On the Requirements for User-Centric Spatial Data Warehousing and SOLAP

Ganesh Viswanathan & Markus Schneider

Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL 32611, USA
`{gv1,mschneid}@cise.ufl.edu`

Abstract. Data warehouses and OLAP systems help to analyze complex multidimensional data and provide decision support. With the availability of large amounts of spatial data in recent years, several new models have been proposed to enable the integration of spatial data in data warehouses and to help analyze such data. This is often achieved by a combination of GIS and spatial analysis tools with OLAP and database systems, with the primary goal of supporting *spatial analysis dimensions*, *spatial measures* and *spatial aggregation operations*. However, this poses several new challenges related to spatial data modeling in a multidimensional context, such as the need for new spatial aggregation operations and ensuring consistent and valid results. In this paper, we review the existing modeling strategies for spatial data warehouses and SOLAP in all three levels: conceptual, logical and implementation. While studying these models, we gather the most essential requirements for handling spatial data in data warehouses and use insights from spatial databases to provide a "meta-framework" for modeling spatial data warehouses. This strategy keeps the *user as the focal point* and achieves a *clear abstraction* of the data for all stakeholders in the system. Our goal is to make analysis more user-friendly and pave the way for a clear conceptual model that defines new multidimensional *abstract data types* (ADTs) and operations to support spatial data in data warehouses.

1 Introduction

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. They contain large repositories of analytical and subject-oriented data, integrated from several heterogeneous sources over a historical time-line [1, 2]. 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 lead to increased challenges in storing such information and analyzing them. Data warehouses 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 large, subject-oriented, repositories of data integrated from a variety sources of a long timeline with native support for spatial objects and advanced 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,"* or map generalization 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 containing all the college towns where more than 2500 units of Kinect were sold in 2010"*. 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.

OLAP operations are often categorized as distributive, algebraic and holistic [3, 4], 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 modeling approaches for spatial data warehousing (Section 2) we found that one of the major shortcomings of existing models is the heavy focus on direct ad-hoc implementation strategies such as a combination of OLAP tools or GIS mapping clients with databases to create a pseudo spatial data warehouse. However, for effective multidimensional data modeling 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) [2], multidimensional OLAP (MOLAP) [1] and hybrid OLAP (HOLAP) [5] is also needed to help complete the design of the spatial data warehouse. Overall, this paper provides

a new insight into the fundamental requirements for designing a user-friendly spatial data warehouse model by providing an objective analysis of the essential requirements for it.

The rest of this paper is organized as follows. Section 2 provides a review of existing literature regarding data warehouse models and OLAP, spatial data modeling, spatial data warehouse models, and user interface tools used to design these systems. Section 3 discusses the essential requirements for spatial data warehouses and OLAP. Section 4 presents our meta-framework as a path for developing an enhanced conceptual model based on the cube metaphor that is 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 modeling 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 spatial data warehouse (Section 4). A survey of the state-of-the-art in each of these domains is the topic of the current section.

Over the past decade several approaches have been proposed for modeling data warehouses. Now, we present a study of the best available conceptual and logical models for data warehousing. Existing conceptual modeling approaches can be broadly classified into *Extensions of Entity Relationship (E/R) models* ([6–10]), *Extensions of Unified Modeling Language (UML)* ([11–13]) and *Ad-hoc* ([14–17]) 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 [4] 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. [18], Vassiliadis et.al. [19] and Pedersen et.al. [20]. Though many of these models aid in the relational representation of aggregate data, contributions like the ALL operator and hierarchies are essential even from a multidimensional perspective. One of the earliest approaches for multidimensional modeling was introduced by Kimball in [21]. This *dimensional modeling* 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 modeling imposes some rules on the modeling but results in a data model that has the access methods defined clearly by virtue of the relationships [21, 1]. 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. in [22] 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.
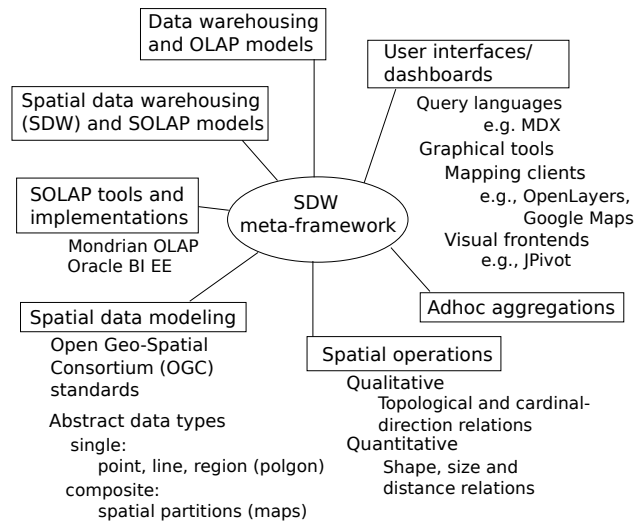
Fig. 1. *An illustration of the various domains considered during the design of the spatial data warehouse meta-framework.*

