spaccapietra, Complex Regions in Topological Queries, Proceedings of the International Conference on Spatial Information Theory, COSIT'97, pp. 175-192, Laurel Highlands, Pennsylvania, USA, October 15-18, 1997.

[Pantazis 96] D. Pantazis, J.P. Donnay, La conception de SIG - Méthode et formalisme, Hermes, 1996.

[Spaccapietra 98] S. Spaccapietra, C. Parent, E. Zimanyi, Modeling Time from a Conceptual Perspective, submitted for publication

[Tryfona 97] N. Tryfona, D. Pfoser, T. Hadzilacos, Modeling Behavior of Geographic Objects: An Experience with the Object Modeling Technique, Proceedings of the 9th International Conference on Advanced Information Systems Engineering, CAiSE'97, Barcelone, Spain, Springer-Verlag, 1997.