Conceptual Modeling for Federated GIS over the Web

Christine PARENT, Stefano SPACCAPIETRA, Esteban ZIMANYI

Ecole Polytechnique Fédérale de Lausanne - Laboratoire de Bases de Données
IN-Ecublens, 1015 Lausanne, Switzerland
{Parent, Spaccapietra, Zimanyi}@di.epfl.ch   http://lbdwww.epfl.ch

Abstract. A conceptual model provides the best support for federating heterogeneous datastores into a unified framework and elaborating a global consistent description of all available data. It therefore plays a central role for exchange of information within the coming information society. This paper introduces a conceptual model for applications using spatio-temporal data. We discuss in particular the features which support spatial and temporal modeling. An example of conceptual design is given using a pedology application. Advantages of our approach for database design are assessed through comparison with traditional geographical information systems modeling techniques.

Keywords: conceptual modeling, federated databases, spatial databases, temporal databases, GIS, database design

1. Introduction

The number of applications using spatial or geographic data has been ever-increasing over the last decade. National as well as local governments are faced with the high complexity of long term decision making processes, where ad hoc issues are to be evaluated in the context of more general policies concerning land, population, resources and environment management, just to mention a few factors to be taken into account. The demand is high for such decision support applications based on factual geographical data.

The technical response from the computer science perspective, in terms of application support, is rapidly evolving. On the one hand, the new object-oriented paradigm materializes an approach which substantially improves understandability as well as system functionalities and even performance. On the other hand, new small scale geographical information systems (GIS), often termed desktop GIS, are coming to the market to challenge traditional dinosaur GIS, i.e. huge software systems so complex that only specialists can really use them. It is therefore foreseeable that the GIS market will rapidly evolve to desktop GIS directly used by application-oriented people, e.g. geographers, economists, managers. By the time these new GIS become available, people as well as application programs will be familiar with using the Web to access distant information. This should also be the case for GIS, as the geographical information of relevance to an application is normally spread over a number of different locations and into heterogeneous data stores. Using the Web as a communication channel is likely to be the easiest way to get data from the heterogeneous stores.

However, for this vision to become reality, GIS technology still needs to achieve a substantial progress in terms of interfaces, which includes in particular the data modeling features they are built on. Current user interfaces are mostly based on the form-filling paradigm. While this approach is very simple for users, it implies that only pre-planned interactions (data acquisition, queries, updates) are possible. Such a fixed pattern for data usage is well suited for the development of applications, where these have to be planned in advance, designed, implemented, tested. Rigidity, on the contrary, is not at all well suited for interactions with casual users, whose requests are abruptly determined and call for on the fly execution. All forms of exploratory data investigation, where users navigate through unplanned paths, also need maximum flexibility in their interface to the GIS.

From the data modeling perspective, while the object-oriented approach represents a significant step forward, it still is not the ultimate response users can expect. More than thirty years of experience in database design have clearly shown that user requirements are best satisfied by conceptual modeling tools and formalisms. With respect to implementation oriented modeling techniques as used in relational and object-oriented design, conceptual modeling has two significant advantages: it allows designers to focus on the problem (i.e. the representation of application data and processes) without any concern for technical constraints, and it provides a long lasting result, where implementation oriented models become obsolete as soon as the techniques evolve. Moreover, conceptual models provide the best support for visual user interfaces. Entity-Relationship (ER) [Spaccapietra 92] or Unified Modeling Language [UML 97] diagrams, for instance, allow users to visualize and easily understand the content of the information systems. These diagrams also support direct manipulation techniques so that users can browse the database or express queries and updates without the burden of obeying the difficult syntax of a textual language [Dennebouy 95]. A conceptual model also provides the best support for federating heterogeneous datastores into a unified framework and elaborating a global consistent description of all available data [Saltor 91]. It therefore plays a central role for exchange of information within the coming information society.

While conceptual modeling has received a great lot of attention in the database community, little effort has been put up to now into the development of a conceptual model for GIS applications. Three major design methodologies have been proposed for GIS, all based on the discrete view of space. MODUL-R [Caron 93, Bédard 96] is the oldest proposal. It extends the original ER formalism [Chen 76] with pictograms representing the geometry and temporality of spatio-temporal objects. It inherits from the ER source its strong limitations in data structuring capabilities, as objects with a complex structure cannot directly be represented and have to normalized into flat structures identical to first normal form relations in the relational model. MODUL-R also suffers from a lack of formal definitions. MECOSIG [Pantazis 96] is a recent proposal. Its object oriented data model is very powerful (and complex): a list of more than 30 different spatial object types is proposed with associated topological integrity constraints. Unfortunately, it does not support any kind of spatial relationship, nor any concept for temporal modeling, a feature of uttermost importance for most spatial data applications. POLLEN [Gayte 96] is an object-oriented design methology based on OMT [Rumbaugh 91] for spatio-temporal information systems. The data model supports five predefined classes: point, line, area, time-interval and time-instant from which users' classes will inherit. POLLEN however does not offer a conceptual model, rather a method to implement a spatio-temporal database on an object-oriented DBMS.

This paper proposes a conceptual model for spatio-temporal data, called MADS, which offers an object-based modeling of data structures enriched with spatial features (a rich variety of geometries), explicit description of topological relationships (whose scope has been extended to apply to objects with complex geometries), and temporal specifications. MADS is supported by formal definitions, establishing a theoretical basis to build manipulation operations, and is being implemented as a visual user interface independent from any underlying GIS. We briefly present the main characteristics of MADS in the next section. Section 3 discusses a concrete application to show an example of conceptual modeling with MADS. Section 4 assesses the benefits of conceptual modeling by reporting results from an experimentation. Section 5 concludes the paper.

2. MADS: a Model for Application Data with Spatio-temporal features

As a **conceptual** model, MADS aims at supporting the expression of information requirements independently of technological and otherwise computer related concepts, thus facilitating man-machine communication. This is in contrast with the current situation where users must adapt to the inherent technological constraints of a particular GIS and transform their intuitive specifications until they conform with the underlying system.

