Spatio-Temporal Conceptual Models: Data Structures + Space + Time

Christine Parent
HEC/INFORGE
Université de Lausanne
1015 Lausanne, Switzerland
Tel. +41 21 693 52 11
christine@lbd.epfl.ch

Stefano Spaccapietra
Database Laboratory
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland
Tel. +41 21 693 52 10
Stefano.Spaccapietra@epfl.ch

Esteban Zimányi
Department of Informatics
Université Libre de Bruxelles
1050 Bruxelles, Belgium
Tel. +32 2 650 31 85
ezimanyi@ulb.ac.be

ABSTRACT

Nowadays, many applications need data modeling facilities for the description of complex objects with spatial and/or temporal facilities. Responses to such requirements may be found in Geographic Information Systems (GIS), in some DBMS, or in the research literature. However, most of existing models cover only partly the requirements (they address either spatial or temporal modeling), and most are at the logical level, hence not well suited for database design. This paper proposes a spatio-temporal modeling approach at the conceptual level, called MADS. The proposal stems from the identification of the criteria to be met for a conceptual model. It is advocated that orthogonality is the key issue for achieving a powerful and intuitive conceptual model. Thus, the proposal focuses on highlighting similarities in the modeling of space and time, which enhance readability and understandability of the model.

Keywords

Conceptual Modeling, Spatial Databases, Temporal Databases, Spatio-temporal Databases, Geographic Information Systems.

1. INTRODUCTION

Nowadays, data management software and tools need more sophisticated facilities to face new requirements from emerging application areas and non-traditional user interactions. In particular, better concepts and tools for manipulating spatio-temporal data are needed. Major DBMS tools are incorporating facilities for spatial or temporal data management (e.g., Oracle's Spatial Cartridge and Informix's Datablades). The GIS scene is also evolving from the huge and uncomfortable old-fashioned systems to more flexible desktop systems based on modern technology (including the object-oriented approach). Temporal systems are still somehow behind, with no generic product on the marketplace, just a few ad hoc systems or application-specific developments (e.g., for time series management).

However, current tools do not match the user perception of and reasoning about the application data. Put into traditional database terms, there is a mismatch between the logical, implementation-oriented view of data supported by the tools, and the application-oriented, conceptual view that users follow in their everyday work. This mismatch is similar to that of traditional database management many years ago, when the market favored the relational approach and the conceptual to logical gap was filled by database design CASE tools based on the entity-relationship (ER) approach. Since then, the advantages of the conceptual approach to data modeling have been extensively demonstrated, in terms of user involvement and of durability of the design specifications. It is thus foreseeable that a similar evolution will lead spatio-temporal data management tools to be complemented with user-oriented design CASE tools. The problem is that nowadays there is no agreed-upon conceptual model on which to build such tools. A plethora of models have been proposed for either spatial or temporal modeling (a few for spatio-temporal modeling), but they fail to show a clean conceptual underlying philosophy. This was, for instance, explicitly identified by a group of experts in temporal databases, who reported that “the time-varying semantics is obscured in the representation schemes by other considerations of presentation and implementation”, which lead the authors to “advocate a separation of concerns, i.e., adopting a very simple *conceptual* data model” [22].

This paper reports on the design of a spatio-temporal conceptual model, called MADS (for Modeling of Application Data with Spatio-temporal features). MADS currently supports spatio-temporal features needed by the applications we have been involved with, mainly related to land management or utility networks. MADS was designed in cooperation with GIS database designers, and its usability was verified through several successful case studies. A water management case study allowed to quantify the benefits of using MADS with respect to a traditional ER model. Implementation of MADS in operational environments has shown that it can be used as a front-end to existing systems. A visual schema editor allowing the definition of MADS schemas through direct manipulation on a screen has been implemented. A companion visual query language is being investigated.

The next section discusses the criteria that we identified as guidelines for building the conceptual model. Existing proposals are briefly positioned against these criteria. The following sections discuss issues for defining spatial and temporal features in a conceptual model. Section 3 discusses the abstract data types supporting space and time description. Section 4 discusses complex data structures. Sections 5 and 6 are devoted to the definition of object and relationship types, respectively. Section 7 highlights some important issues about constraints. An example of MADS schema is shown in Section 8. Finally, Section 9 concludes by pointing at work in progress within a global spatio-temporal framework. The formal definition of MADS and its syntax are available in a separate report [7].

2. CRITERIA AND RELATED WORK

Several comparisons of spatio-temporal models in the literature typically intend to assess the quality of a model with respect to its competitors. Thus, arguments and conclusions vary from one paper to the next one. A few, more generic comparisons are also available (see, for instance, the analysis of object-oriented temporal languages in [24]). Significant work is reported in [12,13], where ten temporal entity-relationship models are analyzed and classified according to 19 design criteria. Most of these criteria are specific to the temporal dimension. We are interested in the more generic question: what characterizes a good conceptual model for spatio-temporal databases?

