

# Multi-granular Spatio-temporal Object Models: Concepts and Research Directions <sup>\*</sup>

E. Bertino<sup>1</sup>      E. Camossi<sup>2</sup>      M. Bertolotto<sup>2</sup>

<sup>1</sup> CERIAS - Purdue University, 250 N. University Street West Lafayette, Indiana,  
USA 47907-2066. Phone: +1 765 496-2399. Fax: +1 765 494-0739. e-mail:  
`bertino@cs.purdue.edu`

<sup>2</sup> School of Computer Science and Informatics - University College Dublin, Belfield,  
Dublin 4, Ireland. Phone: +353 (0)1 7162-944/913. Fax: +353 (0)1 2697-262. e-mail:  
`{elena.camossi,michela.bertolotto}@ucd.ie`

**Abstract.** The capability of representing *spatio-temporal objects* is fundamental when analysing and monitoring the changes in the spatial configuration of a geographical area over a period of time. An important requirement when managing spatio-temporal objects is the support for multiple granularities. In this paper we discuss how the modelling constructs of object data models can be extended for representing and querying multi-granular spatio-temporal objects. In particular, we describe object-oriented formalizations for granularities, granules, and multi-granular values, exploring the issues of value conversions. Furthermore, we formally define an object-oriented multi-granular query language, and discuss the dynamic adapting of multi-granular data. Finally, we illustrate the current open research issues of multi-granular spatio-temporal data handling.

## 1 Introduction

Many relevant application domains, including homeland security, environmental protection, geological and agricultural sciences, require modelling and managing spatial data objects and monitoring their evolution according to time. The capability of representing data objects with respect to both their spatial layout and their temporal evolution, referred to as *spatio-temporal objects*, is fundamental when analysing and monitoring the changes in the spatial configuration of a geographical area over a period of time. An important requirement when managing spatio-temporal objects is the support for multiple granularities. For example, when tracing modifications to spatial areas, the history of the areas under observation has to be maintained and retrieved at multiple temporal granularities (e.g., years, months, decades). When analysing large spatial datasets, one may

---

<sup>\*</sup> Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support. The work of Elena Camossi is supported by the Irish Research Council for Science, Engineering and Technology.

need to zoom-in and zoom-out from the dataset under analysis according to different spatial granularities (e.g., meters, kilometres, feet, yards).

Granularities intuitively represent the units of measure of a dataset, and may be defined on all data dimensions (i.e., space and time for spatio-temporal data). For each dimension, a connected set of granularities must be defined, and the different sets are independent. The choice of proper granularities allows the system to store a minimal amount of data, according to the most appropriate levels of detail. In many applications different granularities may exist, neither of which is inherently better than the others. Therefore, a database system for such applications should support a wide range of granularities and allow to define their own application-specific granularities. Moreover, because the selection of attribute granularities is based on a trade-off between application efficiency and modelling requirements and this trade-off may change over time, the model at hand should support the ability to dynamically set and change the spatio-temporal granularity. For example, in a spatio-temporal database for environmental monitoring, the collection of meteorological parameters like the amount of rainfall, the strength and direction of the wind, the value of atmospheric pressure, must be collected more frequently in the presence of exceptional events like hurricanes and storms. Furthermore, such a granularity modification may involve only specific geographical areas (e.g., those affected by the phenomenon), and is required for limited periods of time (e.g., the time when the phenomenon occurs).

However, even though research in the spatial and temporal data management systems has resulted in many spatial and temporal models, these models are in most cases extensions of the relational data model [17] and are unable to directly represent crucial modelling features of spatio-temporal data objects. As a result, the applications have to implement and maintain mappings between the spatio-temporal objects of interest and low-level data and are unable to efficiently support multiple granularities of object representations in both space and time. We believe that an object data model, because of its modelling features such as complex data types and methods, is better suited for addressing such requirements. However, even with such a model, modelling and managing multiple granularities is not trivial and extensions are required. However, the natural extensibility of object models makes easier developing these extensions. The goal of this paper is to explore in details the notion of multiple granularities for spatio-temporal objects and show how the modelling constructs of an object data model can be extended for representing and querying these objects.

More specifically in the paper we discuss a number of issues, including:

- *The notion of granularities in space and time.* How do we represent them? How do we relate granularities and maintain a granularity lattice? How do we extend the granularity sets and, at the time, preserve granularities relationships?
- *The notion of temporal, spatial and spatio-temporal values.* How do we support granularity conversions? How do we preserve data usability and reduce uncertainty on converted values? Can we combine concepts like topologi-

- cally consistent transformation, probability distributions, invertibility and quasi-invertibility properties to reduce uncertainty?
- *Navigating and querying through multi-granular data.* How do multi-granularity impact object navigation and value comparison? How do we express the access to spatio-temporal object values?
- *Dynamic adapting of multi-granular data.* How do we refine object attributes? Are conventional object specialization models adequate? How do we support object evolution with respect to the object state and the object granularities?

In the discussion, we will refer to the *ST*-ODMG and the *ST*<sup>2</sup>-ODMGe object data models, which have been specifically defined for modelling and querying spatio-temporal objects with multiple dynamically varying granularities [11, 13] and thus can illustrate solutions to some of the above challenges. Both models have been defined as extension of the ODMG model [15], the standard de facto for object-oriented databases; *ST*-ODMG has been recently extended for application to an object-relational data model [12].

