Fundamenta Informaticae CVII (2011) 1001–1026 1001
IOS Press

Querying Cardinal Directions between Complex Objects in Data Warehouses

Ganesh Viswanathan
Department of Computer & Information Science & Engineering
University of Florida
Gainesville FL 32611 USA
gv1@cise.ufl.edu

Markus Schneider *
Department of Computer & Information Science & Engineering
University of Florida
Gainesville FL 32611 USA
mschneid@cise.ufl.edu

Abstract. Data warehouses help to store and analyze large multidimensional datasets and provide enterprise decision support. With an increased availability of spatial data in recent years, several new strategies have been proposed to enable their integration into data warehouses and and perform complex OLAP analysis. Cardinal directions have turned out to be very important qualitative spatial relations due to their numerous applications in spatial wayfinding, GIS, qualitative spatial reasoning and in domains such as cognitive sciences, AI and robotics. They are frequently used as selection and restriction criteria in spatial queries. In data warehouses, cardinal directions can be used to perform spatial OLAP and feature navigation operations. In this article, we introduce and develop the *Objects Interaction Graticule* (*OIG*) approach to query the cardinal direction relations among spatio-temporal objects in data warehouses. First, we apply a tiling strategy that determines the zones belonging to the nine cardinal directions of each spatial object at a particular time and intersects them. This leads to a *collection of grids* over time called the Objects Interaction Graticule (OIG). For each grid cell, the information about the spatial objects that intersect it is stored in a *Objects Interaction Matrix*. In the second phase, an interpretation method is applied to these matrices to determine the cardinal direction between the moving objects. The results obtained for each valid *instant* over the objects' lifetime describe the variation in the objects movement over time. Thisis integrated as a spatio-temporal OLAP operation in a novel *moving objects data warehouse* (*MODW*) that provides an extensible framework for supporting complex structured objects. Finally, we define new directional predicates that extend MDX querying and leverage OLAP between moving objects.

Keywords: Cardinal directions, data warehouse design, spatio-temporal OLAP

Address for correspondence: E301 CSE Building, University of Florida, P.O. Box 116120, Gainesville, FL 32611-6120 USA
*This work was partially supported by the National Aeronautics and Space Administration (NASA) under the grant number NASA-AIST-08-0081.

1. Introduction

A *data warehouse* is a large, subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making processes [14, 17]. Online Analytical Processing (OLAP) is the technology that helps the decision maker perform complex analyses over the information stored in the data warehouse. Data warehouses and OLAP enable organizations to gather overall trends and discover new avenues for growth. With the emergence of new applications in areas such as geo-spatial, sensor, multimedia and genome research, there is a large increase in the amount of spatio-temporal data that needs to be properly managed and analyzed. This data is often complex, of hierarchical, multidimensional nature, and exhibits spatio-temporal variations. A good framework to store, query and mine such datasets involves *moving object data warehouses* (*MODW*) that bring the best tools for data management to support complex, spatio-temporal analysis.

A moving objects data warehouse can be defined as a large, subject-oriented, integrated, time-variant, non-volatile collection of analytical, *spatio-temporal* data that is used to support scientific data analysis and the strategic decision-making process for an enterprise. It is a full-fledged data warehouse which provides native support for spatio-temporal data and advanced spatio-temporal OLAP operations on them. Such OLAP operations on spatio-temporal events can include *basic querying operations*, such as "*Find the states in eastern USA where the rate of change of Xbox sales in any two quarters between 2000 and 2010 was greater than 3%*", or involve spatial analysis operations such as convex hull, "*Find the smallest convex region in the western USA containing the maximum number of college towns where more than 2500 units of Kinect were sold in 2010*", or even advanced spatial OLAP involving geometric union such as "*Return the geometry of the region in south-eastern USA described by the counties where DropBox usage exceeded that of Twitter in the first four months of 2011*". Many other interesting spatial aggregation queries are possible when spatial data is fully integrated into data cubes and an effective approach for multidimensional querying is available on them. Data warehouses thus provide an *integrated system* to store, manage and analyze complex, geo-spatial phenomena over a historical time-range, and can actively (in real-time) support enterprise decision making and scientific data analysis.

Qualitative relations between spatial objects include cardinal direction relations, topological relations and approximate relations. Of these cardinal directions have turned out to be very important due to their application in spatial wayfinding, qualitative spatial reasoning and in domains such as cognitive sciences, robotics, and GIS. In spatial databases and GIS, they are frequently used as selection criteria in spatial queries. However, currently there is no available method to model and query for cardinal directions between moving objects (that exhibit spatio-temporal variations). We have described a novel system to model cardinal directions between spatial regions in databases using the *Objects Interaction Matrix* (*OIM*) model in [5]. This model solves the problems found in existing direction relation models like the unequal treatment of the two spatial objects as arguments of a cardinal direction relation, the use of too coarse approximations of the two spatial operand objects in terms of single representative points or MBRs, the lacking property of converseness of the cardinal directions computed, the partial restriction and limited applicability to simple spatial objects only, and the computation of incorrect results in some cases. The basis of the model is a bounded grid called the *objects interaction grid* which helps to capture the information about the spatial objects that intersect each of its cells. Then, we use a matrix and apply an special interpretation method to determine the cardinal direction between the operand spatial objects.

In this article, we present and develop a novel *Objects Interaction Graticule* (*OIG*) system for modeling cardinal directions between moving objects and querying for such relations. In particular, we

extend the work in [40] by motivating the need for supporting complex objects in data warehouses and presenting a framework for the same, by introducing a complete application for weather event analysis involving cardinal direction relations, and providing additional directional predicates such as *existential* and *complete cover*, which extend MDX queries for spatio-temporal OLAP in the data warehouse. Our method improves upon the OIM model by adding support for moving objects and provides an innovative approach to model cardinal direction relations inside data warehouses. In a first phase, we apply a multi-grid tiling strategy to determine the zones belonging to the the nine cardinal directions of each spatial object at a particular time and intersects them. This leads to a collection of grids over time called the Objects Interaction Graticule. For each grid cell the information about the spatial objects that intersect it is stored in an Objects Interaction Matrix. In the second phase, an interpretation method is applied to these matrices to determine the cardinal direction between the moving objects. These results are integrated into MDX queries [22] using directional predicates.

The rest of this article is organized as follows. In Section 2, we provide a survey of existing techniques to model cardinal directions and discuss their applicability to data warehouses. Further, we deal with models for direction relations between moving objects. In Section 3, we introduce our design framework for a moving objects data warehouse system and provide an overview of the Objects Interaction Graticule model for modeling cardinal direction relations between moving objects in Section 4. The tiling phase of the model, explained in Section 5, helps to generate the OIM matrix, and its interpretation is achieved in Section 6. Section 7 describes the integration of complex data types in the data warehouse and motivates the application of OIG to mixed combinations of simple and complex spatial objects. Section 8 introduces novel direction predicates and supporting MDX query language extensions that complete the user-view using cardinal direction querying in data warehouses. Finally, Section 9 concludes the paper and provides some directions for future research.

2. Related Work

A detailed survey of existing approaches for modeling cardinal directions between region objects is provided in [5]. Here, we provide an overview of these models along with the strategies commonly used for handling spatio-temporal variations among moving objects to illustrate the existing work in this area.

Various strategies have been proposed to model direction relationships between spatial objects. However, they have in-common applied the same design paradigm. First, a decision needs to be made about how to approximate spatial objects, so that a simpler equivalence between the two operand objects is found and handled. Second, a set of direction relations needs to be explored and defined, which need to be mutual exclusive and complete. Third, objects are brought into relations, and a set of rules are applied to determine their direction relationships.

Models that capture directions between region objects have evolved from reducing these objects to points, to the use of minimum bounding rectangles to approximate their extent, and ultimately to the final goal of considering their exact shapes. So far, direction relation models that can precisely capture region objects' shapes do not exist. Therefore, the *point approximation* and the *minimum bounding rectangle approximation* are the two major techniques used to reduce region objects to simpler equivalences before modeling direction relationships between them. Another not so widely used approximation technique is to apply the minimum bounding circle to spatial objects.

Point approximation models ignore the extent of spatial objects and determine their direction relation-

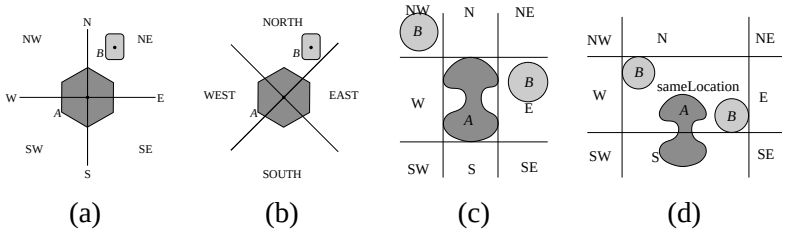


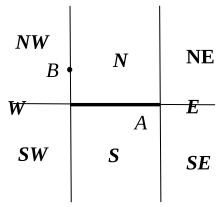
Figure 1. Projection-based (a) and cone-shaped (b) models, and the Direction-Relation Matrix model with *A* as reference object (c) and with *B* as reference object (d)

ships according to their representative points. A general strategy applied by most point approximation models is the *tiling-based* strategy. The *tiling-based* models define cardinal direction relationships by using partitioning lines that subdivide the plane into tiles. They can be further classified into *projection-based* models and *cone-shaped* models. The *projection-based* models create partitioning lines parallel to the coordinate axes, while the *cone-shaped* models partition the space into angular zones. The technique of generalizing the spatial objects into points is easy to use but usually leads to rather inaccurate models.

In *minimum bounding rectangle approximation* (*MBR approximation*) models, minimum bounding rectangles are used to approximate at least one object. The shape of the object is irrelevant to the final result. *MBR approximation* models can be further classified to two categories, one is the *tiling-based* models, and the other is so called *interval-based* models. *Interval-based* models make use of the minimum bounding rectangles of both operand objects and apply Allen's 13 interval relations to the rectangle projections on the *x*- and *y*-axis respectively. Direction relationships are determined according to the interval relations. These models are superior to point approximation models due to the fact that the extents of both operand objects contribute to the final result. However, shapes of spatial objects are still ignored. The tiling-based strategy also applies to MBR approximation models. According to the roles the two operand objects play in the process of partitioning the space, *tiling-based* models can be divided into two categories, the *symmetric* approach and the *asymmetric* approach. The *asymmetric* approach treats the two operand objects differently, only one object is approximated with its minimum bounding rectangle, and is placed in the center for partitioning the space. The intersections of the other operand object with partition zones are recorded and contribute to the final result. In contrast, the *symmetric* approach treats the two operand objects equally. The partition of the space is done by the cooperation of both objects. As a result, *tiling-based* models with the *asymmetric* approach is superior to *interval-based* models due to the fact that it relaxes one object from the minimum bounding rectangle approximation. However, they suffer from the problem raised by the unequal treatment of the two operand objects. From this perspective, we show that our *OIM* model, which applies the symmetric tiling approach, maintains the strength of shape capturing while solving the unequal treatment problem.