In addition, MADS aims at achieving the following general modeling objectives:

- **Completeness (thematic, spatial and temporal information)**: the model must include a set of concepts allowing to describe every type of spatio-temporal application, integrating traditional and spatio-temporal data. Spatial data may be represented through either a continuous view of space –altitude and type-of-soil are typical examples of continuous informations– or a discrete view where spatial objects, e.g. roads, rivers or lakes, are localized in space by their coordinates. Temporal data may include instantaneous events as well as facts lasting over some period of time.
- **soundness**: every concept must rely on a formal definition, to avoid ambiguities due to incomplete or imprecise specifications.
- **user-orientation**: the model must allow easy communication with users, preferably with the support of schema diagrams. It must be understandable with reasonable effort, thanks to a limited number of not too sophisticated concepts.
- **orthogonality**: constructs in the model have to be as independent as possible from each other, to make the model easy to use and easy to implement.
- **implementability**: the model must be directly translatable into logical data models of existing GIS, so that no redesign is needed and the model can be effectively used as the common, pivot data model in a federation of heterogeneous systems.
- **full operationality**: the model must include an associated data manipulation language, so that users can use the data through the same paradigm they use to define the data. In a federation, this manipulation language will act as the common language between the GIS.

The following subsections review the data modeling capabilities of MADS in terms of spatial or temporal features. Process modeling is out of the scope of this paper. As for the conceptual modeling of traditional data structures, MADS supports the nowadays usual set of basic concepts: objects, relationships, attributes, is-a links, aggregation links. As these concepts are well understood, we need not recall their definitions. Let us just express a few remarks on attributes and on aggregation:

- an **attribute** represents a real world property of interest. Both object types and relationship types may be described by attributes. Attributes themselves may be described by component attributes. Attributes are:
  - either complex (composed of other attributes) or simple (intended to bear atomic values).
  - either monovalued (one single value admitted) or multivalued (bearing an unspecified number of values).
  - either mandatory (must be valued in every instance) or optional (may be valued or not).

  The mono/multi valuation and optional/mandatory characteristics of an attribute are expressed using the minimum/maximum cardinality concept. Cardinalities also apply to roles in relationship types, to define how many instances of a relationship type may link an object of the associated type.

  Values for an attribute may be derived from values existing in some other attribute(s) which can belong to the same owner (entity or relationship) or to other objects (objects or relationships). Attributes whose value is automatically computed by the system are called derived attribute. The derivation function may use both computations and navigations through the database.

- an **aggregation link** is a peculiar directed binary relationship whose specific semantics is to express that objects of the first type, called composite objects, correspond to aggregations of objects of the second type, called component objects. Figure 2 shows an example of aggregation on spatial object types. Alternative terms for aggregation in object-oriented models include composition link and part-of relationship.

2.1. Spatial Object Types

Current GIS support either one (or both) of the traditional ways of describing the objects/space relationships: the discrete view (also termed vector view), where the database consists of objects whose location in space may be defined, and the continuous view (also termed raster or field view), where the database consists of space regions over which variables are defined as continuous fields of values.

For purely pragmatic reasons, MADS takes the discrete view, while providing facilities for expressing a continuous view if needed. For the same pragmatic reasons MADS is currently limited to the representation of two-dimensional data. An object whose description includes spatial features is called a spatial object:

- a **spatial object** represents an entity that is perceived as spatial by the application, i.e. that is represented by a point, line, area or by any set of points, lines and/or areas. This spatial description of the object is called its geometry.
- a **spatial object type** is an object type which bears an additional specific characteristic: the spatial type of its instances (point, line, area, ...). Diagrammatically speaking, a spatial object type includes the icon associated to its spatial type.

As shown in Table 1, MADS supports all usual simple spatial types: point, line, oriented line, area. It also supports set types: set of points, set of lines, set of areas (complex areas with holes and islands). Finally, it supports three generic types:

- simple geo, which stands for any simple type. Objects in the population of a simple geo object type may be of any one of the simple spatial types.
- complex geo, which stands for any set type.
- geo, which stands for any type.

All spatial types are abstract data types, i.e. they provide the associated methods to define and manipulate objects of the type. Is-a links among spatial types are illustrated in Figure 1.

Generic types allow users to define spatially heterogeneous object types, where some instances are of a given spatial type and some other instances are of another spatial type. For example, a *River* object type whose spatial type is simple-geo, can have instances (small rivers) described as oriented lines and other instances (large rivers) described as areas. Heterogeneous object types are very much likely to appear in integrated schema of federated GIS, because of the diversity of representations of the same objects in different component databases. From a design methodology perspective, generic types represent a way for users to define the spatial type approximately, whenever at the current design stage they do not know exactly which type they want. For example, using the *geo* type a designer may just denote an object type as being spatial, and leave for a later design step a more precise definition.

2.2. Spatial Relationships

Spatial relationships among spatial objects are an essential part of the information needed by applications managing spatial data. These relationships include: topological relationships (e.g., two countries are neighbors), orientation relationships (e.g., a town lies north of a river), metric relationships (e.g., a town lies at 65 km from a state boundary), aggregation relationships (e.g., a state is composed of a set of counties).

spatial type	**icon**	**dimension**	**definition**
geo		0, 1 or 2	any spatial type
simple geo		0, 1 or 2	any simple spatial type (point, line, oriented line or simple area)
point		0	a point
line		1	any line, whether straight, arc, polyline, closed or not, oriented or not
oriented line		1	any oriented line
simple area		2	any area without holes or islands
complex geo		0, 1 or 2	any set of simple spatial types
point set		0	a set of points
line set		1	a set of lines
oriented line set		1	a set of oriented lines
complex area		2	any area, eventually with holes or islands

Table 1. Spatial types for objects.

