A Multigranular Spatiotemporal Data Model *

Elena Camossi[1] Michela Bertolotto[2] Elisa Bertino[1] Giovanna Guerrini[3]

[1]Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
via Comelico 39/41 - 20135 Milano, Italy
{camossi, bertino}@dico.unimi.it

[2]Department of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland
michela.bertolotto@ucd.ie

[3]Dipartimento di Informatica
Università di Pisa
via F. Buonarroti 2 - 56127 Pisa, Italy
guerrini@di.unipi.it

ABSTRACT

A large percentage of data managed by a variety of different application domains has spatiotemporal characteristics. Unfortunately, traditional geographical information systems do not allow for an easy representation of temporal aspects of spatial data. Moreover, they do not usually support the representation of data at multiple levels of granularity. In this paper we present a multigranular spatiotemporal data model. Our model extends the ODMG model with multiple spatial and temporal granularities. In particular, the model allows for an uniform management of two kinds of spatiotemporal objects: *moving entities* (e.g. cars, planes, etc.) and *temporal maps* (i.e, maps representing the change over time of a given geographic area). It also provides a framework for mapping the movement of an entity such as a car onto an underlying geographic area. The model we propose relies on a standard definition of temporal granularity. On the other hand, the representation of spatial entities at multiple granularities is obtained by applying model oriented map generalization principles. In particular, we consider a set of generalization operators that guarantee topological consistency.

Categories and Subject Descriptors

H.2.1 [**Database Management**]: Logical Design,—*Data models*; H.2.3 [**Database Management**]: Languages—*DDL, Query languages*; H.2.4 [**Database Management**]: Systems—*Object oriented databases*; H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*

General Terms

Design, Standardization, Languages

Keywords

Spatiotemporal databases, spatial and temporal granularities, multiresolution, multirepresentation

1. INTRODUCTION

Recently, it has been estimated that 80% of all available data have spatial characteristics. At the same time, a lot of organizations have realized the importance of exploiting the spatial component of data, for performing statistical analysis, reporting, etc. As a consequence, in many application contexts, there is the need to add spatial functionality to existing non-spatial data. Unfortunately, this is a non-trivial task when two different systems have to be used, one to handle spatial data and the other to manage non-spatial data, as it happens when using traditional GIS systems. Indeed, traditional GIS applications store spatial and non-spatial data attributes separately: usually spatial objects properties are stored in files managed by a file management system, while attribute data are stored in a commercial database. Such an approach, that we refer to as *loosely coupled approach* [18], has the drawback that maintaining data integrity between spatial and attribute data is difficult as the two types of data are not managed by the same engine. Emerging database products, like Oracle Spatial [23] and Postgres [25], follow an integrated approach for managing spatial and non-spatial information, providing the information infrastructure that includes a single database system for managing both types of data. Adding spatial functionality to legacy data already stored in a traditional DBMS is easier if the spatial component can be integrated in a homogeneous way.

Furthermore, since GIS are main memory applications, they do not adequately manage real time updates of large datasets. For example, they are not able to manage real time applications such as an airport control system that requires to retrieve concurrently the real time trajectories of several airplanes, comparing them with a map of the underling geographic area. This scenario requires the system to handle the overlay of representations between moving entities and an underlying map, over time. In order to handle correctly the huge amount of data required by this kind of application, temporal aspects have to be taken into account. Although temporal extensions of GIS systems exist [14], commercial GIS packages still do not properly support temporal aspects of spatial data. Other typical examples

---

*Acknowledgments: The work of Elisa Bertino and Elena Camossi has been partially supported by the EU under the IST Panda project. The work of Elena Camossi has been partially supported by the University of Genova, Italy. The support of the International Collaboration Initiative of Enterprise Ireland is also gratefully acknowledged.

GIS'03, November 7–8, 2003, New Orleans, Louisiana, USA.

of applications that are intrinsically spatiotemporal are systems to control spread of fire over forests, analysis of meteorological phenomenons, tide control systems, cadastral applications, control deforestation systems, etc.

Another critical point relates to the fact that commercial GIS do not provide much support for multirepresentation of spatial data, that is considered as an important functionality when analyzing huge amounts of spatial data, often collected from different sources.

In this paper we address the above need by proposing a framework for representing both spatiotemporal data supporting multiple granularity management for spatial and temporal dimensions. We are particularly interested in modeling situations in which the position of *moving entities* (i.e., entities with a spatial extension that change their position over time) must be related to geographic areas represented by maps. Both moving entities and map data can be specified at different levels of detail, i.e., at different *spatial granularities*. Our notion of spatial granularity partitions the space by taking into account the specific application domain considered. Therefore, this notion is application domain dependent. To compare spatial data expressed at different granularities, we refer to map generalization operators [15, 16, 20], specifically those used in *model-oriented generalization* [17]. In particular, the operators supported by our model are those defined in [2], that guarantee topological consistency. To simplify the examples we consider only two dimensional data and a simple set of hierarchical granularities that rely on standard length units.