Some well-known answers come from past experience in traditional database design. For instance, the word **conceptual** refers to the capability of providing a direct (i.e., with no distortion) mapping between the perceived real world and its representation. Examples of distortions in the proposed spatio-temporal models include unnecessary restrictions due to poor data structuring capabilities (e.g., binary relationships only, no multivalued attributes). Several models (e.g., GeoOM [25] and POLLEN [11]) represent spatial or temporal features through space or time object types (e.g., Point, Line,..., Instant, Time-Interval,...). Such artificial object types do not represent real-world items of interest, thus contradicting the very first rule in conceptual modeling. To avoid distortion, the model has to provide **powerful** constructs. The current standard in this respect (as represented by, e.g., UML [4]) is the support of object types, relationship types, multivalued attributes, complex attributes (i.e., attributes composed of other attributes), is-a links, aggregation (part-of) links, and the associated integrity constraints. Many more features could be thought of, but experience has shown that striving for the highest expressive power leads to unbearable complexity and eventually results in rejection of the model. Also, models with too sophisticated constructs or with constructs having non-standard semantics are likely to be discarded by users. So, **simplicity** is the next important criterion, which also applies to the **visual** notations that support the drawing of schema diagrams. Some models do not provide visual notations [5, 11, 18, 25], while others have non-intuitive notations [10]. Of course, a conceptual model should rely on a sound, **formal** definition and several proposals lack such a sound background. Finally, an associated **data manipulation language** will allow using the same model for both data description and manipulation.

The new, fundamental issue of spatio-temporal models is how the space and time dimensions are added to the model. In our opinion, **orthogonality** is the only correct criterion. This concept refers to the necessary independence among the modeling dimensions: data structures, space, and time. Database designers should be able to determine the most appropriate data structures for an application, without taking into account which information items bear spatial or temporal information. Once the data structure is fixed, space and time features should be freely added whenever appropriate. This approach makes also easier the addition of space or time features to legacy databases, that usually do not contain explicit spatio-temporal specifications. Most of current spatial models lack orthogonality, forcing the designer to a specific representation for spatial information. For instance, GeoOM [25], POLLEN [11], MODUL-R [2], and CONGOO [19], only support the association of space to object types. Thus, an object type cannot include attributes representing spatial information, such as the reservoir attribute in the example shown Figure 3. Conversely, GéO2 [5] has no concept of a spatial object, only attributes may be spatial. From the definition of an object type with several spatial attributes (e.g., Watershed in Figure 3), it is not possible to determine which one, if any, represents the spatiality of the objects. Similarly, not every temporal model supports the association of time with objects, relationships, and attributes. For instance, in their support of temporal object types, [14] ignores the facilities offered by traditional inheritance with redefinition, thus illustrating another way of not achieving orthogonality. Beyond orthogonality, **simplicity** of the model, as well as easiness of use and understanding, is greatly enhanced if modeling spatial and temporal features relies on similar reasoning backgrounds.

Another question is the **comprehensiveness of** a spatio-temporal model, i.e., how many of the perceived space and time phenomena should be directly representable in the model. This concerns issues such as supporting 2.5 or 3 dimensions for spatiality, or supporting valid time or transaction time for temporality. Many such issues may be discussed without coming to a consensus. The more is not necessarily the best, as already stated. Extensible models look like the best answer, but evaluating the extensibility of a model is not an easy task. Again, orthogonality proves to be an important quality as having orthogonal dimensions limits the impact of the addition of a new concept in the model. There is, however, a point in comprehensiveness that we want to stress. Because we believe that conceptual models should explicitly support objects, relationships and attributes, and because we strongly insist on orthogonality, we want to have spatio-temporal objects, spatio-temporal relationships, and spatio-temporal attributes. Current proposals lack of support for spatio-temporal relationships. These relationships have been neglected, relying on the underlying GIS to compute the actual (mostly topological) relationships from objects' coordinates stored in the database. An evident drawback of such an approach is that no property can be attached to a spatial relationship, which contradicts the expressive power expected in the structural dimension.

In conclusion, none of the spatial and temporal models we have examined satisfies all of the above goals. This prompted the development of MADS, an object+relationship conceptual model. The major originality of MADS lies in that it is indeed a conceptual model. Still, it has been implemented through translators converting MADS specifications into operational database models. For instance, a mapping of MADS temporal constructs to TSQL2 [23] has been described in [20]. The MADS model currently covers requirements for traditional land management applications: one or two-dimensional spatial data, priority to the discrete object view versus the continuous-field view, valid time only. Current work on MADS includes an extension to transaction time and multiscale databases.

3. ADTs FOR SPACE AND TIME

MADS structural dimension includes well-known features such as objects, attributes (mono-/multivalued, simple/ complex, derived), methods, integrity constraints, n-ary relationships, is-a links, and aggregation links. As already stated, a clean conceptual solution for supporting space and time features is to provide adequate abstract data types (ADTs). Spatial ADTs (SADTs) provide for the shape and location information; they include representation of points, lines, and simple areas (with holes but without islands). Temporal ADTs (TADTs) support timestamping, i.e., associating a timeframe to a fact; they include representation of instants, intervals, and temporal elements. These ADTs take into account the spatial granularity (i.e., the scale) and the temporal granularity (i.e., second, minute, hour, day, …). Appropriate functions allow the conversion from one granularity to another.