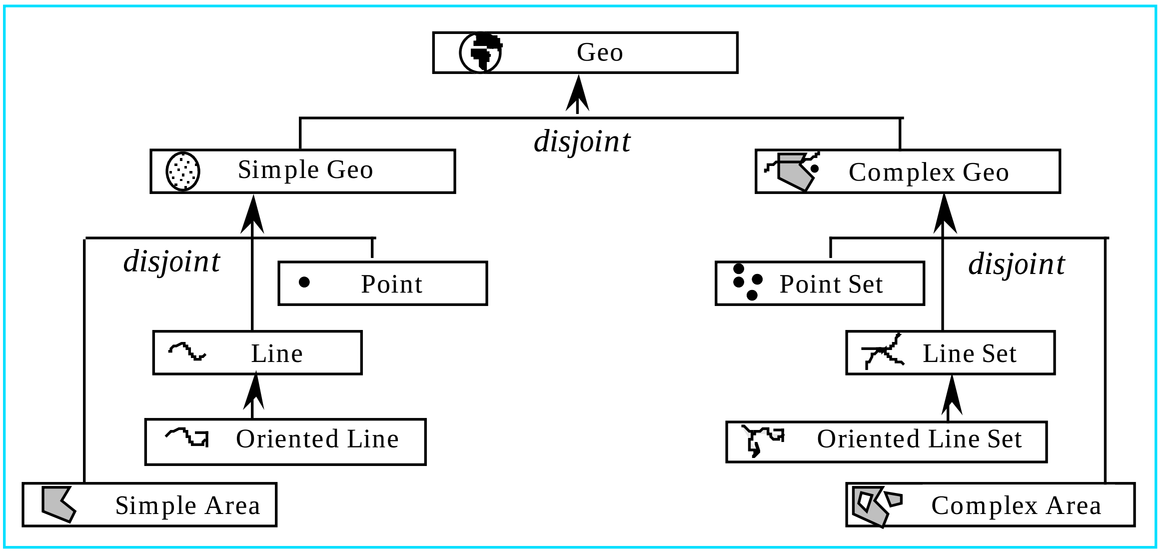

Figure 1: Generalization hierarchy of spatial types for objects.

The current trend in GIS is that information on spatial relationships is automatically computed by the system, on demand, based on the absolute geometry (coordinates) of objects. To GIS users spatial relationships appear as built-in functions. There are, however, several reasons which motivate an explicit description of spatial relationships in a conceptual schema:

- spatial relationships sometimes convey information which is essential to a proper representation and understanding of the data structure of the application. If such is the case, they should appear in the schema. For example, given that a state is composed of a set of counties, it would be misleading for a user to see a conceptual schema in which states and counties are represented as unrelated objects.
- if explicitly defined, spatial relationships can be denoted using semantically meaningful names, e.g., a *crossing* relationship among *route segments*.
- properties can be added to such spatial relationships: attributes, methods, integrity constraints. For example, adjacency constraints on components are common on spatial aggregations (e.g., the route segments composing a route must be connected).
- stating explicitly the spatial relationships allows to verify the coherence and non redundancy of the schema.
- while topological built-in functions provide derived information, explicit spatial relationships define an integrity constraint on the geometry of related objects.

Beyond traditional generic relationships, MADS supports explicit definition for a set of predefined topological relationships. The choice not to include all possible spatial relationships is based on the aim to keep the model reasonably simple.

2.2.1. Topological Relationship Types

As previous theoretical work has shown [Egenhofer 91, 92, Champoux 92], more than a hundred topological relationship types can be defined. This is by far too much for a conceptual model to be used in practice. Fortunately, it has been demonstrated that all these relationships may be meaningfully merged into a few classes [Clementini 93]. Relying on Clementini's classification, MADS supports six elementary topological relationships. These are easy to understand by users and formally defined, based on the concepts of boundary, interior, exterior and dimension. Their definition is given in Table 2.

spatial type	icon	definition
disjunction		no sharing (objects from the related object types must have spatially disjoint geometry)
adjacency		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 boundary (valid for spatial object types of the same dimension)

Table 2 : Spatial types for topological relationships.

2.2.2. Spatial Aggregation

Figure 2 shows an example of aggregation relationship involving spatial objects: a country as an aggregation of districts. Since spatial aggregation is a relationship, cardinalities can be defined for each role (composite, component) and attributes and methods can be attached to the aggregation. As shown in the figure, a country is composed of 10 to 1000 districts and one district is component of exactly one country. A join-date attribute records the date at which the district joined the country.

Spatial aggregation is usually complemented with a spatial constraint forcing a consistency rule between the geometry of the composite object and the geometries of its component objects. If the consistency criterion is stated through an equality, the geometry of the composite object is likely to be a derived attribute. Derivation functions may also link thematic attributes. Usual derivation functions include: sum, average, minimum, maximum, union... For example, area and population of each country are derivable from areas and populations of its districts, the country-name for each district is derivable from the name of its country.

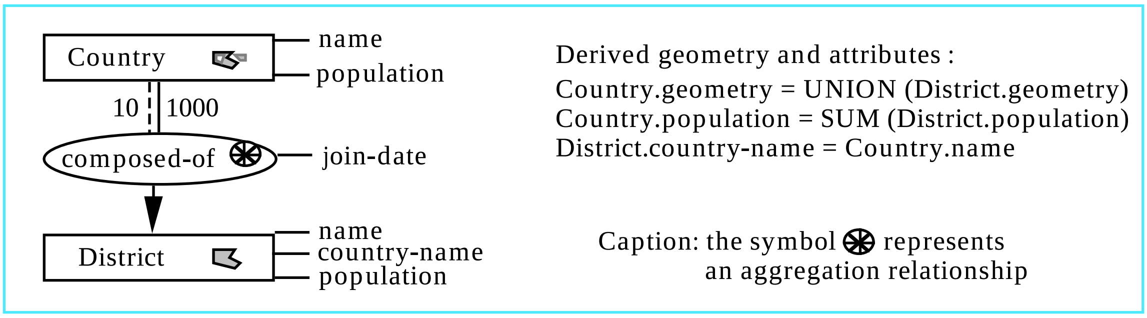

Figure 2: Example of a spatial aggregation.

