Querying multigranular spatio-temporal objects

E. Camossi[1] M. Bertolotto[1] E. Bertino[2]

[1] University College Dublin, Ireland *
[2] Purdue University, Indiana, USA **

Abstract. The integrated management of both spatial and temporal components of information is crucial in order to extract significant knowledge from datasets concerning phenomena of interest to a large variety of applications. Moreover, multigranularity, i.e., the capability of representing information at different levels of detail, enhances the data modelling flexibility and improves the analysis of information, enabling to zoom-in and zoom-out spatio-temporal datasets. Relying on an existing multigranular spatio-temporal extension of the ODMG data model, in this paper we describe the design of a multigranular spatio-temporal query language. We extend OQL value comparison and object navigation in order to access spatio-temporal objects with attribute values defined at different levels of detail.

Key words: Spatio-temporal query language; Spatial and temporal granularities.

1 Introduction

As the available datasets are becoming increasingly large and often unnecessarily detailed due to the development of sophisticated collection technologies, effective methods for presenting information to users are required. In such respect approaches able to present the data at different levels of detail (i.e., *granularities*) represent an effective solution to facilitate the analysis when additional details are only required for specific subsets of the data. For example, zoom-out operations can improve the efficiency of spatio-temporal data mining algorithms, which are time consuming [1]. On the other hand, zoom-in operations can help in refining the mining of specific data subsets. Multigranularity, multiresolution and multiple representation have been investigated first for temporal data [5, 6], and more recently for both spatial [2, 22] and spatio-temporal data [7, 12, 9, 20].

In particular, the *ST*_ODMG data model [9] has been defined as extension of ODMG [10], the reference model for object-oriented databases. *ST*_ODMG

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

** 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

models multigranular spatio-temporal values, relying on the definition of temporal granularity proposed by Bettini et al. [6], which is commonly adopted by the temporal databases and reasoning community. The model also relies on a notion of spatial granularity compliant with the formalization of stratified map spaces proposed by Stell and Worboys [26]. Moreover, unlike other multigranular models, it incorporates a framework for the conversion of spatio-temporal values at different spatial and temporal granularities.

Since the effectiveness of a model greatly depends on the associated query language, several spatio-temporal query languages have been developed [13, 15, 17, 19]. Among these approaches, in [13, 15] SQL:99 has been extended to spatio-temporal support. The extension of SQL proposed by Chen and Zaniolo [13] is based on user-defined spatio-temporal aggregates and functions in order to minimize changes to SQL. Erwig et al. [15] propose an extension of SQL to include spatio-temporal objects defined in terms of abstract data types relying on the abstract framework defined in [18] for moving objects.

By contrast, the approaches by Huang and Claramunt [19] and by Griffiths et al. [17] define how to query spatio-temporal values in an object oriented paradigm. Huang and Claramunt propose an OQL spatio-temporal extension that includes spatio-temporal operators for evaluating spatial queries and topological relationships; Griffiths et al. propose supporting queries against spatio-temporal objects at application level.

This paper focuses on the access of multigranular spatio-temporal data, which has not been discussed in the previous proposals. Specifically, we investigate the impact of multigranularity on the specification and execution of spatio-temporal queries. Spatio-temporal multigranularity may not be simply supported relying only on data types and operators already available in object oriented and relational query languages: a specific query language must be designed in order to support accesses to subsets of data that refer to spatio-temporal *granules* and sets of granules, both explicitly and implicitly represented through constraints against database values. Thus, simple expressions for representing temporal and spatial granules at different granularities must be provided. Furthermore, expressions in a multigranular query language must support multigranular spatio-temporal comparison of attribute values. Moreover, since an attribute can be accessed with granules at a different granularity levels, a suitable set of value conversions to convert spatio-temporal data at different granularities has to be integrated in the query language. Such conversions should support attribute values conversions according to the semantics of the attribute involved in the query. Therefore the types of conversion should vary according to the data types and the semantics of the represented information.

In this paper we propose a multigranular spatio-temporal query language which provides specific solutions to these requirements; its design relies on the multigranular model defined in [9]. The overall conceptual design of a multigranular spatio-temporal model and query language addresses some important issues related to spatio-temporal multigranularity. Taking advantage of the extension capability given by the object oriented paradigm, this design can be applied to extend both object oriented and relational database models to support spatio-temporal multigranularity.

The language herein presented extends OQL, the set-oriented content-based query language associated with the ODMG data model, with features supporting queries against non-homogeneus multigranular spatio-temporal objects. The two mechanisms supported by SQL for querying data, namely value comparison and object navigation, have been extended with multigranular spatio-temporal capabilities: we have extended the conventional path expressions in order to query multigranular spatio-temporal objects, whose attributes can be accessed according to given spatial and temporal elements [16], expressed at multiple granularities.

Whenever attribute values at different granularities have to be accessed and compared in a multigranular path expression, multigranular conversions [9] are applied to obtain values expressed at a common granularity. Throughout the paper, the formalization we propose is described in details and its application is illustrated through significative examples. Moreover, we specify the consistency conditions for accessing multigranular spatio-temporal values.

