
User-centric spatial data warehousing: a survey of requirements and approaches

Ganesh Viswanathan and Markus Schneider*

Department of Computer and
Information Science and Engineering,
University of Florida,
Gainesville, FL 32611, USA
E-mail: gv1@cise.ufl.edu
E-mail: mschneid@cise.ufl.edu
*Corresponding author

Abstract: The increased availability of spatial data in recent years has led to new challenges in the analysis of large multidimensional datasets. One solution is to integrate GIS with OLAP and relational databases. Another strategy has been to leverage existing spatial capabilities of databases to perform spatial OLAP. In this article, we review existing modelling strategies for spatial data warehousing at all three levels: conceptual, logical and implementation. We gather the most essential requirements for handling spatial data and use insights from spatial databases and GIS systems to design a meta-framework that would enable a user-centric modelling of complex data. Our strategy is to keep the user as the focal point in the analysis process and lay the foundation for clear data abstraction at different levels using multidimensional abstract data types and operations and thus support complex spatial data in data warehouses.

Keywords: user-centric data warehousing; spatial OLAP; meta-modelling; requirements for spatial data warehousing.

Reference to this paper should be made as follows: Viswanathan, G. and Schneider, M. (xxxx) 'User-centric spatial data warehousing: a survey of requirements and approaches', *Int. J. Data Mining, Modelling and Management*, Vol. x, No. x, pp.xxx-xxx.

Biographical notes: Ganesh Viswanathan works at Amazon Elastic Block Store (EBS), the cloud data storage offering from Amazon.com. His research interest lies in user-centric spatial data warehousing and OLAP particularly in the context of big data, and moving objects databases. He has published extensively in several journals and conferences related to data engineering. Previously, he has been a Visiting Researcher at Los Alamos National Laboratory (LANL). He holds PhD and MS in Computer Engineering from the University of Florida at Gainesville, and a Bachelor of Technology (BTech) from Anna University, India.

Markus Schneider is an Associate Professor at the Department of Computer and Information Science and Engineering of the University of Florida. He holds an MS in Computer Science from the Technical University in Dortmund, Germany, and a PhD in Computer Science from the University

of Hagen, Germany. His research interests include spatial, spatio-temporal, and moving objects databases, spatial data warehousing and SOLAP, spatial information science, geographical information systems, applied computational geometry, and extensible databases. He is the author of three books and has published more than 100 journal articles, book chapters, and conference papers. He is a co-founder of the ACM Special Interest Group SIGSPATIAL and is on the editorial board of the journal *GeoInformatica*. He is a recipient of the 2004 National Science Foundation (NSF) CAREER Award.

This paper is a revised and expanded version of a paper entitled ‘Requirements for spatial data warehousing and SOLAP’ presented at the 16th International Conference on Database Systems for Advanced Applications (DASFAA), Hong Kong, 2011.

1 Introduction

For more than a decade, data warehouses have been at the forefront of information technology applications as a way for organisations to effectively use information for business planning and decision making. They contain large repositories of analytical and subject-oriented data, integrated from several heterogeneous sources over a historical time-line (Kimball and Rose, 2002; Inmon, 2005). The technique of performing complex analysis over the information stored in the data warehouse is popularly called online analytical processing (OLAP). The large increase in the availability of spatial data in recent years has led to increased challenges in storing such information and analysing them. Data warehouses could provide an effective way to manage spatial information by providing large-scale storage, multidimensional data management and OLAP querying capabilities together in one system.

Spatial data warehouses (SDWs) are full-fledged data warehouses which provide native support for spatial data and advanced spatial online analytical processing (SOLAP) operations on them. These operations on the spatial objects can include *basic querying operations*, such as “Find the city with the largest sales volume for iPads in the state of Florida in 2010”, *map generalisation operations* such as “Find all states where the top five school districts out-performed all others within that state, between 2005 and 2010 in terms of student grades”, or spatial analysis operations such as *convex hull*: “Find the smallest convex region in Western United States containing the maximum number of college towns where more than 2500 units of Kinect were sold in 2010”, and selective *spatial union*: “Return the geometry of the region in Florida described by the counties where DropBox usage exceeded that of Twitter in the last five months. This last query requires a spatial aggregate union on the geometry of the various counties satisfying the condition. 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. An brief comparison of existing models for spatial data warehousing was introduced in Viswanathan and Schneider (2011). In this article, we extend the work by providing a detailed study of the existing state-of-the-art in SDW methodologies, describe the essential requirements for user-centric multidimensional modelling and motivate the need for a generic framework for the design of such as system.

In order to provide built-in support for spatial data in analysis systems it is important to incorporate *spatial data hierarchies*, *spatial data dimensions* and *spatial measures* within the data warehouse. These would help to support *spatial aggregation operations* on them. However, these pose several new challenges related to spatial data modelling in a multidimensional context, such as the need for new spatial data types suited for aggregation operations, inclusion of spatial hierarchies in data dimensions and as measures, the development of new SOLAP operations, ensuring consistent and valid SOLAP, etc. Consider for example the result of the spatial aggregate union query shown above. Since the result can be either – a simple region, a simple region with holes, a complex region with multiple faces, or a complex region with multiple faces that bear holes – based on the selection condition for the constituent counties, a SDW model should provide dynamic, built-in support for such return types.

OLAP operations are often categorised as distributive, algebraic and holistic (Han and Kamber, 2006; Gray et al., 1996), depending on whether the measures of high level cells can be easily computed from their low level counterparts, without accessing base tuples residing at the finest level. For example, in the classic *sales(location,time,product)* data, the total sales of an item at (Florida, 2010) can be calculated by adding up the total sales of (Florida, January 2010) ... (Florida, December 2010), without looking at base data points such as (Florida, 20 March 2010), which means that SUM is a distributive measure. In comparison, AVG is often cited as an algebraic or semi-distributive measure, in that AVG can be derived from two distributive measures: SUM and COUNT, i.e., algebraic measures are functions of distributive measures. Holistic measures such as standard deviation require data at the specific requisite level for all computations. Similarly, spatial querying and aggregation operations such as spatial roll-up, drill-down and selection also involve several levels of data manipulation. For example, consider a drill down operation from country (region) to county (maps) to cities (string labels for points). This complex navigation operator can be very useful in mining several levels of spatial information such as geo-spatial and video data.

Upon reviewing existing modelling approaches for spatial data warehousing (Section 2), we found that one of the major shortcomings of existing models is the heavy focus on direct implementation strategies such as a combination of OLAP tools or GIS mapping clients with databases to create a ad-hoc SDW. However, for effective multidimensional data modelling and analysis what is needed is a refined data warehouse architecture that keeps the *user as the focal point* and achieves a *clear abstraction* of the data for all stakeholders in the system. Hence our proposal is for a sound conceptual model built on abstract data types (ADTs) and using the cube metaphor for OLAP analysis while natively supporting spatial data along the data dimensions and as measures for aggregation. The user view is created by using a generic textual analysis language such as an extension of MDX that helps to write SOLAP queries. Finally, a set of transformation rules from the conceptual model to logical design strategies such as *relational OLAP (ROLAP)* (Inmon, 2005), *multidimensional OLAP (MOLAP)* (Kimball and Ross, 2002) and *hybrid OLAP (HOLAP)* (Pedersen and Jensen, 2002) is also needed to help complete the design of the SDW. Overall, this paper provides a new insight into the fundamental requirements for designing a user-friendly SDW model by providing an objective analysis of the essential requirements for it.