2.3. Spatial Attributes

Every GIS comes with a set of spatial types which are used to characterize the spatiality of object types. As we have seen, MADS improves over current GIS by allowing generic spatial types for object types and by supporting spatial relationship types as a data model construct. Another important new functionality is provided in MADS: the support of **spatial attributes**. This means that attributes of object/relationship types may have their own spatiality, beyond the one, if any, at the object/relationship level. Thanks to such a facility, spatiality appears to be orthogonal to the choice of a modeling construct to represent a given piece of reality. This gives maximum flexibility to application designers.

One more feature in MADS is the possibility of defining a continuous view ot the part of space covered by either the geometry of a spatial entity or by a spatial attribute, provided that their dimension is not null. This continuous view is provided by the concept of **continuous attribute**, whose value is a function of the location within the geometry of the spatial object. The geometry is partitioned by a user defined grid, which cuts a line into segments and an area into elementary cells. The function maps each segment or cell into a value from the domain associated to the attribute. Continuous attributes are visually marked using a specific icon. MADS solution is similar but more general than the one proposed in [Camara 94]. With respect to GIS, it is more flexible as it allows the designer to choose the spatial domain on which the continuous attribute will be defined.

2.4. Temporal features

When modeling real-world applications designers are likely to be confronted to time-varying information. Recording the temporal evolution of data often provides an interesting insight into the dynamics of real-world phenomena. Over the last years there has been an extensive research effort in the area of temporal databases [Tansel 93, Clifford 95]. Many different models have been proposed, mainly for relational databases, although some work has been done for entity-relationship and object-oriented models. In particular, an extension of the SQL language, called TSQL2, has been proposed as a consensus language by the research community [Snodgrass 95] and similar ongoing work is being realized to add temporality to the forthcoming standard SQL3.

Three complementary ways of recording time have been identified. *Transaction-time* consists of system-generated timestamps recording when a fact was actually stored in the database. Values for *valid-time* are user-provided and represent the actual time when a fact occurs or is valid in the real world. Valid-time may span over the past, the present, and the future. *User-defined time* refers to a data type whose domain codes time values, e.g. a DATE domain, but does not have any temporal semantics for the DBMS. Valid-time and transaction-time are two orthogonal time dimensions over which the data evolves independently. MADS approach to temporal modeling stems from the following principles:

- focus is on valid-time, since it is the most common requirement for geographical applications, but the approach should be easily extensible to transaction-time;

- both snapshot (non temporal) and time-varying (temporal) information should be representable;
- orthogonality: temporality can be attached to each construct of the model, i.e., to objects, relationships, and attributes (at any level);
- consistency rules enforcing a correct semantics of temporality have to be defined;
- temporal facts include both instantaneous events and facts lasting over some period of time;
- adopt TSQL2 recommendations whenever applicable and appropriate.

The *granularity* of a temporal specification defines the time unit to be used to record changes, e.g., hour, day, year. Temporal DBMSs provide the necessary functions to convert a temporal specification from one unit to another one.
A temporal specification either defines an instant or an interval or a temporal element. An *instant* or chronon typically represents the time an event happens. An *interval* is a time slice, typically used to specify when a fact is valid. An interval is usually defined by a start chronon and an end chronon, and contains all chronons in between. A *temporal element* is the union of a set of intervals, intended to represent the set of time slices where a fact is valid. A temporal element may also be a set of non consecutive time instants. This is useful for all kinds of discrete events like measures done by discrete instruments, or photographs ...

MADS temporal features are briefly described hereinafter.

2.4.1. Temporal Attributes

Temporal attributes keep the history of their values. For example, if *rate of flow* is defined as a temporal attribute in a *River* object type, for each river object the DBMS will maintain the set of all values of *rate of flow* which have been (or are planned to be) valid at some time. This may be seen in general as the association of a temporal element to each value of a temporal attribute. The set of these temporal elements can be contiguous or disjoint, but they cannot overlap. For example, assuming that the granularity for *rate of flow* is the day, it could be specified that *rate of flow* of a given river had value 1300m^3/second from April 4, 1996 to April 12, 1996 and from June 18, 1996 to June 22, 1996. The value of an attribute is assumed to be undefined (unknown) for the points in time not explicitly specified by the set of temporal elements.

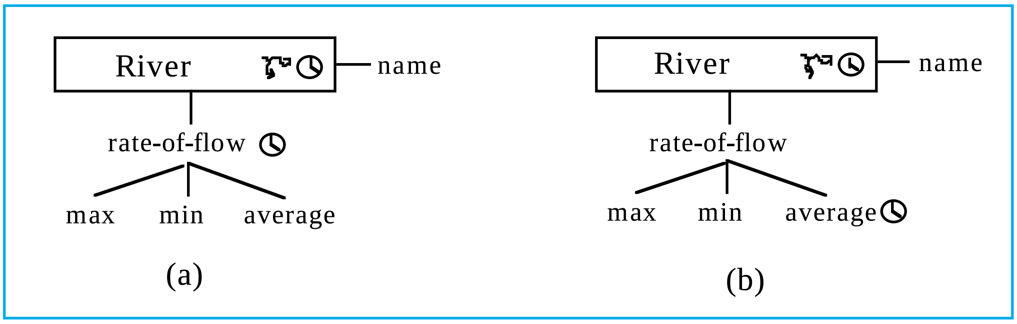

Figure 3: temporal attributes.