The paper is organized as follows. In Section 2 we briefly describe the main characteristics of the *ST*_ODMG model. In Section 3 we specify how multigranular spatio-temporal values are accessed, while in Section 4 we give some illustrative examples of multigranular spatio-temporal queries. The complete syntax of *ST*_ODMG queries is reported in the Appendix. Finally, Section 5 concludes the paper outlining future research directions.

2 *ST*_ODMG: a Multigranular Spatio-temporal Model

Granularities in *ST*_ODMG are specified as mappings from an index set $\mathcal{IS}$ to the power set of the $\mathcal{TIME}$ and the $\mathcal{SPACE}$ domains, respectively. $\mathcal{TIME}$ is totally ordered. The supported $\mathcal{SPACE}$ domain is 2-dimensional (i.e., a proper subset of R^2). 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. Through granules we can specify the validity spatio-temporal bounds of attribute values. For instance, we can say that a value reporting the measure of the daily temperature in Dublin is defined for the first and the second day of January 2000. The granules of interest for this example can be identified by three textual labels: '01/01/2000", '02/01/2000", and '$Dublin$", that respectively identify two temporal and one spatial granules. The interior of granules of the same granularity cannot overlap[1]. Moreover, non-empty temporal granules must preserve the order given by the index set.

[1] Temporal granules, according to the definition in [5], do not overlap, while spatial granules can overlap on the granule boundaries.

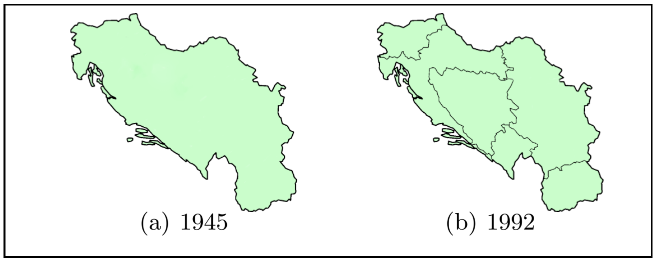

Fig. 1. Example of spatio-temporal geometric value

Spatial and temporal granularities are related by the *finer-than* relationship [5]. Such a relationship formalises the intuitive idea that different granularities correspond to different partitions of the domain, and that, given a granule of a granularity G, usually a granule of a coarser granularity H exists that properly includes it. For example, granularity *days* is finer-than *months*, and granularity *months* is finer-than *years*. Likewise, *municipalities* is finer-than *countries*. If a granularity G is finer-than H, we also say that H is coarser-than G.

Beyond the conventional database values, an *ST*_ODMG database schema can include spatial, temporal, and spatio-temporal values. 2-dimensional geometric vector features can be represented. Multigranular spatial and temporal data are uniformly defined as instances of two parametric types: $\mathit{Spatial}_{SG}(\tau)$ and $\mathit{Temporal}_{TG}(\sigma)$, which are specified according to a granularity (spatial and temporal, respectively) and an inner type. The inner type can be a type without spatio-temporal characteristics, or a geometric type. Moreover, multigranular spatio-temporal types are defined as compositions of *Spatial* and *Temporal*. Figure 1 illustrates the changes of the political boundaries of Balkan nations in different historical periods. A legal *ST*_ODMG type specification for this value is $Temporal_{years}(Spatial_{countries}(\mathrm{set}\langle\texttt{Polygon}\rangle))$.

Granularity conversions are provided in order to represent data at the most appropriate level of detail for a specific task, i.e., to increase or reduce the level of detail used for data representation. In *ST*_ODMG different semantics can be applied when converting values. The conversion of multigranular geometrical features is obtained through the composition of model-oriented and cartographic map generalisation operators [23] that guarantee topological consistency [4, 24], an essential property for data usability. Refinement operators perform the inverse functions. Such operators are classified according to the type of conversion applied [9]. 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 supported operators perform *contraction* and *thinning* (whose inverse is *expansion*); *abstraction* and *simplification* (whose inverse is *addition*).

On the other hand, the model provides also operators for converting quantitative (i.e., not geometrical) attribute values, both temporal and spatial. These conversions are classified in families according to the semantics of the operation performed: *selection* (e.g., `projection`, `main`, `first`), and *aggregation* (e.g., `sum`, `average`) convert values to coarser representation; their inverse functions, *restriction* and *splitting*, convert attribute values to finer representations, according to downward hereditary property [25] or according to a probability distribution, respectively. The different semantics we provided for converting spatio-temporal values at finer granularities have been introduced to address indeterminacy and imprecision that affect such type of conversion.

Granularity conversions have been proven to return legal values of the type system defined [9]. Moreover, those that generalize geometric attribute values to coarser spatial granularities have been proven preserve the semantics of the spatio-temporal data represented [9].

3 Multigranular Spatio-temporal Queries

The query language we propose extends value comparison and object navigation paradigms of OQL [10] to support multigranular spatio-temporal values. *Multigranular spatio-temporal path expressions* are the key concept of the resulting language: indeed, they are used to specify multigranular spatio-temporal queries. The access to multigranular class attributes is performed according to spatial and temporal *elements* and *expressions*, which specify portions of $\mathcal{SPACE}$ and $\mathcal{TIME}$ domains at a given granularity, in explicit or implicit form, respectively. Whenever path expressions involve different granularities, during their evaluation granularity conversions described in the previous section are applied. In the remaining of the section, multigranular spatial and temporal elements, expressions, and path expressions are described.