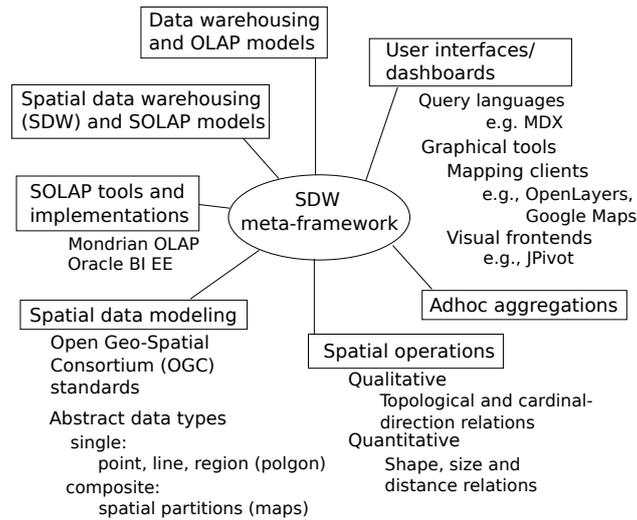
The rest of this paper is organised as follows. Section 2 provides a comprehensive review of the existing literature regarding conceptual data warehouse modelling, spatial data and SDW modelling, and some user interfaces used for such systems. Section 3

discusses the essential requirements for SDWs and OLAP. Section 4 presents our meta-framework as a novel strategy for developing an enhanced conceptual model based on the cube metaphor that is user-centric and capable of natively supporting spatial data and aggregations on them. Section 5 concludes the paper and mentions topics for further research.

2 Related work

In this section, we review existing research on data warehousing and OLAP tools, spatial data modelling and associated implementation strategies, leading to the list of essential requirements for spatial data warehousing (in Section 3). Figure 1 illustrates the various domains that need to be considered for deciding the architecture of a SDW (Section 4). A survey of the state-of-the-art in each of these domains is the topic of the current section.

Figure 1 An illustration of the various domains considered during the design of the SDW meta-framework



2.1 Data modelling for OLAP

Over the past decade several approaches have been proposed for modelling data warehouses to enable OLAP over large-scale enterprise data. Now, we present a study of the best available conceptual and logical models for data warehousing. Existing conceptual modelling approaches can be broadly classified into *extensions of entity relationship (E/R) models* (Franconi and Kamble, 2004; Kamble, 2008; Sapia et al., 1999; Malinowski and Zimányi, 2006; Tryfona et al., 1999), *extensions of unified modelling language (UML)* (Abelló et al., 2006; Luján-Mora et al., 2006; Prat et al., 2006) and *ad-hoc* (Golfarelli et al., 1998; Hüsemann et al., 2000; Zepeda et al., 2008; Viswanathan and Schneider, 2011) design models. Several different logical models have also been proposed to model multidimensional data in the past few years. The data cube operator was formally introduced in Gray et al. (1996) in an attempt to extend the

relational model to suit multidimensional analysis. A complete survey of the properties of several earlier logical design models can be found in the works of Blaschka et al. (1998), Vassiliadis and Sellis (1999) and Pedersen et al. (2001). Though many of these models aid in the relational representation of aggregate data, contributions like the ALL operator and concepts regarding data hierarchies are significant even in a multidimensional context. One of the earliest approaches for multidimensional modelling was introduced by Kimball (2011). This *dimensional modelling* approach proposes an informal methodology to derive the multidimensional schema and provides a way to develop a relational implementation in the form of the star schema. Dimensional modelling imposes some rules on the modelling but results in a data model that has the access methods defined clearly by virtue of the relationships (Kimball, 1997; Kimball and Ross, 2002). Users are also better able to relate to the ‘see measure by dimensional value(s)’ paradigm rather than a simple ‘collection of values’. The approach involves discovering the data-marts for the data-warehouse space, listing all dimensions for each data-mart, using an ad-hoc matrix to capture user requirements, and then designing a fact table with measures added to each grain of detail along the dimension levels. The model presented by Agrawal et al. (1997) is a logical data model for multidimensional databases. The cube is defined as a set of *dimensions* (each associated with a domain) and a set of *elements* (measures). A mapping is provided between the dimensions and the set of elements. The elements of the cube can be 0,1 (the Boolean Cube) or a n tuple of elements. This model does not require the dimensions to have a ranked, discrete domain. Instead the mapping function can be used to provide a symmetric treatment between measures and dimensions. An algebra is also defined over the model with operations such as *push* and *pull* (to transform a dimension into measure and vice-versa), *destroy dimension*, *restriction* (to constraint member values), and *join* (to combine two cubes). Several other operations like *cartesian product*, *natural join*, and *associate* are also mentioned. However, this model does not discuss the handling of explicit multiple hierarchies among dimensions or the problem of imprecision due to double counting during data aggregation.

2.2 Conceptual SDW modelling and SOLAP

Spatial data warehousing has become a topic of growing interest in both the database and GIS communities in recent years. This is primarily due to the explosion in the amount of spatial information available from various sources such as GPS receivers, communication media, online social networks and other geo-spatial applications. Consequently some *SOLAP* tools are now available to help model and analyse such data. The term *SOLAP* was introduced in Rivest et al. (2001), though leveraging decision support systems with spatial data and operations has been the topic of research in database and GIS communities for several years. Rivest et al. (2001) is an early approach to *SOLAP*, which mentions essential *SOLAP* features classified into three requirements. The first requirement is to enable *data visualisation* via cartographic (maps) and non-cartographic displays (e.g., 2D tables), numeric data representation and the visualisation of context data. Secondly, *data exploration* requires multidimensional navigation on both cartographic and non-cartographic displays, filtering on data dimensions (members) and support for calculated measures. The third requirement discussed involves the *structure of the data*, for example, the support for spatial and mixed data dimensions and the support for storage of geometric data over an

extended time period. The conceptual design models for SDWs are extensions of E/R and UML diagrams or ad-hoc design approaches. Among the extensions of E/R models, Malinowski and Zimányi (2004b) presents a clear integration of spatial data for OLAP by extending the MultiDimER and MADS approaches. Among the other ad-hoc design approaches, Ferri et al. (2002) presents a formal framework to integrate spatial and multidimensional databases by using a *full containment* relationship between the hierarchy levels. In Jensen et al. (2004), the formal model from Pedersen et al. (2001) is extended to support spatially overlapping hierarchies by exploiting the partial containment relations among data levels, thus leading to a more flexible modelling strategy. Bimonte et al. (2006, 2010) present the *GeoCube* model for SDW design, based on a formal schema and instance definition for cube elements. GeoCube extends conventional SOLAP operations with five new operations named *classify*, *specialise*, *permute*, *OLAP-buffer* and *OLAP-overlay*. However, one of the shortcomings of this approach is the use of many-to-many mappings between data-dimensions and facts. Since each cell of the data cube is a unique Cartesian product of the associated data dimensions, this many-to-many mapping weakens the cube structure and makes it difficult to apply constraints and dynamic schema changes during OLAP operations.