As shown in Figure 3, the spatiality of an object type can also be temporal. This means that, in particular, the geometry of objects may vary in time and its history be recorded.
Figure 3 also shows that, since attributes can be complex, i.e. composed of other attributes, the temporality can be defined at any level of the composition. For example, *rate of flow* is defined in Figure 3(a) as a temporal complex attribute composed of *minimal rate*, *maximal rate*, and *average rate*. Thus, the value of *rate of flow* will be a set of triples (minimal, maximal and average rates). Each triple bears a temporal element defining the time periods at which the triple is valid. On the contrary, if only the component attribute *average rate* is temporal, as in Figure 3(b), then the value of *rate of flow* will contain one value for *minimal rate*, one for *maximal rate* and a set of couples (value, temporal element) for *average rate*.

2.4.2. Temporal Objects

When associated to object/relationship types, temporality has not to do with values but has to do with the existence of the object/relationship instances within this type. Indeed, objects are created as instances of an object type, but can migrate to another object type, be temporarily suspended as instances of this object type, be resumed in their membership of the type, and eventually be deleted. Relationship instances can be created, suspended, resumed and deleted.
Defining an object type as temporal instructs the DBMS to keep track of the lifecycle of its instances, as defined by the events (creation, suspension, deletion) happening during their lifespan. This information, together with object values, remains available even after the deletion of the object. To that extent, each temporal object is associated with a temporal element stating the validity of the object as member of the type. For example, specifying an object type *Flood* as temporal may be used as a way to store the information about when each flood occured.

The definition of an object type as temporal is orthogonal to defining some of its attributes as temporal. For a temporal object having no temporal attributes, its lifespan is stored as well as one value (the last one) per attribute. On the other hand, for a non temporal object with temporal attributes, its lifespan is not recorded but the history of attribute values is recorded for each temporal attribute, and this history is kept up to the deletion of the object. Whenever a temporal object type has temporal attributes, it is usually suggested that the temporal elements associated to attribute values must be included in the lifespan of the object they belong to. For example, the valid time associated to attributes values for a flood (e.g., rate of flow) must be included in the lifespan of the flood.

2.4.3. Temporal Relationships

As for object types, temporality of relationship types allows to keep track of the lifecycle of its relationship instances. Assume, for example, a temporal relationship type *IsInCharge* linking the *RiverSegment* and *LegalEntity* object types. Temporality of the relationship type implies that we want to keep track of past, present, and future couples of (river segment, legal entity), e.g., to be able to determine at any point in time who was responsible for each river segment.
Notice that a temporal relationship type can only link temporal object types, since otherwise there might be dangling references, i.e., a relationship linking objects for which no information is anymore kept in the database. In addition, the lifespan of a relationship must be included in the intersection of the lifespans of the participating objects.

3. A Pedology Application in MADS

This section provides an example of application modeling with MADS. The application for which a spatio-temporal schema was needed concerns the study of the evolution of alluvial soils in a Swiss floodplain ecosystem. To capture the present spatial distribution of soils, their diversity and their degree of evolution, a soil survey was realised on the whole site. This soil survey was based on 277 points of observation where a number of soil parameters were recorded and samples taken for further analysis. Clustering analysis have subsequently been used to group observation points into similarity groups. Observation points representing each similarity group were described with more details: their complete soil profiles have been physically and chemically analysed.
Altogether, the input data consists in:

- a set of orthogonal aerial photos of the river taken in 1930, ‘43, ‘55, ‘69, ‘80 and ‘92
- a corresponding set of maps of land cover derived from the aerial photos
- a 1988 vegetation map at the scale 1:10,000
- data about the rate of flow of the river
- data about the floods (dates, duration, rate of flows)
- data about the projects of embankments, as well as a map of the different embankments realised along the century
- data about soil properties (pedological analysis), soil profiles and laboratory analysis, for the field survey points.

Figure 5 contains a simplified MADS schema of the application. Attributes are omitted. The object types definitions are:

- River: a stream of water. Since the river channel has evolved over time, and past locations of the channel have to be recorded, the geometry of a river is defined as temporal.
- Embankment: a bank of stone inside a metallic frame, constructed to protect the banks of a river against flood. Embankments may be temporarily out of service. To record periods of activity, embankments are temporal objects. However, as their geometry only undergoes minor changes, only the current geometry is stored.
- Flood: an overflow of river waters onto land. It is defined as temporal to keep the information about the time period where the floods existed.
- Vegetation-Unit-1988: a qualitative class of vegetation defined in 1988 by a field survey, e.g., water, sediments, etc. The precise area covered by a vegetation unit has to be known. Hence, vegetation units are spatial objects. As they have been established once and never updated, they are not temporal.
- Land-Cover-Unit: a qualitative class of land cover defined by interpretation of aerial photographs, e.g. water, alluvial vegetation, etc. Land cover units are less detailed than the vegetation units. Their precise area and its evolution in time is needed. Hence, Land cover units are spatio-temporal objects.
- Soil-Survey-Point: spatial locations (i.e. points, defined by their coordinates and altitude) at which soil samples were extracted by means of a drill and analysed.
- Transect: a set of soil survey points whose locations approximately form a line. In practice, the surveyor first determines a transect of interest and then chooses survey points on the transect. Transects have no temporal feature.
- Soil-Profile-Point: soil survey points for which the analysis were more complete.
- Horizon: a layer of soil material approximately parallel to the land surface and differing from adjacent layers in physical, chemical, and biological properties. At each survey point parameters corresponding to several horizons are collected. A horizon is in fact a set of values for the thematic attributes representing the parameters. It is neither temporal nor spatial, as the location of the point where the values were collected is known through the aggregation relationship linking component horizons to their composite soil survey object.
- Soil-Group: a cluster of soils computed by a clustering analysis resulting in a hierarchical classification of all soil survey points.

Aggregations and topological relationships illustrated in Figure 5 are self-explanatory.