3.1 Spatial and temporal elements and expressions

Temporal elements have been formally introduced by Gadia [16], and then extended with respect to temporal granularities by Bertino et al. [3]. In a multigranular model, a temporal or spatial element is a set of granules expressed at the same granularity. For instance, $\{1999, 2000, 2001\}^{years}$ is a temporal element at granularity $years$, whereas $\{\text{Roma, Berlin, Zurich}\}^{municipalities}$ is a spatial element at granularity $municipalities$. Temporal and spatial elements can be converted to different granularities. Let Υ^H be a temporal (respectively, a spatial) element, with temporal (respectively, spatial) granularity H. Let G be a temporal (respectively, spatial) granularity, such that $G \preceq H$ or $H \preceq G$, then $G(\Upsilon^H)$ denotes the conversion of Υ^H to granularity G. For instance, $months(\{1999, 2000, 2001\})^{years}$ denotes the temporal element at granularity $months$ $\{\text{January 1999, February 1999}, \cdots, \text{December 2001}\}^{months}$.

Temporal and spatial elements at the same granularity can be combined by using the convential set operators: complement ($\setminus$), intersection ($\cap$), and union ($\cup$). Moreover, operators $first$ and $last$, relying on the order of temporal granules, can be applied to temporal elements. For instance, $first(\{1999, 2000, 2005\}^{years})$ returns the year 1999.

Temporal and spatial elements can be represented explicitly, as in the previous examples, or implicitly, by means of *temporal* and *spatial expressions*. Expressions represent conditions that are evaluated on database objects. Intuitively, the temporal and spatial elements resulting from temporal and spatial expressions specify *when* and *where* such conditions are satisfied. Conditions are specified through temporal and spatial variations of conventional comparison operators (i.e., $=, <>, <, >, <=, >=$) and binary topological relationships as defined by Egenhofer and Franzosa [14] (i.e., *equal*, *disjoint*, *meet*, *overlap*, *contains*, *inside*, *cover*, *coveredby*). For instance, each temporal comparison operator (i.e., $=_T, <>_T, <_T, >_T, <=_T, >=_T$) compares (spatio-)temporal and conventional values, and the resulting temporal expression represents the set of instants *when* the comparison is satisfied. Analougously, spatial expressions involving spatial comparison operators (i.e., $=_S, <>_S, <_S, >_S, <=_S, >=_S$) specify *where* a comparison is satisfied. Consider for example the spatio-temporal value:

$$v = \{\langle 2004,\ \{\langle \texttt{France,'Raffarin'}\rangle,\ \langle \texttt{United Kingdom,'Blair'}\rangle\}^{countries}\rangle,$$
$$\langle 2007,\ \{\langle \texttt{France,'Fillon'}\rangle,\ \langle \texttt{United Kingdom,'Brown'}\rangle\}^{countries}\rangle\}^{years}.$$

The temporal expression $v =_T$ 'Brown' returns the temporal element $\{2007\}^{years}$, whereas the spatial expression $v =_S$ 'Brown' returns the spatial element {United Kingdom}^{countries}.

Similarly, temporal and spatial variations of topological relationships are provided. For instance, $equals_T$, $disjoint_T$, $meet_T$, $overlaps_T$, $contains_T$, $inside_T$ can be applied between spatio-temporal values: the resulting temporal expressions represent the set of instants when the values satisfy the topological relationships. By contrast, $equals_S$, $disjoint_S$, $meet_S$, $overlaps_S$, $contains_S$, $inside_S$ are applied between spatial and spatiotemporal values, and return where the specified topological relationships are satisfied.

3.2 Spatio-temporal access and path expressions

Spatio-temporal path expressions are extension of conventional path expressions as used in object-oriented languages and models: the access to object attribute values is specified also according to temporal and spatial elements as described in the previous section. The access to conventional object attribute values is obtained through the usual postfix dot notation. To specify the access to spatio-temporal attribute values the $\downarrow$ operator is provided. As in conventional path expressions, the nesting of attribute accesses is allowed, and internal nodes in a path expression must result in single object identifiers.

Example 1. We define an object type for describing geographical historical maps. For each map the information recorded includes the political boundaries, the capital and the name of the head of government of each country. The definition of class `Map` in *ST*_ODMG syntax is as follows:

```
class Map ((extent Maps, key ...) {
 attribute Temporal_years(Spatial_countries(set<Polygon>)) boundaries {...};
 attribute Temporal_years(Spatial_countries(Point)) capitals {...};
```

attribute $Temporal_{years}(Spatial_{countries}(\texttt{string}))$ head_of_government {...};
};

Consider object o of type Map and value v of attribute head_of_government such that:

$v = \{\langle 2004, \{\langle \texttt{France}, \texttt{'Raffarin'}\rangle, \langle \texttt{United Kingdom}, \texttt{'Blair'}\rangle\}^{countries}\rangle,$
$\langle 2005, \{\langle \texttt{France}, \texttt{'Raffarin'}\rangle, \langle \texttt{Germany}, \texttt{'Merkel'}\rangle,$
$\langle \texttt{United Kingdom}, \texttt{'Blair'}\rangle\}^{countries}\rangle,$
$\langle 2007, \{\langle \texttt{France}, \texttt{'Fillon'}\rangle, \langle \texttt{Germany}, \texttt{'Merkel'}\rangle,$
$\langle \texttt{United Kingdom}, \texttt{'Brown'}\rangle\}^{countries}\rangle\}^{years}.$

Then, the temporal path expression $o.head\_of\_goverment \downarrow \{2007\}^{years}$ returns the spatial value: $\{\langle$France,'Fillon'$\rangle$, $\langle$Germany,'Merkel'$\rangle$, $\langle$United Kingdom,'Brown'$\rangle\}^{countries}$.
By contrast, the spatial path expression $o.head\_of\_goverment \downarrow \{\text{France}\}^{countries}$ returns the temporal value: $\{\langle$2004,'Raffarin'$\rangle$, $\langle$2007,'Fillon'$\rangle\}^{years}$. □

Whenever the spatial or temporal element involved in a spatio-temporal path expression includes more than one granule, the access results in a subset of the specified attribute value that corresponds to the *restriction* of the attribute value to the given element, as illustrated in the following Example.

Example 2. Given object o and value v of Example 1, the temporal path expression $o.head\_of\_goverment \downarrow \{2004, 2005\}^{years}$ returns the spatio-temporal value:

$v = \{\langle 2004, \{\langle \texttt{France}, \texttt{'Raffarin'}\rangle, \langle \texttt{United Kingdom}, \texttt{'Blair'}\rangle\}^{countries}\rangle,$
$\langle 2005, \{\langle \texttt{France}, \texttt{'Raffarin'}\rangle, \langle \texttt{Germany}, \texttt{'Merkel'}\rangle,$
$\langle \texttt{United Kingdom}, \texttt{'Blair'}\rangle\}^{countries}\rangle\}^{years}.$

Moreover, the spatial path expression $o.head\_of\_goverment \downarrow \{\text{France, United Kingdom}\}^{countries}$ returns the spatio-temporal value:

$v' = \{\langle 2004, \{\langle \texttt{France}, \texttt{'Raffarin'}\rangle, \langle \texttt{United Kingdom}, \texttt{'Blair'}\rangle\}^{countries}\rangle,$
$\langle 2005, \{\langle \texttt{France}, \texttt{'Raffarin'}\rangle, \langle \texttt{United Kingdom}, \texttt{'Blair'}\rangle\}^{countries}\rangle,$
$\langle 2007, \{\langle \texttt{France}, \texttt{'Fillon'}\rangle, \langle \texttt{United Kingdom}, \texttt{'Brown'}\rangle\}^{countries}\rangle\}^{years}.$ □

Given a spatio-temporal path expression, the following consistency property holds.

Property 1. (Spatio-temporal path expression consistency) Given object o, attribute a defined for o, and (temporal or spatial) element el, the access $o.a \downarrow el$ resulting in value v verifies the following consistency conditions:

If a is a *temporal (spatial) attribute* and
- el is a spatial (temporal) element, $o.a \downarrow el$ is undefined;
- el is a temporal (spatial) element including a single granule, v is a conventional i.e., non-temporal (non-spatial) value;
- el is a temporal (spatial) element including two or more granules, v is a temporal (spatial) value;

If a is a *spatio-temporal attribute* and
- el is a temporal element including a single granule, v is a spatial value;

– el is a spatial element including a single granule, v is a temporal value;
– el is a spatial element including two or more granules, v is a spatio-temporal value. $\triangledown$

Whenever the granularity in a spatio-temporal path expression differs from that of the value accessed, a granularity conversion is applied. To apply a granularity conversion, the starting and the target granularities must be related according to the finer-than relationship. If for the attribute being accessed a suitable granularity conversion has been defined in the database schema, such a conversion is applied. Otherwise, we can specify which conversion to apply in the path expression by using the access operator of the form $\downarrow^{gconv}$, where $gconv$ is a granularity conversion. To avoid conflicts, granularity conversions specified in path expressions take precedence over those specified in the schema. This enables to convert attribute values according to different semantics. Moreover, even if not required to perform the access, the application of an existing granularity conversion can be forced by specifying the access operator in the form $\downarrow^{G}$, where G is the target granularity of the conversion. For instance, when performing a temporal access, the value can be conveniently converted to a different spatial granularity, and vice versa, to format the access result.

Example 3. Given value v for attribute `head_of_government` of Example 1, the access $o.head\_of\_goverment \downarrow^{\texttt{last}_{years \rightarrow decades}} \{2000\text{-}2009\}^{decades}$ returns the spatial value:

$\{\langle\texttt{France,'Fillon'}\rangle,\ \langle\texttt{Germany,'Merkel'}\rangle,\ \langle\texttt{United Kingdom,'Brown'}\rangle\}^{countries}$.