2.3 Logical SDW modelling

The logical SDW design models aim to provide support for spatial data dimensions (Scotch and Parmanto, 2005), spatial measures (Han et al., 1997; Marchand et al., 2004; Shekhar et al., 2004; Rivest et al., 2005) and spatial aggregations (Gomez et al., 2009). The concept of spatial measures (with a specific geometric part) is either defined as references to spatial objects (Stefanovic et al., 2002; Rivest et al., 2001), as the results of topological, distance or metric operations (Malinowski and Zimányi, 2004b; Rivest et al., 2001), or as values associated with a spatial data dimension in the data cube (Marchand et al., 2004; Han et al., 1998). In Stefanovic et al. (2002), the authors classify spatial dimension hierarchies according to their spatial references as non-geometric (like traditional descriptive data dimensions), geometric to non-geometric, and fully geometric. In addition to supporting spatial objects, most GIS models use both geometric (e.g., the extent of fire spread is shown as a *polygon*) and thematic or descriptive attributes (e.g., state name) to help qualify geometric data objects (Rigaux et al., 2002). This is a very useful feature for supporting spatial aggregation operations and map generalisations (such as moving from *state* level to *country* level in the *location* hierarchy). A discussion of spatial hierarchies and topological operators in a conceptual SDW model is presented in Malinowski and Zimányi (2005). Some techniques to refine ragged and unbalanced hierarchies in logical multidimensional database design by the use of functional dependencies is specified in Niemi et al. (2001). Shekhar et al. (2001) extend the MapCube operator to support spatial data and aggregations, but the model is rather constrained and not easily extendable for user-defined queries.

2.4 Implementations of SDW and SOLAP systems

The major implementations of SOLAP tools can be broadly classified as *OLAP dominant*, *GIS dominant*, or *integrated OLAP and GIS* solutions (Rivest et al., 2005). OLAP approaches provide means for aggregation of data, while GIS approaches focus on geometric operations and visual data selections while limiting multidimensional data analysis. Another approach is the integration of OLAP and GIS systems (Marchand et al., 2004; Scotch and Parmanto, 2005; Bimonte et al., 2006). In Rivest et al. (2005), the authors present requirements and guidelines for implementing *SOLAP* technology and introduce a commercial product called JMAP that combines GIS and OLAP technology and presents an easy-to-use interface for performing analysis for non-technical users. Though GIS systems have traditionally been used for geo-spatial exploration, they have an enormous drawback when considering OLAP analysis requirements. This is primarily because GIS systems are not built to support interactive navigation along data hierarchies and to provide decision support. Instead, they are often transactional systems that import spatial data files and provide a cartographic user interface for visual exploration and specific spatial functionality. However, an integration of GIS and OLAP systems could be an interesting path for developing SDWs that can facilitate complex spatial operations along with decision support functionality. The GeoMondrian Project (2011) aims to develop an open-source implementation of a SOLAP analysis server. Currently, it provides a spatially enabled version of the Mondrian OLAP server (Pentaho Analysis Services: Mondrian Project, 2011). However, in our view the GeoMondrian system lacks a clear underlying spatial data model to help integrate spatial objects into OLAP systems. Instead, it is essentially built ad-hoc, by using a combination of the Java Topology Suite (2011) (which provides spatial operations according to OGC standards) and Mondrian (which provides the OLAP operations on thematic attributes) with PostGIS (which provides the spatial data types). These together create a functional spatial data analysis toolkit supporting the integration of spatial data and operations in an OLAP server.

2.5 Geo-spatial data models and operations

For modelling spatial data there are now several established approaches in the database community. An introduction to basic spatial data types is given in Shekhar and Chawla (2003). The ROSE algebra (Guting and Schneider, 1995; Guting et al., 1995) provides a more robust discussion of spatial data types by introducing types such as *point*, *line* and *region* (Schneider, 1997) for simple and complex spatial objects and describes the associated spatial algebra. Composite spatial objects (collections of points, lines and regions) are presented as *spatial partitions* or *map* objects. Similarly, the Open GIS Consortium also provides a reference model (Open GIS Consortium: Reference Model, 2011) as a standard for representing geo-spatial information. Qualitative spatial operations include topological relations (Schneider and Behr, 2006) such as *disjoint*, *meet*, *overlap*, *equal*, *inside*, *contains*, *covers* and *coveredBy*, and cardinal direction relations. Quantitative relations on spatial objects include metric operations based on the size, shape and metric distances between objects or their components. All these operations can be used to query and analyse spatial data in the data warehouse.

3 Requirements for user-centric SOLAP

For a data warehouse model to be effective in modelling, storing and querying data, some essential requirements need to be met. Blaschka et al. (1998) provide a list of requirements for multidimensional modelling for OLAP applications. Pedersen et al. (2001) present eleven requirements for a multidimensional model using a clinical data warehousing application as an example. These requirements are then used to evaluate fourteen existing models, to classify them into three groups as *simple cube models*, *structured cube models* and *statistical object models*. Finally, an extended multidimensional model is also presented. Tsois et al. (2001) presents ten additional requirements for conceptual models of multidimensional data and compares them against several other earlier models. By studying these works, the existing models for multidimensional data modelling (Section 2), and several new OLAP tools and applications that have emerged in the last few years, we now compile a list of basic features that must be present in any effective user-centric SDW model. For a SDW, we explicitly provide four fundamental requirements, namely, supporting spatial data objects as members and measures of the data cube, supporting spatial data hierarchies, supporting spatial data dimensions (for conceptual modelling and visualisation) and supporting spatial operations and aggregations.

- 1 *Multidimensional data store*: A SDW system must first and foremost qualify as a *multidimensional* data store. The primary reason behind this basic property is to ensure support for increasing data dimensionality over time. Since data warehouses typically integrate data from heterogeneous sources over a broad timeline, it is common to see a large number of attributes for each data object accommodated into a *data mart*. Several such data marts with varied and independent dimensionality are often threaded together to build a single, large enterprise level data warehouse system. Thus, the model should allow for multiple facts, data dimensions and even multiple data cubes to be included in the data warehouse system. Limits to the multidimensionality of the data cube or the granularity of the members or measures should not be based on logical or implementation considerations.
- 2 *Simple user view*: As data warehousing and OLAP systems have become increasingly complex to understand and develop, in recent years the trend has been to get back to a *user-centric* approach for modelling multidimensional data. Thus the second important requirement for a SDWs is that the *user view* be simple and intuitive, yet capable of capturing the full dimensionality of the data. In recent years, many data mining and complex knowledge gathering systems are being built over traditional relational databases and complex scientific file systems. However, for users to be able to perform complex analytics over large data, the foremost requirement is an easy-to-use interface. Such an interface should have the following properties:
 - simple
 - easy to understand the structure of the data (the interface should clearly illustrate the conceptual structure of the data, for example, in terms of classes and associations)

- easily available aggregation functions on particular data types and the means to apply them
- easy to gather results, visualise and export or save them
- ability to perform multi-level queries using the results from previous aggregations
- ability to create new types and specify the syntax and semantics of new operations in an extensible manner.

One example of such an user-interface is an abstract multidimensional view, like a *data cube* or *multidimensional array* with a supporting set of OLAP operations that would make it easier for users to navigate through hierarchical data and perform analysis.