Another approximation technique, the *minimum bounding circle approximation*, creates a circle around the operand object. Based on the interaction between representative circles and the information about where the object locates in the circle, direction relationships between objects can be identified. Examples of *tiling-based* models and *minimum bounding rectangle-based* (*MBR-based*) models proposed to capture cardinal direction relations between simple spatial objects (like *point*, *line*, and *region* objects) as instances of *spatial data types* [32] are shown in Figure 1.

Tiling-based models use partitioning lines that subdivide the plane into tiles. They can be further



(a)

$$\begin{bmatrix} 32 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(b)

	$[TL$	T	TR	R	BR	B	BL	L	$DP]$
NW	$[0$	0	0	1	0	0	0	0	$0]$
N	$[0$	0	0	0	0	0	0	1	$0]$

(c)

Figure 2. The DRM partition for the line *A* and the point *B* (a), the corresponding Deep Direction-Relation Matrix (b), and the vectors recorded in the Deep Direction-Relation Matrix (c)

classified into *projection-based* models and *cone-shaped* models, both of which assign different roles to the two spatial objects involved. The *projection-based* models define direction relations by using partitioning lines parallel to the coordinate axes. The *Direction-Relation Matrix* (DRM) model [10, 35] helps capture the influence of the objects' shapes as shown in Figure 1c. However, this model only applies to spatial objects with non-temporal variations. It also leads to imprecise results with intertwined objects. The *coarse* DRM model is extended by Goyal and Egenhofer in [11] to make it possible to deal with points and lines. They develop a *Deep Direction-Relation Matrix* which increases the resolution of the values of each element in the original matrix. The Deep Direction-Relation Matrix uses a vector for each partition to record not only the intersections of the target objects with the interior of the partitions, but also the intersection of the target objects with the boundary lines and points of the tile. As shown in Figure 2c, the vector captures the intersections of the target object with top (T), top-left (TL), top-right (TR), right (R), bottom-right (BR), bottom (B), bottom-left (BL), left (L) and the interior (DP). Then the decimal values of the vectors are computed and stored into a 3×3 matrix as shown in Figure 2b. This model nicely supports the computation of directions between generic spatial data types. However, it still suffers from all the problems appearing in the original DRM model, and moreover, lacks a strategy to manage the large number of matrices generated as results of a direction query. The completeness and uniqueness of the resultant directions is thus hard to evaluate.

The *cone-shaped* models define direction relations by using angular zones. The *MBR-based* model [26] approximates objects using minimum bounding rectangles and brings the sides of these MBRs into relation with each other using Allen's interval relations [2].

We have introduced an improved modeling strategy for cardinal directions between both simple and multi-component complex region objects using a two-step strategy called the *Objects Interaction Matrix* (*OIM*) model in [5]. This model solves many of the problems in earlier models for gathering cardinal directions among spatial objects such as converseness, limited applicability to simple region objects, incorrect results, etc. However, OIM is only built for spatial objects, and does not consider temporal variations among spatial phenomena. In [4], we have explored a new strategy to evaluate the development of cardinal directions among moving *point objects*, for applications such as tracking the eye of a hurricane over its lifetime.

The design and construction of data warehouses has been a topic of research in both academia and industry, for well over a decade. The technique of performing real-time ("online") analysis has also been studied in several works [30, 3, 33]. We describe some basic approaches here that are relevant to the introduction of our objects interaction graticule model and to motivate its use for spatial OLAP. Over

the past decade several approaches have been proposed for modeling multidimensional data. Existing conceptual modeling approaches can be broadly classified into *extensions of Entity Relationship (E/R) models* ([8, 16, 31, 20, 36]), *extensions of the Unified Modeling Language (UML)* ([1, 19, 29]) and *ad-hoc design models* ([28, 9, 13, 41, 39]). The data warehouse helps to store and query over large, multidimensional datasets and is hence a good choice for storing and querying spatio-temporal data. We have presented a conceptual, user-centric approach to data warehouse design based on *abstract data types* (*ADTs*), called the $\mathcal{BigCube}$ model in [39].

Research in extending OLAP to spatio-temporal events has attracted the attention of both data warehousing and GIS communities. Rivest *et al.* [30] present spatial OLAP as a paradigm that enables "drilling" on layers of spatial data on maps, and allows to perform supporting aggregations. Although they do not provide a formal model for this, a commercial implementation is presented. Shekhar *et al.* [34] introduce the MapCube as a visualization tool for spatial data cubes and specify roll-up and drill-down operations on its aggregation hierarchy. The MultiDim [21] conceptual model classifies data-dimension levels in data warehouses as spatial and non-spatial based on the type of members they contain. Topological relationships are used to relate spatial data levels with one another. Stefanovic *et al.* provide another approach that differentiates between geometric, non-geometric or combination dimension hierarchies and associates operations to each based on the type. Trajectory data warehouses are introduced in [25] to provide advanced reporting capabilities and facilitate the use of data mining algorithms on aggregate data. Another important aspect of handling large, multidimensional objects in data warehouses is the size of the resulting data cubes. The data cubes should be built at levels such that multidimensional querying and aggregation can happen in an online manner. To faciliate this several approaches have been proposed. An interesting discussion on compression of data cubes in the presence of multiple simultaneous hierarchical range queries is presented in [7]. [6] introduces a compressed data structure called KLSA, that implements a technique to support accuracy control in compressed multidimensional data cubes that can be efficiently used in QoA-based OLAP tools. These concepts can be applied to large cubes if, for example, satellite data such as scatterometer and wind speed metrics are used for analysis over a long timeline. Moving objects, their formal data type characterizations and operations have been introduced in [12, 18]. However there is the lack of a model for *qualitative direction relations between moving objects* in all existing works. This paper provides a clear solution to this problem by first describing the basics of the moving objects data warehouse framework in Section 3 and then introducing the objects interaction graticule model for gathering direction relations between moving objects in Section 4. Later sections describe the OIG model in detail and deal with the support for cardinal directions among complex objects in data warehouses.

3. Moving Objects Data Warehouses: An Approach to Modeling Spatio-Temporal Events among Multidimensional Data

For more than a decade, data warehouses have been at the forefront of information technology applications as a way for organizations to effectively use information for business planning and decision making. In recent years, due to the large spread of *location-aware* devices such as smart phones, GPS and wireless sensors, large amounts of spatial data is getting generated. Existing proposals for data warehouses capable of handling complex objects with spatial (and temporal) variations focus on several different functional requirements, but we are yet to find a generalized approach capable of addressing both users'

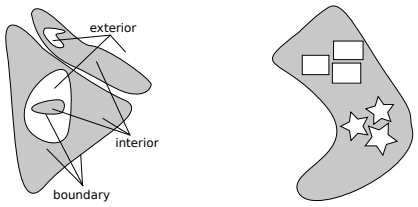


Figure 3. Illustration of (a) a complex region object with three faces and its interior, boundary and exterior point sets, and (b) a single face, also denoted as a simple region with holes.

and scientific analysis requirements. For example, though several approaches exist in both academia and industry for complex data warehousing and OLAPing, the terminology used is often misleading, in the sense that the same terms often mean different things in different models. One reason for this is the overall complexity of modeling multidimensional data with real-world semantics. However, in order to develop a standard for multidimensional OLAP, we need a complete and unified taxonomy of data warehouse components and operations for navigating, querying and extending analysis on such datasets. In this section, we take a step in this direction by proposing a novel moving objects data warehouse framework to help handle spatio-temporal changes among complex multidimensional objects. Later, cardinal directions are used in spatial OLAP queries inside the data warehouse.

Before describing a generic data warehouse framework to support complex spatio-temporal phenomena, we first examine the existing work on describing and warehousing such data. Several models have been proposed to design data warehouses at conceptual, logical and physical design levels. Conceptual models such as the MADS approach from Zimanyi *et al.* [27], focus only on modeling real-world scenarios at an abstract level to meet user requirements and do not consider other system or implementation dependencies. Logical design involves transformations from the conceptual level to a logical (discrete) level that is suitable for meeting computational and system requirements. The exact means of data storage and implementation are considered in physical design. Most conventional approaches for data warehousing in all three levels consider only alphanumeric data and optimize aggregation operations on them. In recent years, however, there have been proposals for new *complex object data warehouses*, which can store, manage and help analyze spatial, temporal and spatio-temporal data. Spatial data refers to data having a geographic location and/or geometric extent. Examples of such data include *point, line* and *region* objects. Such spatial objects can be described as either *simple* or *complex*. Simple spatial objects are single component and hole-free such as a *simple point* describing the location of Westminster Abbey in London, or a *simple line* describing the extent of the German Autobahn A8 between Stuttgart and Munich with no posted speed limits, or a *simple region* showing the extent of spread of radioactivity around Fukushima perfecture in Japan in 2011. Complex spatial objects involve complex structures such as multi-components (the different 'faces' of a region), holes within faces (the Vatican inside the extent of Italy), etc., as illustrated in Figure 3. Additionally, spatial data can also be a composite of several single objects such as *maps* (formally called *spatial partitions*) or *spatial networks*. Some examples of such objects include the map of USA and the traffic network in Manhattan.

An analysis of the trends in data warehousing models and OLAP tools over the past decade reveals an interesting taxonomy in the development of such systems. Data warehouses were originally designed to handle simple numeric aggregations on business data. For example, a traditional OLAP query is to find

the three-year *moving average* in the sales of a product over a broader timeline in some geographic region. Later, enhancements were made to handle the problem of 'slowly changing dimensions' [17] and to deal with data warehouse evolution and versioning. However, these only involve partial solutions (and mostly academic implementations) such as time-stamping dimension tuples with their validity intervals and time-series fact tables to track instance changes, or creating temporal views over non-temporal data and defining 'temporal dimensions' to track schema changes. The changes that happen over a large temporal period in a data warehouse can involve either (i) changes in the multidimensional structure over time, (ii) changes in only the instance values over time, or, (iii) evolving changes in both structure and the instance values over time. The second category involves tracking the *history* of values of data dimensions and is popularly called *TOLAP* for *Temporal OLAP*. The third category includes changes where an instance value can belong to a dimension with two different structures over its valid time interval, or changes in the hierarchical structure along with the aggregation of values over time. Later designs of data warehouses used spatial databases in the backend to provide support for spatial data. Some models also proposed the integration of GIS and traditional OLAP systems for enhancing spatial data analysis. All these systems can be categorized as *Spatial OLAP* or *SOLAP* systems. The integration of TOLAP with spatial data support leads to another class of data warehouses called *Spatial-TOLAP* systems [37]. These data warehouses can handle *static* spatial data with temporal changes in the real-world facts, for example, a real-world sales event that happens in a geographic location described by a geometry that does not change with time. Geographic Information Systems (GIS) provide a specialized interface to visualize, manage and apply operations on the geometry of objects without considering changes over time. The interactive visualization and interpretation of geographically referenced information using GIS helps to discover relationships, patterns and trends among data, which can be useful both in the context of both business strategy improvement and for scientific analysis. However, a GIS is *static*, in that it does not consider evolving changes to object geometries over time as a continuous function. In contrast, data warehouses are 'dynamic' systems that natively handle changes over time in order to discover trends in real-world facts. They allow online (in real-time) data analysis by helping to handle and mine large collective datasets from numerous heterogeneous sources. Ideally, an advanced data warehousing system should be capable of supporting complex objects and thus incorporate features from both GIS (for spatial data analysis) and spatio-temporal databases (for storage and management of data). Additionally, they should also support complex OLAP operations combining both these systems to enhance analysis. Thus, some recent proposals for data warehouses include the integration of GIS and OLAP systems. However, most of these approaches are ad-hoc designs built for specific applications and only provide a subset of data warehouse functionality.

