

Provided by the author(s) and University College Dublin Library in accordance with publisher policies. Please cite the published version when available.

Title	Adaptive management of multigranular spatio-temporal object attributes
Author(s)	Camossi, Elena; Bertino, Elisa; Guerrini, Giovanna; Bertolotto, Michela
Publication Date	2009
Publication information	Mamoulis, N. ... [et al.] (eds.). Advances in Spatial and Temporal Databases : 11th International Symposium, SSTD 2009 Aalborg, Denmark, July 8-10, 2009, Proceedings
Publisher	Springer-Verlag
Link to publisher's version	http://dx.doi.org/10.1007/978-3-642-02982-0_21
This item's record/more information	http://hdl.handle.net/10197/1340

Downloaded 2012-05-16T20:34:02Z

Some rights reserved. For more information, please see the item record link above.



Adaptive Management of Multigranular Spatio-Temporal Object Attributes ^{*}

E. Camossi¹

E. Bertino²

G. Guerrini³

M. Bertolotto¹

¹ 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

² 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

³ DISI - Università degli Studi di Genova, Via Dodecaneso 35, 16146 Genova, Italy. Phone: +39 010 353-6701. Fax: +39 010 353-6699. e-mail: guerrini@disi.unige.it

Abstract. In applications involving spatio-temporal modelling, granularities of data may have to adapt according to the evolving semantics and significance of data. In this paper we define *ST²_ODMGe*, a multi-granular spatio-temporal model supporting *evolutions*, which encompass the dynamic adaptation of attribute granularities, and the deletion of attribute values. Evolutions are specified as *Event - Condition - Action* rules and are executed at run-time. The event, the condition, and the action may refer to a period of time and a geographical area. The evolution may also be constrained by the attribute values. The ability of dynamically evolving the object attributes results in a more flexible management of multigranular spatio-temporal data but it requires revisiting the notion of object consistency with respect to class definitions and access to multigranular object values. Both issues are formally investigated in the paper.

1 Introduction

The ability of representing datasets with respect to both their spatial layout and their historical evolution is crucial when performing analysis and monitoring changes in the spatial configuration of geographical areas. Moreover, approaches able to present data at different granularities [3] represent an effective solution to facilitate information analysis [1].

The granularity according to which information is represented depends on the the data domain and semantics as well as on the application tasks to be performed on them. The selection of the appropriate granularity is based on modelling requirements and on a trade-off between application efficiency and

^{*} 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.

data accuracy. A greater detail (i.e., a finer granularity) reduces data indeterminacy and allows to obtain information as accurate as possible. Conversely, storing data at an unnecessary level of detail, causes waste of space and additional costs in aggregating detailed data to the required abstraction level. Thus, the choice of a less detailed representation (i.e., a coarser granularity) makes it possible to store the minimal amount of data, thus reducing storage costs, and could improve application efficiency. Therefore, the selection of attribute granularities is a crucial task, that in existing multigranular systems is done once for all, at schema definition time. To enhance data flexibility, the model at hand must support the ability of dynamically setting and changing the spatio-temporal granularity. A static definition of attribute granularities in the database schema, as supported by current multigranular models, may not be adequate for many important spatio-temporal applications. For instance, in a spatio-temporal database for environmental monitoring, the collection of meteorological parameters such as 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. Moreover, 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).

In our effort to address these issues we have defined ST^2 -ODMGe (*Spatio-(Bi)Temporal ODMG supporting Evolutions*), a spatio-temporal data model that enables the *evolution* of attributes values, that is, the modification of the granularities used in attribute definitions, and the deletion of attribute values at run-time. ST^2 -ODMGe evolutions reflect modifications about data significance that arise for several reasons, including: 1) periodic phenomena (e.g., rain and snowfall usually increase during predetermined seasons); 2) modification to the value of an attribute, or its occurrence (e.g., in monitoring systems); 3) the execution of an operation (e.g., in diagnostic systems); 4) data aging (e.g., older data may be aggregated and then maintained at coarser granularities); 5) privacy restrictions (e.g., individual information on user locations, which are collected in traffic analysis, must be aggregated to coarser granularities in order to be made publicly available). Hence, evolutions enhance the flexibility in the management of multigranular spatio-temporal data. They allow one to dynamically adapt the granularities to dynamic events and situations resulting from spatio-temporal attribute value updates and operation executions.

The types of evolutions supported by ST^2 -ODMGe include: granularity evolution, granularity acquisition, and value deletion. *Granularity evolution* aggregates existing detailed data at a coarser granularity (e.g., older data that may be stored for future reference), or refines information at a finer granularity (e.g., in data analysis)¹. By contrast, *granularity acquisition* changes at run-time the granularity used when inserting new values in the database, whenever the domain conditions change (e.g., sales recording during Christmas). Finally, *value*

¹ The latter operation increases indeterminacy on converted data, as discussed in the paper and in further detail in [8].

deletion removes attribute values that are no longer useful at a given granularity (e.g., detailed data already aggregated at coarser granularities) from the database.

The ST^2 _ODMGe model design extends our previous models ST _ODMG [7], a multigranular spatio-temporal object model that does not support evolutions, and T _ODMGe [6], a multigranular temporal model supporting dynamic objects. It expands on their data definition languages, type systems, and multigranular conversions to support the evolution of spatio-temporal values and the bitemporal domain. Granularity evolutions and value deletions, originally defined for historical data only in T _ODMGe [6], are herein extended to the spatio-temporal domain: the spatio-temporal type system defined in ST _ODMG [7], further extended to include bitemporal support, has been embedded in the new model ST^2 _ODMGe. A fundamental difference with our previous evolution model [6], where granularity evolutions only allow one to summarize older data at coarser granularities, is that in ST^2 _ODMGe they may be specified also to refine data at finer granularities. Other important differences with T _ODMGe [6] are: 1) evolutions may be specified and executed at run-time, based on the execution model of active databases, instead of being defined statically in the database schema; 2) evolutions of an attribute value may be triggered according to database conditions involving also other attributes, as well as relying on the execution of methods, thus making our evolution approach highly flexible; 3) the introduction of granularity acquisition removes one of the major limiting assumptions of T _ODMGe, where the granularity used for acquiring new data is immutable. The main novelty with respect to ST _ODMG, besides bitemporal support, are the evolution facilities. ST^2 _ODMGe enhances the expressive power of ST _ODMG providing a flexible and comprehensive support for run-time modifications of attribute granularities.