- 3 *Implementation independent conceptual design*: By definition, any *conceptual model* should be completely free of implementation aspects to serve as an effective data model. Since *business intelligence* (BI) systems are most often used by data analysts for decision support, the user view should be independent of implementation aspects to ease analysis. Thus, for example, the use of fact *tables* and dimension *tables* (thereby exposing ROLAP implementation) should be avoided in the user view. The data should be viewable to the analyst at an abstract and high-level, without requiring the understanding of complex logical designs such as star, galaxy or snowflake schema, or physical implementation considerations such as optimisations for materialised views, column stores and indexing requirements. Internal requirements should not dictate the conceptual design of the data warehouse as this can further force users to model data warehouses for system specific implementations and restrict data analysis.
- 4 *Separation of structure and values*: There should be an explicit separation of schema and instances, i.e., the structure of data and their actual values. The distinction between structure and instances of the data cube helps the analyst to apply OLAP operations and manipulate the multidimensional view of data and its contents independently. For example, consider dropping a data dimension in a data cube. Though this is a simple conceptual operation for the user to change the structure of the data cube, it also drastically affects the values inside the cube. The cells of the cube have to be reevaluated and their contents updated with new measures. However, this reformulation of the cube should not affect the view of the data cube itself for the analyst, and should only be noticeable as a change in the *state* of the cube. This helps in keeping analysis *online* and user-friendly, without developing new transformations from conceptual to logical designs for each new state of the multidimensional cube. Further, this also allows for efficient type checking for ensuring the validity of complex OLAP operations.
- 5 *Descriptive attributes*: Thematic or descriptive attributes for members and measures (geometric or otherwise) allow adding additional information about data. For example, applications such as web data warehouses often involve one or more *keyword* or *tag* fields, and GIS and spatial database systems use *labels* to help identify, qualify and correctly represent composite spatial partitions. This must be supported by the SDW model. Additionally, selection, navigation and aggregation

queries over such thematic attributes should also be available to improve analysis capabilities.

- 6 *Explicit hierarchies*: Hierarchies (with several levels of member or measure categories) should be supported explicitly in the *data dimensions* and even for the various *facts of analysis*. Such data hierarchies should be supported as ‘first class citizens’ inside the data warehouse. Moreover, hierarchies should also be supported in their most general form, meaning that ragged, unbalanced and uneven hierarchies of data should be available and usable for analysis. This requirement provides an opportunity to model complex data objects with variable representations in the most generic and user-friendly format without any influence from implementation considerations (Niemi et al., 2001).
- 7 *Multiple hierarchies*: Multiple hierarchies along the data dimensions and even measure values should be supported. Data hierarchies in SDWs can be of two basic kinds: *data dimension hierarchies* and *object hierarchies*. The former set includes hierarchies along the data dimensions, which can allow users to ‘roll-up’ or ‘drill-down’ along the levels of the hierarchy. The object hierarchies are complex hierarchies representing the internal structures of basic data types such as a *region*. Thus, in their most generic form, SDWs should support object hierarchies as members of data hierarchies. However, the uniqueness and constraints for each of them should be uniquely maintained, for example, it must be possible to ensure that a region object does not contain dangling lines or independent points in its structure. The support for multiple *data dimension* hierarchies is a well known requirement for data warehouses (Pedersen et al., 2001; Malinowski and E. Zimányi, 2004a; Tsois et al., 2001). However, in this article, we also motivate the need for *object hierarchies*, to enable native support for hierarchical UDTs in DWs. Consider, for example, Figure 2(a) that illustrates a *complex region* object which consists of three regions with one of them inside the *hole* of another. Another example in Figure 2(b) displays a single *face* of a region object (which can also be regarded as a *simple region*) with multiple holes. Such complex data objects can require several hierarchies for correct representation of their structure, attribution of internal types to measure values (“Find the quantity of sales of coffee beans in *mainland USA* (one face of entire USA object) in 2010”) and for performing efficient operations on them. Figure 3(a) provides a more detailed visualisation of a complex region object with three faces labelled as F1, F2 and F3. The interior, exterior and boundary point sets of the region are also displayed. After performing a scan operation, the cyclic order of the region’s boundary is stored to represent each face uniquely. A secondary hierarchy linking the sibling lists of outer cycles can help to optimise operations such as intersections and unions that involve computations on the object’s geometry. Figure 3(b) shows the detailed tree structure of a *region* object. In the figure, *face*[], *holeCycle*[], and *segment*[] represent a list of faces, a list of hole cycles and a list of segments respectively. In the tree representation, the root node represents the structured object itself, and each child node represents a component named *sub-object*. A sub-object can further have a structure, which is represented in a sub-tree rooted with that sub-object node. For example, the region object in Figure 3(a) consists of a label component and a list of face components. Each face in the face list is also a structured object that contains a face label, an outer

cycle, and a list of hole cycles, where both the outer cycle and the hole cycles are formed by segments lists. While storing such a region object in the data warehouse, it is necessary to provide the ordered segment lists for the *face cycles* for performing efficient plane-sweep operations. However, we would also like to store secondary hierarchies in the structure such as the basic region hierarchy as illustrated in Figure 3(b). This can only be achieved if multiple hierarchies are allowed in the SDW.

- 8 *Support for irregular hierarchies:* There must be support for non-conformant (non-onto, non-strict and ragged) hierarchies and generalisation/specialisation (is-a) relationships (Malinowski and E. Zimányi, 2004a; Niemi et al., 2001). For example, consider a location hierarchy that exists in a Sales data warehouse: $\langle \text{City} \rightarrow \text{County} \rightarrow \text{State} \rightarrow \text{Country} \rangle$. If the user would later like to include another hierarchy such as $\langle \text{School District} \rightarrow \text{City} \rightarrow \text{Voting Zone} \rightarrow \text{Country} \rangle$ by creating new, independent levels along with some existing levels, this should be allowed by the abstract model. Note that the two paths of the location hierarchy illustrated above, specify different aggregation semantics on the measure values. Such scenarios are often encountered in data warehouses where updates to the structure of data emerge with the inclusion of heterogeneous datasets over time.
- 9 *Support for attribute aggregations:* The model must provide good support for aggregation on both geometric and alphanumeric attributes apart from basic numeric and statistical computations on the members and measures of the OLAP cube. Examples of possible aggregate operations are shown in Table 1. There should be also support for aggregations along attributes that are not part of the data dimensions, hierarchies or measures themselves, such as thematic attributes. Further, apart from aggregations inside just one data store, analytics should also be available between different OLAP cubes to enable the the seamless integration of the various data marts.
- 10 *User-defined aggregates, extensible OLAP:* User defined aggregation functions should be supported. These may even include ad-hoc operations such as *ratio* (metric) and multi-level buffer (geometric) operations. For example, consider a query to find the ‘moving buffer’ in ranges of 10 km over *Fukushima Prefecture* in Japan to assess the extent of spread of radioactivity through the atmosphere and to aid in relocating the population (Figure 4) after the 2011 earthquake. Since the rate of contamination reduces with the range from the affected region, the buffers and the rate of contamination in each zone in this case are aggregations computed over the previous buffer extents based on the rate of decay of nuclear activity with spatio-temporal variations. Thus, support for ad-hoc, user-defined geo-spatial operations on both spatial measures, members and their thematic attributes provides for an *extensible data analysis system*.
- 11 *Online aggregation:* The model should allow for multiple levels of *online* aggregation, i.e., dynamic, multi-level query design. This allows, for example, to navigate along a spatial data dimension of the cube while thus aggregating the measure values inside it and determining a valid analysis result.
- 12 *Handling data imprecision and summarisability conditions:* An important property of any multidimensional data model is ensuring *correct summarisability*. This