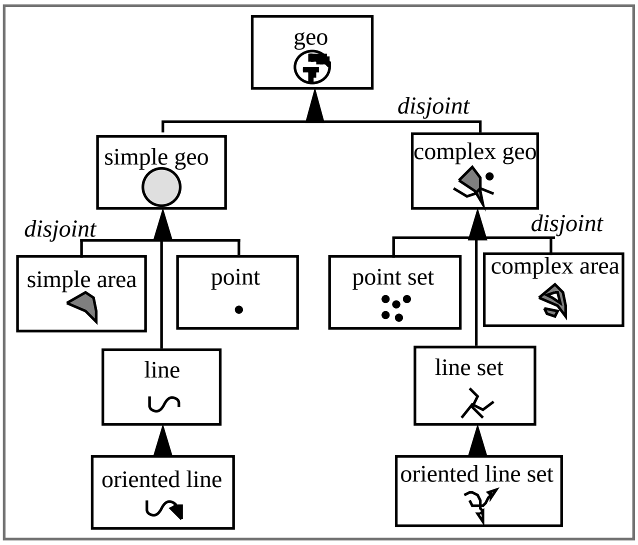


Figure 1: MADS basic hierarchy of spatial abstract data types.

Spatial and temporal ADTs may be implemented as generalization hierarchies. Figure 1 shows our spatial ADT hierarchy (more details in [21]). MADS uses set ADTs (sets of points, sets of lines, complex areas), as well as generic types (e.g., simple geo, complex geo, geo). Figure 1 also shows the icons denoting each SADT. The situation is similar for temporal types. The hierarchies of spatial and temporal ADTs may be extended with user-defined ADTs as appropriate.

Generic ADTs are particularly useful to deal with spatial and temporal features whose instances may have different types. Figure 2c shows that, since Town objects may be represented as points or areas depending on the subtype they belong to, the simple geo SADT has been associated with the generic object type Town. Similarly, lifespans of natural phenomena may be instants or intervals.

The above ADTs support the so-called discrete (or object-based) view of space and time. However, more concepts are needed. First, from the application point of view, space is usually not infinite and time of interest may be limited to a given span. These limits must be known to check the validity of the data or to infer correct space/time extents associated to the data. In MADS the DBSPACE and DBTIME parameters define, respectively, the area and the time frame described by the database.

Applications also need to express that some information varies in space and/or in time. An example of *time-varying* information is to keep the evolution of salaries of employees. Storing the values of altitude and soil cover over a given geographic area (cf. Figure 3) is an example of *space-varying* information. This concept corresponds to the so-called continuous (or field-based) view of space. Finally, keeping track of rainfall values in a given area over time is an example of information varying in both space and time, simultaneously. A purely conceptual representation of time/space-varying information is as functions from time/space to value domains, e.g., soil cover is a function from an area to the domain {field, forest, urban, …}. The choice of the sampling technique (e.g., regular grid, TIN) and of the interpolation functions is a concern of the logical schema definition.

Any attribute can be time-varying. Specifying a spatial attribute as time-varying allows to describe objects that move and change their shape. Figure 2c) illustrates this: each GPS-Car object has a time-varying geometry whose semantics is that the position of the car is recorded at each instant.

To provide domains for varying information, MADS defines two sets: SPACEZONES and TIMEZONES. SPACEZONES is the set of all spatial elements (i.e., simple areas, complex areas, lines, and line sets) contained in DBSPACE. TIMEZONES is the set of all temporal elements (i.e., intervals, sets of disjoint intervals, sets of instants, heterogeneous sets of disjoint intervals and instants) contained in DBTIME. It contains also the special chronon *now* representing the current time. The next section shows how these sets are used in the definition of attributes.

Spatial and temporal ADTs support the definition of domains for attribute values. To enforce orthogonality, it should also be possible to associate space and time to object and relationship types, for expressing an inherent spatiality or temporality that does not rely on their semantic attributes. As other models, MADS associates a system-generated attribute to object and relationship types specified as spatial or temporal. The spatiality of an object or relationship type is described by a predefined attribute, *geometry*, whose value domain is a SADT. Similarly, temporality associated to an object or relationship type is described by a predefined attribute, *status*, describing the temporal behavior of the membership of an instance to its type. MADS allows object and relationship instances to be suspended and resumed in their membership. Hence, the *status* attribute is time-varying: it is a function from DBTIME to STATUS, where STATUS is a predefined domain made up of four values: not-yet-existing, active, suspended, and disabled. The first value relates to an object known to exist at some later time. The active value describes an object that can be used within the associated timeframe. Properties of suspended objects can be read but not updated. Finally, the disabled value qualifies an object that existed in some past time but it is not accessible within the disabled timeframe [9]. Transitions between the values obey the following constraints: not-yet-existing -> active, active <-> suspended, active -> disabled, and suspended -> disabled. Functions associated to *status* include the lifespan (the interval from birth to death) and the activespan (the set of intervals/instants associated to the active status).

4. COMPLEX DATA STRUCTURES

Complex objects have attributes that may be recursively decomposed into other attributes. MADS uses the concept of structure to convey all the characteristics of an attribute, independently from its association to the item (object, relationship, or another attribute) it describes. A structure can be seen as a higher-level ADT for an attribute.

A structure is defined by a tuple **(Name, Cardinality, Spacezone, Timezone, Domain)** such that:

Name is the name of the structure.

Cardinality defines the min-max cardinalities. If the structure is multivalued, cardinality specifies whether the values form a set, a bag, or a list. For time/space-varying structures, this cardinality is the snapshot/local cardinality (it applies for each instant/point). If desirable, the model can easily be extended with an additional lifespan/space-extent cardinality to constrain the number of allowed value changes for time/space-varying attributes.