Evolutions introduce additional issues that are addressed in the formal design of the model. As a result of the execution of granularity evolutions and acquisitions, the run-time type of an object attribute is a Cartesian product of multigranular types at different granularities. The semantic consistency of the evolved attributes values must therefore be guaranteed and the strategies to access attribute values must be redefined to take advantage of these composed types. Moreover, even if a value is deleted, the access to attribute values may be preserved by considering evolved values at different granularities. In particular, in the paper we formally revise the notion of object consistency, and redefine the strategies to access evolved multigranular attribute values.

The rest of the paper is organized as follows. We first discuss related work (Section 2). In Section 3 we introduce the ST^2 _ODMGe type system, objects and classes. Then, in Section 4 we address the definition of evolutions for spatio-temporal data, by means of illustrative examples. In Section 5 we investigate how object consistency is affected by evolutions. We define in Section 6 the access strategies to take advantage of attribute run-time values at multiple granularities, and we show that, under certain assumptions, object access is invariant

with respect to the execution of evolutions. Finally, in Section 7 we conclude the paper by outlining future research directions.

2 Related Work

ST^2 _ODMGe assumes and extends previous work on efficiently computing historical aggregates for on-line analytical processing (OLAP) of spatio-temporal data streams [16,13,11]. Zhang et al. [16], defined a spatio-temporal extension of the *SB-Tree* [15] structure, that, like our previous work [6], proposes an aggregated indexing approach whereby older data are stored using coarse granularities. Tao and Papadias [13] proposed over the years several indexing structures for the efficient historical aggregation of spatio-temporal data. Recent work focuses on aggregates for trajectories of moving objects [11]. Unlike those approaches, ST^2 _ODMGe supports different time granularities and multiple levels of aggregation and refinement, that is, different indexing forms. Moreover the appropriate granularity level can be selected on a per-attribute basis, thus supporting different semantics (i.e., different queries). Furthermore, our notion of evolution refers to the bounds of granules at a given granularity, instead of referring to a given amount of time. Finally, ST^2 _ODMGe relies on an widely adopted notion of temporal granularity [5] that considers granularities as data integrity constraints and formalises how different granularities are related to each other.

The approach to deletion we adopt has been inherited from research in the area of temporal databases. In this area data deletion is a crucial issue because answers against historical queries must be preserved [9,14,12]. Garcia-Molina et al. [9] have addressed data deletion in historical databases by proposing an approach whereby data may be removed (i.e., data *expire*) without affecting related views. A similar approach has been proposed by Toman [14] for historical data warehouses, whereby automatic data deletion is supported by preserving answers to a known and fixed set of first-order queries. This approach assumes that conditions for data evolution are inferred from a given set of queries. This approach may complement our work, since conditions for data evolution may be inferred for those attributes for which they are not known at schema definition time. A different approach is proposed by Skyt et al. [12], who address the faithful encoding of data history in temporal databases in the presence of vacuumed data. Such an approach relies on query modification, that is, on accompanying query results with additional information about how the required data may be affected by vacuuming. Unlike that approach, we allow queries be performed on vacuumed data whenever a result at different granularities exists.

3 Preliminaries

In this section we illustrate the main characteristics of ST^2 _ODMGe, namely, the spatio-temporal dimensions, the granularities formalization, the multigranular type system, and describe granularity conversions and their properties. We then present ST^2 _ODMGe classes and objects.

3.1 Time, Space, and Granularities

The ST^2 _ODMGe model is a 4-dimensional multigranular spatio-temporal model that supports two-dimensional space and two temporal dimensions: *valid time* and *transaction time*. In the following, valid time dimension in ST^2 _ODMGe (denoted by \mathcal{VT}) refers to the time a fact is true in the reality [10]. Transaction time dimension (denoted by \mathcal{TT}) represents the time at which database transactions are executed [10]. Moreover, ST^2 _ODMGe supports *two-dimensional space*, denoted by \mathcal{S} , that refers to the space in which the modelled objects are actually located. Spatio-temporal attributes values refer to valid time and to the space dimensions. By contrast, modification actions, including those triggering the evolutions of attributes, refer to transaction time. Unlike the valid time dimension, transaction time includes references to the current time denoted by variable *NOW*.

In each ST^2 _ODMGe database a set of temporal granularities [5] \mathcal{G}_T and a set of spatial granularities \mathcal{G}_S are defined. We further distinguish between valid time granularities $\mathcal{G}_{\mathcal{VT}}$ and transaction time granularities $\mathcal{G}_{\mathcal{TT}}$. Temporal and spatial granularities are mappings from an index set \mathcal{IS} to the power sets of the temporal [5] and the spatial domains, respectively. For instance, *days* and *weeks* are temporal granularities; *meters*, *yards* and *provinces* are spatial granularities. The temporal dimensions are totally ordered. Temporal and spatial granularities are used to represent ST^2 _ODMGe objects attributes and modification actions at different levels of detail.

A *granule* is a subset of a domain corresponding to a single granularity mapping, 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 of the same granularity have disjoint interiors. Moreover, non-empty temporal granules preserve the order of the temporal domains.

Valid time granules bound attribute values, while transaction time granules refers to modification actions. Similarly, spatial granules specify the geographical areas where spatio-temporal attribute values are defined. For instance, consider the value of the daily temperature in Rome the first and the second day of January. To model this value, one can use the labels “01/01”, “02/01”, and “Rome” to denote two temporal granules at granularity *days* and one spatial granule at granularity *municipalities*, respectively.

Granularities differ according to how their granules partition a domain. In this respect, granularities are related by the *finer-than* transitive relationship and its inverse *coarser-than* [5]. For example, granularity *days* is finer-than *months*, which in turn is finer-than *years* (symmetrically, *years* is coarser-than *months*, which is coarser-than *days*). Likewise, *municipalities* is finer-than *countries*. The finer-than relationship is denoted by \preceq , while \prec denotes the anti reflexive finer-than.

Given two multigranular values, one at granularity G and one at granularity H such that G and H are not directly related by finer-than, these values may be compared if they are converted, i.e., represented, at the same granularity K , that is finer-than G and H . K is chosen as the granularity that minimizes the number

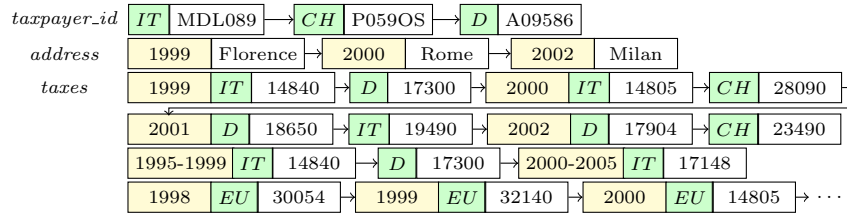


Fig. 1. Example of object state

of conversion steps. 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 (denoted as $GLB(G,H)$) [5].