property was first introduced in the domain of statistical databases in Lenz and Shoshani (1997). A SDW model should be able to handle data imprecision so that double-counting of data is avoided, and non-additive data are not summarised. This is particularly relevant for semi-additive and non-additive aggregate operations on SOLAP data. For example, consider averages computed for the sales percentages of an item at the *city* granularity. When rolling up to the next level, i.e., *state*, one must ensure that the average is re-computed by taking into account the new sales percentage figures at the new granularity. Moreover, the association of spatial geometry to measure values must be evaluated correctly while performing aggregations across spatial hierarchies.

- 13 *Drill-across capability*: The model should support drilling across dimensions, i.e., sharing of dimensions among different fact cubes. For example, consider a sales data cube with the following data dimensions: product, time and location. For the location information consider a level called *district* in one cube. The same level is named as *prefecture* in another data cube due to localisations. Thus we require capabilities to relate and drill-across data cubes along varied granularities by means of suitable association functions. However, the aggregations over such operations should be correct and should yield meaningful results. This can be achieved by explicitly monitoring the current *state* of a cube during OLAP navigation.
- 14 *Drill-through capability*: The model should support drilling through capability to be able to query the base level (raw) data. This means that access to the *base* data cube and the stored low-level data (in databases, spreadsheets or complex scientific formats) must be available to the user.
- 15 *Handling uncertainty*: The model should also be able to handle the uncertainty in the data using techniques such as *data lineage tracking* or special null or \perp values.
- 16 *Handling changes over time*: Another requirement for data warehouses which has been the subject of research in several domains including statistical databases, multidimensional data warehouses and temporal OLAP systems, is the ability to handle updates and deletions over time. Since data warehouses typically collect data over a long time period, the system should ensure that (re)calculations of measure values are consistent and correct over time.
- 17 *Complex abstract types and (spatial) data objects*: The model should support the basic set of data types such as alphanumeric types (int, char, etc.) and more complex types such as geo-spatial types (point, line, region, etc.), temporal types (time interval, instant, etc.). This helps to integrate existing types and operations into the system for OLAP analysis. Further, the model should be extensible to support abstract *user defined types* (UDTs) and operations on them. There must be facilities to specify the syntax and semantics of such UDTs, along with any additional constraints to ensure meaningful aggregations on such data. These complex objects can reside as measures of analysis or as the members of the data dimensions in the multidimensional data warehouse. Additionally, the model should also provide support for multiple (composite) and complex members and 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 2, with its several faces and half-segment cycles. Thus, it is essential that the spatial data cube be capable of storing and managing spatial members and measures as both simple, complex and composite (map) spatial objects.

- 18 *Support for spatial hierarchies*: The model should support generalisation and specialisation hierarchies on spatial objects. This would, for example, enable roll-up operations from a *city* level to a *state* level to a *country* level in the *location* hierarchy. Further allowing inter-linking of spatial hierarchies with thematic attribute hierarchies allows for improved multidimensional data analysis. For example, a query such as: *Find the trajectories of hurricanes labelled Category 3 or higher that traversed the state of Florida in 2005* involves returning spatial line objects (*trajectories*) selected through spatial (*Florida*), temporal (*year*) and thematic constraints (Hurricane Category).
- 19 *Support for spatial dimensions*: Hierarchies can be used as the data dimensions defining the spatial data cube structure of the data warehouse. This native support for spatial dimensions will help users to perform selection, navigation and aggregation operations on both members and measure values easily. The support for spatial data and spatial hierarchies along the data dimensions is one of the essential requirements for any integrated spatial decision support system. Additionally, the model should allow one or more spatial hierarchies to be combined as a single data dimension to define the cube structure. A unique separation of data dimensions as fully geometric, semi-geometric and non-geometric dimensions was introduced in Rivest et al. (2001). Bédard et al. (2001) describe a SOLAP system supporting these three types of spatial dimensions (Figure 5). Geometric spatial dimensions are said to comprise geometric shapes in all levels of the data dimension. Non-geometric or descriptive dimensions essentially contain only alphanumeric data in their dimension members and mixed spatial dimensions comprise some spatial shapes and non-spatial data along the hierarchical levels. Though this distinction of types sounds intuitive, it is often difficult to formalise data types and apply constraints for closure and uniqueness based on such generic structures. Another approach to handling spatial data along data dimensions is to have unique geometric and non-geometric data dimensions followed by an association operator between them. For example, one can follow the path from a *city* object (*polygon*) to a *state* name (*string*) in GIS systems by first using the map generalisation operator and then selecting the *map label* for that state. A similar notion could be applied in spatial data cubes to generalise spatial dimensions with non-spatial thematic attributes.
- 20 *Support for spatial operations and aggregations*: The model should explicitly support operations and aggregations on the spatial measures and members. Operations on spatial data can include *qualitative operations* such as finding topological and cardinal direction relations between the interacting spatial objects, or *quantitative operations* such as metric and distance finding. Other operations on spatial data can include *feature specific operations* such as manipulation operations on point, line and region, map operations (on a collection of spatial

features) such as map generalisation, map fusion, etc. Aggregation operations on spatial data can include both conventional set operations (such as union, intersection and difference), aggregations based on the thematic properties of spatial objects (such as the sum of areas of selected counties) or geometric aggregations such as creating minimum convex polygons (convex hull, minimum bounding circle and minimum bounding rectangle), buffering, etc. Consider, for example, the following aggregate operation on a SDW: “Compute a *convex hull* on cities having the top-k highest sales of iPads in every state in 2010”.

Figure 2 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

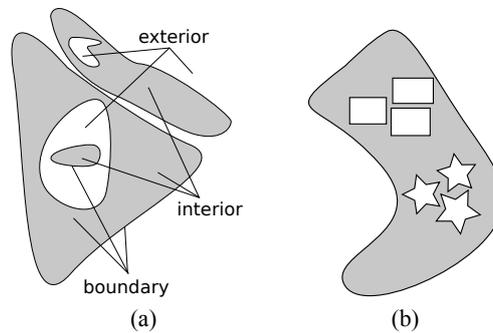


Figure 3 Illustration of a complex structured region showing faces F1 (containing outerCycle C1 and holeCycle C2), F2 (with cycle C3) and F3 (with cycle C4), and a hierarchical representation for the *region* (or multi-polygon) object