Spacezone exists only for a space-varying structure. It is an element in SPACEZONES (e.g., an area, a line, or a set of lines) and defines the extent of the spatial domain of the structure. Thus, every value of the structure is a function from Spacezone into the domain of the structure. Values of attributes associated with the structure can be queried at any point in Spacezone, the returned value depending on the chosen point.

The Spacezone concept supports the representation of continuous field information. Whereas in most GISs such information is implicitly defined over DBSPACE, MADS allows attaching to each continuous field attribute the relevant area, usually (but not necessarily) the area of the owner of the attribute. This enforces orthogonality and greatly enhances the flexibility of the model.

Timezone exists only for a time-varying structure. It is an element of TIMEZONES (e.g., a time interval, a set of disjoint time intervals, or a set of instants) and defines the extent of the temporal domain of the structure. Thus, every value of the structure is a function from Timezone into the domain of the structure. Values of attributes associated with the structure can be queried at any instant in Timezone, the returned value depending on the chosen instant.

Domain defines the domain of the structure. A structure defines an atomic attribute if its domain is an elementary domain like Integer, Real, String ... or any of the SADTs or TADTs. Otherwise, the structure defines a complex attribute and its domain is the set of structures of the component attributes.

The set of legal values associated to a structure is computed according to the composition of the structure and its spatial and temporal features. It may be an atomic value, a Cartesian product, a powerset, a set of bags, a set of lists, or, if the structure is space and/or time varying, a set of functions (from Spacezone or Timezone or from the Cartesian product of Spacezone and Timezone). This semantics is formally defined in [7].

5. OBJECT TYPES

Spatial and/or temporal information may be associated to objects, independently from the characteristics of their attributes. Consequently, an object type can be plain (neither spatial nor temporal), spatial, temporal, or spatio-temporal.

An object type O is defined by a tuple O = **(Name, Geometry, Status, Attributes, Methods, Super, Pop)** as follows.

Name is the name of the object type.

Geometry is a peculiar attribute defining the spatiality of O objects, if any. The domain for *geometry* is one of the SADTs. As any attribute, *geometry* may be time-varying and can be inherited, possibly with refinement or redefinition (see hereinafter), or derived. In the current version of MADS only single geometry values are possible (*geometry* is monovalued). However, having multiple geometry values for the same object is desirable for supporting multiscale databases or federated spatial databases. A proposal for a multiscale extension of MADS may be found in a companion paper [26].

Status is a peculiar time-varying attribute defining the lifecycle of O objects, if any. The domain for *status* is a function from DBTIME to the STATUS domain. The attribute is monovalued as an object may only be in a given status at each instant. The *status* attribute can be inherited, possibly with refinement or redefinition, or derived. Although formally possible, we could not find a meaningful example for a space-varying *status*.

Attributes defines the structures of the attributes of O. This set of structures is the union of the structures for the locally defined attributes, and the structures for the attributes inherited by O with refinement and/or redefinition. A redefined attribute allows keeping two values: the one inherited from the supertype and another local to the type in which it is redefined.

Refinement and redefinition are traditional mechanisms in object-oriented databases. However, few spatial or temporal models allow such facilities for the spatial or temporal features [3]. These mechanisms have been proved extremely useful when modeling spatio-temporal applications. For instance, redefinition of objects' temporality allows to keep track of lifecycles of objects in a specific role, thus complementing the information about the lifecycle of an object in its generic role, as in Figure 2 (b). Redefinition also provides an easy way to represent different geometries of an object at different scales, e.g., the same road line at 1/100'000 and at 1/200'000 scales, as in Figure 2 (d). Refinement, on the other hand, allows attaching a more specific spatial or temporal ADT to the subtype, as in Figure 2 (c).

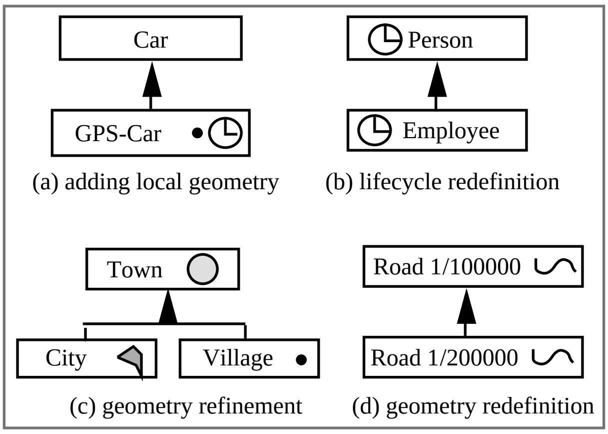


Figure 2: Refinement and redefinition of spatio-temporal features.

Methods defines the set of methods of O. Each method defines a method name and a signature specifying the types of the input and output parameters of the method. To ensure substitutability, refined attributes must satisfy the sub-type relationship.

Super is the (possibly empty) set of supertypes of O.

Pop is the population of the object type. Each object is a couple made up of the object identity and its value. In case of a temporal entity type, its population is time-varying.

In the semantics defined in [7] a set of legal objects is associated to each object type. Appropriate functions are used to derive the set of attributes inherited by O from its supertypes, and to derive the whole set of attributes of O. Constraints are defined to avoid name clashes between local and inherited attributes in case of multiple inheritance. Further constraints define the attribute inheritance rules associated to the generalization links for ensuring substitutability and population inclusion (see Section 7).