3.2 Multigranular Types and Conversions

Besides conventional database values, a multigranular spatio-temporal database schema includes multigranular spatial, temporal, and spatio-temporal values. Multigranular values are defined as partial functions from the set of granules of the corresponding granularity(ies) to the set of values of a given inner type. Fig. 1 illustrates examples of ST^2 -ODMGe multigranular attribute values: *taxpayer_id* is an alphanumeric spatial attribute with type $Spatial_{countries}(\mathbf{string})$; *address* is an alphanumeric temporal attribute with type $Temporal_{years}(\mathbf{string})$; finally, *taxes* is a numeric spatio-temporal attribute. The ST^2 -ODMGe type of the first value *taxes* in Fig. 1 is $Temporal_{years}(Spatial_{countries}(\mathbf{int}))$.

In a multigranular database data may be converted at different granularities to increase or reduce their level of detail. In ST^2 -ODMGe, the conversion of multigranular geometrical features is obtained through the composition of model-oriented and cartographic map generalisation operators that guarantee topological consistency (e.g., *merge*, *abstraction*), and refinement operators that perform the inverse functions (e.g., *split*, *add feature*) [7]. On the other hand, to retrieve, for instance, the annual trend of a phenomenon having a daily representation (e.g., the values of sales in shops located in several countries), also conversion for non geometric attribute values are provided. For example, *average* and *sum* aggregate numeric values; *selection* and its specializations (e.g., *first*, *main*) coerce alpha-numeric values; *restriction* and *split* refine values.

An interesting property that will be used in the paper to evaluate the correctness of attribute access refers to the *invertibility* of conversions [4]. Indeed, even if when converting a temporal value to a different granularity, and then performing the inverse conversion, we would expect the original value to be returned. However, unfortunately, when converting from a finer to a coarser granularity, we lose some details that we cannot usually recover by applying the inverse conversion to the finer granularity. By contrast, when converting from a coarser to a finer granularity, we introduce some details that we should be able to forget; thus we can recover the original value. Given a pair of conversion functions, we

denote them as *quasi-inverse* or *inverse* [4], according to whether the conversions refer to the first or the second situation, respectively. In the first case, a measurable indeterminacy is introduced. For example, the pair (**average,split**) is quasi-inverse, while the pair (**split,sum**) is inverse.

Granularity conversions may be enriched with user defined functions to suite specific domain requirements (e.g., the RGB spectrum, user credentials). In this case, the user must provide both the conversions to finer and coarser granularities, and he/she has to take care of the invertibility aspect, according to the domain semantics.

3.3 ST^2 _ODMGe Classes and Objects

In the following example we illustrate an ST^2 _ODMGe class specification.

Example 1. Given an object type for describing *taxpayers*, reporting the value of taxes paid by a person over time in different countries, its definition includes: a spatial attribute **taxpayer_id** at granularity *countries* to store the fiscal identifiers that a taxpayer holds in different countries; a temporal attribute **address** at granularity *years* to store the history of his/her fiscal domiciles; and a spatio-temporal attribute **taxes**, defined at temporal granularity *years* and at spatial granularity *countries*, which stores the amount of taxes the taxpayer pays every year in each country where he/she works (for simplicity we suppose the values are stored according to the same currency in €). \square

Given an ST^2 _ODMGe class, an ST^2 _ODMGe object is defined as follows.

Definition 1. (*ST^2 _ODMGe Object*). Given a class c , an ST^2 _ODMGe object o of c is defined as a 6-tuple $(id, N, v, c, \Upsilon_{VT}^{GIT} \times \Upsilon_S^{GIS}, \Upsilon_{TT}^{GIT})$ where: id is the object identifier, unique in the database; N is the set of object names; v is the object state, given as a tuple of attribute values: $(a_1:v_1, \dots, a_n:v_n)$, where a_i is an attribute name, and v_i is an attribute value, with $1 \leq i \leq n$; c is the class to which the object belongs; $\Upsilon_{VT}^{GIT} \times \Upsilon_S^{GIS}$ is the spatio-temporal object lifespan, represented as set of granules at the temporal chronon and the spatial quantum granularities², with respect to valid dimensions; Υ_{TT}^{GIT} is the transactional temporal lifespan of o at the temporal chronon granularity. \diamond

Example 2. Let o be an object of class **taxpayer** as described in Example 1. According to Definition 1, the values of attributes **taxpayer_id**, **address** and the first value of **taxes** at granularities *years* and *countries* in Fig. 1 define a legal object state v for o . For instance, according to Fig. 1, the contributor in 1999 paid € 14,840 in Italy and € 17,300 in Germany. An example of spatio-temporal lifespan for o is $\{1 \text{ Jan } 1999 \text{ 00:00:1}, \dots, 31 \text{ Dec } 2030 \text{ 23:59:59}\}_{VT}^{seconds} \times \{IT, D, CH\}_S^{countries}$, $\{1 \text{ Jan } 1999 \text{ 00:00:1}, \dots, NOW\}_{TT}^{seconds}$, where *IT* is for Italy, *D* is for Germany, and *CH* is for Switzerland, assuming *seconds* and *countries* as chronon and spatial quantum granularities. \square

² These are the finest granularities on the spatio-temporal domain.

4 Evolutions

Evolutions are defined and executed at run time on ST^2 _ODMGe objects. Our model supports three different types of evolutions: *granularity evolution*, *granularity acquisition*, and *value deletion*.

Granularity evolutions and acquisitions modify the granularity of an attribute. The granularity evolution operation, previously introduced by us [6] in a more limited form and restricted to the valid temporal domain, allows one to define a new portion of an attribute value, specified at a different granularity. The new value, referred to as *target*, is obtained by converting values, referred to as *source*, already stored in the database at a different granularity through the application of granularity conversions. By contrast, granularity acquisitions do not change the database state, but re-define the granularity(ies) that can then be used to insert new attribute values. They have the same effect of a modification of the database schema, as if an SQL ALTER statement were executed. Finally, a value deletion eliminates portions of an attribute value at a given granularity.