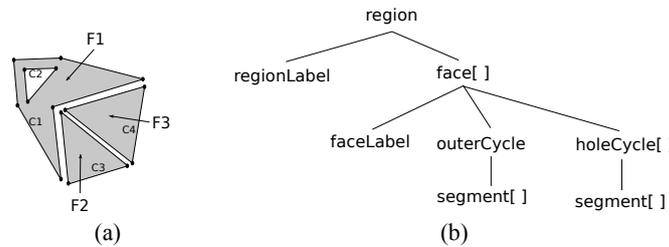


Table 1 Examples of non-spatial and spatial aggregation operators

Type	Bigcube aggregation operator
Additive	Sum, count, max or apex, min or base, concatenate, convex hull, spatial union, spatial intersection
Semi-additive	Average, variance, standard deviation, MaxN, MinN, centroid, centre of gravity, centre of mass
Non-additive	Median, MostFrequent, rank, LastNonNullValue, FirstNonNullValue, minimum bounding box, nearest neighbour, equi-partition

Figure 4 Illustration of three 10 km range buffers for Fukushima Prefecture in Japan to access spread of contaminated material during the 2011 earthquake

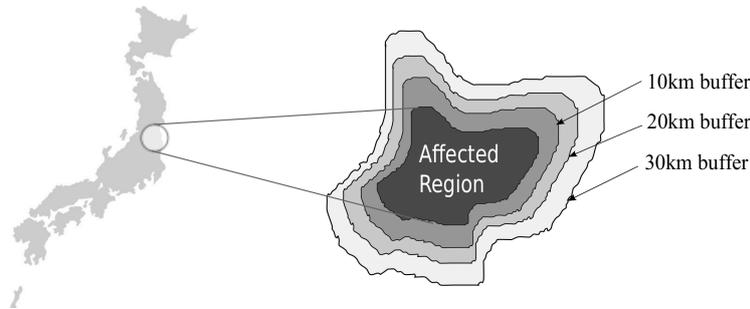
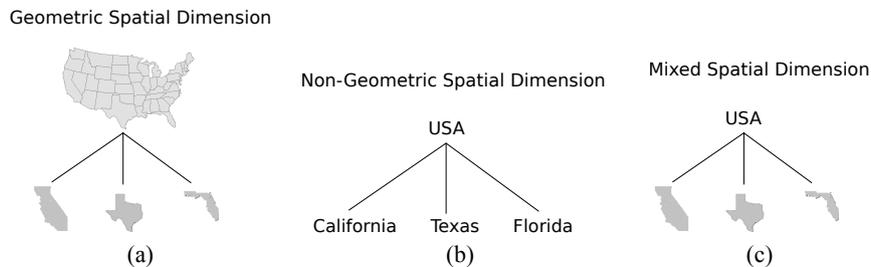


Figure 5 Types of spatial dimensions, (a) geometric (b) non-geometric (c) mixed



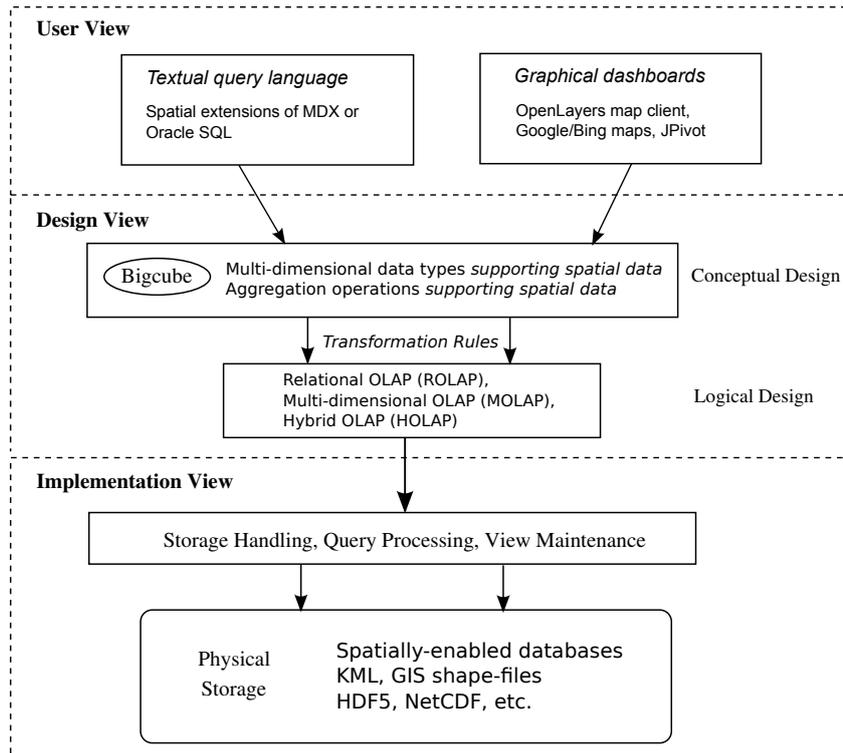
4 A meta-framework for SDW design

After reviewing the existing data warehouse and SOLAP modelling approaches and generating the list of essential requirements for an effective SDW model, we now provide a broad insight into how a SDW architecture should be constructed for supporting user-centric OLAP. For providing user-friendly spatial data analysis it is essential to use an abstract data model to design and construct the data warehouse. This can only be provided by a conceptual design view that fully abstracts from the underlying implementation details. To allow users to interact with the conceptual cube, a user view (query language or visual map interface) can be used to expose the set of data types and operations for OLAP analysis. At each level, explicit support for spatial data must be provided using spatial data types which can represent single objects such as *points*, *lines* or *regions*, or a combination of these in terms of *spatial partitions* or *map* objects. Figure 6 illustrates such a meta-framework that we propose for SDW design.

A conceptual model for SDWs should provide built-in support for spatial objects by using ADTs or by extending multidimensional data types such as *perspectives* (data dimensions) and *analysis subjects* (facts) to include spatial values. Examples of such approaches include the *BigCube* model (Viswanathan and Schneider, 2010) which provides ADTs arranged over different levels to create the conceptual cube or E/R and UML, or other ad-hoc design models such as Jensen et al. (2004), Malinowski and Zimányi (2004b), Gomez et al. (2009) and Bimonte et al. (2010). Later, additive,

semi-additive and holistic classes of aggregation operations can be defined over them (Han and Kamber, 2006; Gray et al., 1996). For example, Ruiz and Times (2009) names several possible spatial aggregate operations in a broad list. However the exact syntax, signature and semantics of these operations is still unclear.

Figure 6 A meta-framework for SDW design illustrating the distinct conceptual and logical design levels and the user view for OLAP analysis



A set of transformation rules are needed from the conceptual model to the logical design level. The logical design can be done in one of three ways. Data warehouse star, snowflake or galaxy schema can be constructed and the corresponding relational tables are stored in a database linked by foreign keys and other functional dependencies. This is called Relational OLAP or ROLAP. In multidimensional OLAP design, data cubes can be constructed in memory to store and operate over the data warehouse. This is very similar to the cube model used for conceptual design. However, though multidimensional querying is often faster in comparison to relational querying, this approach can lead to increased memory and storage requirements. A balance between these two approaches is achieved in Hybrid OLAP by using a combination of relational and multidimensional design strategies. For example, in-memory multidimensional arrays can be used for constructing the materialised views that enable faster query processing on frequently accessed measures and data dimensions, while base level data (at highest granularity) is still stored in relation datasets. A drill-through operation can be used to retrieve the raw data when required.