The rest of the paper is organized as follows. We first discuss related work on modelling approaches for multi-granularity. We then illustrate how multi-granular spatio-temporal objects are represented and queried in the *ST*-ODMG model. In Section 5 we describe two different solutions for evolving multi-granular objects, i.e., attribute redefinition and dynamic objects. Afterwards, in Section 6 we discuss the issues of spatio-temporal multi-granularity. Finally, we conclude the paper illustrating future research directions.

## 2 Related Work

Spatio-temporal multi-granularity has been mostly investigated by separately considering the temporal and the spatial domains. The pioneering research work on temporal granularities is by Anderson [2]. A consensus among the different disciplines interested in temporal granularity representation is the formalization by Bettini et al. [9], who give a comprehensive discussion on temporal granularities for databases, data mining, and temporal reasoning.

Temporal granularity issues related to temporal databases have been investigated both for the relational and the object-oriented data models [30, 20]. The introduction of multiple temporal granularities in an object-oriented data model poses additional issues with respect to the relational context, due to the semantic richness of such a model. Bertino et al. [6] investigate the impact of temporal granularities in an object-oriented model compliant with the ODMG standard.

The representation of data at multiple levels of details, that is, at multiple granularities, is a topic of relevant interest also in modelling spatial entities. In the context of Geographical Information Systems (GIS), much research addresses the development of data models for the multiresolution representation of geographic maps [25, 31].

Research on multiple resolutions addresses in particular model-oriented generalization [32], which applies techniques used in cartography for representing spatial data at different levels of abstraction, by taking into account also the

semantics of data and some notion of consistency to preserve data usability, as, for example, the preservation of topological relationships.

Other proposals address specifically the multi-granular representation of spatio-temporal data [16, 11, 3]. Griffiths et al. [23] define the Tripod spatio-historical model. It includes a definition for granular histories at different granularities. However, no operators are provided to convert multi-granular data, but the histories are always internally represented at the *chronon* [24] granularity. Katri et al. [26] define an annotation-model for the specification of spatio-temporal data at multiple granularities. Such a granularity formalization relies on the concepts of temporal indeterminacy [21] and spatial imprecision [19]. However, the resulting model and the granularity systems are effective only for data specification, because the conversion from a granularity to another is completely left to the user. The European project *MurMur* [27] addresses multiple resolutions through multiple representations, supporting *perceptions*, which include different point of views and spatial resolutions. Belussi et al. [3] define spatio-temporal granularities as historical evolution of spatial granules, to ease the search of valid spatial granules in a given instant. Their approach relies on the mapping of spatial multiple granularities and granules onto graph structures (multidigraphs), which encompass labelling functions for granules and their mutual (topological) relationships, disregarding value conversions.

### 3 Modelling Multi-granular Spatio-temporal data

The exploitation of multiple granularities for spatio-temporal data entails the definition of a multi-granular spatio-temporal type system and of conversions to represent spatio-temporal data at different granularities. These elements, defined based on formal definition for granularities, granules, and granular elements, are the key components of a multi-granular algebra for representing and managing multi-granular spatio-temporal data. Relying on such an algebra, a multi-granular spatio-temporal query language, which will be discussed in the next section, may be designed as well.

In what follows, we discuss the formal definition of granularities, types, values, and multi-granular conversions we adopt in the design of *ST-ODMG* [11]. We refer the interested reader to [14] for a more complete discussion of granularity implementation challenges.

#### 3.1 Spatial and Temporal Granularities

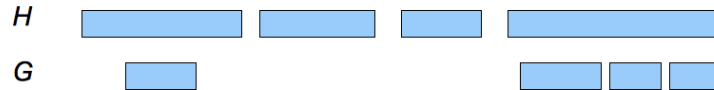
According to [9], temporal granularities may be formally represented as mappings from an ordered index set  $\mathcal{IS}$  to the power set of the temporal domain (i.e.,  $\mathcal{TIME}$ ), which is totally ordered. Both Khatri et al. [26] and Camossi et al. [11] apply the same definition to spatial granularities, which are defined as mapping from an index set  $\mathcal{IS}$  to subset of  $\mathcal{SPACE}$ , the spatial domain.  $\mathcal{SPACE}$  is two-dimensional (that is, a proper subset of  $R^2$ ). Spatial granularities can include

2-dimensional granules (e.g., units of area:  $m^2$ , *acre*, etc.; administrative boundaries classifications: *municipalities*, *countries*, etc.), or 1-dimensional granules (e.g., measures of length: *km*, *mile*, etc.; map scales: 1 : 24000, 1 : 62500, etc.). For instance, *days*, *weeks*, *years* are temporal granularities; *meters*, *kilometers*, *feet*, *yards*, *provinces* and *countries* are spatial granularities.

Each subset of the temporal and spatial domains corresponding to a single granularity mapping is referred to as a temporal or spatial *granule*, i.e., given a granularity  $G$  and an index  $i \in \mathcal{IS}$ ,  $G(i)$  is a granule of  $G$  that identifies a subset of the corresponding domain. Granules are used to specify the *valid* spatio-temporal bounds on attribute values, as well as the temporal occurrence of database events. For instance, we can say that a value reporting the measure of the daily temperature in Rome is defined for the first and the second day of January. The granules of interest for this example can be identified by three textual labels: “01/01”, “02/01”, and “*Rome*”, that respectively identify two temporal and one spatial granules. The interior of granules of the same granularity cannot overlap<sup>1</sup>. Moreover, non-empty granules of the same temporal granularity must preserve the order of the temporal domain.