Evolutions are performed according to the general execution model of active databases, and have the form: `ON Event [IF Condition] DO Action`. 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. Then, the corresponding conditions, if present, are evaluated. For the triggered evolutions whose conditions evaluate to TRUE, the corresponding actions are executed. An evolution action is a sequence of operations that may modify attribute granularities and delete attribute values. As a consequence, the database state (or schema, in case of granularity acquisition) may be modified.

The temporal behaviour of ST^2 _ODMGe evolutions differs according to their occurrence: ST^2 _ODMGe supports *periodic* and *non-periodic* evolutions. These different behaviours may be further characterized with the support of *spatio-temporal bounds*, that may apply to each of the elements of an evolution, thus restricting the occurrence of the evolution event, the evaluation of the condition, and the effects of the action to given temporal periods and geographical areas.

As a consequence of the execution of evolutions, the type of an object state in the ST^2 _ODMGe model, that is, of the values of their attributes, changes dynamically. Let a be a multigranular attribute defined in class c . In the general case, the run-time type of a is a Cartesian product of multigranular types, as illustrated by the following example.

Example 3. Let o be the identifier of an object of class `taxpayer` we described in Example 1. A legal state for o is shown in Fig. 1. The value of attribute `taxes` in Fig. 1 is a set of spatio-temporal values at different granularities. The first value is the value corresponding to the attribute definition, and is given at temporal granularity *years* and at spatial granularity *countries*. The other two values are obtained from this value through granularity evolutions. They are specified at granularities *5years* and *countries*, and *years* and *ecAlliances* (i.e., economic alliances), respectively. According to the different granularities,

they temporally and spatially aggregate the first value (at granularities *years* and *countries*). The domain of attribute **taxes** is: $Temporal_{years}(Spatial_{countries}(\mathbf{int})) \times Temporal_{5years}(Spatial_{countries}(\mathbf{int})) \times Temporal_{years}(Spatial_{ecAlliances}(\mathbf{int}))$. \square

An evolution defined on an attribute *a* is specified on one of the granular values that compose the value of *a*. Each value is referred to as a *granularity level*, and is identified by its granularity. More precisely, given an object *o* of class *c*, the value of attribute *a* at a given (either temporal or spatial) granularity *G* (at temporal granularity G_t and at spatial granularity G_s , respectively) is referred to as the *granularity level* $\langle G \rangle$ of *a* ($\langle G_t, G_s \rangle$ if the attribute is spatio-temporal). Given for instance attribute **taxes** of Example 3, with the object state depicted in Fig. 1, we have three different granularity levels: $\langle years, countries \rangle$, $\langle 5years, countries \rangle$, $\langle years, ecAlliances \rangle$. In the following example we illustrate the syntax to define evolutions.

Example 4. Given class **taxpayer** of Example 1, the following evolution summarises the older annual record of taxes at a coarser temporal granularity *5years*.

```
ON update taxpayer.taxes< years, countries > during {1999, ..., 2014} $_{\sqrt{T}}^{years}$ 
IF every  $5_{\sqrt{T}}^{years}$ 
DO evolve < years, countries > to < 5years, countries > in {IT} $^{countries}$ 
    using average $_{years \rightarrow 5years}$ , restriction $_{5years \rightarrow years}$  .
```

The evolution is defined for attribute **taxes**, which evolves from granularity level $\langle years, countries \rangle$ to granularity level $\langle 5years, countries \rangle$. It is triggered by the updates of the evolution source granularity level, and involves at each execution 5 years of data, as specified by the periodic condition **every** $5_{\sqrt{T}}^{years}$. The evolution involves only the taxes paid between year 1999 and 2014 in Italy, according to the event temporal bound $\{1999, \dots, 2014\}_{\sqrt{T}}^{years}$, and by the action spatial bound **in** $\{IT\}^{countries}$. Granularity conversion **average** $_{years \rightarrow 5years}$ is applied for creating the target level, while conversion **restriction** $_{5years \rightarrow years}$ may be used to recover the original values whenever these are deleted from the database.

Now suppose that the following evolution is specified from granularity level $\langle years, countries \rangle$ to $\langle years, ecAlliance \rangle$, where *ecAlliance* represents (non-overlapping) economical alliances among different countries:

```
ON update taxpayer.taxes< years, countries >
IF after  $1_{\sqrt{T}}^{years}$ 
DO evolve < years, countries > to < years, ecAlliance > in countries({EC} $^{ecAlliance}$ )
    using sum $_{countries \rightarrow ecAlliance}$ , split $_{ecAlliance \rightarrow countries}$  .
```

The notation $G(\mathcal{Y}^{G'})$ denotes the conversion of the set of G' -granules $\mathcal{Y}^{G'}$ to granularity *G*. Note that the spatial bound **in** $countries(\{EC\}^{ecAlliance})$ constrains the action execution, and accordingly the evolution aggregates only the tax logs that refer to European Countries. The evolution is executed periodically, according to the condition **after** $1_{\sqrt{T}}^{years}$. Fig. 1 is an example of object state, after the execution of the evolutions. \square

5 Object Consistency

ST^2 _ODMGe evolutions affect the conventional notion of object consistency, because at run-time the object state may no longer match their class definitions, as illustrated by Example 4. However, evolved multigranular attribute values are created starting from source granularity levels, which rely in turn on the original granularity level defined for the attribute. The evolved values are semantically consistent with the original ones, to which they are related by a chain of (quasi) inverse granularity conversions. Therefore, the original value may be recomputed from an evolved one when needed (for example when the original data are deleted from the database), with a bounded imprecision. For the same reason, the introduction of granularity acquisition does not pose problems, i.e., the original and the refined granularity levels are related by a pair of (quasi) inverse granularity conversions.

In the following, ST^2 _ODMGe object consistency is formalized, taking into account how evolutions modify the object state and relying on the relationships among the granularity levels of object attributes. Such relationships are formalized by the notion of *Granularity Levels Graph* (GLG) of an attribute, which is preliminary to the consistency formalization.

5.1 Attribute Granularity Level Graph

Let a be a multigranular attribute defined in class c . The granularity levels that compose the value of a are pairwise linked by pairs of quasi-inverse granularity conversions, to form a graph that we refer to as the *Granularity Levels Graph* (GLG) of the attribute, which is formalised by the following definition. This structure, that must not be confused with the database granularity lattices we may define for G_{VT} , G_{TT} and G_S relying on finer-than [5], is specific for each multigranular attribute, and relates its granularity levels enabling to navigate among them when accessing the attribute value. In this definition, and in the rest of the paper, for simplicity we consider a multigranular attribute that refers to either the spatial or the temporal domain, whenever the case of spatio-temporal values may be inferred straightforwardly. Whenever needed, we point out the differences of the spatio-temporal case.