6. RELATIONSHIP TYPES

Relationships are an essential part of conceptual design. MADS supports several kinds of relationships. As for objects, relationships may be located in space and time, via the *geometry* and *status* attributes. In this case they are referred to as spatial and/or temporal relationships. Locating relationships in space is usually not supported by current proposals, hence worth an example. Assume a cyclic binary relationship type Accident, linking twice the object type Car. To keep track of both the time of occurrence and the location of the accident, the relationship type is defined as spatial (with point type geometry) and temporal (with instant type timestamp). The object type Car may or may not be spatial and/or temporal according to the application needs.

MADS supports the following kinds of relationships: traditional (i.e., ER-like) relationships, aggregation relationships, constrained relationships, and dynamic relationships.

Constrained relationship types convey spatial and temporal constraints on the objects they link. Well-known examples are topological relationships, e.g., a relationship type Inside linking object types City and County expressing that the geometry of a city is within the geometry of the related county. Metric and directional relationships are other examples related to space. All of these are usually called "spatial relationships" but this term is used in MADS for relationships located in space. On the temporal side, synchronization relationships are useful to constrain the lifecycle of linked objects, e.g., a relationship type During linking object types Landslide and Typhoon to express that the lifecycle of a landslide is within the lifecycle of the associated typhoon. The During relationship may also hold the topological constraint that the landslide occurs within the area covered by the typhoon.

Dynamic relationship types describe real-world phenomena where time and/or space play a significant role. MADS identified **transition** relationships to model the migration of objects from a source to a target object type (e.g., a relationship type Graduation relating object types Student and Alumnus), and a **generation** relationship to represent processes that lead to the emergence of new objects (e.g., in a survey application, a relationship Generate keeping track of which parcel(s) generated other(s) parcel(s) by splitting and/or merging).

A relationship type R is defined by a tuple R = **(Name, Kind, Roles, Geometry, Status, Attributes, Methods, Pop)** as follows.

Name is the name of the relationship type.

Kind defines the kind of the relationship (see below).

Roles defines the participating object types, the role names, their cardinality, and the collection type (set, bag, or list). Usual constraints apply, e.g., role names must be uniquely identifiable.

Geometry, Status, **Attributes**, and **Methods** are defined as for object types.

Pop is the population of the relationship type. Each relationship is a tuple made up of the relationship identifier, the set of linked objects, and the relationship value. The population of a temporal relationship type is time varying.

The **Kind** of a relationship R is defined by the tuple (**Type**, **Constraint**, **Membership**) as follows.

Type $\in$ {plain, aggregation, transition, generation} $\cup$ Topological $\cup$ Synchronization, where

> Topological = { disjoint, adjacent, intersect, cross, inside, equal, Spatial-User-Defined }, and
>
> Synchronization = { before, equal, meets, overlaps, during, starts, finishes, Temporal-User-Defined }

Spatial-User-Defined and Temporal-User-Defined types denote application-specific relationship types that convey user-defined constraints. Synchronization relationships are defined using Allen's operators [1] extended for temporal elements. They take into account either the entire lifecycle of the objects or the active periods only. Other predefined spatio-temporal relationships, e.g., AlwaysDisjoint, AlwaysInside, SometimesCross for moving geometries [15] can be easily added.

Constraint is an optional predicate over the geometry and/or the lifecycles of objects linked by R. It can be either a predefined or a user-defined predicate. Consider two objects o1 and o2 linked by a relationship r. The predicate for the predefined topological relationship disjoint is Disjoint(o1.geometry, o2.geometry). The predicate for the predefined synchronization relationship before is Before(o1.lifecycle, o2.lifecycle). Finally, the predicate for the transition relationship is (o1.birth≤o2.birth $\wedge$ o2.birth $\in$ r.activespan $\wedge$ o1.oid=o2.oid).

Membership is a Boolean determining whether the constraint defines a sufficient condition for belonging to the relationship (the constraint always defines a necessary condition). If membership = true, for every instant in DBTIME every tuple of objects satisfying the constraint defines a relationship belonging to R.

7. CONSTRAINTS

The issue of consistency of specifications naturally arises when spatial and temporal features are distributed over objects, relationships, and attributes. Space does not automatically generate constraints. For instance, the spatiality of an object does not constrain the spatiality of its attributes: the spatiality of a building is unpredictably related to the spatiality of the nearest fire station, even if the latter is a component attribute of the former. On the other hand, there should be an easy way to specify spatial integrity constraints, e.g., for stating that the spatiality of a country topologically contains the spatiality of its capital. MADS follows this policy.

Time, instead, is seen as prone to constraints. Common assumptions about timeframes in related facts include:

- the validity period of an attribute must be within the life cycle of the object it belongs to (e.g., [3,27]);
- the validity period of a complex attribute is the union of the validity periods of its components (e.g., [8]);
- the lifespan of a relationship should be included in the intersection of the lifespans of the related objects [14].

These assumptions are based on the intuitive view that an attribute cannot exist if its owner does not exist, and a relationship cannot link objects that do not exist simultaneously. However, in this case intuition is unfortunately misleading, and the above constraints prove to be unnecessary strong limitations in expressive power. Consider, for instance, a relationship Biography linking the author of a biography to the famous person the biography is about. Clearly, the famous person may have been in existence long before his/her biographer. Thus, MADS does not enforce any of the constraints stated above.