To perform such an access, value v has been first converted to granularity *decades* according to the granularity conversion $\texttt{last}_{years \rightarrow decades}$. Then, the resulting spatio-temporal value has been accessed according to the temporal element $\{2000\text{-}2009\}^{decades}$. $\square$

4 Querying Multigranular Spatio-temporal Objects

The spatio-temporal query language described in the previous section extends the OQL syntax [10] to multigranular spatio-temporal path expressions as described above. Queries have the usual OQL `select-from-where` form. According to the OQL specification, spatio-temporal path expressions can be used in the target list to specify the data to retrieve, and in the where clause to express the conditions against multigranular spatio-temporal objects.

The complete syntax for the specification of multigranular spatio-temporal queries is presented in the Appendix. In this section, we describe its application through some illustrative examples, emphasizing the use of granularity conversions when querying multigranular attribute values. In particular, we focus on the access to spatio-temporal values, performed converting both spatio-temporal elements and attribute values. The access to temporal and spatial values follows straightforwardly. Moreover, we demonstrate the expressiveness of the spatio-temporal extensions of comparison and topological operators, which can be used to restrict the constraints used in the search, as well as to characterize the query results.

In these examples, the database schema including the class `Map` of Example 1 is extended with class `Flight`, representing passenger aircrafts. For each flight performed by an aircraft, we record its flight number, the departure and arrival airports, and the name of the pilot who flew the plane. Moreover, we record the spatial location of the aircraft during the flight. According to this specification, the class `Flight` in *ST*_ODMG is defined as follows:

$$
\begin{array}{l}
\texttt{class Flight (extent Flights, key ..) \{} \\
\quad \texttt{attribute } Temporal_{minutes}(\texttt{string})\texttt{ flightNum;} \\
\quad \texttt{attribute } Temporal_{minutes}(\texttt{string})\texttt{ departure;} \\
\quad \texttt{attribute } Temporal_{minutes}(\texttt{string})\texttt{ arrival;} \\
\quad \texttt{attribute } Temporal_{hours}(\texttt{string})\texttt{ pilot \{} \\
\qquad\qquad\qquad \texttt{main}_{hours \rightarrow months}\texttt{, restr}_{hours \rightarrow minutes}\texttt{ \};} \\
\quad \texttt{attribute } Temporal_{minutes}(Spatial_{meters}(\texttt{set}\langle\texttt{Region}\rangle))\texttt{ trips \{} \\
\qquad\qquad\qquad \texttt{r\_merge}_{meters \rightarrow provinces}\texttt{, r\_contr}_{provinces \rightarrow countries}\texttt{ \};} \\
\quad \texttt{\};}
\end{array}
$$

In particular, the spatio-temporal attribute `trips` reports the spatial location of the aircraft over time. For such attribute two spatial multigranular conversions have been specified, which support converting the aircraft location from the spatial granularity *meters* to the spatial granularity *provinces*, and from *provinces* to *countries*. The first conversion specified, i.e., `r_merge`, merges multiple regions in order to give the spatial representation of the aircraft with respect to a single region; by contrast, `r_contr` (i.e., region contraction), collapses regions in single points preserving topological consistency.

Other conversions have been specified for the temporal attribute `pilot`. Granularity conversion `main` converts the temporal value of this attribute to granularity *months* selecting the more frequent pilot name recorded for each month; by contrast, granularity conversion `restr` converts the same values to finer granularity *minutes* by applying the downward hereditary property [25], according to which if a multigranular value assumes the value v in a granule g, value v also refers to any *finer* granule g' included in g. For instance, given a value representing the address of a person, defined with temporal granularity *years*, each value referring to a year Y is the valid address of that person for every day of year Y.

Given the above specification for class `Flight`, the following query retrieves (the names of) the pilots flying the flight with number ‘AZ555’ during December 2007:

$$
\begin{array}{l}
\texttt{select distinct a.pilot } \downarrow hours(\{12/2007\}^{months}) \\
\texttt{from Flight f} \\
\texttt{where f.flightNum } \downarrow minutes(\{12/2007\}^{months})\texttt{ = 'AZ555'}
\end{array}
$$

We can further refine the query asking which pilot has more often flown such a flight, during the same period of time. The query has to be modified as follows:

$\texttt{select a.pilot} \downarrow \{12/2007\}^{months}$
$\texttt{from Flight f}$
$\texttt{where f.flightNum} \downarrow minutes(\{12/2007\}^{months})\ \texttt{= 'AZ555'}$

The granularity conversion `main` is automatically applied for converting the value of attribute `pilot` to granularity *months*. According to the semantics of this conversion, the value of `pilot` occurring more often in the selected month is the value for this month.