Sets of temporal or spatial granules expressed at the same granularity are referred to as *temporal* or *spatial elements* [13], respectively. An element at granularity  $G$  is denoted as  $\mathcal{T}^G$ . For instance,  $\{1999, 2000, 2001\}^{years}$  is a temporal element at granularity *years*, and  $\{\text{Rome}, \text{Berlin}\}^{municipalities}$  is a spatial element at granularity *municipalities*.

Different granularities provide different partitions of their domain of reference. The reason is that diverse relationships may hold among granularities, depending on the inclusion and the overlapping of granules [9]. For instance, in *ST.ODMG* [11] we assume that spatial and temporal granularities are related by the *finer-than* relationship: given two granularities  $G$  and  $H$  such that  $G$  is finer-than  $H$ , every granule  $g$  of  $G$  is properly included in a granule  $h$  of the coarser granularity  $H$  (cf. Fig. 1). If  $G$  is finer-than  $H$ , we also say that  $H$  is coarser-than  $G$ . For example, temporal granularity *days* is finer-than *months*, and granularity *months* is finer-than *years*. Likewise, spatial granularity *municipalities* is finer-than *countries*.



**Fig. 1.** The *finer-than* relationship:  $G$  is finer-than  $H$ .

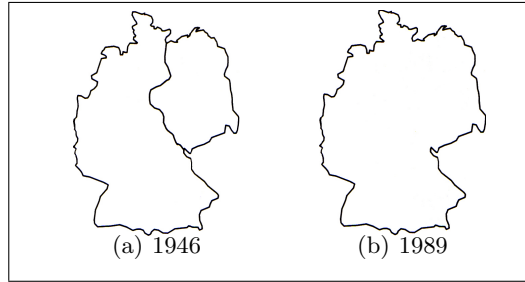
<sup>1</sup> Temporal granules, according to the definition by Bettini et al. [8], do not overlap, while spatial granules can overlap on the granule boundaries.

Relationships among different granularities are fundamental for enabling the comparison of multi-granular values in queries. For instance, in a query we might require to compare the values of seasonal sales of two similar products, one stored at spatial granularity *countries* and one at temporal granularity *provinces*, to decide which one to sell at a shop chain. To perform a meaningful comparison, these values have to be expressed at the same spatial granularity. In the following sections we will describe granularity conversions supported by *ST.ODMG* for converting granular values at different temporal and spatial granularities related by finer-than. In this case, we observe that *provinces* is finer-than *countries*; therefore we may likely apply some conversions to these values (e.g., the value at granularity *countries* may be split among the different provinces of each country or viceversa). By contrast, if the granularities were, for instance, *feet* and *meters*, these values could not be directly converted (we are not able to represent granule portions), but we have to convert both values to a common representation, different from both of them (e.g., *μms*, which is finer-than both granularities). Therefore, Given two multi-granular values, one at granularity  $G$  and one at granularity  $H$  such that  $G$  and  $H$  are not directly related by finer-than, such values may be compared if the two values may be represented (i.e., converted) at the same granularity  $K$ , that is finer-than  $G$  and  $H$ .  $K$  is chosen as the granularity that minimizes the number of conversions applied. If  $K$  is the coarsest, among the granularities finer-than  $G$  and  $H$ ,  $K$  is referred to as the *greatest lower bound (GLB)* of  $G$  and  $H$ .

Furthermore, by relying on granularity relationships, we may design efficient representations for granularities. For instance, we may implement granules of temporal granularity *years* relying on the representation provided for *months* and exploiting the uniform relationship between these two granularities. indeed, the mapping onto the temporal domain of a given granule of *years* is obtained retrieving the mappings of the twelve *months* of that year. Iterating the same technique onto the set of temporal granularity, we map the representation of most of the granularity onto the most finer one the model assumes (e.g., *mms*), and avoid the exhaustive mapping onto the temporal domain for most of the granularities (see [14] for further details).

### 3.2 Multi-granular Types, Values and Conversions

Beyond the conventional database values, a spatio-temporal database schema can include spatial, temporal, and spatio-temporal values. Multi-granular values in *ST.ODMG* are defined as partial functions from the set of granules of the corresponding granularity(ies) to the set of values of a conventional (i.e., literal or object types without any spatio-temporal capability) or geometric (i.e., two- and three-dimensional vector features) inner type. *ST.ODMG* provides two multi-granular parametric types: *Spatial<sub>SG</sub>(σ)* and *Temporal<sub>TG</sub>(τ)*, where *SG* and *TG* are a spatial and a temporal granularity, respectively;  $\sigma$  is a conventional or a geometric type;  $\tau$  is a conventional or a *Spatial* type. These types may be functionally combined to define multi-granular spatio-temporal types, as in the following example.



**Fig. 2.** A spatio-temporal geometric value

*Example 1.* Suppose a class **Europe** is defined to describe geo-political properties of European countries. The following is an example of a spatio-temporal value storing some of the names of the Heads of Government of European countries. Its type is  $Temporal_{years}(Spatial_{countries}(\mathbf{string}))$ .