The standard notion of temporal granularity [3] is also supported by the model. The framework we present extends the ODMG [6] model, by taking into account the work done in a previous temporal extension [1]. This multigranular framework can be considered as a basis for developing an object oriented spatiotemporal model with an expressive query language, as extension of OQL and the temporal path expression language defined in [1], for performing the analysis of spatiotemporal data expressed at multiple spatial and temporal granularities.

Several proposals providing an integrated approach for the management of spatial and temporal information have been presented in the recent past, as temporal extensions of GIS [14], or as independent frameworks [21]. A growing interest has been devised in the area of moving [11] and geometric [8] objects. In [11] a framework for modeling moving points and regions has been formalized, by proposing abstract data types that can be integrated in relational and object relational models. In [8], the closure properties of a set of spatiotemporal objects (rectangle and convex polygons) have been discussed. All these proposals (except for [11]) involve abstract modeling. Recently, also spatiotemporal extensions of SQL99 [7] and of ODMG model [10, 12] have been proposed. [10] reports on the design of the Tripod spatiotemporal database system, that extends the ODMG type system with temporal and spatial types, handling past representations of attributes through the concept of *histories*. Huang and Claramunt [12] extend the ODMG set of literal types with spatial types, by defining also a parameterized temporal type that can be instantiated using a spatial type for modeling spatiotemporal information. Such proposals, however, do not address issues related to multigranular representation of spatiotemporal data.

An attempt in this direction is discussed in [13], that presents an annotation-model (extension of the Unifying Semantic Model formalism) allowing for the specification of spatiotemporal data at multiple granularities. The granularity systems presented in [13] rely on the concepts of temporal *indeterminacy* and spatial *imprecision* [9]. Indeed, a huge amount of work has been done during the last few years to formally define the notion of spatial granularity. Recently, this work has focused on issues related to the concepts of vagueness, imperfection and imprecision of spatial information, in particular in the reasoning research area [4]. In this paper, we do not follow this approach, although we take into account domain semantic dependency in the general design of spatial granularities. Furthermore, in contrast with [13], in our model we provide a specific set of functions for converting values at different granularities. In [19] a theoretical framework for the specification of a spatial granularity lattice, that establishes how to relate different granularities and is closer to our approach, is presented.

The paper is organized as follows. Section 2 describes the model, discussing the set of spatial and temporal granularities we consider and the spatiotemporal types we define. Section 3 reports on how the model handles multirepresentation of spatiotemporal data. Section 4 presents an example, and Section 5 concludes the paper, outlining how the model will be extended.

2. THE SPATIOTEMPORAL DATA MODEL

The spatiotemporal data model we propose manages uniformly both spatial and temporal information. We have extended the ODMG [6] type system with specific types for representing spatial data by means of vector features (e.g. points, lines and regions), and with parametric constructors for specifying spatial, temporal and spatiotemporal information at multiple granularities. Spatial and temporal information is specified at attribute level. The temporal dimension we consider in our discussion is the valid time dimension [3], i.e., data stored in the database refer to the time the represented facts were true in the reality.

The model has been designed for handling *moving entities* (e.g. cars, planes, walking people, etc.) and *temporal maps* (e.g. road, land coverage, city, population distribution historical maps, etc.) uniformly. Such homogeneous management easily allows for relating the trajectory of moving entities to the geographical areas represented by maps. The history of updates of maps over time, such as modifications of land parcel boundaries, road paths, and other topological changes, is also managed.

The moving entities and maps representation can be given by means of their geometric shapes, expressed in two dimensions. A moving entity can also be represented by a point, i.e., by referring its centroid. We assume that geodetic coordinates at the maximum resolution are used to specify points coordinates. By using the geometric shape for moving entities instead of representing them by point abstraction, we can answer some interesting queries that could not be expressed otherwise. Given, for example, two airplanes, as illustrated in Figure 1(a), the model allows for the detection of crashes involving their spatial extension but not involving their centroid trajectories. Moreover we can easily check if a moving entity changes direction, and distinguish between two opposite movements having the same trajectory. Given for example the planes in Figure 1(b), we can recognizing that plane A is moving forward, while plane B

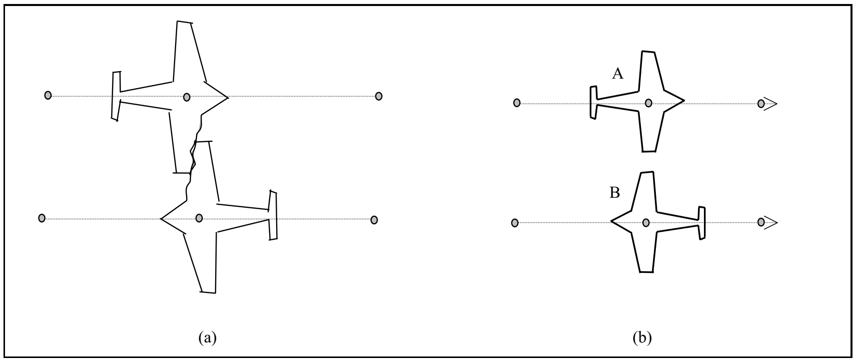

Figure 1: Queries examples for moving entities solved by the model

is moving backward (suppose it is making a reverse motion when maneuvering), although both planes are moving in the same direction, with the same trajectory. Such kind of distinction is possible only if the representation of the moving entity is not symmetric.