Beyond quantitative queries, multigranular spatio-temporal expressions that return geometric data can also be defined. For example, what follows is an example of a more complex query which retrieves the capitals and the corresponding countries that have been flown over by flight number 'AZ555' flown on Christmas 2007:

$\texttt{select distinct m.capitals,}$
$\quad \texttt{(f.trips}\downarrow^{countries}\texttt{((f.flightNum}\downarrow minutes(\{\texttt{25/12/2007}\}^{days}))=_T\texttt{'AZ555'))}$
$\qquad <>_S \perp$
$\texttt{from Map m, Flight f}$
$\texttt{where f.flightNum} \downarrow minutes(\{25/12/2007\}^{days})\ \texttt{= 'AZ555' and}$
$\quad \texttt{f.trips} \downarrow minutes(\{25/12/2007\}^{days})\ overlaps_S$
$\quad \texttt{m.capitals} \downarrow^{restr_{years \to days}} \{25/12/2007\}^{days}.$

Starting from the where clause, the topological test:

$\quad \texttt{f.trips} \downarrow minutes(\{25/12/2007\}^{days})\ overlaps_S$
$\quad \texttt{m.capitals} \downarrow^{restr_{years \to days}} \{25/12/2007\}^{days}$

compares two spatial values at granularity *countries*. The first value represents the trajectory of aircrafts, given through the position of their centroid over time, which is returned executing the temporal access on attribute `trips`. The temporal element specified for the access is given at granularity *minutes* as expected for such attribute, but granularity conversions `r_merge` and `r_contr`, defined in class `Flight`, are applied to convert its value to granularity *countries*. The result is further refined thanks to the access to attribute `flightNum` specified in the where clause. This spatial value is compared in the topological test with the capitals of the countries represented in some map on Christmas 2007, which are returned accessing attribute `capitals`. Note that the spatio-temporal value of attribute `capitals` is converted from granularity *years* to granularity *days* before evaluating the expression by appling granularity conversion `restr`, which supports downward hereditary property [25]. The spatial element resulting from the topological comparison includes the granules at granularity *countries* where the trajectories of the aircraft overlapped some country capitals.

To answer the query, however, this value is not sufficient, and we can note no attribute is defined in the database schema to represent the countries given in the maps. However, this information is implicitly stored because of the multigranular support, and through multigranular spatio-temporal path expressions we can make it explicit. Indeed, the required value is obtained by the spatial expression specified in the target list:

$$\texttt{(f.trips} \downarrow^{countries}$$
$$\texttt{((f.flightNum} \downarrow minutes(\{\texttt{25/12/2007}\}^{days}\texttt{))}=_T \texttt{ 'AZ555'))} <>_S \perp$$

This path expression is a little more complex than the previous ones: we have two nested path expressions, and a temporal and a spatial expression. First, the temporal path expression `f.flightNum` $\downarrow$ $minutes(\{25/12/2007\}^{days})$ returns the flight numbers of the flights flown during Christmas 2007. This value (let it be v) is temporally compared with the flight number 'AZ555': the temporal expression $v =_T$ 'AZ555' returns the temporal element at granularity *minutes* specifying the exact time of occurrence for the flight 'AZ555' during Christmas 2007. Then, this temporal element, namely te, is used to access the attribute `trips` in the temporal path expression `f.trips` $\downarrow^{countries}$ te. This temporal access returns the trips performed by the flight already selected through the where clause during the period of time specified by temporal element te. Before performing the access, the value of `trips` is converted at spatial granularity *countries* by appliying granularity conversions `r_merge` and `r_contr`. The spatial value at granularity *countries* resulting from the access (let it be v') is then involved in the spatial comparison $v' <>_S \perp$. This straightforward comparison simply returns the spatial element including the granules where v' is defined, at granularity *countries*, which at the end represents the (granules of) countries whose capitals have been flown over by the aircrafts selected through the where clause.

5 Conclusions

In this paper we have defined a multigranular spatio-temporal object oriented query language. The language is based on the spatio-temporal model we previously defined in [9]. The query language extends OQL, the ODMG query language, to support multigranular spatio-temporal access. The access to spatio-temporal values is given in terms of spatial and temporal elements and expressions. Their use in multigranular spatio-temporal path expressions, combined with multigranular conversions, allows one to access and compare spatio-temporal values expressed at different granularities.

Unlike previous work on spatio-temporal query languages, in this paper we provide a formalization that addresses several important issues arising from the introduction of spatio-temporal multigranularity. Despite the importance of multigranularity and of the integrated management of spatio-temporal information, no currently available DBMS include suitable tools for dealing with spatio-temporal data at different levels of detail. The overall design proposed in [9] and in this paper is suitable for the development of both object-oriented and relational multigranular spatio-temporal models and query languages. Our design relies on ODMG and the related query language OQL: the features we introduce to support spatio-temporal multigranularity extend the model type systems with specific types and suitable operators for handling time and space, as well as spatial and temporal granularities. Accordingly, the model Data Definition Language, value comparison and object navigation have been extended to support multigranular spatio-temporal types and expressions.

The work presented in this paper can be refined and extended in different ways. First of all, we plan to develop a software prototype to assess the effectiveness and to test the performance of our approach. In this respect, a suitable indexing system for spatio-temporal values (such as the one proposed in [8] for dealing with the expiration of historical values, and the one proposed in [27]), an efficient representation for those values to support value coalescing, and an efficient implementation for granularity conversions, are desirable. Moreover, in order to build a comprehensive spatio-temporal information system, conventional spatial operations, like intersections and overlay, should be integrated in the formalization provided. Finally, we plan to extend spatio-temporal comparison operators to support qualitative semantics both for time and space.