The user view can include generic textual query languages, a visual graphical dashboard of map clients such as OpenLayers (OpenLayers Mapping Client, 2011), Google or Bing maps or tabular representations using tools such as JPivot. A combination of these tools is often required for effective data visualisation and user-friendly analysis to design multiple levels of queries.

Such a meta-framework defined over several unique, cognitive levels can help users to easily design and develop data warehouses. Translations to logical and physical structures can be handled internally and this results in an effective, user-centric strategy to let users focus on performing spatial and thematic data analysis instead of being burdened by physical system aspects. For example, the following query can be interesting in a SDW context: “Find all states neighbouring those states where more than 5,000 black iPhone units were sold in March 2010”. This involves a selection on the thematic ‘colour’ attribute of the product, followed by a test for the topological relation *meet* on the spatial partitions resulting from the basic query.

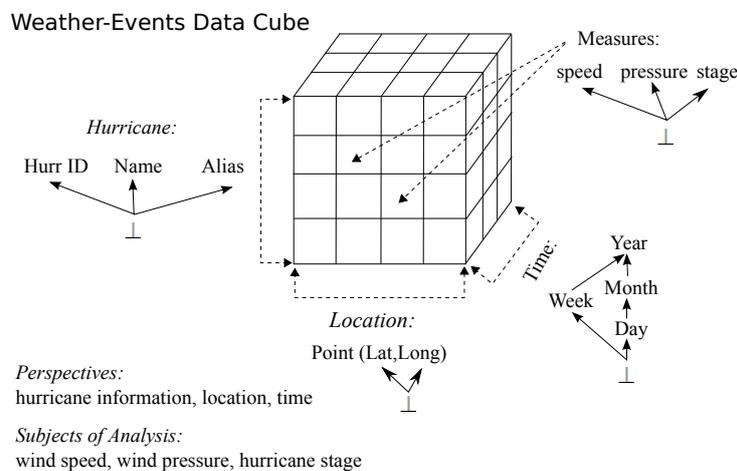
4.1 Case study

To illustrate the application of such a user-centric, conceptual data warehouse meta-framework, we have used an example from *tropical weather events* research. The US National Hurricane Centre (NHC – Season Archives, 2011), NOAA Hurricane Research Division (NOAA, 2011), and Joint Typhoon Warning Centre (JTWC, 2011) collect data about hurricane events in the North Atlantic and Pacific Ocean using a combination of satellite, weather balloons and flight telemetry systems. These datasets 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 SDW with data cubes describing the hurricane trajectory and other associated attributes (an example is shown in Figure 7). 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 5 km 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 employed the generic SDW meta-modelling approach to design the example data cube. 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 visualisation. 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 SDW for performing OLAP operations on large-scale datasets. The generic meta-model for such a SDW 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.

Figure 7 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



5 Conclusions and future work

In this paper, we present an overview of existing conceptual, logical and implementation strategies for SDWs. By studying these models we arrive at a set of essential requirements for incorporating spatial data in data warehouses. These are used to propose a ‘meta-framework’ for modelling SDWs. This framework consists of a user-friendly conceptual cube model that abstracts over logical design details such as star or snowflake schemas and implementation details such as the maintenance of materialised views. Further, user-friendly views are proposed for the SDW by means of a generic textual query language like a spatial extension to MDX, and graphical dashboards or cartographic mapping tools such as JPivot, OpenLayers or Google map visualisations. Overall, this provides a comprehensive view of the existing state-of-the-art in SDW modelling and lays the foundation for incorporating spatial cubes as ADTs in the modelling of complex multidimensional data.

Acknowledgements

This work was partially supported by the National Aeronautics and Space Administration (NASA) under the grant number NASA-AIST-08-0081.

References

- Abelló, A., Samos, J. and Saltor, F. (2006) 'YAM2: a multidimensional conceptual model extending UML', *Information Systems*, Vol. 31, No. 6, pp.541–567.
- Agrawal, R., Gupta, A. and Sarawagi, S. (1997) 'Modeling multidimensional databases', in *13th Int. Conf. on Data Engineering*, pp.232–243.
- Bédard, Y., Merrett, T. and Han, J. (2001) '3 Fundamentals of spatial data warehousing for geographic knowledge discovery', *Geographic Data Mining and Knowledge Discovery*, Vol. 2, p.53.
- Bimonte, S., Tchounikine, A. and Geocube, M. (2006) 'A multidimensional model and navigation operators handling complex measures: application in spatial OLAP', *Advances in Information Systems*, pp.100–109.
- Bimonte, S., Tchounikine, A., Miquel, M. and Pinet, F. (2010) 'When spatial analysis meets OLAP: multidimensional model and operators', *Int. Journal of Data Warehousing and Mining (IJDWM)*, Vol. 6, No. 4, pp.33–60.
- Blaschka, M., Sapia, C., Höfling, G. and Dinter, B. (1998) 'Finding your way through multidimensional data models', in *9th Int. Workshop on Database and Expert Systems Applications*, p.198.
- Franconi, E. and Kamble, A. (2004) 'A data warehouse conceptual data model', in *Scientific and Statistical Database Management*, pp.435–436.
- Ferri, F., Pourabbas, E., Rafanelli, M. and Ricci, F. (2002) 'Extending geographic databases for a query language to support queries involving statistical data', in *Int. Conf. on Scientific and Statistical Database Management*, IEEE, pp.220–230.
- GeoMondrian Project (2011) Available at <http://www.spatialytics.org/projects/geomondrian/> (accessed on 12 October 2011).
- Golfarelli, M., Maio, D. and Rizzi, S. (1998) 'The dimensional fact model: a conceptual model for data warehouses', *Int. Journal of Cooperative Information Systems*, Vol. 7, pp.215–247.
- Gomez, L., Haesevoets, S., Kuijpers, B. and Vaisman, A. (2009) 'Spatial aggregation: data model and implementation', *Information Systems*, Vol. 34, No. 6, pp.551–576.
- Gray, J., Bosworth, A., Layman, A. and Pirahesh, H. (1996) 'Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals', *Int. Conf. on Data Engineering*, No. 152.
- Guting, R. and Schneider, M. (1995) 'Realm-based spatial data types: the ROSE algebra', *The VLDB Journal*, Vol. 4, No. 2, pp.243–286.
- Guting, R., De Ridder, T. and Schneider, M. (1995) 'Implementation of the ROSE algebra: efficient algorithms for realm-based spatial data types', in *Advances in Spatial Databases*, Springer, pp.216–239.
- Han, J. and Kamber, M. (2006) *Data Mining: Concepts and Techniques*, Morgan Kaufmann.
- Han, J., Koperski, K. and Stefanovic, N. (1997) 'GeoMiner: a system prototype for spatial data mining', in *ACM SIGMOD Int. Conf. on Management of Data*, pp.553–556, ACM.