The model allows for the definition of data at different spatial and temporal granularities, i.e., at different levels of detail with respect to spatial and temporal dimensions. In particular, data representing moving entities or maps, that have spatiotemporal characteristics, can be specified with respect to both different spatial and temporal granularities.

In the remainder of the Section, the notions of spatial and temporal granularities we adopt are described. Then, we define the types that can be assigned to object attributes in order to describe spatiotemporal information.

2.1 Spatial and Temporal Granularities

To represent data at different temporal levels of detail, we consider the definition of temporal granularity commonly adopted by the temporal database community [3]. A temporal granularity is defined as a mapping from an ordered index set $\mathcal{IS}$ to the set of possible subsets of the *time domain*. We assume that the time domain is a discrete set of time instants, on which a total order relationship is defined. Intuitively, a granularity is a partition, possibly non-total, of the time domain. Examples of granularity are *days*, *weeks*, *years*, with the common notion used by the Gregorian calendar. Every portion of the time domain obtained by such a mapping is called *granule* (e.g., using a textual representation, "15/05/2003" is a granule for the granularity *days*, and "2002" is a granule for the granularity *years*). Granules of the same granularity cannot overlap and must keep the same order given by the index set.

The set of temporal granularities managed by the model is denoted by $\mathcal{G_T}$. Granularities in $\mathcal{G_T}$ are related by the *finer-than* relationship. A granularity G is said to be *finer-than* a granularity H, denoted by $G \preceq H$, if, for each index i, an index j exists such that $G(i) \subseteq H(j)$ [3] (e.g., *days* is finer-than *months*). We also say that H is *coarser-than* G. The symbol "$\prec$" denotes the anti-reflexive finer-than relationship.

For representing the Earth surface we consider a two-dimensional Euclidean space. Spatial entities[1] can be represented at different granularities by considering hierarchical representations that can be devised from subdivisions of the reference space into regular grids, or from some of their semantic characteristics, e.g. administrative boundaries, roads categories, lands use classifications. In this paper, due to the lack of space, we refer only to regular subdivisions of space, that are more immediate to understand and to focus on. However, when different hierarchies are used in a database schema to describe both temporal maps and moving entities, equivalences between different spatial levels of detail must be specified in the database schema, for comparing and querying data at different granularities [5]. Example of granularities we consider are *ms*, *Dms*, and *kms*, representing the international standard measures of length meters, decameters, and kilometers, that in a two-dimensional space represent squares with 1 meter, 10 meters, and 1 kilometer long sides. A possible straightforward interpretation for these granularities can be, for instance, that for a spatial entity represented at *ms* granularity, all its spatial components that are less than 1 meter long are not represented. Usually, the semantics of spatiotemporal granularities sets can vary with respect to the application domain. However, it must be fixed for a specified database schema. The semantics of spatial granularities is given by specifying the conversions of spatial and spatiotemporal values at different spatial granularities.

The set of spatial granularities managed by the model is denoted with $\mathcal{G_S}$. Granularities in $\mathcal{G_S}$ are related, like temporal granularities, by finer-than relationship (and by its inverse coarser-than). We can say, for instance, that *mms* (i.e., millimeters) is finer-than *ms* and that *Dms* is coarser-than *cms* (i.e., centimeters).

2.2 Types and Values

Object attributes we model can be spatial, temporal, spatiotemporal or conventional attributes, that is, attributes without any spatiotemporal characteristics.

Let $\mathcal{T}$ be the set of ODMG types [6], including class and literal types. Conventional attributes are defined having type $\tau \in \mathcal{T}$.

For representing spatial data we extended the ODMG types set with the interface type *GeometricFeature* and its implementations *Point*, *Line*, and *Region*. These object types represent the well-known vector feature types in two-dimensions. In the model, each point is implemented by using two coordinates; each line is represented by an ordered set of points, classified into *endpoints* and *shape points* of the line; and each region is represented by the ordered set

---

[1] Entities that are related to the reference space, by having a spatial extension or position.

of its boundary lines, with the usual convention on the order of coordinates to represent internal and external boundaries (clockwise for internal boundaries and counterclockwise for external boundaries). The *GeometricFeature* type hierarchy is represented in Figure 2, where the types data structures are reported.

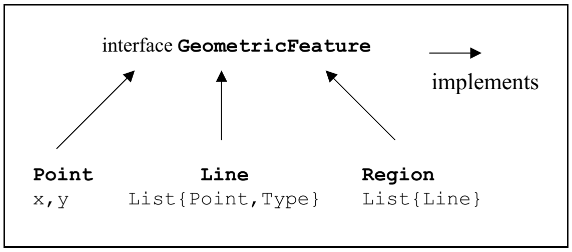

Figure 2: *GeometricFeature* **types**

To represent spatial data at multiple granularities we define the *Spatial* parametric type. For each geometric type γ (i.e., γ is a *GeometricFeature* type or γ is obtained using ODMG collection types constructors, such as `set<>`, and *GeometricFeature* types), and for each spatial granularity $G_s \in \mathcal{G}_S$, the spatial type $Spatial < G_s, \gamma >$ is defined. A legal value of type $Spatial < G_s, \gamma >$ is a legal value of the type γ.