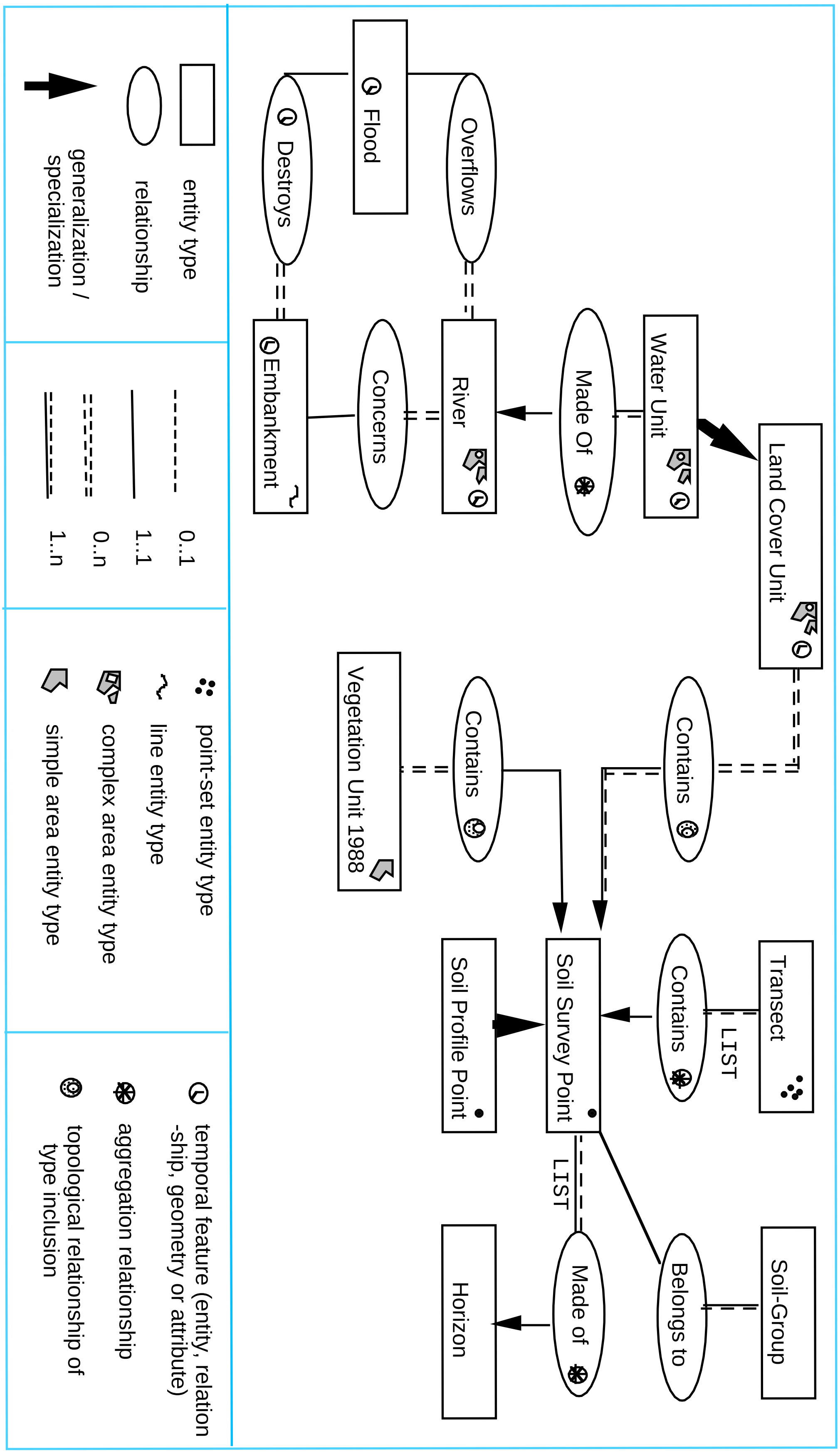

Figure 5: a simplified MADS schema for the pedology application.

4. Experimental Results

By now, MADS has been used to model several applications, including the one reported in the previous section. In one case MADS has been used to redesign an existing application, thus providing an easy and realistic way to compare different modeling techniques. The testbed application, known as the GESREAU project, supports the management of water resources of the Vaud county in Switzerland. GESREAU was choosen as a test application because of its complexity. The operational version had been modeled using the IE methodology [IEF 88], an entity-relationship approach by Texas Instruments Software. The result of the remodeling task is evident through comparison of the following figures. The GESREAU schema in IE has 85 entity types, grouped into 9 main subjects, and linked by 74 relationship types. The corresponding schema in MADS has 66 entity types (minus 22%) and 57 relationship types (minus 23%).

A closer analysis of such experimental results pointed out several reasons which make MADS more effective than traditional models in current GIS. A well-known reason is that most GIS describe the world using simple first normal form structures, and thus real-world objects with complex structures must be fragmented into several simple non-structured objects. With MADS (as with recent object-oriented models) a real-world entity is directly described as a single database object. Replacing many fragments with a single object type enhances the lisibility of the schema and provides a view closer to real-world perception.
The following MADS features participate in the simplification of the resulting schema:

- hierarchy of complex attributes: every object/relationship type bearing many attributes benefits from the possibility to structure the set of attributes into groups of semantically related attributes. For example, in GESREAU the entity type *Wildlife-Conservation-Branch* bears 38 own attributes, in addition to those inherited through generalization. In the MADS schema, these 38 attributes were structured into 12 complex attributes.
- direct representation of multivalued attributes, also a well-known problem of relational-based systems. It allows to delete all artificial types created to normalize multivaluation.
- n-ary relationship types, possibly with attributes.
- explicit specification of the spatiality of data, which makes useless to have object types merely representing spatial types. In addition, the schema gives thus information which is useful to the user for understanding it. This information is also crucial for the translation modules generating the schemas for the target GIS on which the spatial database is implemented.
- a conceptual model allows the designer to abstract from the many restrictions dictated by GIS for the description of spatiality, like areas must form a partition of space; areas cannot averlap each other; the geometry of all instances of the same object type must be of the same type ...
- the organization of spatial types in a hierarchy is very useful each time the spatial type is not well-known or variable. For example, in GESREAU the entity type *Waterway-Cross-Over-Section* has two subentities of spatial type line and point, respectively. In MADS the spatiality of the super-object type is geo, which is the spatial type generalizing line and point.
- the availibility of spatial relationships: in GESREAU 2/3 of the relationships are spatial (adjacency, covering, ...). In particular, the spatial aggregation at the conceptual level turns out to be very useful, since implementing aggregation in current GIS introduces implementation constraints which are not revelevant for the modeler.
- continuous attributes are very useful for representing at a conceptual level data which is usually represented in raster mode on GIS. For example, a continuous attribute *Land-Covering* of object type *Catchment* allows to describe for each point of the surface its land cover unit, e.g., forest, agriculture, or urban.
- temporal information is very frequent in geographical applications. The temporal features of MADS allow to describe time-varying information at a conceptual level, independently of how this information is implemented. In GESREAU many attributes describe temporal measures: rainfall, rates of flow of waterways, height of water in water basins ...

In summary, the main advantage of MADS for complex applications is that it gives a modeling perspective closer to the real world, in particular due to the possibility to model explicitly spatial and temporal aspects.

In the context of a federation of GIS in which several juridical entities collaborate for the management of some geographical data, as was the case in the GESREAU project, each of these juridical entities is responsible for the management of its geographical data, according to its scope of activity. However, the information maintained in one GIS must be made available to the others juridical entities. Thus, a global schema at the conceptual level must be used to abstract away the particularities and restrictions of each individual GIS. MADS as a conceptual language provides rich and powerful mechanisms for this task.

5. Conclusion

This paper presented the MADS conceptual model for describing spatio-temporal data. Thanks to its powerful constructs for the description of thematic, spatial and temporal features, MADS has proven to be very efficient for designing complex applications. It represents a significant step forward compared to current GIS modeling capabilities. MADS provides orthogonal fonctionalities including:

- for thematic description:
  - complex and composite objects
  - n-ary relationships with attributes
  - complex and multivalued attributes
- for spatial description:
  - a hierarchy of spatial types
  - explicit spatial topological relationships and spatial aggregation

spatial attributes
continuous attributes for field-oriented description
• for temporal description:
temporal objects, relationships, and attributes
instantaneous and interval temporal types

MADS concepts are formally defined, which provides a sound basis for further developments, in particular regarding query and update languages. Nevertheless, its goals are also very pragmatic, focusing on:

• user-orientation: this is very important in a federation of GIS since each user must be able to understand and interact with the global conceptual schema as well as with the local GIS schema.
• implementability: all MADS features can be supported using appropriate mappings onto the functionalities provided by the current generation of GIS.

A visual interface layer implementing MADS is being developed in a UNIX environment using Java. This layer includes:

• a visual schema editor allowing the user to draw its schema on the screen.
• a set of translators for MADS schemas into GIS schemas such as ArcInfo, Interlis (a Swiss geographical data exchange norm), or IEF.
• an import/export facility for the exchange of textual definitions.
• a tool for automatic generation of HTML documentation for MADS schemas.
• a visual query language mapping to its underlying algebra.

Research work in progress focuses on multiscale representations (based on multi-instanciation), spatial integrity constraints, temporal imprecise specifications, and the visual query language facilities.

Acknowledgements

We deeply thank Lou Mendonça for making available the pedology application. We appreciated in particular her patience for discussing with us and answering all our questions. Similar thanks extend to Pierre-André Crausaz for the GESREAU application.

References

[Bédard 96] Y. Bédard, C. Caron, Z. Maamar, B. Moulin, D. Vallière, *Adapting data models for the design of spatio-temporal databases*, in Computer, Environment and Urban Systems, Vol. 20, N°1, 1996

[Camara 94] G. Camara, et al., *A model to cultivate objects and manipulate fields*, Proc. of the 2nd ACM Workshop on Advances in Geographic Information Systems, N. Pissinou, K. Malki, Eds., Gaithersburg, Maryland, USA, December 1-2, 1994

[Caron 93] C. Caron, Y. Bédard, P. Gagnon, *MODUL-R: un formalisme Individuel adapté pour les SIRS*, Revue de Géomatique, Vol.3, N°3, septembre 1993

[Champoux 92] P. Champoux, *Notions fondamentales d'analyse spatiale et d'opérateurs spatiaux*, in Revue des Sciences de l'Information Géographique et de l'Analyse Spatiale, Vol.2, n°2, 1992

[Chen 76] P.P. Chen, The *Entity-Relationship Model - Towards a unified view of data*, ACM Transactions on Database Systems, Vol.1, N°1, 1976

[Clementini 93] E. Clementini, P. Di Felice, P. van Oosterom, *A Small Set of Formal Topological Relationships Suitable for End-User Interaction*, in SSD'93, LNCS 692, Springer Verlag, 1993

[Clifford 95] J. Clifford and A. Tuzhilin (Eds) *Recent Advances in Temporal Databases.* Springer, 1995.