Several issues are related to the coexistence of temporal and non-temporal facts. If a non-temporal object contains a time-varying attribute, the time-varying information will disappear from the database when the object is deleted. Consider also a temporal object having an attribute that is not time-varying. Retroactive or proactive queries querying, respectively, the value of the attribute in the past or in the future, only have available the current value. This value is accurate in the past or in the future only if the attribute is known to be constant (e.g. a person's date of birth). Currently, MADS returns the existing value for retroactive queries, and no value for proactive queries.

Considering temporality, the population inclusion constraint associated to is-a links is revisited in MADS as follows. If both the supertype A and the subtype B are temporal, the constraint reads: at any time t, objects active in B must be active in A, objects suspended in B must be either suspended or active in A, objects disabled in B must be either disabled, suspended, or active in A. If the supertype A is non-temporal and the subtype B is temporal, the inclusion constraint only holds, at time *now*, between active objects in B and objects in A. In particular, B may have a larger number of objects than A.

Cardinalities of roles in relationships also deserve being revisited. We already mentioned that snapshot and lifespan cardinalities may be defined. These apply when a temporal object type participates into a non-temporal relationship type. If a non-temporal object type participates into a temporal relationship type, the cardinality of its role may be understood as applying to active instances of the relationship type. This is our choice for MADS, although formally it would be meaningful to define four cardinalities, one for each set of relationships in the same status (not-yet-existing, active, suspended, disabled). The case where both the object type and the relationship type are temporal is a conjunction of the two previous cases.

8. VISUAL NOTATIONS AND TOOLS

Figure 3 shows a MADS diagram for an excerpt of a river monitoring application. MADS retains the flavor of ER diagrams. Lines express cardinalities based on an intuitive notation: dotted means optional, two lines means multivalued. Temporal and spatial icons are embedded into object and relationship types. They may also be associated to attributes. The temporal icon (symbolizing a clock) on the left-hand side of the object/relation type (cf. M-Station in Figure 3) expresses that the lifecycle information is to be kept. Spatial icons are shown to the right-hand side. They may have an adjacent temporal icon when the spatial information is time varying.

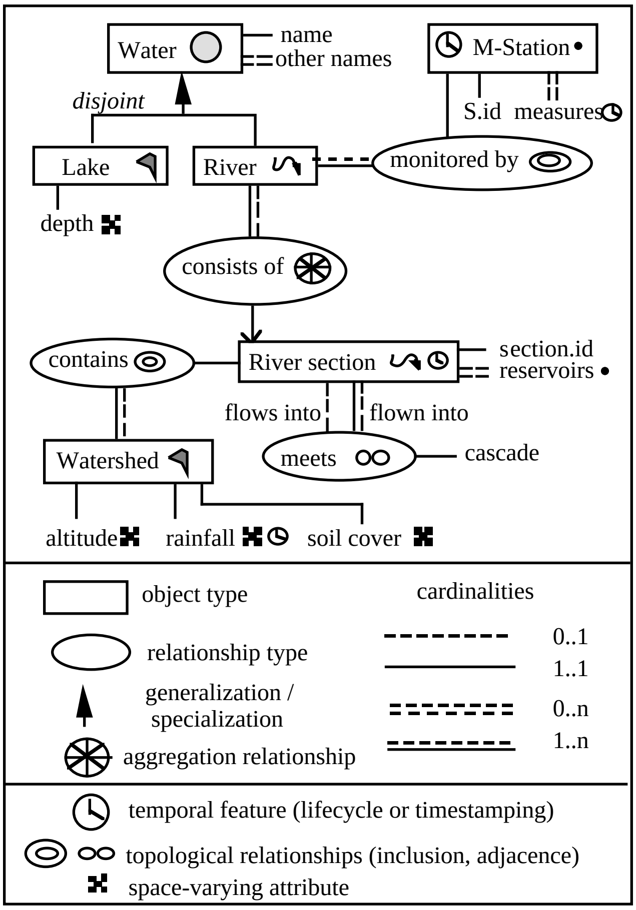
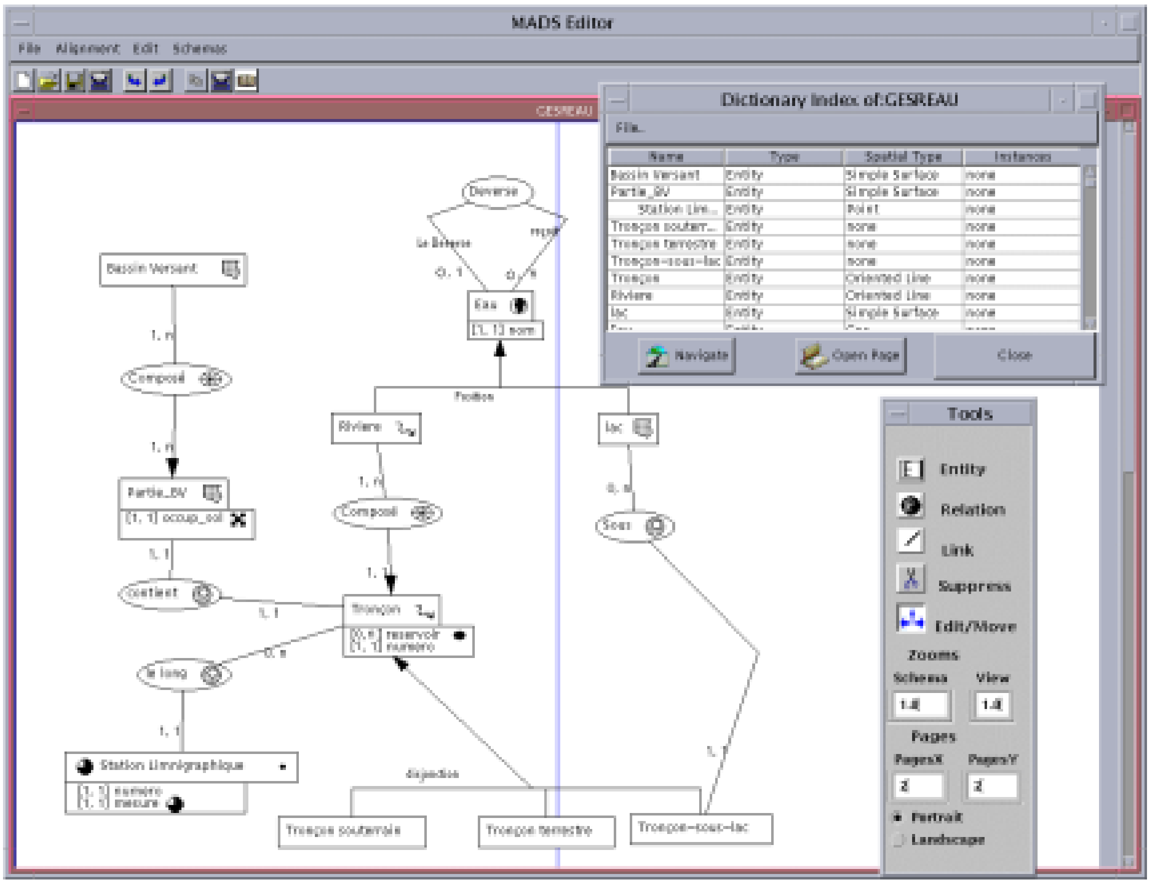