Example 1. *The following class specification defines the object type* `city`*:*

```
class city {
  attr² int population;
  attr Spatial < hms, set < Region >> ext;
  ... }
```

Given an ordinary city, this can be described by an object of type `city`*. The numeric attribute* `population` *records the current value for its population, while the spatial attribute* `ext` *describes its geographic extension, that, at granularity hms (i.e., hectometers, 100 meters), is represented as a set of regions. A legal value for attribute* `ext` *is depicted in Figure 3(a), where the city is represented by a set of adjacent regions, representing the neighborhoods of the city.* ◇

The set of spatial types that can be specified by using both *Spatial* constructor and *GeometricFeature* type is denoted by $\mathcal{S}$.

For describing temporal and spatiotemporal information, we defined the parametric type *Temporal*. For each type $\tau^+ \in \mathcal{T} \cup \mathcal{S}$, called inner type, and for each temporal granularity $G_t \in \mathcal{G}_\mathcal{T}$, a corresponding type $Temporal < G_t, \tau^+ >$ is defined. If $\tau^+ \in \mathcal{T}$, then type $Temporal < G_t, \tau^+ >$ is simply temporal, whereas if the inner type is a spatial type (i.e., is $\tau^+ \in S$), then the type $Temporal < G_t, \tau^+ >$ is spatiotemporal. With $\mathcal{TT}$ we denote the set of (spatio)temporal types that can be specified by using *Temporal* constructor.

A (spatio)temporal value of a temporal type $Temporal < G_t, \tau^+ >$ is defined as a partial function that maps G_t-granules (referred to by their indices) to τ^+ values. We refer to the set of time instants for which these partial functions are defined as the *domain* of the (spatio)temporal value.

In the following, given a type $\tau^* \in \mathcal{T} \cup \mathcal{TT} \cup \mathcal{S}$, the notation $[\![\tau^*]\!]$ denotes the set of legal values for type τ^*.

Example 2. *The class specification for object type* `city` *of Example 1 has been modified, by specifying that both attributes* `population` *and* `ext` *have temporal characteristics.*

```
class city {
  attr Temporal < days, int > population;
  attr Temporal < decades, Spatial < hms,
     set < Region >>> ext;
  ... }
```

The attribute `population` *has been specified with granularity days, while attribute* `ext` *has been specified with granularity decades. Intuitively, such specification means that the semantics of these attributes requires that they are updated, respectively, at most once a day and once a decade*[3]*. This formalizes the intuition that, for any city, the population is a characteristic that is more dynamic than its spatial extension. An example of legal value for attribute* `population` *for an object of class* `city` *representing a small city is: {<06/05/2003, 20,456>, <07/05/2003, 20,488>, <18/08/2003, 20,475>}*$_{days}$*. This value specifies that the value for population for that city was 20,456 on the 6th of May 2003, 20,488 on the 7th of May and 20,475 on the 18th of August. Note that the legal value of the* `population` *attribute has been represented as a set of pairs, that correspond to the points that describe the graph of the value function. since the temporal type for attribute* `ext` *has a spatial type as inner type, the set of legal values for this attribute can be represented as a set of pairs in which the second component is a legal value for the spatial type specified at granularity hms. An example of legal value for attribute* `ext` *is shown in Figure 4, where the geometric representation of a city over three decades, "1980-1989" (a), "1990-1999" (b), "2000-2099" (c), expressed at spatial granularity hms, is illustrated.* ◇

3. MULTIREPRESENTATION

Temporal and spatial multirepresentation of objects attributes are orthogonal characteristics of the model. They are obtained by taking into account temporal and spatial granularities sets separately, and scaling a temporal/spatial value to a different granularity. In particular, for spatiotemporal attributes for which a spatial granularity is specified, spatial multirepresentation is achieved by considering the spatial representation of the attribute with respect to a specific temporal granule. Temporal and spatial multirepresentations are generated on the fly, when a query or a user request is specified, according to the specification given in the database schema.

Spatial multirepresentation is obtained by constructing generalized views of spatial data. A spatial data expressed at a certain granularity is converted to a coarser granularity by applying generalization operators usually considered by model-oriented approaches [17]. In particular, we refer to the set of operators defined in [2], since these operators are continuous mappings that preserve topological consistency, an essential property for usability of spatial data. These operators are only a possible subset of those that can be used to generalize spatial data, and in fact they do not allow to perform some of the traditional generalization operations,

---

[2] This is a shortening for the keyword *attribute* used by ODMG DDL syntax.

[3] The temporal granularity expresses a constraint on the *maximum* frequency of update for an attribute, but it can be updated less frequently.