Definition 2. (Granularity level graph - GLG) *Given a set of temporal or spatial granularity levels $\langle G_i \rangle$ defined for attribute a , where $\forall i = 1 \dots n$, $G_i \in \mathcal{G}$ is either a temporal or a spatial granularity, the granularity level graph of a , denoted by a_{GLG} , is a graph (V, E) such that $V = \{\langle G_1 \rangle, \dots, \langle G_n \rangle\}$, and $E = \{\langle G_j \rangle \rightleftharpoons \langle G_k \rangle\}$, if G_j and G_k are related by the finer-than relationship, and two (quasi)inverse granularity conversions $f_{G_j \rightarrow G_k}$ and $g_{G_k \rightarrow G_j}$ have been defined; $1 \leq j \leq n$, $1 \leq k \leq n$. \diamond*

Similarly, given a spatio-temporal attribute a and the set of its spatio-temporal granularity levels $\langle G_{t_i}, G_{s_i} \rangle$, and given the granularity conversions defined among these granularity levels through evolution specifications, a GLG is defined for a . Given an attribute a , let a_{GLG} denote its GLG. Moreover, given a

GLG a_{GLG} the set of its nodes and edges are denoted by $a_{GLG}.V$ and $a_{GLG}.E$, respectively.

Example 5. Given attribute **taxes** of class **taxpayer** of Example 1, suppose the evolutions of Example 4 have been specified. Hence, $taxes_{GLG} = (V, E)$ is specified as follows:

$$\begin{aligned} taxes_{GLG}.V &= \{ \langle years, countries \rangle, \langle 5years, countries \rangle, \langle years, ecAlliance \rangle \}; \\ taxes_{GLG}.E &= \{ \langle years, countries \rangle \rightleftharpoons \langle 5years, countries \rangle, \\ &\quad \langle years, countries \rangle \rightleftharpoons \langle years, ecAlliance \rangle \}. \end{aligned} \quad \square$$

The following property ensures that, given two granularity levels in an attribute GLG, it is always possible to compare them even if the granularity levels are not directly linked through granularity conversions in the GLG. Therefore, to solve an attribute access we may navigate among the values defined in the attribute GLG by using the defined granularity conversions, as we will see in the following section.

Property 1. Let $\langle G_i \rangle$ and $\langle G_j \rangle$ be two granularity levels in a_{GLG} . Then, one of the following conditions holds:

- $G_i \prec G_j$;
- $G_j \prec G_i$;
- $GLB(G_i, G_j) \in a_{GLG}.V$. ▽

With the following definition we introduce also the concepts of *bottom* and *top* granularities for a multigranular attribute.

Definition 3. (Bottom and top granularities in an attribute GLG). *Given an object o and a multigranular attribute a defined for o , G^\perp is the set of the (temporal or spatial) bottom granularities of a , that is the finest granularities in a_{GLG} , i.e., no granularity G , with $\langle G \rangle \in a_{GLG}.V$ exists such that, $\forall G' \in G^\perp, G \prec G'$. Symmetrically, G^\top is the set of the (temporal or spatial) top granularities of a , that is, the coarsest granularities in a_{GLG} for which no granularity H , with $\langle H \rangle \in a_{GLG}.V$ exists such that, $\forall H' \in G^\top, H' \prec H$. ◇*

Example 6. Given attribute **taxes** of Example 1 with the GLG of Example 5, $G_{VT}^\perp = \{years\}$ and $G_{VT}^\top = \{5years\}$; $G_S^\perp = \{countries\}$, while $G_S^\top = \{ecAlliance\}$. □

5.2 Consistency Conditions for ST^2 _ODMGe Objects

Relying on attribute GLGs we now revisit the consistency of ST^2 _ODMGe objects. We define the constraints that are useful to define the access strategies and must therefore be preserved when manipulating object states. Such constraints are expressed with respect to all the dimensions supported by the model.

To guarantee object consistency, every ST^2 _ODMGe attribute value must satisfy the following conditions: 1) each attribute value belongs to the set of legal values of the corresponding type; 2) whenever the attribute value is an object

identifier, the referred object exists in the database sometimes during the temporal transactional lifespan of the object; 3) the value of a multigranular attribute is a tuple of multigranular values, whose spatial and the temporal domains do not exceed the spatial and temporal lifespan of the object; thus for each defined value, the corresponding granule intersects the object lifespan³; 4) for a multigranular attribute a GLG is defined according to Definition 2, and its edges preserve the relationships holding among granularities of the granularity levels. The previous constraints are formalised by Definition 4, which expresses the notion of run-time consistency for objects in an ST^2 -ODMGe database.

Definition 4. (ST^2 -ODMGe Consistent Instance). *Let c be a class and $attr$ its attribute specification $\{(b_1, \tau_1), \dots, (b_m, \tau_m)\}$, where $\forall j, 1 \leq j \leq m, b_j$ an attribute name and τ_j an attribute type. Let \mathcal{LT} and \mathcal{OT} be the sets of literal and object types, respectively, and let \mathcal{T}_{geom} be the set of geometric vector types (e.g., point, line, polygon). Let $\llbracket \tau \rrbracket$ be the set of legal values for type τ , and $\llbracket \tau' \rrbracket_i^{G_{IT}}$ be the set of legal values defined for τ' in granule $G_{IT}(i)$. Let o be a ST^2 -ODMGe object defined as $(id, N, (a_1 : v_1, \dots, a_p : v_p), c, \Upsilon_{VT}^{G_{IT}} \times \Upsilon_S^{G_{IS}}, \Upsilon_{TT}^{G_{IT}})$. Then, object o is a consistent instance of c if all the following conditions hold:*