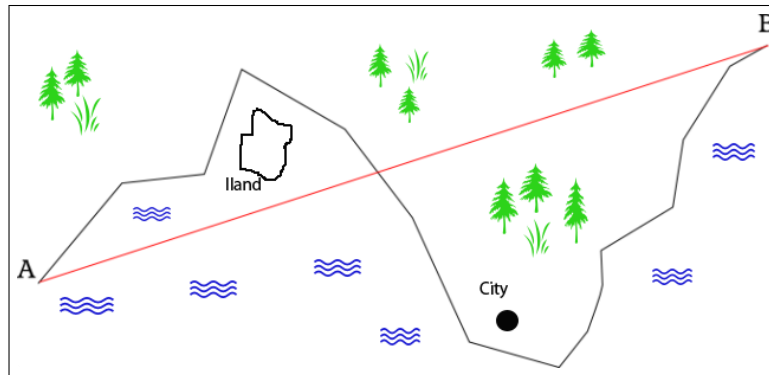
$$v = \{ \langle 2007, \{ \langle \text{France}, 'F. Fillon' \rangle, \langle \text{Germany}, 'A. Merkel' \rangle \}^{countries} \rangle, \langle 2008, \{ \langle \text{France}, 'N. Sarkozy' \rangle, \langle \text{Germany}, 'A. Merkel' \rangle \}^{countries} \rangle \}^{years}.$$

In Fig. 2, a value of type  $Temporal_{years}(Spatial_{countries}(\mathbf{set}(\mathbf{Polygon})))$  illustrates the historical changes in the German political boundaries: each country is represented through a polygon or a closed polyline.

To improve or reduce the level of detail of a multi-granular value, the value has to be converted to a different granularity. To address this requirement, in *ST-ODMG* we introduce the notion of *granularity conversions*, which include temporal and spatial coercion [6] and refinement [4] functions. We note here that the notions of coercion and refinement functions, that are basic notions in the object-oriented paradigm, directly address the requirement of spatio-temporal value conversion from a conceptual point of view. However, in a spatio-temporal setting, these functions must account for the additional semantics provided by granularities.

An important issue in the use of spatio-temporal coercion and refinement functions is represented by data consistency. For instance, if one first coerce a value  $v$  from a spatial granularity  $G$  into a value  $v'$  at a spatial granularity  $H$ , one would expect the relationships  $v$  has with other spatial objects be preserved by  $v'$ . To address this issue, model-oriented and cartographic map generalisation operators that guarantee topological consistency [7, 28], an essential property for data usability, may be applied. For example, *merge* operators merge adjacent features of the same dimension into a single one, while *splitting* operators subdivide single features in adjacent features of the same dimension. Other operators perform *contraction* and *thinning* (whose inverse is *expansion*); *abstraction* and *simplification* (whose inverse is *addition*). Their application avoids situation like the one shown in Figure 3, where a non-topologically consistent line simplification algorithm [18] is applied to coarse a coast line in a map (in black the original coast, in red the coarser one). Such a simplification would require a post-process

revision, to correct the location of the island, which has been incorporated into the land, and of the city, which has been moved into the sea.



**Fig. 3.** Topologically inconsistent geometric transformation.

The *ST*-ODMG model also provides operators for converting spatio-temporal quantitative (i.e., non-geometrical) attribute values. These operators perform *selection* (e.g., **projection**, **main**, **first**), and *aggregation* (e.g., **sum**, **average**) to convert values to coarser representation; their inverse functions, *restriction* and *splitting*, convert attribute values to finer representations, according to downward hereditary property [29] or according to a probability distribution, respectively.

Granularity conversions in *ST*-ODMG have been proven to return legal values of the *ST*-ODMG type system [11]. Conversions that generalize geometric attribute values to coarser spatial granularities have been proven preserve the semantics of the spatio-temporal data represented [11]. Furthermore, the conversions we provided for converting spatio-temporal values at finer granularities address indeterminacy [21] and imprecision [19] that always affect this type of conversion (see also [4] for a more comprehensive discussion on *invertibility* and *quasi-invertibility* of multi-granular values).

## 4 Querying Multi-granular Spatio-temporal data

A model for multi-granular spatio-temporal queries in *ST*-ODMG has also been defined [13]. The language extends the value comparison and object navigation paradigms of OQL [15] to support multi-granular spatio-temporal values. The key concept of the language is the *multi-granular spatio-temporal path expression* (*MST Path-Expr*), which extends the conventional notion of object-oriented path expression to multi-granular spatio-temporal values. In a *MST Path-Exprs* the access to multi-granular attribute values is specified by referring to portions of spatio-temporal domain, through the use of a specific operator ( $\downarrow$ ).

References to the spatio-temporal domain are given explicitly through spatial and temporal *elements* we introduced in the previous section, as illustrated in the following example.

*Example 2.* Given the spatio-temporal value of Example 1, representing the name of European Heads of Government, the temporal path expression  $v \downarrow \{2007\}^{years}$  returns the spatial value:

$\{(France, 'F. Fillon'), (Germany, 'A. Merkel')\}^{countries}$ .

By contrast, the spatial path expression  $v \downarrow \{France\}^{countries}$  returns the temporal value:

$\{(2007, 'F. Fillon'), (2008, 'N. Sarkozy')\}^{years}$ . □

In *MST Path-Exprs*, we also make use of multi-granular spatio-temporal *expressions* (*Exprs*), which are implicit representations of spatio-temporal elements. *Exprs* are given as conditions that are evaluated on database objects. They result in temporal and spatial elements, which intuitively represent *when* and *where* such conditions are satisfied. Conditions are specified through temporal and spatial variations of conventional comparison operators (e.g.,  $=_T$ ,  $<>_S$ ) and binary topological relationships as defined by Egenhofer and Franzosa [22] (e.g.,  $equals_T$ ,  $overlaps_S$ ).