The idea behind *moving objects data warehouses* (*MODW*) is to provide a system capable of representing moving entities in data warehouses and be able to ask queries about them. Moving entities could be *moving points* such as people, animals, all kinds of vehicles such as cars, trucks, air planes, ships, etc., where usually only the time-dependent position in space is relevant but not the extent. However, moving entities with an extent, e.g., hurricanes, fires, oil spills, epidemic diseases, etc., could be characterized as *moving regions*. Such entities with a continuous, spatio-temporal variation (in position, extent or shape) are called *moving objects*. With a focus on cardinal direction relations, moving regions are more interesting because of the change in the relationship between their evolving extents over time. In this paper, we provide a novel approach to gather direction relations between such objects over time, and apply this as a data warehouse OLAP operation. A *moving objects data warehouse* is defined by a conceptual data cube that provide an abstract view containing moving objects in the data dimensions. These data dimensions

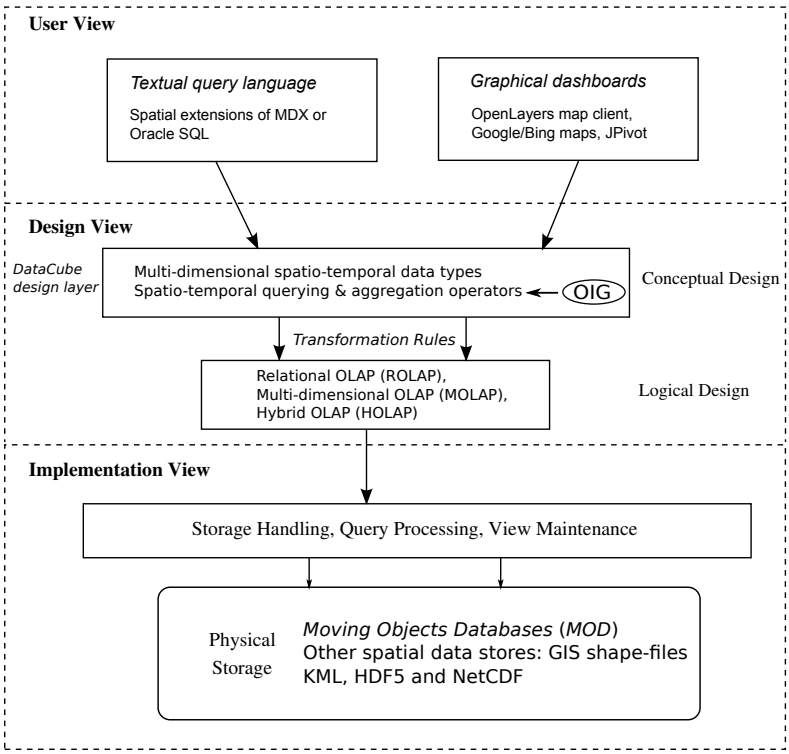


Figure 4. The design framework for a *Moving Objects Data Warehouse* (*MODW*) System

are called the *perspectives for analysis*, and the cube structure itself is defined as an injective functions from the multidimensional space composed of these perspectives to the various *subjects of analysis* in the cells of the cube. The cell instances of the cube contain measure values that quantify real-world facts. The measures and members include instances of moving object data types [12]. The conceptual data cube includes a set of multidimensional *abstract data types* (*ADTs*) to support both simple and complex objects in side the data warehouse. The basic set of data types include alphanumeric types (such as *int*, *real*, *char* and *string*), temporal types (such as *interval*, *instant* and *range*) and boolean types. The complex types include geo-spatial types (such as point, line, region, maps) and moving object types (such as moving point, moving line and moving region). Section 7 provides a more detailed view of the complex types defined in the data cube. Our goal is to make the model extensible enough to support new *user defined types* and semantics of aggregate operations on them. These complex objects can reside as measures of analysis or as the members of the data dimensions in the multidimensional datacube. Additionally, such a MODW framework should also provide support for multiple (composite) and complex members or measures. For example, a cell in a *sales* data cube can conceptually include several measures such as *sales quantity*, *inventory* and/or *sales profit*. Location can be a complex object such as a *polygon* representing Italy with the Vatican as a *hole* inside it. An example of a *complex region* object is illustrated in Figure 3, with its several faces and segment cycles (representing the object boundary). Thus, the MODW spatial data types should be capable of modeling data warehouse members and measures to include the simple, complex and composite spatial objects that are encountered in real-world scenarios. The $\mathcal{BigCube}$ [39] is an example of a conceptual, user-centric data warehouse model that is extended here for moving objects data warehouse design. The overall design framework for our moving objects

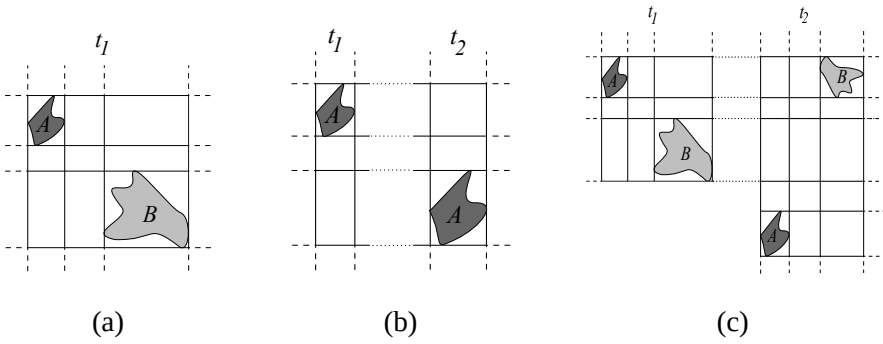


Figure 5. Possible configurations between two objects: spatial variation *(a)*, spatio-temporal variation in one object *(b)*, spatio-temporal variation in both objects *(c)*.

data warehouse system is shown in Figure 4, which is an extension of the three-tier approach often used in the design of scientific data management systems. The objects interaction graticule approach for cardinal directions is modeled within the conceptual data cube view in the data warehouse and is available to the user for spatio-temporal OLAP. In this paper, our OIG model lies at the conceptual level and the predicates provide the means to implement the model using any logical approach [38]. However, we also provide an application scenario for tropical storm event analysis to design the 'weather events' data cube and supporting MDX queries to help illustrate the versatility of the model in querying for direction relations between moving objects.

4. Overview of the Objects Interaction Graticule (OIG) Approach for Modeling Cardinal Directions

The goal of the objects interaction graticule model is to enable a data warehouse user to query for cardinal directions between moving objects. To achieve this goal, we need to take the various possible moving objects' configurations into account and model direction relations in all of the cases to arrive at the overall direction relation. This is because the direction relation for two moving objects between two queried time instances can be determined only by considering all the direction relations between them during their *lifetimes*. The possible configurations between moving objects that we need to consider include the following. First, two objects could be at different spatial locations at the same instant of time (dual object, spatial variation) as shown in Figure 5a. Second, an object could be at two different spatial locations at two different instances of time (single object, spatio-temporal variation) as shown in Figure 5b. Third, two objects could be at two different spatial locations at two different time instances (dual object, spatio-temporal variation) as shown in Figure 5c. The dotted lines between the configuration of objects across time represent the *flux* in the intersection of the coordinate systems used over the spatio-temporal variation of the moving objects. We include this in our model to be able to capture the locations of objects across the valid time extents. However, the dashed lines indicate the part not bounded by the objects interaction graticule (OIG). The OIG is a closed, bounded region and the dotted lines do not signify any *holes* in the spatio-temporal variation of the moving objects.

Figure 7 shows the two-phase strategy of our model for calculating the cardinal direction relations between two moving objects A and B at time instances t_1 and t_2. We assume that A and B (in the general

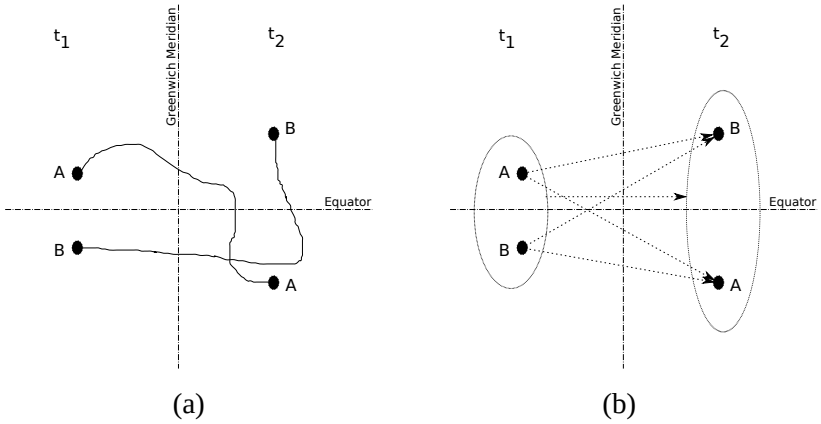


Figure 6. The actual movement of two moving objects over *valid* time instants t_1 and t_2 (a), and tracking the change in their cardinal directions over time (b)

case) are non-empty values of the complex spatio-temporal data type *mpoint*, *mline*, or *mregion* [12]. Here, we describe the OIG model for moving regions and later shown how it can be extended to support even other types such as moving points using an example application for weather-events analysis in Section 7.