Acknowledgement

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.

References

1. G. Andrienko, D. Malerba, M. May, M. Teisseire, Mining spatio -temporal data. In *Journal of Intelligent Information Systems*. 27(3):187-190, 2006.
2. S. Balley, C. Parent, and S. Spaccapietra. Modelling Geographic Data with Multiple Representations. In *International Journal of Geographical Information Science*. Taylor & Francis. 18(4):327-352, 2004.
3. E. Bertino, E. Ferrari, G. Guerrini, and I. Merlo. *T_ODMG*: An ODMG Compliant Temporal Object Model Supporting Multiple Granularity Management. In *Information Systems*, 28(8):885-927, 2003.
4. M. Bertolotto. Geometric Modeling of Spatial Entities at Multiple Levels of Resolution. Ph.D. Thesis, Università degli Studi di Genova, 1998.
5. C. Bettini, C. Dyreson, W. Evans, R. Snodgrass, and X. Wang. A Glossary of Time Granularity Concepts. In *Proc. of Temporal Databases: Research and Practice*, LNCS 1399:406-413, 1998.
6. C. Bettini, S. Jajodia, and X. Wang. Time Granularities in Databases, Data Mining, and Temporal Reasoning. Springer-Verlag, 2000.
7. T. Bittner. Reasoning about qualitative spatio-temporal relations at multiple levels of granularity. In *Proc. of the* 15^{th} *European Conf. on Artificial Intelligence*, F. van Harmelen (Ed.). IOS Press. 2002.
8. E. Camossi, E. Bertino, G. Guerrini, M. Mesiti. Handling Expiration of Multigranular Temporal Objects. In *Journal of Logic and Computation*, Oxford University Press. 14(1):23-50, 2004.

9. E. Camossi, M. Bertolotto, and E. Bertino. A multigranular Object-oriented Framework Supporting Spatio-temporal Granularity Conversions. In *International Journal of Geographical Information Science*. Taylor & Francis. 20(5): 511-534, 2006.
10. R. Cattel, D. Barry, M. Berler, J. Eastman, D. Jordan, C. Russel, O. Schadow, T. Stanienda, and F. Velez. The Object Database Standard: ODMG 3.0. Morgan-Kaufmann, 2000.
11. C. Claramunt and M. Thériault. Managing Time in GIS: an event oriented approach. In *Proc. of the Int'l Workshop on Temporal Databases: Recent Advances in Temporal Databases*, J. Clifford and A. Tuzhilin (Eds.). Springer-Verlag. Workshops in Computing. 23-42, 1995.
12. C. Claramunt and B. Jiang. Hierarchical Reasoning in Time and Space. In *Proc. of the 9^{th} Int'l Symposium on Spatial Data Handling*, 41-51, 2000.
13. C.X. Chen and C. Zaniolo. SQLST: A Spatio-Temporal Data Model and Query Language. In *Proc. of 19^{th} Int'l Conf. on Conceptual Modeling / the Entity Relational Approach*. Springer-Verlag Berlin. Lecture Notes in Computer Science. 1920:96-11, 2000.
14. M.J. Egenhofer, and R.D. Franzosa, Point-Set Topological Spatial Relations. In *International Journal of Geographical Information Science*, 5(2):161-174, 1991.
15. M. Erwig, and M. Schneider, STQL A Spatio-Temporal Query Language. In *Mining Spatio-Temporal Information Systems*, R. Ladner, K. Shaw, and M. Abdelguerfi (eds.), Kluwer Academic Publishers, Chapter 6, 105-126, 2002.
16. S.K. Gadia. A homogeneus relational model and query languages for temporal databases. In *ACM Transactions on Database Systems*, 13(4):418-448, 1988.
17. T. Griffiths, A.A.A. Fernandes, N.W. Paton, R. Barr. The Tripod spatio-historical data model. In *Data Knowledge and Engineering*, Elsevier. 49(1): 23-65, 2004.
18. R.H. Güting, M.H. Bhölen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M. Shneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. In *ACM Transaction On Database Systems*, 25:1-42, 2000.
19. B. Huang and C. Claramunt. STOQL: An ODMG-based Spatio-Temporal Object Model and Query Language. In *Proc. of the 10^{th} Int'l Symposium on Spatial Data Handling*, 2002.
20. V. Khatri, S. Ram, R.T. Snodgrass, and G. O'Brien. Supporting User Defined Granularities and Indeterminacy in a Spatio-temporal Conceptual Model. In *Annals of Mathematics and Artificial Intelligence*, 36(1):195-232, 2002.
21. D.H. Kim, K.H. Ryu, and H.S. Kim. A spatio-temporal database model and query language In *Journal of Systems and Software*, Elsevier, 55(2):129-149, 2000.
22. L. Kulik, M. Duckham, and M.J. Egenhofer. Ontology driven Map Generalization. In *Journal of Visual Language and Computing*, 16(2):245-267, 2005.
23. J-C. Muller, J.P. Lagrange, and R.Weibel (eds.) GIS and Generalization: methodology and practice. Taylor and Francis, 1995.
24. A. Saalfeld. Topologically consistent line simplification with the Douglas-Peucker algorithm. In *Cartography and Geographic Information Science*, 26(1):7-18, 1999.
25. Y. Shoham, Temporal Logics in AI: Semantical and Ontological Considerations. In *Artificial Intelligence*, 33(1):89-104, 1987.
26. J.G. Stell and M. Worboys. Stratified Map Spaces: A Fomal Basis for Multi-Resolution Spatial Databases. In *Proc. of 8^{th} Int'l Symposium on Spatial Data Handling*, 1998.
27. D. Zhang, D. Gunopulos, V.J. Tsotras, B. Seeger. Temporal Aggregation over Data Streams using Multiple Granularities. In *Proc. of 8^{th} Int'l Conf. on Extending Database Technology*. 643-663, 2002.