*Example 3.* Given the spatio-temporal value of Example 1, the temporal expression  $v =_T$  'N. Sarkozy' returns the temporal element  $\{2008\}^{years}$ , whereas the spatial expression  $v =_S$  'N. Sarkozy' returns  $\{France\}^{countries}$ . □

Queries have the usual OQL *select-from-where* form. *MST Path-Exprs* are applied in the target list to specify the data to retrieve, and in the where clause to express the conditions against multi-granular spatio-temporal objects. Whenever *MST Path-Exprs* involve different granularities, during their evaluation granularity conversions described in the previous section are applied.

*Example 4.* Given class **Europe** of Example 1, and given a class **Nation** describing the properties of interest for a single country, the following query retrieves the name of the Head of Government of West Germany in 1980:

```
select e.head_of_government ↓ {1980}^{years}
from Europe e, Nation n
where n.name ↓ {1980}^{years} = 'West Germany'.
```

It returns the name 'H. Shmidt'. By contrast, the following query retrieves when Angela Merkel was Head of Government of Germany:

```
select e.head_of_government =_T 'A. Merkel'
from Europe e, Nation n
where (n.name ↓ e.head_of_government =_T 'A. Merkel') = 'Germany'.
```

It returns the temporal element representing the period from 2005 and 2009. □

## 5 Adaptive Spatio-temporal Multi-granular Models

Adaptivity support is a crucial requirement for almost all applications we may think of. In a spatio-temporal setting with multiple granularities, an added dimension to the problem of adaptation is represented by the *evolution* of attribute granularity. To date this problem has not been much investigated. In what follows, we discuss two preliminary solutions to adapt attribute granularities: object-oriented attributes redefinition, and evolution models. The first approach, which is discussed in [5], provides a weaker granularity adaptability, which is limited to attribute redefinition along the inheritance hierarchy; then, granularity modifications are pre-arranged in the database schema. Conversely, the  $ST^2$ \_ODMG model [10], which is described at the end of the section, adopts a flexible solution by which run time evolutions may be specified and executed.

### 5.1 Multi-granular Attribute Redefinition

The idea behind multi-granular attribute refinement is that the granularity at which an attribute value is stored can be changed in a subclass, to better reflect the application evolution needs. In the subclass the attribute values may be maintained at a coarser or at a finer level of detail. For instance, if at the superclass only the monthly values are recorded, in the subclass the daily changes can be maintained, improving the level of detail for the attribute. By contrast, we may reduce the detail coarsening the attribute value in the subclass.

The most critical requirement in attribute refinement is to preserve object substitutability. Whenever an object instance of a subclass is found in a context where a superclass object was expected, its attribute values must be converted to the expected granularity, leaving the whole procedure completely transparent to the user. Therefore, multi-granular conversions including both coercion and refinement functions, such those described in Section 3.2, must be provided by a multi-granular model. Supplying a variety of conversions with different semantics enables to choose, for each attribute and situation, the conversion that better reflects the attribute semantics.

Substitutability impacts both attribute accesses and updates. In case of object access, granularity conversions are used to compute the value to be considered in the superclass, given the value of the attribute in the subclass. By contrast, in case of object updates, granularity conversions are applied to convert the value to assign to the granularity required in the subclass.

*Example 5.* The following multi-granular spatio-temporal schema includes an example of multi-granular attribute refinement<sup>2</sup>. We give a partial definition for class **Nation**, reporting the specification of attribute **population**, which stores

---

<sup>2</sup> The syntax we use in this example has been first introduced in  $ST$ \_ODMG: it extends the ODMG Data Definition Language to spatio-temporal multi-granularity. The same syntax has been further extended in [5] to support attribute refinement.

the daily updates of the amount of population recorded in each municipality of a country.

```
class Nation (...) {
    attribute Temporaldays(Spatialmunicipalities(int)) population;
    ...
};
```

Then, we define a class `NationStatistic`, which *extends* `Nation`, to collect statistical information on the countries in the database. In particular, attribute `population` is refined at temporal granularity *years* and at spatial granularity *countries*.

```
class NationStatistic extends Nation (...) {
    ref attribute Temporalyears(Spatialcountries(int)) population
        < splitcountries→municipalities, summunicipalities→countries >,
        < restryears→days, avgdays→years > }
    ...
};
```

Two pairs of granularity conversions are specified for this attribute: the first refers to the spatial refinement, whereas the second deals with the temporal refinement. In each pair of conversions  $\langle af, uf \rangle$ , *af* is the granularity conversion used to access the attribute value from an object that at compile time has type `Nation`: the attribute value has to be converted from the sub-class granularity (e.g., *countries*) to that used in the super-class (e.g., *municipalities*). In this example both the spatial and the temporal granularities have been refined in the sub-class, therefore we have two refinement conversion functions to use in the access: both `split` and `restr` (i.e., split and restriction) granularity conversions have to be applied to the value, that is converted from granularities *years* and *countries* to finer granularities *days* and *municipalities*. *uf* conversions are applied when updating this attribute from an object whose run-time type is `NationStatistic`, while at compile time it has type `Nation`: in this case, the conversions `sum` and `avg` (i.e., sum and average) are applied to coarser the finer value to granularities *years* and *countries*.  $\square$

To preserve data consistency, both compile and run-time checks may be applied. At compile time, the consistency of the database schema must be verified, checking first that the granularities in the superclass and in the subclass are related by some granularity relationship. In *ST-ODMG* we consider the finer-than relationship, but other relationships may be applied as well. Then, we have to check that two *inverse* or *quasi-inverse* [4] granularity conversions have been specified, one to use for the attribute access and one for the attribute update. However, at run-time we may allow one to apply a different granularity conversion for the attribute access, whenever the user needs a different conversion semantics, and also in this case the granularity conversion must be compliant with the attribute refinement.