To compute the cardinal directions between complex moving objects, we track the *change in the movement* of each object and of the entire system (both objects at each valid instant) with each other over time. This is done by evaluating the direction relations at valid snapshots. Before we delve deeper into the OIG concept, let us revisit the definition of a moving object type and understand the semantics of *cardinal directions between moving objects* and what this phrase exactly means. A moving object is defined by a set of partial functions, that represent the spatio-temporal variation of the object over a finite sequence of disjoint time intervals. These ordered time intervals are called the *periods of validity* of the moving object, and these together represent the *lifetime* of the moving object. The *lifetime* of a moving object is defined as $L = \{i_1, i_2, ..., i_n\}$, where each $i = (t_p, t_q), t_p < t_q$ represents a valid interval, i.e., a period or range of time when the object is defined. Given two time intervals $i_m, i_n \in L$, where $i_m = (t_p, t_q)$ and $i_n = (t_r, t_s)$, with $t_p < t_q$ and $t_r < t_s$, any time interval i_u is defined as an *invalid time interval* for the moving object *iff* $\exists t_u \in i_u$, s.t. $t_q < t_u < t_r$, and t_u is called an *invalid time instant*. Conceptually, this means that the moving object is undefined, invalid, or unknown during the instant t_u. As a consequence the cardinal direction for the moving object cannot be gathered at any such instant.

To help understand the semantics of cardinal directions between moving objects across valid time instants, consider, without loss of generality, the two *moving point* objects shown in Figure 6a, across two time instants t_1 and t_2. Object A has moved in a south-easterly direction between $t1$ and $t2$. Object B has moved in a north-easterly direction between t_1 and t_2. However, there could exist an invalid time interval i_u during the lifetimes of A and B, and lying between $t1$ and $t2$. At such an interval, neither object is defined. This means that, for example, if they are hurricanes, A could have been active in t_1, dissipated during $t_u \in i_u$, and re-emerged in t_2 with a south-east cardinal direction. Hence, it is incorrect to consider the trajectory of A as a straight *line* connecting A^{t_1} (read: A at time instant t_1) and A^{t_2} (read: A at time instant t_2). A similar situation arises if a is a complex point, such as a multi-point object that has several components forming and dissipating between two valid time intervals. To accommodate such a "null" state of a moving object in the modeling of cardinal directions, we consider the *moving direction*

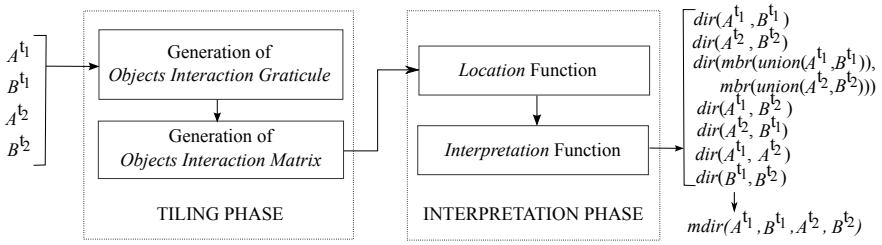


Figure 7. Overview of the two phases of the Objects Interaction Graticule (OIG) model

itself as a discrete function that gives the direction relations at any valid instant during the lifetimes of each of the operands.

For computing the direction relation between two moving objects' snapshots, we need to consider *all possible direction relations* between the *various combinations of objects* in the interacting system. First, we consider the scenario at each snapshot t_1 and t_2, and also the case when $t_1 = t_2$. For these, we default to the OIM approach for directions between objects without temporal variation and gather the direction relations between them. This is given by $dir(A^{t_1}, B^{t_1})$ and $dir(A^{t_2}, B^{t_2})$. The second case arises if $t_1 \neq t_2$. Then five more direction relations can be computed as shown in Figure 7. This includes four combinations for the two objects at t_1 and t_2, given by $dir(A^{t_1}, B^{t_2})$, $dir(A^{t_2}, B^{t_1})$, $dir(A^{t_1}, A^{t_2})$ and $dir(B^{t_1}, B^{t_2})$. In addition, we also relate the entire system (both objects) at each of the different time instances used to determine the query result. This is given by $dir(mbr(union(A^{t_1}, B^{t_1})), mbr(union(A^{t_2}, B^{t_2})))$. Using all these direction relations, we can now define the semantics of moving direction relations between the two objects over time. The semantics of the moving cardinal direction between two moving objects is thus defined as the combination of the direction relation of each object over time, the direction relation between each object over time, and the direction relation of both A and B *as a single system* over time, as illustrated in Figure 6b. Note that the change in the cardinal directions is shown by dotted lines because we do not consider it to evolve continuously with time, rather depending totally on the nature of the moving object and the partial functions that define with spatio-temporal variations.

Notice that, for clarity, we have used the notation A^t instead of $A(t)$ to refer to the temporal development of the moving region A (A is actually defined by a continuous function $A : time \rightarrow region$). We will use this notation through the rest of the paper. The *tiling phase* in Section 5 details our novel tiling strategy that produces the *objects interaction graticule* and shows how they are represented by *objects interaction matrices*. The *interpretation phase* in Section 6 leverages the objects interaction matrix to derive the directional relationship between two moving region objects.

4.1. Case Study

In order to illustrate the application of a moving-objects data warehouse, we use a real-world example from *tropical weather events* research. The US National Hurricane Center (NHC) [23], NOAA Hurricane Research Division (NOAA) [24], and Joint Typhoon Warning Center (JTWC) [15] collect data about hurricane events in the North Atlantic and Pacific Ocean using a combination of satellite, weather balloons and flight telemetry sytems. These data sets contain historical hurricane trajectory information (from 1997-2010) along with about 150 other relevant attributes such as wind speed, pressure, hurricane-stage (category), etc. Using this information and the shapefiles for US State boundaries, we create a moving objects data warehouse with data cubes describing the hurricane trajectory over a timeline and

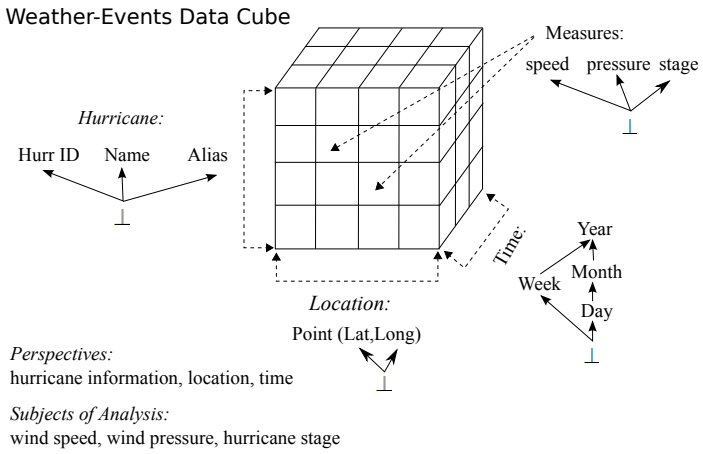


Figure 8. Illustration of the *structure* of a *Weather-Events* data cube showing three *perspectives*: Hurricane Data, Location (spatial *point*) and Time that define three *subjects of analysis*: wind-speed, wind-pressure and hurricane-stage.

other associated attributes (an example is shown in Figure 8). This framework can now be used to execute spatial analysis queries such as, "*find the hurricane that crossed the state of Louisiana in 2005 with maximum monthly wind speed averages*", "*determine all wind speeds for a 5km radius for hurricanes classified as category-3 or higher from 2003-2010*", and "*determine a heat map for all US States based on the number of hurricanes that affected each of them from 1990-2010*".

We employ a generic data warehouse meta-modelling approach to design the data cube and use it in the rest of the article for illustrating directional queries. In this approach, *hierarchies* are defined as *first class citizens* of the multidimensional structure. Hierarchies of *data categories* exist in both the *perspectives* (often called data dimensions) and the *subjects of analysis* or *metrics* (often called facts) of the cube. Measure values are instances of subjects of analysis and members are instances of the cube's perspectives of visualization. The measures and members of the data cube can be both alphanumeric values, spatial objects or a combination of these. For example, we store the location of the eye of the hurricane as a spatial point object. The spatial *point* is defined by latitude and longitude on geographic WGS94 coordinate system. The execution of the first query (above) included the following steps. First, we select all hurricanes that had a topological *is-cross* relation with Louisiana with a slice on year (2005). For these hurricanes, we gathered the winds speeds at the location of the eye of the hurricane and then computed the hurricane-specific wind-speed averages for each month in 2005. Finally, we select the name of the hurricane with the maximum wind speed average (*Katrina*). The ability to perform thematic selections, spatial topological relations, aggregations on measures (such as the average on wind speed values) over data integrated from heterogeneous sources over the historical time period allows for the execution of such a query. This illustrates the versatility and usefulness of a spatial data warehouse for performing OLAP operations on large-scale datasets. The generic meta-model for such a spatial data warehouse allows the system to completely capture and store the multidimensional structure while dormant, and easily recreate, pivot and query the relevant perspectives and analysis-subjects of the data cube while queries are being processed.

5. The Tiling Phase: Representing Interactions of Objects with the Objects Interaction Graticule and Matrix

In this section, we describe the *tiling phase* of the model. The general idea of our tiling strategy is to superimpose a graticule called *objects interaction graticule* (*OIG*) on a configuration of two moving spatial objects (regions). Such a graticule is determined by four vertical and four horizontal *partitioning lines* of *each* object at the available time instances. The four vertical (four horizontal) partitioning lines of an object are given as infinite extensions of the two vertical (two horizontal) segments of the object's minimum bounding rectangle at each of the two time instances. The partitioning lines of both objects create a partition of the Euclidean plane consisting of multiple mutually exclusive, directional *tiles* or *zones*. Further, these lines partition an object into non-overlapping components where each component is located in a different tile.

In the most general case, all partitioning lines are different from each other, and we obtain an overlay partition with central, bounded tiles and peripheral, unbounded tiles (indicated by the dashed segments in Figure 9a). The final OIG matrices obtained from this tiling is presented in Figure 9b and the process if defined formally below. The unbounded tiles do not contain any objects and therefore, we exclude them and obtain a graticule space that is a bounded proper subset of $\mathbb{R}^2$, as Definition 5.1 states. This is in contrast to the partitions of all other tiling-based models that are unbounded and equal to $\mathbb{R}^2$.