Figure 3: An example of a schema diagram in MADS.

The example diagram shows watersheds containing river sections into which rivers are decomposed. Measurement stations provide water quality information on the rivers. These stations may not operate full time: their lifecycle information allows to record periods of operation. Each station provides several measures, whose validity period is also recorded. Reservoirs may be located on river sections: their position is recorded as points. Each watershed records space-varying information (e.g., altitude). While the history of rainfall is kept, the history of soil covers is not.

We have developed a visual schema editor, in Java, that allows MADS schema diagrams to be defined through direct manipulation on the screen. A design session consists of drag and drop operations to place schema elements in a diagram, complemented with interactions through form-like windows for the detailed specification of the elements. A screen capture of the editor is shown in Figure 4. The development of the schema editor was based on the experience acquired in the SUPER project on the design and implementation of conceptual visual interfaces for classical databases [6]. Existing translation modules allow to transform a MADS schema into an equivalent IEF (a simple ER model) [16],



Name	Type	Spatial Type	Instances
Bassin Versant	Entity	Simple Surface	none
Partie_BV	Entity	Simple Surface	none
Station Lim...	Entity	Point	none
Tronçon souterr...	Entity	none	none
Tronçon terrestre	Entity	none	none
Tronçon-sous-lac	Entity	none	none
Tronçon	Entity	Oriented Line	none
Rivière	Entity	Oriented Line	none
lac	Entity	Simple Surface	none

Figure 4: Schema definition with the MADS editor.

INTERLIS [17] (a Swiss exchange norm for GIS) or relational schema. The spatio-temporal characteristics of data are kept in the target specification provided that the GIS or DBMS can handle space or time. Otherwise, spatio-temporal integrity constraints are generated and implemented as active rules. An automatic generator of HTML documentation complements the basic functionality of the editor. A companion tool for visual querying is being investigated.

9. CONCLUSION

This paper proposes a sound basis for the development of spatio-temporal conceptual models. Indeed, an analysis of existing models shows that such a basis is weakly defined. Spatial models use ad hoc ways of embedding space within data structures. This lack of orthogonality results in restricting the designer's freedom with no conceptual reason. Temporal models tend to be poor in the supported data structures and/or include unnecessary constraints. Few models address both space and time, showing similar drawbacks. To the best of our knowledge, the model we propose, MADS, is the first one to explicitly obey the orthogonality principle in adding space and time to data structures: spatio-temporal features may be associated to objects, attributes, and relationships. The space and time combination has an immediate desirable effect: MADS naturally supports modeling of moving objects (e.g., cars, borders, pollution clouds). The spatial features of MADS support both the discrete and the continuous views of space, where the space domain for space-varying information is the geometry of any selected item. This approach is far more flexible than what many GIS offer. The temporal features of MADS are mainly characterized by the fact that no constraint is enforced over timeframes of related facts.

Moreover, MADS allows the specification of relationships that have space- and/or time-related semantics. Thus, topological relationships may be defined as named schema elements, and bear attributes and methods, an important feature that GIS-computed relationships cannot offer. Similarly, synchronization relationships may be defined to constrain the lifespan of related objects. Transition and generation relationships have been included in MADS to better respond to user requirements.