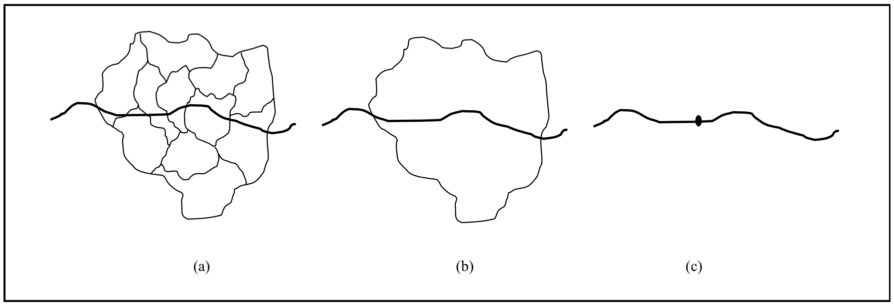

Figure 3: Example of values for attribute `ext` of class `city` at granularities hms (a), kms (b), Mms (c)

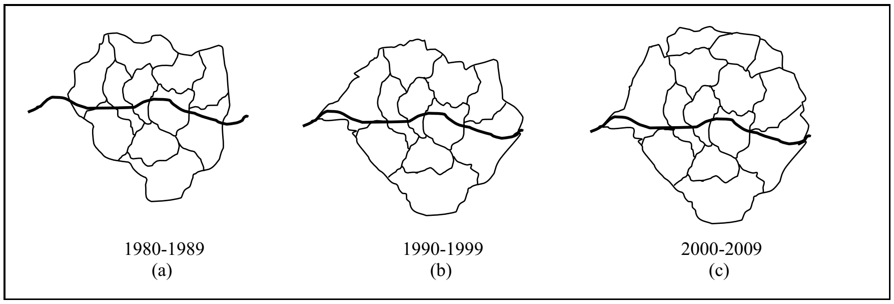

Figure 4: Example of legal value for attribute `ext` of class `city`

i.e., those that do not guarantee preservation of topological consistency (e.g., aggregation, line simplification, etc.). However, we have chosen these operators because they have been proved to be sufficient to perform all generalizations that preserve topological consistency through composition [2], i.e., they can be composed to obtain macro-operators with the same characteristics. The set of these operators, that are shown in Figure 5, is denoted by $\mathcal{O}p_s$. The $\mathcal{O}p_s$ elements are:

(a) `l_contr`, that contracts an open line, endpoints included, to a point;

(b) `r_contr`, that contracts a simple connected region and its boundary to a point;

(c) `r_thinning`, that reduces a region and its bounding lines to a line;

(d) `l_merge`, that merges two lines sharing an endpoint into a single line;

(e) `r_merge`, that merges two regions sharing a boundary line into a single region;

(f) `p_abs`, that eliminates (abstracts) an isolated point inside a region;

(g) `l_abs`, that eliminates (abstracts) a line inside a region.

The conversion of a spatial attribute in the model is expressed by specifying a composition of the operators just described. We can eliminate, for example, San Marino and Vaticano from a map of Italy by applying a composition of operators `r_cont` and `p_abs`. Given two spatial types $\mathit{Spatial} < G_s, \gamma >$ and $\mathit{Spatial} < H_s, \gamma' >$ such that $G_s \prec H_s$, and given $f = \{f_1 \circ f_2 \circ \ldots \circ f_n\}$, where, $\forall i = 1, \ldots, n$, $f_i \in \mathcal{O}p_s$,

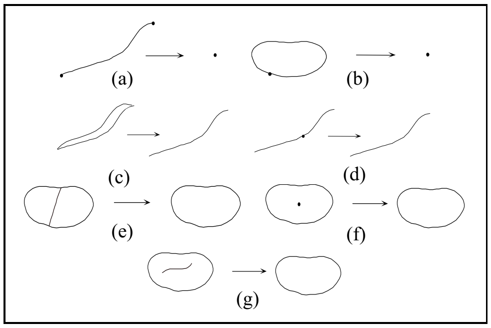

Figure 5: Generalization operators

$$f : [\![\mathit{Spatial} < G_s, \gamma >]\!] \rightarrow [\![\mathit{Spatial} < H_s, \gamma' >]\!]$$

is a partial function that maps values of type $\mathit{Spatial} < G_s, \gamma >$ into values of type $\mathit{Spatial} < H_s, \gamma' >$.

Sometimes, the same generalization can be obtained by composing operators in a different order. Moreover, the generalization process of a spatial entity is usually subjective and strongly domain dependent. Therefore, the sequence of operators to be applied to a spatial attribute for scaling its value, from at certain granularity to a coarser one, must be explicitly expressed in the database schema, by using a language specification, as exemplified in the following.