Definition 5.1. Given $A, B \in mregion$, and a function $at\_instant(m, t_i) \rightarrow m^{t_i}, m \in \{A, B\}$, let $R = (A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2})$, $A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2} \in region$ with $A^{t_1} \neq \varnothing, B^{t_1} \neq \varnothing, A^{t_2} \neq \varnothing,$ and $B^{t_2} \neq \varnothing$, and let $min_x^r = \min\{x \mid (x,y) \in r\}$, $max_x^r = \max\{x \mid (x,y) \in r\}$, $min_y^r = \min\{y \mid (x,y) \in r\}$, and $max_y^r = \max\{y \mid (x,y) \in r\}$ for $r \in \{A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}\}$. The *objects interaction graticule space* (*OIGS*) of $A^{t_1}, B^{t_1}, A^{t_2}$ and B^{t_2} is given as:

$$\begin{aligned} OIGS(R) = \{(x,y) \in \mathbb{R}^2 \mid & \min(min_x^{A^{t_1}}, min_x^{B^{t_1}}, min_x^{A^{t_2}}, min_x^{B^{t_2}}) \leq x \leq \\ & \max(max_x^{A^{t_1}}, max_x^{B^{t_1}}, max_x^{A^{t_2}}, max_x^{B^{t_2}}) \wedge \min(min_y^{A^{t_1}}, min_y^{B^{t_1}}, \\ & min_y^{A^{t_2}}, min_y^{B^{t_2}}) \leq y \leq \max(max_y^{A^{t_1}}, max_y^{B^{t_1}}, max_y^{A^{t_2}}, max_y^{B^{t_2}})\} \end{aligned}$$

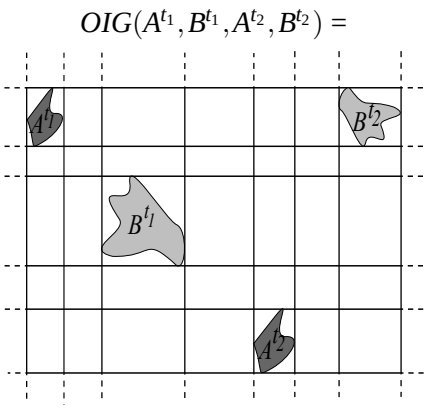
Definition 5.2 determines the bounded graticule formed as a part of the partitioning lines and superimposed on $OIGS(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2})$.

Definition 5.2. Let *seg* be a function that constructs a segment between any two given points $p, q \in \mathbb{R}^2$, i.e., $seg(p,q) = \{t \mid t = (1-\lambda)p + \lambda q, 0 \leq \lambda \leq 1\}$. Let $H_r = \{seg((min_x^r, min_y^r), (max_x^r, min_y^r)), seg((min_x^r, max_y^r), (max_x^r, max_y^r))\}$ and $V_r = \{seg((min_x^r, min_y^r), (min_x^r, max_y^r)), seg((max_x^r, min_y^r), (max_x^r, max_y^r))\}$ for $r \in \{A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}\}$. We call the elements of $H_{A^{t_1}}$, $H_{B^{t_1}}$, $H_{A^{t_2}}$, $H_{B^{t_2}}$, $V_{A^{t_1}}$, $V_{B^{t_1}}$, $V_{A^{t_2}}$ and $V_{B^{t_2}}$ *objects interaction graticule segments*. Then, the *objects interaction graticule* (*OIG*) for *A* and *B* is given as:

$$OIG(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}) = H_{A^{t_1}} \cup V_{A^{t_1}} \cup H_{B^{t_1}} \cup V_{B^{t_1}} \cup H_{A^{t_2}} \cup V_{A^{t_2}} \cup H_{B^{t_2}} \cup V_{B^{t_2}}.$$

In the OIG of an object, there are two constituent *object interaction coordinate systems* (*OICS*) for each temporal state of the moving objects. These are defined as follows:

$$\begin{aligned} &OICoordS(A^{t_1}, B^{t_1}) = H_{A^{t_1}} \cup V_{A^{t_1}} \cup H_{B^{t_1}} \cup V_{B^{t_1}}, \text{ and} \\ &OICoordS(A^{t_2}, B^{t_2}) = H_{A^{t_2}} \cup V_{A^{t_2}} \cup H_{B^{t_2}} \cup V_{B^{t_2}}. \end{aligned}$$



(a)

$$OIM(A^{t_1}, B^{t_2}) = \begin{pmatrix} 1 & 0 & 2 \end{pmatrix}$$

$$OIM(A^{t_2}, B^{t_1}) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$OIM(mbr(union(A^{t_1}, B^{t_1})), mbr(union(A^{t_2}, B^{t_2}))) = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 0 & 2 \end{pmatrix}$$

$$OIM(A^{t_1}, A^{t_2}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$OIM(B^{t_1}, B^{t_2}) = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix}$$

(b)

Figure 9. The objects interaction graticule $OIG(A,B)$ for the two region objects A and B in Figures 1c and 1d (a), and the derived objects interaction matrices (*OIM*) for OIG components described in Definition 5.3 (b).

The definition of OIG comprises the description of all graticules that can arise. In the most general case, if $t_1 = t_2$ and $H_{A^{t_1}} \cap H_{B^{t_1}} = \varnothing$ and $V_{A^{t_1}} \cap V_{B^{t_1}} = \varnothing$, we obtain a bounded 3×3-graticule similar to that for a non-temporal variation in the objects configurations. Special cases arise if $H_{A^{t_1}} \cap H_{B^{t_1}} \neq \varnothing$ and/or $V_{A^{t_1}} \cap V_{B^{t_1}} \neq \varnothing$. Then equal graticule segments coincide in the union of all graticule segments. As a result, depending on the relative position of two objects to each other, the objects interaction graticule can be of different sizes. However, due to the non-empty property of a region object, not all graticule segments can coincide. This means that at least two horizontal graticule segments and at least two vertical graticule segments must be maintained. Definition 5.3 gives a formal characterization for the OIG.

Definition 5.3. An objects interaction graticule $OIG(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2})$ consists of two objects interaction coordinate systems, at t_1 and t_2, each of size $m \times n$, with $m,n \in \{1,2,3\}$, if $|H_A \cap H_B| = 3 - m$ and $|V_A \cap V_B| = 3 - n$. Further, it also consists of four objects interaction grids for each of the spatio-temporal combinations of the two moving objects and a fifth for the overall system. Together, these are called the *objects interaction graticule components*.

The objects interaction graticule partitions the objects interaction graticule space into *objects interaction graticule tiles* (*zones*, *cells*). Definition 5.4 provides their definition for each of the time instances uniquely, using the objects interaction coordinate systems.

Definition 5.4. Given $A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2} \in region$ with $A^{t_1} \neq \varnothing, B^{t_1} \neq \varnothing, A^{t_2} \neq \varnothing,$ and $B^{t_2} \neq \varnothing$, $OIGS(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2})$, and $OIG(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2})$, we define $c_H = |H_A \cup H_B| = |H_A| + |H_B| - |H_A \cap H_B|$ and c_V correspondingly at a time instant. Let $H_{AB} = H_A \cup H_B = \{h_1, \ldots, h_{c_H}\}$ such that (i) $\forall 1 \leq i \leq c_H : h_i = seg((x_i^1, y_i), (x_i^2, y_i))$ with $x_i^1 < x_i^2$, and (ii) $\forall 1 \leq i < j \leq c_H : h_i < h_j$ (we say that $h_i < h_j :\Leftrightarrow y_j < y_i$). Further, let $V_{AB} = V_A \cup V_B = \{v_1, \ldots, v_{c_V}\}$ such that (i) $\forall 1 \leq i \leq c_V : v_i = seg((x_i, y_i^1), (x_i, y_i^2))$ with $y_i^1 < y_i^2$, and (ii) $\forall 1 \leq i < j \leq c_V : v_i < v_j$ (we say that $v_i < v_j :\Leftrightarrow x_i < x_j$).

Next, we define four auxiliary predicates that check the position of a point (x,y) with respect to a graticule segment:

$$\begin{aligned}
\mathit{below}((x,y),h_i) &\Leftrightarrow x_i^1 \le x \le x_i^2 \wedge y \le y_i \\
\mathit{above}((x,y),h_i) &\Leftrightarrow x_i^1 \le x \le x_i^2 \wedge y \ge y_i \\
\mathit{right_of}((x,y),v_i) &\Leftrightarrow y_i^1 \le y \le y_i^2 \wedge x \ge x_i \\
\mathit{left_of}((x,y),v_i) &\Leftrightarrow y_i^1 \le y \le y_i^2 \wedge x \le x_i
\end{aligned}$$

An *objects interaction graticule tile* $t_{i,j}$ with $1 \le i < c_H$ and $1 \le j < c_V$ is then defined for a particular time instant as

$$\begin{aligned}
t_{i,j} = \{(x,y) \in \mathit{OIGS}(A,B) \,|\, &\mathit{below}((x,y),h_i) \wedge \mathit{above}((x,y),h_{i+1}) \wedge \\
&\mathit{right_of}((x,y),v_j) \wedge \mathit{left_of}((x,y),v_{j+1})\}
\end{aligned}$$

The definition indicates that all tiles are bounded and that two adjacent tiles share their common boundary. Let $\mathit{OIGT}(A,B)$ be the set of all tiles $t_{i,j}$ imposed by $\mathit{OIG}(A,B)$ on $\mathit{OIGS}(A,B)$. An $m \times n$-graticule contains $m \cdot n$ bounded tiles.

By applying our tiling strategy, an objects interaction graticule can be generated for any two region objects A and B. It provides us with the valuable information which region object intersects which tile across the temporal variations. With each time event t_1 and t_2, Definition 5.5 provides us with a definition of the *interaction* of A and B with a tile.

Definition 5.5. Given $A,B \in \mathit{region}$ with $A \neq \varnothing$ and $B \neq \varnothing$ and $\mathit{OIGT}(A,B)$, let ι be a function that encodes the *interaction* of A and B with a tile $t_{i,j}$, and checks whether no region, A only, B only, or both regions intersect a tile. We define this function as

$$\iota(A,B,t_{i,j}) = \begin{cases} 0 \text{ if } & A^\circ \cap t_{i,j}^\circ = \varnothing \wedge B^\circ \cap t_{i,j}^\circ = \varnothing \\ 1 \text{ if } & A^\circ \cap t_{i,j}^\circ \neq \varnothing \wedge B^\circ \cap t_{i,j}^\circ = \varnothing \\ 2 \text{ if } & A^\circ \cap t_{i,j}^\circ = \varnothing \wedge B^\circ \cap t_{i,j}^\circ \neq \varnothing \\ 3 \text{ if } & A^\circ \cap t_{i,j}^\circ \neq \varnothing \wedge B^\circ \cap t_{i,j}^\circ \neq \varnothing \end{cases}$$

We use the *mbr* and *union* functions for computing the minimum bounding rectangle and the spatial union of two objects, respectively. To support both objects interaction coordinate systems we extend ι to accept $\mathit{mbr}(\mathit{union}(A^{t_1},B^{t_1}))$ and $\mathit{mbr}(\mathit{union}(A^{t_2},B^{t_2}))$ as operands. The operator $^\circ$ denotes the point-set topological *interior* operator and yields a region without its boundary. For each graticule cell $t_{i,j}$ in the ith row and jth column of an $m \times n$-graticule with $1 \le i \le m$ and $1 \le j \le n$, we store the coded information in an *objects interaction matrix* (*OIM*) in cell $\mathit{OIM}(A,B)_{i,j}$, as illustrated in Figure 9b.

$$\mathit{OIM}(A,B) = \begin{pmatrix} \iota(A,B,t_{1,1}) & \iota(A,B,t_{1,2}) & \iota(A,B,t_{1,3}) \\ \iota(A,B,t_{2,1}) & \iota(A,B,t_{2,2}) & \iota(A,B,t_{2,3}) \\ \iota(A,B,t_{3,1}) & \iota(A,B,t_{3,2}) & \iota(A,B,t_{3,3}) \end{pmatrix}$$

6. The Interpretation Phase: Assigning Semantics to the Objects Interaction Matrix

The second phase of the objects interaction graticule model is the *interpretation phase*. This phase takes an objects interaction matrix (OIM) obtained as the result of the tiling phase as input and uses it to generate a set of cardinal directions as output. This is achieved by separately identifying the locations of both objects in the objects interaction matrix and by pairwise interpreting these locations in terms of cardinal directions. The union of all these cardinal directions is the result. This phase is similar to the interpretation phase of the OIM model [5].