1. $\forall i, 1 \leq i \leq p, \exists (b, \tau) \in attr$ such that $b = a_i$;
2. $\forall (b, \tau) \in attr, \exists k, 1 \leq k \leq p$, such that $b = a_k$ and the following conditions hold:
 - (a) if $\tau \in \mathcal{LT}$, $v_k \in \llbracket \tau \rrbracket$ (Cf. condition 1);
 - (b) if $\tau \in \mathcal{OT} \cup \mathcal{T}_{geom}$, $v_k \in \bigcup_{\Upsilon_{TT}^{G_{IT}}} \{\llbracket \tau \rrbracket_h^{G_{IT}} \mid h \in \mathcal{IS}\}$ (Cf. condition 2);
 - (c) if τ is a multigranular type at granularity G , all the following conditions hold (Cf. conditions 3 and 4):
 - i. $v_k = (v_{k_1}, v_{k_2}, \dots, v_{k_n})$, with $n \geq 1$;
 - ii. $\forall j, 1 \leq j \leq n$, such that v_{k_j} is defined,
 - A. $\exists \tau_j$, where τ_j is a multigranular type at granularity G_j , such that $v_{k_j} \in \llbracket \tau_j \rrbracket$;
 - B. $\forall i \in \mathcal{IS}$ such that $v_{k_j}(i)$ is defined, $G_j(i) \cap (\bigcup_{\Upsilon_{VT}^{G_{IT}} \times \Upsilon_S^{G_{IS}}} \{G_{IS}(h) \mid h \in \mathcal{IS}\}) \neq \emptyset$;
 - iii. a granularity (level) graph (V_k, E_k) is defined, such that:
 - A. $V_k = \{ \langle G_1 \rangle, \dots, \langle G_n \rangle \mid v_{k_j} \in \llbracket \tau_j \rrbracket \text{ is defined, with } 1 \leq j \leq n \}$;
 - B. $E_k = \{ \langle G_q \rangle \rightrightarrows \langle G_r \rangle, \text{ with } G_q \prec G_r \text{ or } G_r \prec G_q, 1 \leq q \leq n, 1 \leq r \leq n \}$ and two (quasi)inverse granularity conversions $f_{G_q \rightarrow G_r}$ and $g_{G_r \rightarrow G_q}$ have been defined. \diamond

Example 7. Given object o of Example 3, we assume the evolutions of Example 4 have been executed on $o.taxes$, with $taxes_{GLG}$ as defined in Example 5. Given $\{1 \text{ January } 1999 \text{ } 00:00:1, \dots, 31 \text{ December } 2030 \text{ } 23:59:59\}_{VT}^{seconds} \times \{IT, D, CH\}_S^{countries}$, $\{1 \text{ January } 1999 \text{ } 00:00:1, \dots, NOW\}_{TT}^{seconds}$, the lifespan of o , assuming *seconds* and *countries* as chronon and spatial quantum granularities,

³ Border granules may not be completely included in the object lifespan, but their intersection with it must be non-empty.

and assuming updates on o have been executed after 1998, then object o , with the object state of Fig. 1, is a consistent instance of class `taxpayer` according to Definition 4. By contrast, it would be inconsistent if its lifespan were $\{1 \text{ January } 2000 \text{ 00:00:1}, \dots, 31 \text{ December } 2002 \text{ 23:59:59}\}_{\mathcal{VT}}^{\text{seconds}} \times \{IT\}_{\mathcal{S}}^{\text{countries}}$, $\{1 \text{ January } 1999 \text{ 00:00:1}, \dots, NOW\}_{\mathcal{TT}}^{\text{seconds}}$, because it would intersect neither the values defined before year 2000, nor the countries different from Italy. \square

6 Object Access

In this section we discuss the access to ST^2 -ODMGe multigranular attribute values. To simplify the presentation, we introduce a basic access that requires the attribute value defined in a single granule. Access to multiple granular values follows straightforwardly. We further distinguish between two forms of access, *qualified* and *unqualified*, depending on a granularity conversion being specified or not, respectively. The strategies to solve them are discussed separately. Therefore, we discuss the invariance of object accesses with respect to evolutions, and characterize unsolvable object accesses.

6.1 Qualified and Unqualified Access

The concept of object access we consider is formalised by the following definition.

Definition 5. (ST^2 -ODMGe object access). *Let o be an object identifier, and let a be the name of an attribute defined for o . If a is a multigranular temporal attribute, let G be a temporal granularity. If a is a multigranular spatial attribute, let G be a spatial granularity. Given a granule label l^G , an object access is an expression of the form $o.a \downarrow^{[f]} l^G$, requiring the value of attribute a of object o in granule l^G . If a granularity conversion f is specified, f is applied to compute the access result. In the latter case, the access is referred to as *qualified*. Otherwise it is *unqualified*. \diamond*

The access to a multigranular spatio-temporal attribute a is expressed as $o.a \downarrow^{[f]} l^{G_t} \downarrow^{[f']} l^{G_s}$, where G_t and G_s are a temporal and a spatial granularity, respectively; l^{G_t}, l^{G_s} are two granule labels for G_t and G_s ; f and f' are granularity conversions.

Example 8. Given class `taxpayer` of Example 1 and object o with the state in Fig. 1, $o.\text{taxes} \downarrow \{1998\}_{\mathcal{VT}}^{\text{years}} \downarrow \{IT\}_{\mathcal{S}}^{\text{countries}}$ is the unqualified access to the payments made by the contributor during 1998 to the Italian Revenue service. By contrast, object access $o.\text{taxes} \downarrow \{1999\}_{\mathcal{VT}}^{\text{years}} \downarrow^{\text{split}[p(x)]} \{IT\}_{\mathcal{S}}^{\text{countries}}$, where $p(x)$ is the probability distribution: $p(x) = \{(IT, 0.5), (D, 0.5)\}$, is the qualified access to the same payments, requiring the application of the refinement function $\text{split}[p(x)]$. \square

6.2 Solving Unqualified Object Access $o.a \downarrow l^G$

To solve the unqualified object access $o.a \downarrow l^G$ we check whether the requested value is available, i.e., if $\langle G \rangle$ is a granularity level defined for a and if the value of $o.a$ for granule l^G is defined. If so, such value, that we denote as $o.a_G(l^G)$, where $o.a_G$ is the granularity level $\langle G \rangle$ defined for a , is returned.

Otherwise, the requested value must be computed starting from the values, stored in other granularity levels, that intersect l^G . In this case, two different strategies may be applied for solving the access, depending on whether the user wants to maximize the accuracy of the result or the access efficiency. The *efficiency* maximization strategy minimizes the number of intermediate accesses needed to solve $o.a \downarrow l^G$. According to this strategy, the application of conversion functions from coarser to finer granularities is preferred, because just one value is accessed for each of the granularity levels involved. By contrast, when maximizing *accuracy*, the highest precision is required in computing the result. Therefore the application of granularity conversions from finer to coarser granularities takes precedence, because they minimize the indeterminacy in the returned values (cf. Section. 3).