## 5.2 Evolutions of Spatio-temporal Multi-granular Objects

Being able to dynamically adapt the spatial and temporal granularities to respond to dynamic events and situations and to reflect changes in data significance is crucial in many contexts: e.g., periodic phenomena, modifications to attribute values, operation execution, data aging or privacy restrictions. Specific operations required for supporting dynamic adaptation of granularity include: 1) *granularity evolution*, which aggregates existing detailed data at a coarser granularity (e.g., older data that may be stored for future reference), or even refines information at a finer granularity (e.g., in data analysis); 2) *granularity acquisition*, which changes at run-time the granularity used when inserting new values in the database; 3) *value deletion*, which removes attribute values from the database, whenever they are no longer useful at a given granularity (e.g., detailed data).

The recently defined  $ST^2\_ODMGe$  (Spatio-(Bi)Temporal ODMG supporting Evolutions) addresses these requirement by supporting the modification of the granularities used in attribute definitions, and the deletion of attribute values at run-time. Evolutions have the form: **ON** *Event* [**IF** *Condition*] **DO** *Action*. Example of events are: update, delete, etc., that is occurrences that modify the database state, including evolution actions, and may have a periodic or an extemporary behaviour. Conditions are specified against database attribute values, and include also periodic checks, evaluated on valid time. Finally, evolution actions are sequences of operations that may modify the attribute granularities and delete the attribute values.

Evolutions are defined and executed at run-time and conform to the execution model of active databases. Given an instance of an  $ST^2\_ODMGe$  database and a set of evolutions specified for it, the database is continuously monitored. The execution of database transactions modifies the database state and triggers the evolutions whose events refer to such transactions. Therefore, the corresponding conditions are evaluated. For those triggered evolutions whose conditions evaluate to TRUE, the corresponding actions are executed. As a consequence, the database state (or schema, in case of granularity acquisition) may be modified.

*Example 6.* Given class **Nation** we defined in Example 5, the following are two examples of evolutions we may specify to periodically obtain summarized values of the amount population of the countries in the database.

```
ON update Nation.population < days, municipalities >
IF every 1years/VT
DO evolve < days, municipalities > to < days, countries > using
    summunicipalities→countries, splitcountries→municipalities;

ON update Nation.population < days, countries >
DO evolve < days, countries > to < years, countries > using
```

```
avgdays→years, restryears→days;
```

The first evolution is triggered by the updates of attribute `population` as originally defined in class `Nation`, i.e., at temporal granularity `days` and at spatial granularity `municipalities`. Once one years of values (i.e.,  $\mathcal{VT}$  denotes valid time) have been recorded for this value, the evolution is executed, and the first time a new value is created for this attribute: specifically, a new *granularity level* (see [10]) at granularity `days` and `countries` is defined for attribute `population`. For each country, it stores the daily amount of population, given as the sum of the population of every municipality in the country. This evolution is executed periodically, every  $1_{\mathcal{VT}}^{years}$ . Every time this new granularity level is updated (i.e., once a year), the second evolution is triggered. It results in the creation of a new granularity level, at granularity `years` and `countries`, that stores the annual amount of population of the country. This value is obtained as the average of the daily amount stored in the previous granularity level.

Consequently to the execution of evolutions, the run-time type of `population` is  $Temporal_{days}(Spatial_{municipalities}(int)) \times Temporal_{days}(Spatial_{countries}(int)) \times Temporal_{years}(Spatial_{countries}(int))$ . Note that the last granularity level has the same type of attribute `population` we defined in class `NationStatistic` of Example 5. However, in this case the value is automatically computed, and belongs to the same object of type `Nation` it refers to. By contrast, in the case of Example 5, for each country at least two objects has to be created to maintain the same information.  $\square$

After the execution of evolutions, the run-time type of attribute values are Cartesian products of multi-granular types as defined in Section 3. Therefore at run-time the state of objects in the database is no longer consistent with their class definition. We formally revisited the notion of object consistency, weakening the conditions on attribute values and on objects spatio-temporal lifespan to include the side-effects of evolutions. In particular, we require that each evolution specification includes a pair of *inverse* and *quasi inverse* granularity conversions, enabling to navigate among portions of the same attribute value expressed at different granularities.

Moreover, we take advantage of attribute run-time values at multiple granularities to enhance the access strategies to multi-granular values. We demonstrate that, under certain assumptions, object access is invariant to the execution of evolutions. In particular, the stored information may be preserved after value deletion, because the same value may be present in the database at a different granularity, and retrieved when needed. Furthermore, object access may benefit from evolutions with respect to both effectiveness and efficiency. The values resulting from the execution of granularity conversions are already materialized in the database, thus improving the performance of queries involving aggregates and granularity refinement. The existence in the database of values at different granularities makes it possible to apply two different strategies for object access. Such strategies optimize, respectively, *execution efficiency*, minimizing the

retrieval time, and *result accuracy*, minimizing the indeterminacy of granular values.