We use an interpretation function to determine the basic cardinal direction between any two object components on the basis of their (i, j)-locations in the objects interaction matrix. The composite cardinal relation between A and B is then the union of all determined basic relations.

In a first step, we define a function *loc* (see Definition 6.1) that acts on one of the region objects A or B and their OIM and determines all locations of components of each object in the matrix for both temporal extents individually. Let $I_{m,n} = \{(i, j) \,|\, 1 \leq i \leq m, 1 \leq j \leq n\}$. We use an index pair $(i, j) \in I_{m,n}$ to represent the location of the element $M_{i,j} \in \{0, 1, 2, 3\}$ and thus the location of an object component from A or B in an $m \times n$ objects interaction matrix.

Definition 6.1. Let M be the $m \times n$-objects interaction matrix of two region objects A and B. Then the function *loc* is defined as:

$$loc(A, M) = \{(i, j) \,|\, 1 \leq i \leq m, 1 \leq j \leq n, M_{i,j} = 1 \vee M_{i,j} = 3\}$$
$$loc(B, M) = \{(i, j) \,|\, 1 \leq i \leq m, 1 \leq j \leq n, M_{i,j} = 2 \vee M_{i,j} = 3\}$$

In a second step, we define an *interpretation function* ψ to determine the cardinal direction between any two object components of A and B on the basis of their locations in the objects interaction matrix. We use a popular model with the nine *basic cardinal directions*: *north* (N), *northwest* (NW), *west* (W), *southwest* (SW), *south* (S), *southeast* (SE), *east* (E), *northeast* (NE), and *origin* (O) to symbolize the possible cardinal directions between *object components*. Definition 6.2 provides the interpretation function ψ with the signature $\psi : I_{m,n} \times I_{m,n} \to CD$.

Definition 6.2. Given $(i, j), (i', j') \in I_{m,n}$, the *interpretation function* ψ on the basis of the set $CD = \{N, NW, W, SW, S, SE, E, NE, O\}$ of *basic cardinal directions* is defined as

$$\psi((i, j), (i', j')) = \begin{cases} N & \text{if } i < i' \wedge j = j' \\ NW & \text{if } i < i' \wedge j < j' \\ W & \text{if } i = i' \wedge j < j' \\ SW & \text{if } i > i' \wedge j < j' \\ S & \text{if } i > i' \wedge j = j' \\ SE & \text{if } i > i' \wedge j > j' \\ E & \text{if } i = i' \wedge j > j' \\ NE & \text{if } i < i' \wedge j > j' \\ O & \text{if } i = i' \wedge j = j' \end{cases}$$

The main difference compared to the OIM approach however is in the following third and final step. We temporally *lift* the cardinal direction relation function *dir* to include objects over their temporal extents. Here, we specify the *cardinal direction function* named *mdir* (moving-direction) which determines the *composite moving cardinal direction* for two moving region objects A and B. This function has the signature $mdir : region_{t_1} \times region_{t_2} \rightarrow 2^{CD}$ and yields a set of basic cardinal directions as its result. In order to compute the function *dir*, we first generalize the signature of our interpretation function ψ to $\psi : 2^{I_{m,n}} \times 2^{I_{m,n}} \rightarrow 2^{CD}$ such that for any two sets $X, Y \subseteq I_{m,n}$ holds: $\psi(X,Y) = \{\psi((i,j),(i',j')) \,|\, (i,j) \in X, (i',j') \in Y\}$. We are now able to specify the cardinal direction function *mdir* in Definition 6.3.

Definition 6.3. Let $A, B \in mregion$, and $A^t, B^t \in region$ at a valid instant t, during the *lifetime* of A and B, we define $dir(A^t, B^t) = \psi(loc(A^t, OIM(A^t, B^t)), loc(B^t, OIM(A^t, B^t)))$. Then the *cardinal direction function mdir* is defined as

$$\begin{aligned} mdir(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}) = \; & dir(A^{t_1}, B^{t_1}) \cup dir(A^{t_2}, B^{t_2}) \cup dir((A^{t_1}, B^{t_1}), (A^{t_2}, B^{t_2})) \cup \\ & dir(A^{t_1}, B^{t_2}) \cup dir(A^{t_2}, B^{t_1}) \cup dir(A^{t_1}, A^{t_2}) \cup dir(B^{t_1}, B^{t_2}) \end{aligned}$$

We apply this definition to our example in Figure 9. With $loc(A^{t_1}, OIM(A^{t_1}, B^{t_1})) = \{(1,1)\}$ and $loc(B^{t_1}, OIM(A^{t_1}, B^{t_1})) = \{(3,3)\}$, and so on, we obtain

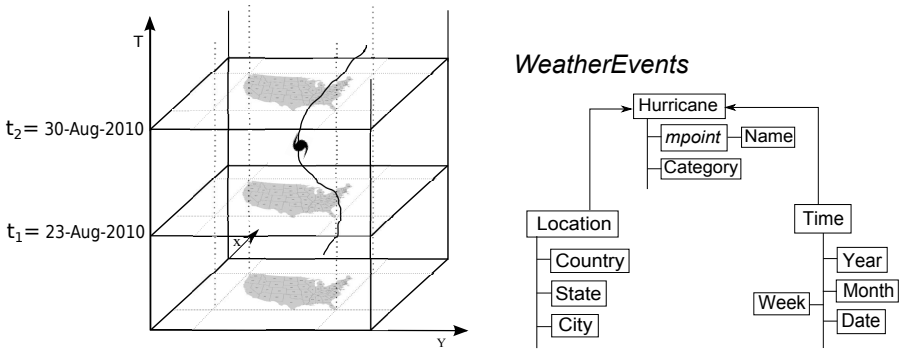
$$\begin{aligned} mdir(A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}) &= \{\psi((1,1),(3,3)), \psi((3,1),(1,3)), \psi(\{(1,1)\}, \\ &\quad \{(1,3),(2,3)\}), \psi((1,1),(1,3)), \psi((3,1),(1,2)), \\ &\quad \psi((1,1),(3,1)), \psi((3,1),(1,3))\} \\ &= \{NW, SW, W, SE\} \end{aligned}$$

Finally we can say regarding Figure 9 that "Object A is *partly northwest, partly southwest, partly west,* and *partly southeast* of object B over the period from time t_1 to t_2". Each of the individual directions between the moving objects for the three possible configurations described in Section 4 can also be provided by using the results from each application of *dir*, that is used to finally arrive at the moving direction relations (given by *mdir*). Note that, here, we define the cardinal directions for the two moving objects at two snapshot instances t_1 and t_2. This strategy is later extended to the entire lifetimes of both objects and the cardinal directions are evaluated using each *valid snapshot* to track the objects' movement over time. This last step is explained further in the next section.

7. Application of the Objects Interaction Graticule Model (OIG) for Spatio-Temporal OLAP

In this section, we introduce the data warehouse design for the management of weather event data collected from "best-track" data sources in the US. These sources include the National Hurricane Center (NHC), National Oceanic and Atmospheric Administration (NOAA) and the Joint Typhoon Warning Center (JTWC), and the data is collected for the period from 1960 till 2010.

The 'weather events' data warehouse integrates and manages data about tropical storm events from various sources, and uses the $\mathcal{BigCube}$ approach [39] for data modeling. This conceptual cube has the following data dimensions: hurricane, location and time. Each of these data dimensions is called



(a) (b)

Figure 10. Determining cardinal directions at valid time instants t_1 and t_2 for a moving point (Hurricane Katrina) and composite region object (USA) (a), and logical design of the WeatherEvents OLAP cube using a relational snowflake schema (b).

a *perspective for analysis* in the weather events data cube. The hurricane perspective stores *member values* which are *moving points* signifying the location of a hurricane object. Such a moving point is created by a constructor $mpoint : instant \rightarrow point$ that is a continuous representation of the location of the moving eye of the hurricane over its duration of existence. We define a function called *lifetime*: $mpoint \rightarrow period$ that gives the entire duration of validity of this moving point object. Another function *at_instant*: $instant \rightarrow location$ gives the location of the moving point at that particular instant of time. The *label* for each moving point refers to the name of the hurricane it represents. The hurricane data has an additional descriptive attribute called *Category*, that displays the highest intensity of the hurricane during a particular time period as defined in the source dataset. We use the Saffir-Simpson scale to represent the hurricane intensity for the Atlantic and Pacific regions. The *location* perspective stores the spatial partitions or maps for the regions (States) in USA. These are spatial region objects and have corresponding labels for each of the State names. The *time* perspective stores a hierarchy of member values for the day, week, month and year time values. There are two hierarchies for time, $day \rightarrow month \rightarrow year$, and $day \rightarrow week \rightarrow year$. Both paths in the hierarchy are separate and these display a partial ordering in the lattice defining the time perspective. Now, the cardinal directions between the moving hurricane (more clearly, the location of the eye of the hurricane system defined as a continuous function over time) and its 'crossing' over the US (represented as composite spatial regions) can be determined using the OIG approach. For this, we evaluate the cardinal directions for the two objects at each valid time interval t_i and determine *mdir*. For n ordered time instants $\{t_1, t_2, ..., t_{i-1}, t_i, ..., t_n\}$ defined over the lifetime of the two objects A and B, we can thus determine *every* $mdir(A^{t_{i-1}}, B^{t_{i-1}}, A^{t_i}, B^{t_i})$, which gives the direction relations at the valid instants, as illustrated in Figure 10a. Note, however, that this does not provide the 'state' of the cardinal direction *between* those time instants. We take this approach to define mdir as a discrete function over time, because the geometric properties of the operand objects solely depends on their 'state' and how the objects are defined. If the operands are moving objects, then *at_instant* gives spatial information only at every valid instant. However, between two valid instants t_1 and t_2, the geometric properties of an object can be invalid, unspecified or unknown. We take this state of the object as *null*. The null state specification allows the OIG model to be generic and applicable to combinations of both spatial and spatio-temporal objects, and also meets the requirements for hierarchical lattices defined in the perspectives of the data warehouse [39]. Moreover, since we handle directions using the basic spatial