Spatial data warehousing (SDW) has become a topic of growing interest 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 spatial OLAP tools are now available to help model and analyze such data.

An early approach to spatial online analytical processing (SOLAP) is [23], which mentions essential SOLAP features classified into three areas of requirements. The first is to enable *data visualization* via cartographic (maps) and non-cartographic displays (e.g., 2D tables), numeric data representation and the visualization of context data. Second, *data exploration* requires multidimen-

sional navigation on both cartographic and non-cartographic displays, filtering on data dimensions (members) and support for calculated measures. The third area discussed involves the *structure of the data*, for example, the support for spatial and mixed data dimensions, support for storage of geometric data over an extended time period, etc. The conceptual design models for spatial data warehouses are extensions of E/R and UML diagrams or ad-hoc design approaches. Among extensions of E/R models, [24] presents a clear integration of spatial data for OLAP by extending the MultiDimER and MADS approaches. Among other ad-hoc design approaches, [25] presents a formal framework to integrate spatial and multidimensional databases by using a *full containment* relationship between the hierarchy levels. In [26], the formal model from [20] is extended to support spatially overlapping hierarchies by exploiting the partial containment relations among data levels, thus leading to a more flexible modeling strategy. Bimonte et.al. [27, 28] present the GeoCube model for spatial datawarehouse design, based on a formal schema and instance definition for cube elements. GeoCube also extends conventional SOLAP operations with five new operations namely, *classify*, *specialize*, *permute*, *OLAP-buffer* and *OLAP-overlay*. However, one of the shortcomings of this approach is the use of n-n mappings between data-dimensions and facts. Since each cell of the data cube is a unique catesian product of the associated data dimensions, this n-n mapping weakens the cube structure and makes it difficult to apply constraints and dynamic schema changes during OLAP operations.

The logical SDW design models aim to provide support for spatial data dimensions [29], spatial measures [30–33] and spatial aggregations [34]. The concept of spatial measures (with a specific geometric part) is either defined as references to spatial objects [35, 23] or as the results of topological, distance or metric operations [24, 23], or as values associated with a spatial data dimension in the data cube [31, 36]. 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 [37]. This is a very useful feature for supporting spatial aggregation operations and map generalizations (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 [38]. Sekhar et.al. [32] extend the MapCube operator to support spatial data and aggregations, but the model is rather constrained and not easily extendable for user-defined queries.

The major implementations of SOLAP tools can be broadly classified as *OLAP dominant*, *GIS dominant*, or *integrated OLAP and GIS* solutions [33]. 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 [31, 29, 27]. The GeoMondrian [39] project aims to develop an opensource implementation of a SOLAP analysis server. Currently, it provides a spatially enabled version of the Mondrian OLAP server [40]. However, it is unclear if GeoMondrian has a clear underlying spatial data model with SOLAP operators.

It seems to be essentially built ad-hoc, by using a combination of the Java Topology Suite [41] (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.

For modeling spatial data there are now several established approaches in the database community. An introduction to basic spatial data types is given in [42]. The ROSE algebra [43, 44] provides a more robust discussion of spatial data types by introducing types such as *point*, *line* and *region* 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 [45] as a standard for a representing geo-spatial information. Qualitative spatial operations include topological relations [46] 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 analyze spatial data in the data warehouse.

3 Requirements for User-Centric Spatial OLAP

For a data warehouse model to be effective in modeling, storing and querying data there are several essential requirements. Blaschka et.al. [18] provide a list of requirements for multidimensional modeling for OLAP applications. In [20], Pedersen et.al. present eleven requirements for a multidimensional model, using a clinical data warehousing application as an example and then present a formal model that accommodates those requirements. By studying the popular models for multidimensional data (Section 2) and several new applications, we now compile a list of basic requirements for an effective user-centric spatial data warehouse model.