Fig. 2 reports the algorithm to solve $o.a \downarrow l^G$. The spatio-temporal access $o.a \downarrow l^{G_t} \downarrow l^{G_s}$ follows straightforwardly. We assume that the granularity levels in a_{GLG} are ordered according to the finer-than relationship. Spatio-temporal granularity levels are ordered first according to temporal granularities, and then with respect to spatial granularities. **ACCURATE** denotes that an accurate answer is preferred, whilst efficiency is the default.

```

if  $\exists o.a_G(l^G) \neq \perp$  then return  $o.a_G(l^G)$ 
else if ACCURATE then
  while  $\exists o.a_K$  s.t.  $K \preceq G$  and  $\forall l_k^K \in K(l^G)$ 
    s.t.  $o.a_K(l_k^K) \neq \perp$ 
    return  $f_{K \rightarrow G}(o.a_K)(l^G)$ 
  return null
else while  $\exists o.a_H$  s.t.  $G \preceq H$ 
  return  $g_{H \rightarrow G}(o.a_H)(l^G)$ 
while  $\exists o.a_K$  s.t.  $K \preceq G$ 
  return  $f_{K \rightarrow G}(o.a_K)(l^G)$ 
return null

```

Fig. 2. Algorithm for object access $o.a \downarrow l_G^i$

The computational complexity of the algorithm in Fig. 2 is $O(n)$. Indeed, assuming that the set of granularity levels defined for each attribute value is finite, and the time required for the application of granularity conversions is linear, the complexity of the algorithm is mainly given by the sequential access to a given value in a granularity level. If we assume that indexing is applied on granularity levels (e.g., BTree⁺ for temporal values and R-Tree for spatial values), the complexity may decrease to $O(\log(n))$ if the internal nodes of the

R-Tree do not overlap. An optimal worst-case complexity is guaranteed also if the indices for spatial data are, for example, PR-Trees [2].

An important result of our work is thus that the introduction of evolutions does not increase the complexity of the access with respect to the conventional multigranular case. Furthermore, complexity may improve whenever the access involves values at granularities among those defined for the attribute, because the access result may be already pre-computed in the database. Indeed, in both execution strategies, the access follows an iterative approach, and to solve it we may need to move across several granularity levels. Once a value is found (or a set of values, in the accuracy maximization strategy) that satisfies the access, a sequence of conversions must be performed. If some precomputed value is already available at an intermediate granularity, these values need not be recomputed, thus improving performance.

Example 9. Given access $o.\text{taxes} \downarrow \{1998\}_{\mathcal{Y}\mathcal{T}}^{\text{years}} \downarrow \{IT\}_{\mathcal{S}}^{\text{countries}}$ introduced in Example 8, and object o of class `taxpayer` whose state is shown in Fig. 1, the access results in € 14,840. \square

6.3 Solving Qualified Object Access $o.a \downarrow^f l^G$

If the access is qualified by a granularity conversion f , this function will be used to compute the access result, taking precedence over the functions already specified in granularity evolutions and acquisitions. Differently from unqualified access, if the accuracy maximization strategy is adopted, an existing value for the specified granule is discarded, if it was constructed with a different function. The value would be used instead by the efficiency maximization strategy. If this value is not defined, we distinguish whether f is a conversion to a coarser granularity (CF), or to a finer granularity.

Fig. 3 reports the algorithm for solving a qualified access $o.a \downarrow^f l^G$. The spatio-temporal object access $o.a \downarrow^f l_t^G \downarrow^{f'} l_s^G$ follows straightforwardly. As above, **ACCURATE** denotes that an accurate answer is preferred. As in the case of unqualified access, the algorithm for qualified access shown in Fig. 3 has computational complexity $O(n)$, which may reach the optimum if indexing is used on the granularity level values as in the previous case.

Example 10. Given the access $o.\text{taxes} \downarrow \{1998\}_{\mathcal{Y}\mathcal{T}}^{\text{years}} \downarrow^{\text{split}[p(x)]} \{IT\}_{\mathcal{S}}^{\text{countries}}$, with $p(x) = \{(IT, 0.5), (D, 0.5)\}$, and object o of class `taxpayer` with the object state depicted in Figure 1. When accuracy is required, the access results in € 15,027. This value is computed starting from the aggregate value at granularities $< \text{years}, ecAlliances >$. \square

6.4 Evolution Invariant Object Access

In order to preserve the consistency of query answers, evolution execution must not affect access results. In what follows, after a preliminary definition introducing the notion of evolution invariant access, we show that unqualified object

```

if  $\exists o.a_G(l^G) \neq \perp$  then
  if ACCURATE then
    if  $\exists o.a_K$  s.t.  $K \preceq G$ 
      and  $f_{K \rightarrow G}$  is defined between  $o.a_K$  and  $o.a_G$ 
      then return  $o.a_G(l^G)$ 
    else return  $o.a_G(l^G)$ 
  if  $f$  is a CF then
    while  $\exists o.a_K$  s.t.  $K \preceq G$ 
      if ACCURATE then
        if  $\forall l_k^K \in K(l^G)$  s.t.  $o.a_K(l_k^K) \neq \perp$  then
          return  $f_{K \rightarrow G}(o.a_K)(l^G)$ 
        else return null
      else return  $f_{K \rightarrow G}(o.a_K)(l^G)$ 
    return null
  else while  $\exists a_H$  s.t.  $G \preceq H$ 
    return  $f_{H \rightarrow G}(o.a_H)(l^G)$ 

```

Fig. 3. Algorithm to solve the qualified object access $o.a \downarrow^f l_G^i$

access is invariant with respect to the three forms of evolution discussed in this paper, given a bounded approximation introduced by granularity conversions.

Suppose that $\langle G \rangle$ is one of the granularity levels defined for attribute a , and suppose that from $\langle G \rangle$ an evolution has been executed involving granule l^G . In the case of acquisitions we consider the insertion of new values in the target granularity level. Suppose the evolution has not been performed yet. Assuming that no updates occurred, if the access $o.a \downarrow l^G$ results in the same value when executed just before and just after the evolution execution, the access is referred to as *evolution invariant*. Considering how we build granularity levels, and the specification of granularity conversions, the following result holds.