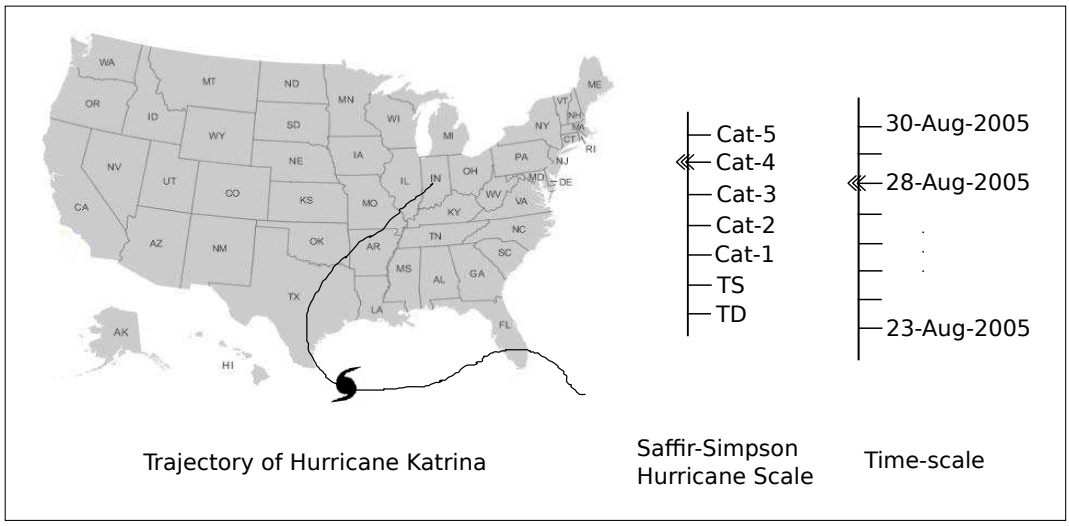


Figure 11. Illustration of the analysis user-view for the moving objects data warehouse system. The example shows the *state of weather events* corresponding to a selection on Hurricane Katrina, in the US, on 28-Aug-2010.

types (e.g., region) and include the union of both operands (i.e., the system of interacting objects) at each time instant, the model can be trivially extended to handle complex objects such as multi-component regions, regions with holes or point clouds. For example, each component ('face') of the mregion is treated as a unique simple mregion object and the OIG is applied.

The translation from conceptual cube to relational tables is done as shown in the Figure 10b. The hurricane system is the real-world fact that we use as the subject of analysis. The perspectives for analysis are time and location with their constituent hierarchies. Multiple hierarchies are supported on the time dimension by using the *snowflake schema* [14] for relational data warehouse design. The primary keys from the time and location *perspective tables* are used as foreign keys inside the hurricane *subject table*. Given, such a data cube and the logical design, we can now pose several queries that involve "navigation" along the hierarchical structure of the perspectives and the subjects of analysis. Conventionally, this can be roll-up or drill-down aggregations on the measures of the data cube. Aggregate operations such as cardinal direction querying is also incorporated as part of the OLAP interface. As shown in Figure 4, the user-view for such MODW queries can involve wither a graphical dashboard presenting a map of the region along with the hurricane trajectories and intensities with a time-line, or a query language extension supporting spatio-temporal predicates. Consider a query to *find the trajectory of Hurricane Katrina and the various US States affected by it in 2005*. Figure 11 shows a graphical dashboard that provides the user view for the spatio-temporal variation in the trajectory of Hurricane Katrina along with hurricane category classification (Saffir-Simpson scale) and the time-scale. Note that we always keep track of the 'state of the cube', i.e., the current level of aggregation in the multidimensional space. For e.g., in Figure 11, the current position of the eye of the hurricane is referenced on the map using the hurricane symbol for the particular time (28-Aug-2010) and intensity (Category-4). The *state* of the cube is a two-tuple $\langle S, i \rangle$ where S represents the cube structure (for example, $\{hurricane \times category \times time \times location\}$), and i is a valid cube instance (containing member and measure values) corresponding to that structure. An example of a query language extension for MDX for querying cardinal direction relations is presented in Section 8.

Some of the additional OLAP queries that can be incorporated into the weather events data warehouse using the objects interaction graticule approach are "*Find all hurricanes, ordered according to their intensity, which affected the western part of the US, during 2009*" that compares the trajectories of hurricanes that lie in the *interior* of the region described as *western* in the US, "*Find all States in the south-eastern US that were affected by a Category 3 or higher hurricane between 2005 and 2010*", that restricts the resulting set of US states affected a Category-3 or higher hurricane to the south-eastern region, and "*Find the hurricane with the maximum intensity that ever crossed the southern-most tip of Florida during the period from 1990 till 2010*" that performs a selection of the hurricane based on its trajectory overlapping with the southern-most point in Florida with the highest intensity.

8. Directional Predicates for OLAP Querying in Moving Objects Data Warehouses

Based on the OIG model and the interpretation mechanism described in the previous sections, we can identify the cardinal directions between any given two moving region objects. To integrate the cardinal directions into moving object data warehouses as selection and join conditions in spatial queries, binary *directional predicates* need to be formally defined. For example, a query like "Find all hurricanes that affect states which lie strictly to the north of Florida" requires a directional predicate like *strict_north* as a selection condition of a spatial join.

The *mdir* function, which produces the final moving cardinal directions between two complex region objects A and B across temporal variation, yields a subset of the set $CD = \{N, NW, W, SW, S, SE, E, NE, O\}$ of *basic cardinal directions*. As a result, a total number of $2^9 = 512$ cardinal directions can be identified. Therefore, at most 512 directional predicates can be defined to provide an *exclusive* and *complete* coverage of all possible directional relationships. We can assume that users will not be interested in such a large, overwhelming collection of detailed predicates since they will find it difficult to distinguish, remember and handle them. Instead we provide a mechanism for the user to define and maintain several levels of predicates for querying. As a first step, in Definition 8.1, we propose nine *existential directional predicates* that ensure the existence of a particular basic cardinal direction between parts of two region objects A and B.

Definition 8.1. Let $R = (A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}), R \in region$. Then the existential directional predicate for north is defined as:

$$exists\_north(R) \equiv (N \in mdir(R))$$

Eight further existential direction predicates for S, E, W, O, NE, SE, NW, and SW are also defined correspondingly. Later, by using $\neg$, $\vee$ and $\wedge$ operators, the user will be able to define any set of composite *derived directional predicates* from this set for their own applications. We provide three examples for these.

The first set of predicates is designed to handle *similarly oriented directional predicates* between two regions. *Similarly oriented* means that several cardinal directions facing the same general orientation belong to the same group. Definition 8.2 shows an example of *northern* by using the existential predicates.

Definition 8.2. Let $R = (A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}), R \in region$. Then *northern* is defined as:

$$northern(R) = exists\_north(R) \vee exists\_northwest(R) \vee exists\_northeast(R)$$

The other similarly oriented directional predicates *southern*, *eastern*, and *western* are defined in a similar way.

The second set of predicates is designed to handle *strict directional predicates* between two region objects. *Strict* means that two region objects are in exactly one basic cardinal direction to each other. Definition 8.3 shows an example of *strict_north* by using the existential predicates.

Definition 8.3. Let $R = (A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}), R \in region$. Then *strict_north* is defined as:

$$\begin{aligned} strict\_north(R) = {} & exists\_north(R) \wedge \neg exists\_south(R) \wedge \neg exists\_west(R) \wedge \\ & \neg exists\_east(R) \wedge \neg exists\_northwest(R) \wedge \quad \neg exists\_northeast(R) \wedge \\ & \neg exists\_southwest(R) \wedge \neg exists\_southeast(R) \wedge \neg exists\_origin(R) \end{aligned}$$

The other strict directional predicates *strict_south*, *strict_east*, *strict_west*, *strict_origin*, *strict_northeast*, *strict_northwest*, *strict_southeast*, *strict_southwest*, *strict_northern*, *strict_southern*, *strict_eastern*, and *strict_western* are defined in a similar way.

Now, we define a function *dirnum* that gives the cardinality of the result of *dir*. This function has the signature $dirnum : dir \rightarrow \mathbb{N}$, and gives the number of unique existential cardinal directions obtained as a result of applying the OIG on the two operand objects. For the directional results between moving objects, we also define *mdirnum*, as a function that gives the cardinality of the result of *mdir*. This function has the signature $mdirnum : dir \rightarrow \mathbb{N}$, and gathers the number of unique existential cardinal directions between two valid time instances for the two moving objects, obtained as a result of *mdir*.

The third set of direction predicates represents the commonly used *cover* relation between two objects. Cover means that one object surrounds another object, either partly or fully. We define two different types of coverage. *Existential coverage*, means that an object *partially surrounds* another object. For partial coverage atleast three different directions out of the eight basic cardinal directions (except origin) must exist in the result set. Thus, existential cover means that the result of mdirnum should be greater than 0.375. The *exists_cover* predicate gives the result of the existential cover relation. The second type is *complete coverage* which is defined as a strict-surround relation where one object fully surrounds the other. This means that at some valid instant of time, we get a value for *mdirnum* greater than or equal to 1. The value is greater than one when the *origin* relation is also valid. Thus complete coverage can include just surround, or both surround and origin relations. Thus these three levels of directional predicates cover the entire spectrum of directional relations between two interacting moving objects.

Definition 8.4. Let $R = (A^{t_1}, B^{t_1}, A^{t_2}, B^{t_2}), R \in region$. Then *complete_cover* is defined as:

$$\begin{aligned} complete\_cover(R) = {} & (exists\_north(R) \wedge exists\_northeast(R) \wedge exists\_east(R) \wedge \\ & exists\_southeast(R) \wedge exists\_south(R) \wedge exists\_southwest(R) \wedge \\ & exists\_west(R) \wedge exists\_northwest(R)) \vee exists\_origin(R) \end{aligned}$$

We can now employ these predicates in MDX queries over the example presented in Section 4.1 with the moving objects data warehouse framework. Consider the weather events data cube presented in Section 7. This *WeatherEvents* cube is analogous to an extended spreadsheet table with hurricane names (ordered in categories according to their intensity) over several years and containing specific geographic

information for describing the location of the eye of the tropical storm event at particular time instances. The predicate *IsCross* checks whether a given spatial object crosses the interior of another spatial object. Now, we can pose the following queries:

Q1: Find the states in south-eastern US that were crossed by Hurricane Katrina, along with the intensity of the hurricane at that time, during the year 2005.
The corresponding MDX query is as follows:

```
SELECT {[Geography].[USA].[State].MEMBERS} ON COLUMNS,
       {[Date].2005.[Month].[Day].MEMBERS} ON ROWS,
       NON EMPTY Filter({[Measures].[Hurricane].[Category].MEMBERS,
       IsCross([Measures].[Hurricane].Katrina, [Geography].[USA].[State].MEMBERS)}) ON PAGES,
FROM WeatherEvents
WHERE ( exists_southeast( [Geography].[Country].[State].MEMBERS ),
        [Geography].[Country].[USA] ))
```

The result of this query is shown below. Note that in the representation of hurricane intensity, TD stands for 'tropical depression' and TS for 'tropical storm', according to the Saffir-Simpson Hurricane Scale.