- Han, J., Stefanovic, N. and Koperski, K. (1998) 'Selective materialization: an efficient method for spatial data cube construction', in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, pp.144–158.
- Hüsemann, B., Lechtenböcker, J. and Vossen, G. (2000) 'Conceptual data warehouse design', in *Workshop on Design and Management of Data Warehouses*, pp.3–9.
- Inmon, W. (2005) *Building the Data Warehouse*, Wiley.
- Java Topology Suite (JTS) (2011) Available at <http://www.vividsolutions.com/jts/> (accessed on 12 October 2011).
- Jensen, C., Kligys, A., Pedersen, T. and Timko, I. (2004) 'Multidimensional data modeling for location-based services', *The Int. Journal on Very Large Data Bases (VLDBJ)*, Vol. 13, No. 1, pp.1–21.
- Joint Typhoon Warning Centre (JTWC) (2011) Available at <http://metocph.nmci.navy.mil/jtwc> (accessed on 12 October 2011).
- Kamble, A. (2008) 'A conceptual model for multidimensional data', in *5th Asia-Pacific Conf. on Conceptual Modelling*, Vol. 79, pp.29–38.
- Kimball, R. (1997) 'A dimensional modeling manifesto', *DBMS Magazine*, Vol. 10, No. 9, pp.58–70.
- Kimball, R. and Ross, M. (2002) *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, 2nd ed., Wiley & Sons, New York.
- Lenz, H. and Shoshani, A. (1997) 'Summarizability in OLAP and statistical databases', in *Proceedings of the Intl. Conf. on Scientific and Statistical Database Management Conference (SSDBM)*, p.132, IEEE Computer Society.
- Luján-Mora, S., Trujillo, J. and Song, I. (2006) 'A UML profile for multidimensional modeling in data warehouses', *Data Knowledge Engineering*, Vol. 59, No. 3, pp.725–769.
- Malinowski, E. and Zimányi, E. (2004a) 'OLAP hierarchies: a conceptual perspective', in *Advanced Information Systems Engineering*, pp.19–35, Springer.
- Malinowski, E. and Zimányi, E. (2004b) 'Representing spatiality in a conceptual multidimensional model', in *12th ACM Int. workshop on Geographic Information Systems*, pp.12–22, ACM.
- Malinowski, E. and Zimányi, E. (2005) 'Spatial hierarchies and topological relationships in the spatial MultiDimER model', in *BNCOD*, pp.17–28.
- Malinowski, E. and Zimányi, E. (2006) 'Hierarchies in a multidimensional model: from conceptual modeling to logical representation', *Data Knowledge Engineering*, Vol. 59, No. 2, pp.348–377.
- Marchand, P., Brisebois, A., Bédard, Y. and Edwards, G. (2004) 'Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis', *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 59, Nos. 1–2, pp.6–20.
- National Hurricane Centre (NHC) – Season Archives (2011) Available at <http://www.nhc.noaa.gov/pastall.shtml> (accessed on 12 October 2011).
- National Oceanic and Atmospheric Administration (NOAA) (2011) Available at <http://www.aoml.noaa.gov/hrd> (accessed on 12 October 2011).
- Niemi, T., Nummenmaa, J. and Thanisch, P. (2001) 'Logical multidimensional database design for ragged and unbalanced aggregation hierarchies', in *Int. Workshop on Design and Management of Data Warehouses. Interlaken, Switzerland*, Citeseer.
- Open GIS Consortium: Reference Model (2011) Available at <http://openlayers.org> (accessed on 12 October 2011).
- OpenLayers Mapping Client (2011) Available at <http://openlayers.org> (accessed on 12 October 2011).

- Pedersen, T. and Jensen, C. (2002) 'Multidimensional database technology', *Computer*, Vol. 34, No. 12, pp.40–46.
- Pedersen, T., Jensen, C. and Dyreson, C. (2001) 'A foundation for capturing and querying complex multidimensional data', *Information Systems*, Vol. 26, No. 5, pp.383–423.
- Pentaho Analysis Services: Mondrian Project (2011) Available at <http://mondrian.pentaho.org/> (accessed on 12 October 2011).
- Prat, N., Akoka, J. and Wattiau, I. (2006) 'A UML-based data warehouse design method', *Decision Support Systems*, Vol. 42, No. 3, pp.1449–1473.
- Rigaux, P., Scholl, M. and Voisard, A. (2002) *Introduction to Spatial Databases: With Application to GIS*, Morgan Kaufmann, San Francisco, CA.
- Rivest, S., Bedard, Y. and Marchand, P. (2001) 'Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing (SOLAP)', *Geomatica*, Vol. 55, No. 4, pp.539–555.
- Rivest, S., Bédard, Y., Proulx, M., Nadeau, M., Hubert, F. and Pastor, J. (2005) 'SOLAP technology: merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data', *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 60, No. 1, pp.17–33.
- Ruiz, C. and Times, V. (2009) 'A taxonomy of SOLAP operators', *XXIV Simpósio Brasileiro de Banco de Dados, Fortaleza, CE*.
- Sapia, C., Blaschka, M., Höfling, G. and Dinter, B. (1999) 'Extending the E/R model for the multidimensional paradigm', in *ER '98: Workshops on Data Warehousing and Data Mining*, pp.105–116, Springer-Verlag.
- Schneider, M. (1997) *Spatial Data Types for Database Systems – Finite Resolution Geometry for Geographic Information Systems*, Vol. LNCS 1288, Springer-Verlag, Berlin Heidelberg.
- Schneider, M. and Behr, T. (2006) 'Topological relationships between complex spatial objects', *ACM Transactions on Database Systems (TODS)*, Vol. 31, No. 1, pp.39–81.
- Scotch, M. and Parmanto, B. (2005) 'SOVAT: spatial OLAP visualization and analysis tool', in *38th Hawaii Int. Conf. on System Sciences (HICSS)*, p.142b, IEEE.
- Shekhar, S. and Chawla, S. (2003) *Spatial Databases: A Tour*, Prentice Hall, Upper Saddle River, New Jersey.
- Shekhar, S., Lu, C., Tan, X., Chawla, S. and Vatsavai, R. (2001) 'MapCube: a visualization tool for spatial data warehouses', *Geographic Data Mining and Knowledge Discovery*, p.73.
- Stefanovic, N., Han, J. and Koperski, K. (2002) 'Object-based selective materialization for efficient implementation of spatial data cubes', *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 12, No. 6, pp.938–958.
- Tryfona, N., Busborg, F. and Christiansen, J. (1999) 'starER: a conceptual model for data warehouse design', in *ACM 2nd Int. Workshop on Data Warehousing and OLAP*, pp.3–8.
- Tsois, A., Karayannidis, N. and Sellis, T. (2001) 'MAC: conceptual data modeling for OLAP', in *Int'l. Workshop on the Design and Management of Data Warehouses (DMDW)*, pp.28–55. Citeseer.
- Vassiliadis, P. and Sellis, T. (1999) 'A survey of logical models for OLAP databases', *SIGMOD Record*, Vol. 28, No. 4, pp.64–69.
- Viswanathan, G. and Schneider, M. (2010) 'BigCube: a MetaModel for managing multidimensional data', in *Proceedings of the 19th Int. Conf. on Software Engineering and Data Engineering (SEDE)*, pp.237–242.
- Viswanathan, G. and Schneider, M. (2011) 'Requirements for spatial data warehousing and SOLAP', in *Proceedings of the 16th Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, pp.144–155, Springer-Verlag.

Zepeda, L., Celma, M. and Zatarain, R. (2008) 'A mixed approach for data warehouse conceptual design with MDA', in *Int. Conf. on Computational Science and its Applications*, pp.1204–1217.