[Dennebouy 95] Y. Dennebouy, et al., *SUPER: Visual Interfaces for Object + Relationship Data Models*, Journal of Visual Languages and Computing, Special Issue on Visual Query Languages, Vol.5, 1995

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

[Egenhofer 92] M. J. Egenhofer, J. Herring, *Categorizing Binary Topological Relationships between Regions, Lines and Points in Geographic Databases*, Department of Survey Engineering, Uni. of Maine, 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 (in French)

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

[Rumbaugh 91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object oriented modeling and design, Prentice Hall, 1991

[Pantazis 96] D. Pantazis, J.P. Donnay, *La conception de SIG Méthode et formalisme*, Hermes, 1996 (in French)

[Saltor 91] F. Saltor, M.G. Castellanos, M. Garcia-Solaco, *Suitability of Data Models as Canonical Models for Federated Databases*, in ACM SIGMOD Record, Vol. 20, No. 4, 1991

[Snodgrass 95] R.T. Snodgrass (Ed): *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995

[Spaccapietra 92] S. Spaccapietra, C. Parent, *ERC+: An Object based Entity Relationship Approach*, in Conceptual Modelling, Database and Case: An integrated View of Information Systems Development, P. Loucopoulos and R. Zicari Eds., John Wiley, 1992

[Tansel 93] A.U. Tansel et al: *Temporal Databases, Theory, Design and Implementation*. Benjamin/ Cummings Publishers 1993

[UML 97] *UML Document Set*. Rational Corporation, available at http://www.rational.com/uml/references

Appendix

Pedology Schema

SPATIAL-ENTITY Transect POINT-SET**:**
T.id: STRING
lengh: REAL
reference-mark: POINT
/ reference-mark is a spatial attribute of domain POINT/
azimuth: REAL
remarks: STRING
Integrity constraint: the points composing each transect must be roughly in line

SPATIAL-ENTITY Soil-Survey-Point POINT**:**
S.id: STRING
local-geomorphology: (date: DATE, type: STRING, source-map: STRING)
local-vegetation: STRING
altitude: REAL
nb-of-horizons: INTEGER
depth: REAL
remarks: STRING

SPATIAL-ENTITY Soil-Profile-Point ISA Soil-Survey-Point**:**
/ Soil-Profile-Point inherits its geometry (a POINT) from Soil-Survey-Point /
humus: STRING
soil-type: ENUM= (simple-fluviosol, typical-fluviosol, polyphase-fluviosol, brown-fluviosol)

SPATIAL-ENTITY Vegetation-Unit-1988 AREA**:**
VU.id: STRING
type: ENUM= (water, construction, cultivation, meadow, pine forest, alder forest, ...)
remarks: STRING
Integrity constraints: the geometry of the instances of Vegetation-Unit-1988 form a partition of the space; the instances can have holes, but cannot have islands.

SPATIAL-ENTITY Land-Cover-Unit INSTANT COMPLEX-AREA **:**
/ Land-Cover-Unit is spatial but not temporal. However, its geometry is temporal (of type INSTANT) /
LC.id: STRING
type: ENUM= (water, peebles, shrub, alluvial vegetation, no alluvial vegetation, grassland, construction)
remarks: STRING
Integrity constraints: the geometry of the instances of Land-Cover-Unit of the same date form a partition of the space; for each date and each value of type, there is at most one instance of Land-Cover-Unit.

SPATIAL-ENTITY Water-Unit ISA Land-Cover-Unit **:**
/ Water-Unit inherits its geometry (a temporal COMPLEX-AREA) from Land-Cover-Unit /
/ As Land-Cover-Unit, the geometry of Water-Unit is temporal /
Integrity constraint: for every instance, the value of type is "water".

SPATIAL-ENTITY River INSTANT COMPLEX-AREA **:**
/ River is spatial, but not temporal. However, its geometry is temporal (of type INSTANT) : a set of geometry values will be recorded for different instants/
name: STRING
rate-of-flow : INSTANT (min : REAL, average : REAL, max : REAL)
/ rate-of-flow is a complex temporal attribute: a set of three values (min, average and max) will be recorded for different instants /
historic-max-flow: INSTANT REAL
/ historic-max-flow is a temporal attribute: a set of rates will be recorded for different instants /
catchment-capacity: REAL
geological-substratum: STRING
remarks: STRING

SPATIO-TEMPORAL-ENTITY Embankement INTERVAL**,** LINE **:**
E.id: STRING
type: ENUM= (longitudinal, transversal)
construct-date : DATE
repair-dates (0:N) : DATE
*/repair-dates is a non temporal multivalued attribute: a set of values will be recorded; but their temporal meaning will have to be managed by the application */
remarks: STRING

TEMPORAL-ENTITY Flood INTERVAL**:**
rate-of-flow : INSTANT (max-flow : REAL, return-period : INTEGER)
/ rate-of-flow is a complex temporal attribute: a set of values (max-flow and return-period) will be recorded for different instants /
remarks: STRING

ENTITY Horizon :
H.id: STRING,
name: STRING,
type: ENUM= (A, Jp, Js, C, D, M)
depth: REAL
color: STRING

HCl: ENUM = (null, little, medium, strong)

ENTITY Soil-Group :
SG.id: STRING
nb-of-horizons: INTEGER
depth: (min: REAL, max: REAL)
organic-matter: (min: REAL, max: REAL)

AGGREGATION Contains :
COMPOSITE: Transect (1:N, LIST)
COMPONENT: Soil-Survey-Point (1:1)

RELATIONSHIP Belongs-to :
ROLE: Soil-Group (1:N)
ROLE: Soil-Survey-Point (1:1)

AGGREGATION Made-of :
COMPOSITE: Soil-Survey-Point (1:N, LIST)
COMPONENT: Horizon (1:1)

TOPOLOGICAL RELATIONSHIP Contains INCLUSION :
CONTAINER: Land-Cover-Unit (0:N)
CONTAINED: Soil-Survey-Point (1:N)

TOPOLOGICAL RELATIONSHIP Contains INCLUSION :
CONTAINER: Vegetation-Unit (0:N)
CONTAINED: Soil-Survey-Point (1:1)

AGGREGATION Made-of :
COMPOSITE: Water-Unit (1:N)
COMPONENT: River (1:1)

RELATIONSHIP Concerns :
ROLE: River (0:N)
ROLE: Embankement (1:1)

RELATIONSHIP Overflows :
ROLE: River (0:N)
ROLE: Flood (1:1)

TEMPORAL-RELATIONSHIP Destroys :
/ As Destroys is temporal, its instances will be kept forever /
ROLE: Flood (0:N)
ROLE: Embankement (0:N)