		State
Cat-5	28-Aug-2010	Louisiana
Cat-4	29-Aug-2010	Alabama
Cat-1	26-Aug-2010	Florida

Q2: Find the hurricanes ordered by their intensities, which had a path moving towards the east from their point of origin, and affected states strictly in the southern part of the US, during the period from 2005 to 2009.
The corresponding MDX query is as follows:

```
WITH
  MEMBER Date.[05 to 09] AS '[Date].[2005] : [Date].[2009]'
SELECT {NON EMPTY Filter( {[Measures].[Hurricanes].[Category].MEMBERS},
   exists_east( [Measures].[Hurricanes].CurrentMember,
  [Measures].[Hurricanes]))} ON AXIS(1),
  {Date.[05 to 09]} ON AXIS(2),
  {[Geography].[Country].[State]} ON AXIS(3),
FROM WeatherEvents
WHERE ( strict_southern([Geography].[Country].[State].MEMBERS,
        [Geography].[Country].[USA].[FL] ))
```

A sample result of this query is shown below.

		2005	2006	2007	2008	2009
Georgia	Cat-2	Kevin	Bronco		Tracy	Nobel
	Cat-3	Cindy	Alberto	Barry	Fay	Ida
	Cat-4	Katrina	Alberto			Ida
North Carolina	Cat-2	Cindy	Alberto		Hought	Jives
	Cat-3	Katrina	Ernesto	Sabley	Hanna	Vorice

9. Conclusions and Future Work

In this article, we introduce the concept of a *Moving Objects Data Warehouse* (*MODW*) for storing and querying multidimensional, spatial-temporal data. We introduce a three-tier design framework that provides a generalized view of the data warehouse system from conceptual to logical and physical design, and demonstrate the features of a user-view for the system based on both graphical dashboards and query languages. We also present a novel approach called the *Objects Interaction Graticule* (*OIG*) to determine the cardinal direction relations for moving objects. Using the MODW framework and the OIG model, we extend cardinal direction queries to include complex spatial objects and present a scenario to handle mixed combinations of complex objects. Further, we have shown how different kinds of directional predicates can be derived from the cardinal directions and how these predicates can be employed in MDX queries and used to perform OLAP querying using the moving objects data warehouse framework.

In the future, we plan to handle fuzzy and incomplete spatial data, and provide an efficient implementation of the moving objects data warehouse supporting cardinal direction queries. Further work involves the design of spatial reasoning techniques for direction relations by using the objects interaction graticule model.

References

[1] Abelló, A., Samos, J., Saltor, F.: YAM^2: A Multidimensional Conceptual Model Extending UML, *Information Systems*, **31**(6), 2006, 541–567.

[2] Allen, J. F.: Maintaining Knowledge about Temporal Intervals, *Journal of the Association for Computing Machinery*, **26**(11), 1983, 832–843.

[3] Bimonte, S., Tchounikine, A., Miquel, M.: Spatial OLAP: Open Issues and a Web Based Prototype, *10th AGILE Int. Conf. on Geographic Information Science*, May 2007, 1–11.

[4] Chen, T., Liu, H., Schneider, M.: Evaluation of cardinal direction developments between moving points, *Proceedings of the 18th ACM SIG-SPATIAL Int. Conf. on Advances in Geographic Information Systems*, 2010, 430–433.

[5] Chen, T., Schneider, M., Viswanathan, G., Yuan, W.: The Objects Interaction Matrix for Modeling Cardinal Directions in Spatial Databases, *Proceedings of the 15th Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, 2010, 218–232.

[6] Cuzzocrea, A.: Accuracy control in compressed multidimensional data cubes for quality of answer-based OLAP tools, *18th IEEE Int. Conf. on Scientific and Statistical Database Management*, 2006, 301–310.

[7] Cuzzocrea, A.: Top-down compression of data cubes in the presence of simultaneous multiple hierarchical range queries, *Proceedings of the 17th Int. Conf. on Foundations of Intelligent Systems*, Springer-Verlag, 2008, 361–374.

[8] Franconi, E., Kamblet, A.: A data warehouse conceptual data model, *Scientific and Statistical Database Management, 2004. Proceedings. 16th Int. Conf. on*, IEEE, 2004, 435–436.

[9] Golfarelli, M., Maio, D. and Rizzi, S.: The Dimensional Fact Model: A Conceptual Model for Data Warehouses, *Int. Journal of Cooperative Information Systems*, **7**(2), 1998, 215, ISSN 0218-8430.

[10] Goyal, R., Egenhofer, M.: Cardinal Directions between Extended Spatial Objects, 2000, Unpublished manuscript.

[11] Goyal, R. K., Egenhofer, M. J.: Consistent Queries over Cardinal Directions Across Different Levels of Detail, *11th Int. Conf. on Database and Expert Systems Applications*, 2000, page 876.

[12] Guting, R., Bohlen, M., Erwig, M., Jensen, C., Lorentzos, N., Schneider, M., Vazirgiannis, M.: A Foundation for Representing and Querying Moving Objects, *ACM Transactions on Database Systems (TODS)*, **25**(1), 2000, 42.

[13] Hüsemann, B., J.Lechtenbörger, G.Vossen: Conceptual Data Warehouse Design, *Workshop on Design and Management of Data Warehouses*, 2000, 3–9.

[14] Inmon, W.: *Building the Data Warehouse*, John Wiley & Sons, New York, 2005.

[15] Joint Typhoon Warning Center (JTWC): 2011, http://metocph.nmci.navy.mil/jtwc.

[16] Kamble, A.: A Conceptual Model for Multidimensional Data, *Fifth Asia-Pacific Conf. on Conceptual Modelling*, 79, 2008, 29–38.

[17] Kimball, R., Ross, M.: The Data Warehousing Toolkit, *John Wiley& Sons, New York*, 1996.

[18] Lema, C., Antonio, J., Forlizzi, L., Guting, R., Nardelli, E., Schneider, M.: Algorithms for Moving Objects Databases, *The Computer Journal*, **46**(6), 2003, 680.

[19] Luján-Mora, S., Trujillo, J., Song, I.: A UML Profile for Multidimensional Modeling in Data Warehouses, *Data Knowledge Engineering*, **59**(3), 2006, 725–769.

[20] Malinowski, E., Zimányi, E.: Hierarchies in a Multidimensional Model: From Conceptual Modeling to Logical Representation, *Data Knowledge Engineering*, **59**(2), 2006, 348–377.

[21] Malinowski, E., Zimanyi, E.: *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*, Springer, 2008.

[22] Microsoft Corporation: Multidimensional Expressions (MDX) Reference, Accessed: 28 April 2011, http://msdn.microsoft.com/en-us/library/ms145506.aspx.

[23] National Hurricane Center (NHC) - Season Archives: 2011, http://www.nhc.noaa.gov/pastall.shtml.

[24] National Oceanic and Atmospheric Administration (NOAA): 2011, http://www.aoml.noaa.gov/hrd.

[25] Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., Silvestri, C.: Spatio-temporal aggregations in trajectory data warehouses, *Data Warehousing and Knowledge Discovery*, 2007, 66–77.

[26] Papadias, D., Egenhofer, M.: Algorithms for Hierarchical Spatial Reasoning, *GeoInformatica*, **1**(3), 1997, 251–273.

[27] Parent, C., Spaccapietra, S., Zimányi, E.: *Conceptual modeling for traditional and spatio-temporal applications - the MADS approach*, Springer, 2006.

[28] Pedersen, T., Jensen, C., Dyreson, C.: A Foundation for Capturing and Querying Complex Multidimensional Data, *Information Systems*, **26**(5), 2001, 383–423.

[29] Prat, N., Akoka, J., Wattiau, I.: A UML-based Data Warehouse Design Method, *Decision Support Systems*, **42**(3), 2006, 1449–1473.

[30] Rivest, S. and Bedard, Y. and Marchand, P.: Toward Better Support for Spatial Decision Making: Defining the Characteristics of Spatial On-line Analytical Processing (SOLAP), *Geomatica*, **55**(4), 2001, 539–555.

[31] Sapia, C., Blaschka, M., Höfling, G., Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm, *ER '98: Workshops on Data Warehousing and Data Mining*, Springer-Verlag, 1999, 105–116.

[32] Schneider, M.: *Spatial Data Types for Database Systems - Finite Resolution Geometry for Geographic Information Systems*, vol. LNCS 1288, Springer-Verlag, 1997.

[33] Scotch, M., Parmanto, B.: SOVAT: Spatial OLAP Visualization and Analysis Tool, *38th Hawaii Int. Conf. on System Sciences (HICSS)*, IEEE, 2005, page 142b.

[34] Shekhar, S., Lu, C., Tan, X., Chawla, S., Vatsavai, R.: MapCube: A Visualization Tool for Spatial Data Warehouses, *Geographic Data Mining and Knowledge Discovery*, 2001, 74–109.

[35] Skiadopoulos, S., Koubarakis, M.: Composing Cardinal Direction Relations, *Artificial Intelligence*, **152**(2), 2004, 143–171.

[36] Tryfona, N., Busborg, F., Christiansen, J.: starER: A Conceptual Model for Data Warehouse Design, *Proceedings of ACM 2nd Int. Workshop on Data Warehousing and OLAP*, 1999, 3–8.

[37] Vaisman, A. A., Zimányi, E.: What Is Spatio-Temporal Data Warehousing?, *Data Warehousing and Knowledge Discovery (DaWaK), 11th International Conference*, 2009, 9–23.

[38] Vassiliadis, P., Sellis, T.: A Survey of Logical Models for OLAP Databases, *SIGMOD Record*, **28**(4), 1999, 64–69.

[39] Viswanathan, G., Schneider, M.: BigCube: A MetaModel for Managing Multidimensional Data, *19th Int. Conf. on Software Engineering and Data Engineering (SEDE)*, 2010, 237–242.

[40] Viswanathan, G., Schneider, M.: The Objects Interaction Graticule for Cardinal Direction Querying in Moving Objects Data Warehouses, *Advances in Databases and Information Systems*, Springer, 2010, 520–532.

[41] Zepeda, L., Celma, M., Zatarain, R.: A Mixed Approach for Data Warehouse Conceptual Design with MDA, *Int. Conf. on Computational Science and Its Applications*, 2008, 1204–1217.