EXAMPLE 3. *Given the class definition for object type* `city` *of Example 1, where the attribute* `ext`*, representing the extension of a city, is defined at granularity hms as a set of regions, we can generalize the attribute value representing the extension of a city at granularity kms as a single region and at granularity Mms (i.e., 1000 kilometers) as a point. The three different representations of the attribute value are shown in Figure 3. The specification of these representations must be added in the database schema, after the class* *definition, as follows:*

```
convert city.ext from Spatial < hms, set < Region >>
  to Spatial < kms, Region > applying r_merge;
convert city.ext from Spatial < kms, Region >
  to Spatial < Mms, Point > applying r_contr;
```

The first specification implies that, for converting attribute `ext` *from granularity hms to granularity kms, the* `r_merge` *operator must be applied. The second specification implies that, for converting attribute* `ext` *from granularity kms to granularity Mms, the application of* `r_contr` *operator to the region representing the city is required. Note that the generalization process does not involve the non-spatial attribute* `population`*, but only concerns the city spatial extension. Moreover, as shown in Figure 3, the generalization process as specified above does not involve other spatial objects that are in some relationship with the city we generalize, such as a river crossing the city.* ◇

Temporal multirepresentation is obtained by extending the mechanisms proposed in [1] for converting temporal values to different granularities to ensure object substitutability.

The conversion of a temporal value from a given granularity to a finer one is achieved by applying what is known in the database community as the *downward hereditary property*. Given this property, a single granule value defined for a temporal value can be considered as a single granule value also for the finer granules included in the coarser one. For example, if we consider a temporal attribute representing the address of a person specified with granularity *years*, the value of the address for a particular year can be considered as the address value for every day of that year.

EXAMPLE 4. *Given the class definition for object type* `city` *provided in Example 2, and the example temporal values presented for attributes* `population` *and* `ext`*, we assume downward hereditary property. We can say that the value of the population of the city at 12:30 p.m. on the 6th of May 2003 was 20,456, while the spatial extension of the city during year 2000 is the value depicted in Figure 4(c).* ◇

Note that downward hereditary property must be applied carefully, depending on the semantics of the attribute considered. We are investigating how to improve the conversion mechanism to finer granularities, taking into account research results achieved for temporal indeterminacy [13].

The conversion of temporal values to coarser granularities is obtained by applying *coercion functions*, that convert temporal values from a given granularity into values of a coarser granularity in a meaningful way. Given two temporal types $Temporal < G_t, \tau^+ >$ and $Temporal < H_t, \tau'^+ >$ such that $G_t \prec H_t$, a *coercion function*:

$$C : [\![Temporal < G_t, \tau^+ >]\!] \rightarrow [\![Temporal < H, \tau'^+ >]\!]$$

is a total function that maps values of type $Temporal < G_t, \tau^+ >$ into values of type $Temporal < H_t, \tau'^+ >$. Coercion functions can be classified into three categories: *selective*, *aggregate*, and *user-defined* coercion functions. Selective coercion functions are `first`, `last`, `proj(index)`, `main`, `all` and their spatiotemporal variations for spatiotemporal values specified with spatial granularities. Coercion function `proj(index)` returns, for each granule in the coarser granularity, the value corresponding to the granule of position `index` at the finer granularity. Coercion functions `first` and `last` are the obvious specializations of the previous one. Coercion function `main` returns, for each granule in the coarser granularity, the value which appears most frequently in the included granules at the finer granularity. Coercion function `all` returns, for each granule in the coarser granularity, the value which always appears in the included granules at the finer granularity if this value exists, the null value otherwise.

If these coercion functions are applied to spatiotemporal values with spatial granularity, their application can require that a spatial conversion be applied too. Aggregate coercion functions are `min`, `max`, `avg`, and `sum` corresponding to the well-known SQL aggregate functions and can be applied only to temporal values with temporal type specified with an ODMG inner type. In computing aggregate coercion functions we consider undefined values as *null* values in OQL. Finally, user-defined coercion functions correspond to methods declared in the classes of the database schema and can have any semantics.

EXAMPLE 5. *Given the class definition for object type* `city` *of Example 2, we are interested in computing the maximum value for* `population` *attribute of class city during every year, and subsequently the average maximum value with respect to every decade. In order to obtain such values, we can specify two conversions for this attribute: one from granularity days to granularity months using coercion function* `max` *and one from granularity months to years using coercion function* `avg`*. For what concerns spatiotemporal attribute* `ext`*, we are interested in retrieving the last representation of the city recorded in a century, at spatial granularity kms. The corresponding specification is:*

```
convert city.population from Temporal < days, int > to
  Temporal < months, int > applying max;
convert city.population from Temporal < months, int >
  to Temporal < years, float > applying avg;
convert city.ext from
  Temporal < decades, Spatial < hms, set < Region >>>
  to Temporal < century, Spatial < kms, Region >>
  applying last; r_merge;
```

Given the above specification and the values for attributes `population` *and* `ext` *of Example 2, querying for the average maximum value of population for year 2003 results in 20,481.5. By contrast, querying for the 20th century extension of the city results in the representation of Figure 4(b), converted to kms granularity by using* `r_merge` *operator, i.e., only the city boundary is reported as result. Note that the representation of Figure 4(b) is selected according to the semantics of* `last` *coercion function.* ◇