Proposition 1. *Given a granularity level $\langle G \rangle$ defined for a , and provided that a granularity level $\langle G' \rangle$ exists such that $o.a_{G'}(G'(l^G))$ is defined, every object access $o.a \downarrow l_G^i$ is evolution invariant.* \diamond

Evaluating the access just before and just after the execution of an evolution defined from $\langle G' \rangle$ to $\langle G \rangle$, the access results in the same value. By contrast, for granularity acquisitions and deletions the access is evolution invariant but with a bounded imprecision, which is due to the application of granularity conversions. Indeed, if the value defined for granule l^G is deleted, we can recover it if the value has been involved in a granularity evolution to granularity level $\langle G' \rangle$. In the case of granularity acquisition, the old and the new acquisition levels are related by a pair of (quasi)inverse granularity conversions, which guarantees the value consistency among the two levels, modulo a bounded error.

6.5 Unsolvable Object Access

We may characterize ST^2 _ODMGe object accesses that can be statically detected as unsolvable. As usual, **null** is returned whenever not enough information is available to solve the access. However, we can distinguish between accesses that

are statically known to be unsolvable, that is, for which no database state exists such that these accesses will produce a value different from `null`, and accesses that can produce or not an answer depending on the actual content of the database. Detecting object accesses that are statically unsolvable reduces query execution times, because the system does not need to execute them, but it may return immediately `null`. Given an object access $o.a \downarrow l^G$ (the case of $o.a \downarrow l_t^G \downarrow l_s^G$ follows straightforwardly), the following result holds.

Proposition 2. *Given attribute a defined for an object o , and given value v for a , such that a_{GLG} includes the granularity levels $\langle G_1 \rangle, \dots, \langle G_n \rangle$, the object access $o.a \downarrow l^G$ is unsolvable if one of the following conditions holds:*

- G is not related by \preceq to any of G_1, \dots, G_n ;
- $G \prec K, K \in G^\perp$;
- $H \prec G, H \in G^\top$.

◇

7 Concluding Remarks

In this paper we have investigated issues related to the evolution of multigranular spatio-temporal objects. The main contribution of this paper is the definition of ST^2_ODMGe , a multigranular spatio-temporal model supporting the adaptive management of multigranular spatio-temporal attributes. Our approach to evolutions allows one to model a large variety of situations. Consistency constraints on attribute values have been relaxed, because the run-time value of a multigranular attribute is a Cartesian product of multigranular values, linked in a connected acyclic graph through the specification of granularity conversions. Relying on such a structure, object accesses may be solved according to different strategies and error tolerances.

The ST^2_ODMGe model may be considered as a basis for future investigations on issues concerning evolutions of multigranular spatio-temporal objects. In particular, the development of a prototype of the model will allow us to investigate the trade off between the flexibility, provided by the model, and the consistency that is guaranteed by the statical specification of evolutions.

Efficient and comprehensive implementations are crucial. Several alternatives can be investigated including 1) implementation of the required features as class libraries on top of an existing DBMS, and 2) extensions to a DBMS engine. Both approaches have shortcomings. The former approach may not be able to support all required features; it may also have performance problems, as it may be impossible to allow the inclusion of specialized indexing techniques or query optimization techniques. The latter approach may require extensive implementation efforts and may also not support all required features, especially the ones depending on the application domain, like specialized spatial conversion operators.

Moreover, since evolution specifications are formulated according to the active database paradigm, it is important that tools for the analysis of evolution triggers be supported to detect non-terminating as well as non-deterministic executions. Note that such issues have been extensively investigated in the area of

active DBMS and no general solutions exist. However, for specialized domains, such as, in our case, the evolution of granularities, effective solutions to these issues could be found.

References

1. G. Andrienko, D. Malerba, M. May, and M. Teisseire. Mining spatio-temporal data. *J. of Intelligent Information Systems*, 27(3):187–190, 2006.
2. L. Arge, M. de Berg, H. J. Haverkort, and K. Yi. The Priority R-Tree: A Practically Efficient and Worst-Case Optimal R-Tree. In *Proc. of SIGMOD Int'l Conf. on Management of Data*, pages 347–358. ACM, 2004.
3. A. Belussi, C. Combi, and G. Pozzani. Towards a Formal Framework for Spatio-Temporal Granularities. In *Proc. of 15th Int'l Symp. 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 *Proc. of 6th Int'l Conf. On Flexible Query Answering Systems*, number 3055 in LNCS, pages 320–333. Springer-Verlag Berlin Heidelberg, 2004.
5. C. Bettini, S. Jajodia, and X. Wang. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer-Verlag Berlin Heidelberg, 2000.
6. E. Camossi, E. Bertino, G. Guerrini, and M. Mesiti. Handling Expiration of Multigranular Temporal Objects. *J. of Logic and Computation*, 14(1):23–50, 2004.
7. E. Camossi, M. Bertolotto, and E. Bertino. A multigranular Object-oriented Framework Supporting Spatio-temporal Granularity Conversions. *Int'l J. of Geographical Information Science*, 20(5):511–534, 2006.
8. E. Camossi, M. Bertolotto, and E. Bertino. Multigranular spatio-temporal models: Implementation challenges. In *Proc. of 16th SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems*. ACM, 2008.
9. H. Garcia-Molina, W.J. Labio, and J. Yang. Expiring Data in a Warehouse. In *Proc. of 24th Int'l Conf. on Very Large Data Bases*, pages 500–511. ACM, 1998.
10. 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 LNCS, pages 367–405. Springer-Verlag Berlin Heidelberg, 1998.
11. S. Orlando, R. Orsini, A. Raffaeta, A. Roncato, and C. Silvestri. Spatio-temporal Aggregations in Trajectory Data Warehouses. In *Proc. of 9th Int'l Conf. on Data Warehousing and Knowledge Discovery*, number 4654 in LNCS, pages 66–77. Springer-Verlag Berlin Heidelberg, 2007.
12. J. Skyt, C.S. Jensen, and L. Mark. A Foundation for Vacuuming Temporal Databases. *Data & Knowledge Engineering*, 44(1):1–29, 2003.
13. Y. Tao and D. Papadias. Historical spatio-temporal aggregation. *ACM Transactions on Information Systems*, 23(1):61–102, 2003.
14. D. Toman. Expiration of Historical Databases. In *Proc. of 8th Int'l Symp. on Temporal Representation and Reasoning*. IEEE Computer Society, 2001.
15. J. Yang and J. Widom. Incremental computation and maintenance of temporal aggregates. *The Int'l J. on Very Large Databases*, 12(3):262–283, 2003.
16. D. Zhang, D. Gunopulos, V.J. Tsotras, and B. Seeger. Temporal and spatio-temporal aggregation over data streams using multiple time granularities. *Information Systems*, 28(1-2):61–84, 2003.