## 6 Open Research Challenges

Even though *ST*\_ODMG and *ST*<sup>2</sup>\_ODMG<sub>e</sub> represent some important initial steps towards the problem of developing adaptive multi-granular spatio-temporal object models and systems, many open research challenges are still open.

***Foundations of formal models and type systems.*** Because of the complexity of an adaptive multi-granular spatio-temporal object model, it is crucial for formal definitions exist for both static and dynamic features of the model. Suitable type systems, that are also relevant for object-oriented programming languages manipulating spatio-temporal objects, need to be developed, perhaps as extension of conventional type systems. For example, when declaring a variable one may have to specify, in addition to the variable type, the spatio-temporal granularity of the variable. Assignments of a value to a variable must then take into account not only the types of the value and the variable, but also their spatio-temporal granularities. Static type checking of programs would then need to be extended by, for example, allowing such an assignment provided that a conversion function be defined for the granularities of the value and the variable, respectively. Consistency properties, such as assuring the correct combination of spatial and temporal type constructors, would also need to be devised and techniques for their analysis be devised.

***Analysis tools for evolutions.*** If evolution specifications are formulated according to the active database paradigm, it is important not only that object-oriented models and systems be equipped with triggers, but also that tools for the analysis of these “evolution” triggers be supported to detect non-terminating executions and indeterministic executions. Note that such issues have been extensively investigated in the area of active DBMS and no good solutions exist. However because we deal with a specialized domain, that is, the evolution of granularities, effective solutions to these issues could perhaps be found.

***Implementation strategies.*** Efficient and comprehensive implementations are crucial. Several alternatives can be investigated including implementation of the required features as class libraries on top of an existing object DBMS (ODBMS) and extensions to the engine of the ODBMS. Both approaches have shortcomings. The first approach may not be able to support all required features and also have performance problems, in that it may not make it possible to include specialized indexing techniques or query optimization techniques. The second approach may require extensive implementation efforts and may still not able to cover all required features, especially the ones depending on the application domain, like specialized spatial conversion operators. Perhaps the best approach

would be to extend the engine of an existing ODBMS with some basic functions, supporting for example the organization of value domains according to multiple granularities, by at the same time allowing the applications to define their own application-depending granularities and specialized conversion functions, perhaps through the use of methods.

***Multi-granular volumetric objects.*** The geographic information systems and spatial community is showing a growing interest in systems for managing three-dimensional (3D) data. This is demonstrated also by commercial GIS and Spatial DBMS products, that offer support for representing and analysing volumetric information. However, those products do not offer instruments for dealing with multi-granular information. One of the main issues in supporting multi-granularity for volumetric information is the definition of meaningful multi-granular conversions, able at the same time to preserve topological consistency. Moreover, because the computational complexity of the analysis and conversion algorithms is very high and the explicit storage of spatial relationships result in huge data sets, techniques are needed to optimize both temporal and storage costs.

***Multi-granular exploitation of legacy data.*** When analysing existing data sets, one may require to make explicit the granularity according to which data are represented, in particular when integrating or matching data sets from heterogeneous sources. The automatic exploitation of the level of detail of a data set is a challenging problem that so far has not been investigated. Semantics driven methods that make use of both implicit and explicit semantics of data, such the ones discussed by Albertoni et al. [1] for the extraction of the levels of detail used in data representation, may provide valid suggestions to develop a solution to this problem.

## 7 Conclusions

In this paper we have discussed concepts and approaches for handling multi-granular spatio-temporal data. In our discussion we refer to recent work on spatio-temporal multi-granularity, and illustrate the design of the *ST*-ODMG and the *ST*<sup>2</sup>-ODMG<sub>e</sub> models, which extend the ODMG model to provide multi-granular spatio-temporal support. In particular we have discussed the formal design for several key concepts encompassing: granularities, granules, multi-granular values, multi-granular conversions, multi-granular spatio-temporal querying, multi-granular attribute refinement and evolutions. Furthermore, we have discussed some open problems of interest for multi-granular spatio-temporal data management, including the definition of formal multi-granular models and type systems; the development of analysis tool for evolution models; the definition of implementation strategies; the design of multi-granular volumetric data models; the handling of legacy data. Many other challenges can be devised when

considering different application domains. Addressing these challenges typically requires extensible data management systems, like object DBMS, that must however be equipped with specialized features in order to be able to support complex application-specific object models, like the dynamic multi-granular spatio-temporal data we have discussed in this paper.