4. AN EXAMPLE

This Section describes in detail an example that involves both moving entities and maps. We consider the example of a plane that is used for European flights. The database schema to model this situation contains the specifications for class `plane` to model planes and class `EuropeMap` to model a map of European countries.

```
class plane{
 attr string model;
 attr Temporal < months, string > companyName;
 attr Temporal < minutes, string > flightNums;
```

```
  attr Temporal < minutes, string > departures;
  attr Temporal < minutes, string > arrivals;
  attr Temporal < minutes, Spatial < ms, set < Region >>
     trips;
}

class EuropeMap{
  attr Temporal < years, Spatial < hms, set < Region >>>
     states;
  attr Temporal < decades, Spatial < hms, set < Line >>>
     rivers;
  attr Temporal < years < Spatial < kms, set < Point >>>
     cities;
}
```

For each plane, we are interested in storing information about the model of the plane, the airline that owns it, and the flights that are organized for it. For each flight we record number, departure and arrival airports, and its path. The attribute `trips` gives the spatial extension of the plane over time, registering the trips made by it. In the specification of the attribute `trips`, the granularity *ms* specifies that the plane components considered are those longer than 1 meter.

For the map of Europe, we are interested in considering European country boundaries, rivers, and cities.

The following specification formalizes the generalizations of `trips` attribute: the conversion from granularity *ms* to granularity *hms* specifies that the attribute, first represented by a set of regions, is generalized to a single region; the conversion from granularity *hms* to granularity *kms* specifies that the attribute is represented by a single point:

```
convert plane.trips from
  Temporal < minutes, Spatial < ms, set < Region >>>
  to Temporal < minutes, Spatial < hms, Region >>
  applying r_merge;
convert plane.trips from
  Temporal < minutes, Spatial < hms, Region >>
  to Temporal < minutes, Spatial < kms, Point >>
  applying r_contr;
```

With this specification we can retrieve, for instance, which countries, and which cities in each country, have been flown over by the plane with flight number 'AZ505' on the 12th of May 2003. The above query can be expressed in extended OQL as follows:

```
SELECT distinct e.states, e.cities
FROM EuropeMap e, plane p
WHERE p.flightNums ↓^days 12/05/2003 = 'AZ505'
  and (p.trips^hms OVERLAPS e.states) ↓
    {{p.flightNums_time ↓ 'AZ505'}↓^days 12/05/2003}^minutes
  and (p.trips^kms EQUALS e.cities) ↓
    {{p.flightNums_time ↓ 'AZ505'}↓^days 12/05/2003}^minutes
```

The specified research condition
$\{\{\texttt{p.flightNums}_{time} \downarrow \texttt{'AZ505'}\}\downarrow^{days} \texttt{12/05/2003}\}^{minutes}$
retrieves the set of granules at granularity *minutes* that represents the duration of the specific trip we are interested in. The two representations of the plane trip, as a plane shaped polygon, at granularity *hms*, and as a single point, at granularity *kms*, that have to be compared with the map of Europe for evaluating the query, are reported in Figure 6. The two operators `OVERLAPS` and `EQUALS` are used for comparing region and point representations, respectively.

Figure 6: Comparing different representations of plane and map

5. CONCLUSIONS AND FUTURE WORK

In this paper, a framework for the specification of a spatiotemporal extension of ODMG data model with support for multiple spatial and temporal granularities has been presented. By designing a database model including both spatial and attribute data, we adopt an integrated approach that uniformly handles all data and differentiates itself from traditional GIS systems. Although the specification discussed is not complete, we believe that it presents some interesting aspects. First of all, we give uniform specifications for both multigranular spatial and temporal types, that can be easily combined to define multigranular spatiotemporal types. In particular, the support for multirepresentation is an important functionality in models designed for analyzing spatial data, potentially collected from different sources. Most of the issues discussed here have individually already been addressed in other papers [10, 12, 11, 8, 19]. However, ours represents the first unified framework that combines multirepresentation for spatial and temporal information in a object data model.

We are currently working on several extensions of the model we presented in this paper [5]. We are specifying the model query language by extending OQL with spatiotemporal path expressions, as extension of temporal path expressions defined in [1]. We are also considering an integration of the query language with map overlay operations. We are extending the data definition language to allow for the explicit storage of topological relationships of spatial data represented in temporal maps, that is a critical issue for speeding up queries. Moreover, we want combine the static conversion specification in the database schema with an more flexible on-demand specification system, by extending the query language with ad-hoc conversion specification.

We are improving the multirepresentation mechanism of the model, by producing theoretical proofs of our approach and by considering alternatives to the specification language presented in this paper. We distinguish between the *value* *correctness* of the conversion mechanism, i.e., a composition of (basic) conversions always produces a legal spatial (or spatiotemporal) value respect to the type system of the model, and its *semantic correctness*, i.e., a conversion does not modify the topological relationships among the data represented in a spatiotemporal database. We are extending the conversion language with a *condition* clause specification, to specialize the application of conversions only to a subset of objects of the same spatial type, i.e., those satisfying a certain condition.