28. ORACLE™, Oracle Corp. http://www.oracle.com.
29. PostgreSQL, Inc. http://www.postgresql.org.

Appendix: Syntax of multigranular spatio-temporal queries

In the following, we report the syntax of *ST*_ODMG queries in Backus-Naur form.

```
<query> ::= select [distinct] <target_list> from <q_source> where
              <search_condition>;
<target_list> ::= [<target_list>, ] <value>
<q_source> ::= [<q_source>] class_name class_alias
<search_condition> ::= [not] <cond> |
              <search_condition> <bin_bool_op> <search_condition>
<value> ::= [<path>].value | <conv_value> | <t_value> | <s_value> | <st_value>
<path> ::= object_name.<internal_path>
<internal_path> ::= [<internal_path>.]attribute_name
<conv_value> ::= conv_attr_name | <t_value> ↓[<g_conv>] t_granule | [not] <conv_value> |
              <s_value> ↓[<gconv>] s_granule | <conv_value> <bin_op> <conv_value> |
              <t_value> <comp_op> <conv_value> | <s_value> <comp_op> <conv_value>
<t_value> ::= temp_value | temp_attr_name [<t_access>] | [not] <t_value> |
              <t_value> <bin_op> <t_value> | <st_value> ↓[<g_conv>] s_granule
<s_value> ::= spat_value | spat_attr_name [<s_access>] | [not] <s_value> <s_value>|
              <bin_op> <s_value> |<st_value> ↓[<g_conv>] t_granule
<st_value> ::= spatio-temp_value | st_attr_name [<access>]
<access> ::= <s_access> | <t_access>
<t_access> ::= ↓[<g_conv>] <temp_elem>
<temp_elem> ::= explicit_temp_elem | <t_expr> | t_gran_name (<temp_elem>) |
              first(<temp_elem>) | last(<temp_elem>)
<t_expr> ::= <temp_value> <bin_op_T > <conv_value>
<s_access> ::= ↓[<g_conv>] <spat_elem>
<spat_elem> ::= explicit_spat_elem | <s_expr> | s_gran_name (<spat_elem>)
<s_expr> ::= <spat_value> <bin_op_S > <conv_value>
<bin_op> ::= <comp_op> | <toprel> | <arithm_op> | <bool_op> | <set_op>
<arithm_op> ::= + | - | / | *
<bool_op> ::= and | or
<set_op> ::= \ | ∪ | ∩
<comp_op> ::= = | <> | < | > | <= | >=
<toprel> ::= equal | disjoint | meet | overlap | contains | inside | cover | coveredby
<bin_op_T > ::= <comp_op_T > | <toprel_T >
<comp_op_T > ::= =_T | <>_T | <_T | >_T | <=_T | >=_T
<toprel_T > ::= disjoint_T | meet_T | overlaps_T | equals_T | contains_T | inside_T|
              cover_T | coveredby_T
<bin_op_S > ::= <comp_op_S > | <toprel_S >
<comp_op_S > ::= =_S | <>_S | <_S | >_S | <=_S | >=_S
<toprel_S > ::= disjoint_S | meet_S | overlaps_S | equals_S | contains_S| inside_S|
              cover_S| coveredby_S
```

Non terminal symbols *conv_attr_name*, *spatial_attr_name*, *temporal_attr_name*, *st-_attr_name* are names for conventional, spatial, temporal and spatio-temporal attributes, while *attr_name* is a generic attribute name, used in a nested path. Nested paths must result in a single object identifier; *temp_value*, *spat_value*, *spatio-temp_value* are explicit temporal, spatial and spatiotemporal values, respectively; *class_name* is a class name, and *class_alias* is the class alias. *s_granule* and *t_granule* are a spatial and a temporal granule, respectively; *t_gran_name* and *s_gran_name* are a temporal and a spatial granularity name, respectively; *explicit_temp_element* and *explicit_spat_element* are a temporal and a spatial element, respectively, represented explicitly. Terminal elements `<temp_elem>` and `<spat_elem>`, when referred to in temporal and spatial access (`<t_access>` and `<s_access>`), include more than one granule. Finally, terminal element <*g_conv*> is a granularity conversion. Note that <*g_conv*>, when applied in attribute access, represents a granularity conversion that should be applicable to the attribute value.