## References

1. R. Albertoni, E. Camossi, M. De Martino, F. Giannini, and M. Monti. Semantic granularity for the semantic web. In R. Meersman, Z. Tari, and P. Herrero, editors, *Proceedings of the IFIP WG 2.12 and WG 12.4 International Workshop on Web Semantics, OTM Workshops 2006 - Volume 2*, volume 4278 of *Lecture Notes in Computer Science*, pages 1863–1872. Springer-Verlag Berlin Heidelberg, 2006.
2. T.L. Anderson. Modeling time at the conceptual level. In *Proceedings of the International Conference on Databases: Improving Usability and Responsiveness*, pages 273–297, 1982.
3. A. Belussi, C. Combi, and G. Pozzani. Towards a Formal Framework for Spatio-Temporal Granularities. In *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning*, pages 49–53. IEEE Computer Society, 2008.
4. E. Bertino, E. Camossi, and G. Guerrini. Access to Multigranular Temporal Objects. In *Proceedings of the 6th Int. Conference On Flexible Query Answering Systems (FQAS 2004)*, number 3055 in *Lecture Notes in Artificial Intelligence*, pages 320–333. Springer-Verlag Berlin Heidelberg, 2004.
5. E. Bertino, E. Camossi, and G. Guerrini. Attribute refinement in a multigranular temporal object data model. Technical report, 2009.
6. E. Bertino, E. Ferrari, G. Guerrini, and I. Merlo. T\_ODMG: An ODMG Compliant Temporal Object Model Supporting Multiple Granularity Management. *Information Systems*, 28(8):885–927, 2003.
7. M. Bertolotto. *Geometric Modeling of Spatial Entities at Multiple Levels of Resolution*. PhD thesis, Università degli Studi di Genova, 1998.
8. C. Bettini, C.E. Dyreson, W.S. Evans, and R.T. Snodgrass. A Glossary of Time Granularity Concepts. In *Proceedings of Temporal Databases: Research and Practice*, volume 1399 of *Lecture Notes in Computer Science*, pages 406–413, 1998.
9. C. Bettini, S. Jajodia, and X. Wang. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer-Verlag Berlin Heidelberg, 2000.
10. E. Camossi, E. Bertino, G. Guerrini, and M. Bertolotto. Adaptive management of multigranular spatio-temporal object attributes. Technical report, 2009.
11. E. Camossi, M. Bertolotto, and E. Bertino. A multigranular object-oriented framework supporting spatio-temporal granularity conversions. *International Journal of Geographical Information Science*, 20(5):511–534, 2006.
12. E. Camossi, M. Bertolotto, and E. Bertino. Multigranular Spatio-temporal Models: Implementation Challenges. In *Proceedings of the 16<sup>th</sup> International Symposium on Advances in Geographic Information Systems (ACM GIS08)*. ACM, November 2008.
13. E. Camossi, M. Bertolotto, and E. Bertino. Querying multi-granular spatio-temporal objects. In S.S. Bhowmick, J. Küng, and R. Wagner, editors, *Proceedings 19<sup>th</sup> International Conference on Database and Expert Systems Application*, volume 5181 of *Lecture Notes in Computer Science*, pages 390–403. Springer-Verlag Berlin Heidelberg, September 2008.

14. E. Camossi, M. Bertolotto, and E. Bertino. Implementation challenges in spatio-temporal multi-granularity. Technical report, 2009.
15. R. Cattell, 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, Academic Press, 2000.
16. C. Claramunt and B. Jiang. Hierarchical Reasoning in Time e Space. In *Proceedings of 9<sup>th</sup> International Symposium on Spatial Data Handling*, pages 41–51, 2000.
17. E.F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, June 1970.
18. D.H. Douglas and T.K. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Line or its Caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.
19. M. Dukham, 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.
20. C.E. Dyreson, W.S. Evans, H. Lin, and R.T. Snodgrass. Efficiently Supporting Temporal Granularities. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):568–587, 2000.
21. C.E. Dyreson and R.T. Snodgrass. Supporting Valid-time Indeterminacy. *ACM Transactions on Database Systems*, 23(1):1–57, 1998.
22. M.J. Egenhofer and R.D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Science*, 5(2):161–174, 1991.
23. T. Griffiths, A.A.A. Fernandes, N.W. Paton, and R. Barr. The Tripod spatio-historical data model. *Data Knowledge and Engineering*, 49(1):23–65, 2004.
24. C.S. Jensen, C.E. Dyreson, M. Bohlen, J. Clifford, and al. A Consensus Glossary of Temporal Database Concepts. In O. Etzion, S. Jajodia, and S.Sripada, editors, *Temporal Databases: Research and Practice*, number 1399 in Lecture Notes in Computer Science, pages 366–405. Springer-Verlag Berlin Heidelberg, 1998.
25. B. Jiang and C. Claramunt. A structural approach to the model generalization of a urban street network. *Geoinformatica*, 8(2):157–171, 2004.
26. V. Khatri, S. Ram, R.T. Snodgrass, and G. O’Brien. Supporting User Defined Granularities and Indeterminacy in a Spatiotemporal Conceptual Model. *Annals of Mathematics and Artificial Intelligence, Special Issue on Spatial and Temporal Granularity*, 36(1-2):195–232, 2002.
27. C. Parent, S. Spaccapietra, and E. Zimányi. The murmur project: Modeling and querying multi-representation spatio-temporal databases. *Information System*, 31(8):733–769, 2006.
28. A. Saalfeld. Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm. *Cartography and Geographic Information Science*, 26(1):7–18, 1999.
29. Y. Shoham. Temporal Logics in AI: Semantical and Ontological Considerations. *Artificial Intelligence*, 33(1):89–104, 1987.
30. R.T. Snodgrass. *The TSQL2 Temporal Query Language*. Kluwer Academic Press, 1995.
31. J.G. Stell and M. Worboys. Stratified Map Spaces: A Formal Basis for Multi-Resolution Spatial Databases. In T.K. Poiker and N. Chrisman, editors, *Proceedings of 8<sup>th</sup> International Symposium on Spatial Data Handling (International Geographical Union)*, pages 180–189, 1998.
32. R. Weibel and G. Dutton. Generalizing Spatial Data and Dealing with Multiple Representations. In P.A. Longley, D.J. Maguire, M.F. Goodchild, and D.W. Rhind, editors, *Geographical Information Systems: Principles, techniques, management and applications*, chapter 10. John Wiley New York, 1999.