We are investigating how to represent spatiotemporal objects and conversion functions in the object relational model, that is the model used by commercial DBMS. We are also going to develop an XML-based version of our model, following specifications and international initiatives such as the Open GIS Consortium [24] and the INSPIRE project [22]. Finally, we are going to extend the set of generalization operators we consider in this framework with others that are commonly used in model-oriented generalization and cannot be represented with the operators defined in [2] since they do not preserve topological consistency, as, for example, *aggregation*, that allows for merging non-contiguous spatial features in a unique one, or *line simplification*.

We are planning to implement and test the spatiotemporal model and its query language. In particular, we are interested in developing a GUI for visualizing spatiotemporal representation of queries results. We are also interested in optimizing performance by maintaining materialized views of different levels of representation of spatiotemporal data.

6. REFERENCES

[1] E. Bertino, E. Ferrari, G. Guerrini, and I. Merlo. *T*_ODMG: An ODMG Compliant Temporal Object Model Supporting Multiple Granularity Management. *Information Systems*, 2003 *(to appear).*

[2] M. Bertolotto. Geometric Modeling of Spatial Entities at Multiple Levels of Resolution. PhD Thesis, Università degli Studi di Genova, 1998.

[3] C. Bettini, S. Jajodia, and X. Wang. Time Granularities in Databases, Data Mining, and Temporal Reasoning. Springer-Verlag, 2000.

[4] T. Bittner, and B. Smith. A Unified Theory of Granularity, Vagueness and Approximation. In *Proc. of COSIT Workshop on Spatial Vagueness, Uncertainty, and Granularity*, 2001.

[5] E. Camossi, M. Bertolotto, E. Bertino, and G. Guerrini. *ST*_ODMG: A Multigranular Spatiotemporal Extension of ODMG Model. Technical Report DISI-TR-03-09, Università degli Studi di Genova, 2003.

[6] R. Cattel, D. Barry, M. Berler, J. Eastman, D. Jordan, C. Russel, O. Schadow, T. Stanienda, and F. Velez. The Object Database Standard: ODMG 3.0. Morgan-Kaufmann, 1999.

[7] C. Chen, and C. Zaniolo. SQLST: A Spatio-Temporal Data Model and Query Language. In *Proc. of Int'l Conference on Conceptual Modeling/the Entity Relational Approach*, 2000.

[8] J. Chomicky, and P. Revesz. Parametric Spatiotemporal Objects. Periodico dell'Associazione Italiana per l'Intelligenza Artificiale. 14(1):41-47, 2001.

[9] M. Dunkham, K. Mason, J.G. Stell, and M.F. Worboys. A formal Approach to Imperfection in Geographic Information. *Computer, Environment and Urban Systems*, 25:89-103, 2001.

[10] T. Griffiths, A. Fernandes, N. Paton, K. Mason, B. Huang, M. Worboys, C. Johnsonon, and J.G. Stell. Tripod: A Comprehensive System for the Management of Spatial and Aspatial Historical Objects. In *Proc. of 9th ACM Symposium on Advances in Geographic Information Systems*, 2001.

[11] R.H. Güting, M.H. Bhölen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M. Shneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transaction On Database Systems*, 25:1-42, 2000.

[12] B. Huang, and C. Claramunt. STOQL: An ODMG-based Spatio-Temporal Object Model and Query Language. In *Proc. of 10th Int'l Symposium on Spatial Data Handling*, 2002.

[13] V. Katri, S. Ram, R.T. Snodgrass, and G. O'Brien. Supporting User Defined Granularities and Indeterminacy in a Spatiotemporal Conceptual Model. *Special Issue of Annals of Mathematics and Artificial Intelligence on Spatial and Temporal Granularity*, 36(1-2):195-232, 2002.

[14] G. Langran. Time in Geographic Information Systems. Taylor & Francis, 1992.

[15] R.B. McMaster, and K.S. Shea. Generalization in Digital Cartography. Association of American Geographer, 1992.

[16] J-C. Muller. Generalization of Spatial Databases. *Geographical Information Systems: Principles and applications*, D.J. Maguire, M.F. Goodchild, and D.W. Rhind. Longman, 1991.

[17] J-C. Muller, J.P. Lagrange, and R.Weibel (eds.) GIS and Generalization: methodology and practice. Taylor and Francis, 1995.

[18] Philippe Rigaux, Michel Scholl, and Agnès Voisard. Spatial Databases with Application to GIS. Morgan Kaufmann, Academic Press, 2002.

[19] J.G. Stell, and M. Worboys. Stratified Map Spaces: A Fomal Basis for Multi-Resolution Spatial Databases. In *Proc. of 8th Int'l Symposium on Spatial Data Handling*, 1998.

[20] R. Weibel, and G. Dutton. Generalizing Spatial Data and Dealing with Multiple Representations. *Geographical Information Systems: Principles, techniques, management and applications*, P.A. Longley, D.J. Maguire, M.F. Goodchild, and D.W. Rhind. John Wiley, 1999.

[21] M. Worboys. A Unified Model for Spatial and Temporal Information. *The Computer Journal*, 37(1):26-34, 1994.

[22] *http://inspire.jrc.it*

[23] *http://www.oracle.com*

[24] *http://www.opengis.org/*

[25] *http://www.postgresql.org*