MADS was developed in an application framework and has been used for modeling several real-world applications: oil management in Colombia, management of the networks of clear and used waters of the Geneva city, study of the evolution of the watershed of the upper part of the Sarine river, and the management of water resources of the Vaud county. The users' feedback received from using MADS in these applications was very encouraging. In the water management case we were able to measure the benefits of using MADS with respect to a traditional ER model. In terms of simplicity of the schema, MADS remodeling reduced the number of object and relationship types by a factor of 23%. Understandability of the schema was greatly improved by the explicit display of spatio-temporal features and removal of artifacts due to non-conceptual constructs. Finally, using MADS led the application designers to discover the importance of temporal information within their application.

A design editor has been implemented for the visual definition of MADS schemas. Automatic translation of MADS specifications into equivalent specifications of commercial tools has been implemented for two target systems, with more target systems (GIS or DBMS) to come in the near future.

Many extensions are being investigated. We are currently developing an extension to address the need for geographical multiscale databases, as well as an algebraic language that would contribute to the development of a visual manipulation interface for spatio-temporal databases.

10. REFERENCES

[1] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11): 832-843, 1983.

[2] 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.

[3] E. Bertino, E. Ferrari, G. Guerrini. T-Chimera: A Temporal OO Data Model, *Theory and Practice of Object Systems*, Vol. 3, No 2, 1997.

[4] G. Booch, J. Rumbaugh, I. Jacobson. *Unified Modeling Language: User Guide*. Addison-Wesley, 1998.

[5] B. David, L. Raynal, G. Schorter. GeO2: Why objects in a geographical DBMS? In *Proceedings of the 3rd International Symposium on Advances in Spatial Databases, SSD'93*, p. 264-276. LNCS 692, Springer, 1993.

[6] Y. Dennebouy et al. SUPER: Visual Interfaces for Object+Relationship Data Model, *Journal Of Visual Languages and Computing*, March 1995.

[7] P. Donini, C. Parent, S. Spaccapietra, E. Zimányi, *Syntax and Semantics for the MADS conceptual model*. Technical Report TR-99-02, Database Laboratory, Ecole Polytechnique Fédérale de Lausanne, February 1999.

[8] R. Elmasri, G. Wuu, and V. Kouramajian. A temporal model and query language for EER databases. In A.U. Tansel et al. (Eds.) *Temporal databases. Theory, design, and implementation.* Benjamin/Cummings, 1993.

[9] O. Etzion, A. Gal, A. Segev. Extended Update Functionality in Temporal Databases, In O. Etzion et al. (Eds), *Temporal Databases: Research and Practice*, LNCS 1399, Springer 1998.

[10] S. Ferg. Modeling the Time Dimension in an Entity-Relationship Diagram. In *4th International Conference on the Entity-Relationship Approach, ER'85*. Computer Society Press 1985.

[11] O. Gayte, T. Libourel, J.P. Cheylan, S. Lardon. *Conception des systèmes d'information sur l'environnement*, Hermès, Paris, 1997.

[12] H. Gregersen, C.S. Jensen. Temporal Entity-Relationship Models - A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(3): 464-497, 1999.

[13] H. Gregersen, C.S. Jensen, and L. Mark. Evaluating Temporally Extended ER Models. In *Proceedings of the 2nd CaiSE'97 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, 1997.

[14] H. Gregersen and C. S. Jensen. *Conceptual Modeling of Time-varying Information*. TimeCenter Technical Report TR-35, September 1998.

[15] R.M. Guting et al. *A Foundation for Representing and Querying Moving Objects.* FernUniversitaet Hagen, Informatik-Report 238, September 1998

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

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

[18] M.R. Klopprogge and P.C. Lockeman. Modeling Information Preserving Databases: Consequences of the Concept of Time. In *Proceedings of the 9th International Conference on Very Large Databases, VLDB'93*, 1983.

[19] D. Pantazis, J.P. Donnay. *La conception de SIG - Méthode et formalisme*, Hermès, 1996.

[20] C. Parent, S. Spaccapietra, E. Zimányi, Modeling time from a conceptual perspective. In *Proceedings of the 7th International Conference on Information and Knowledge Management, CIKM'98*, p. 432-440. 1998.

[21] C. Parent, S. Spaccapietra, E. Zimányi, P. Donini, C. Plazent, C. Vangenot Modeling Spatial Data in the MADS Conceptual Model. In *Proceedings of the 8th International Symposium on Spatial Data Handling, SDH'98*, p. 138-150. 1998.

[22] N. Pissinou et al. Towards an Infrastructure for Temporal Databases, *SIGMOD Record*, 23(1):35-51, 1994.

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

[24] R. Snodgrass. Temporal object oriented databases: A critical comparison. In *Modern Database Systems: The Object Model, Interoperability, and Beyond*, p. 386-408. Addison-Wesley, 1995.

[25] N. Tryfona, D. Pfoser, T. Hadzilacos. Modeling Behavior of Geographic Objects: An Experience with the object modeling technique. In *Proceedings of the 9th International Conference on Advanced Information Systems Engineering, CAiSE'97*, Springer-Verlag, 1997.

[26] C. Vangenot, *Multiscale support in the MADS conceptual model*. Technical Report TR-99-01, Database Laboratory, Ecole Polytechnique Fédérale de Lausanne, February 1999.

[27] G.T.J. Wuu and U. Dayal: A Uniform Model for Temporal and Versioned Object-oriented Databases. In A.U. Tansel et al (Eds.), *Temporal databases. Theory, design, and implementation.* Benjamin/Cummings, 1993