The basic requirements for an effective multidimensional model for data warehouse design can be summarized as follows:

1) *Simple user view*: The user view of the data warehouse must be simple and intuitive, yet capable of capturing the full dimensionality of the data. This user view should be independent of implementation aspects. Thus, for e.g., the use of facts and dimension *tables* (thereby exposing ROLAP implementation) should be avoided in the user view. Instead an abstract view, like a *data cube* with available OLAP querying operations on it would make it easier for users to navigate across data dimensions and perform analysis.

2) *Implementation independent conceptual design*: The conceptual model for the SDW should also be free from implementation aspects (such as star schema design) to be able effectively model data for the analyst's needs.

3) *Separation of structure and values*: There should be an explicit separation of schema and instances, i.e., structure and values. A formal conceptual data model should allow for independence between specification and implementation, and the ability to alter implementations without affecting the user's view of the OLAP system.

4) *Explicit hierarchies*: Hierarchies (with several levels of member or measure categories) should be supported explicitly in the *data dimensions* and *facts*.

5) *Multiple hierarchies*: Multiple hierarchies along the data dimensions and even measure values should be supported.

6) *Descriptive attributes*: Thematic or descriptive attributes for members and measures (geometric or otherwise) should be supported for enabling selection, navigation and aggregation queries over them during analysis.

7) *Support for attribute aggregation*: The model must provide good support for aggregation on both geometric and thematic attributes apart from metric computations.

8) *Complex measures*: The model should support multiple (composite) and complex members and measures. Example, a cell in the cube can include several measures such as sales quantity and profit. Location can be a complex object such as a *polygon* representing Italy with the Vatican as a *hole* object inside it.

9) *Handling different levels of granularity*: The model should be able to handle data with multiple levels of granularity (dynamic multi-level hierarchies.)

10) *Support for irregular hierarchies*: There must be support for non-conformant (non-onto, non-strict and ragged) hierarchies and generalization/ specialization (is-a) relationships.

11) *User-defined aggregates*: User defined aggregation functions should be supported. These may even include ad-hoc operations such as *ratio* (metric) and multi-level buffer (geometric) operations.

12) *Handling data imprecision/ summarizability*: The model should be able to handle data imprecision so that double-counting of data is avoided, and non-additive data are not summarized.

13) *Handling uncertainty*: The model should also be able to handle the uncertainty in the data using techniques such as *data lineage tracking*.

14) *Handling change over time*: The model should be able to handle updates and deletions over time. Any re-calculations of measure values should be consistent and correct.

15) *Multiple facts/ cube schema*: The model should allow for multiple facts, data dimensions and data cubes to be present in the schema.

16) *Drill-across capability*: The model should support drilling across dimensions, i.e., sharing of dimensions among different fact cubes.

17) *Drill-through capability*: The model should support drilling through capability to be able to query the base level (raw) data.

18) *Dimensionless aggregation*: There should be support for aggregations along attributes that are not part of the data dimensions or hierarchies themselves, such as thematic attributes.

19) *Measureless aggregation*: Aggregation on thematic attributes of facts (measures) should be supported.

20) *Online aggregation*: The model should allow for multiple levels of *online* aggregation, i.e., dynamic, multi-level query design.

21) *Support for spatial hierarchies*: The model should support generalization and specialization hierarchies on spatial objects. This would, for example, enable roll-up operations from *cities* to *states* and *country* level in the *location* hierarchy. The linking of spatial hierarchies to thematic attribute hierarchies should also be allowed.

22) *Support for spatial dimensions*: Data dimensions and hierarchies should natively support spatial data types and operations on them such as roll-up and drill-down on spatial hierarchies (for example, from *state* to *city* level).

23) *Support for spatial measures*: The data cube should be capable of storing and managing spatial measures as both simple, complex and a composite (map) spatial objects.

24) *Support for spatial aggregations*: The model should explicitly support aggregation operations on the spatial measures and members. For e.g., "a *convex hull* on cities having the top-k highest sales of iPads in every state in 2010".

25) *Support for adhoc geo-spatial aggregations*: The model should support ad-hoc, user-defined geo-spatial operations on both spatial measures, members and their thematic attributes.

4   A Meta-Framework for Spatial Data Warehouse Design

After reviewing the existing data warehouse and SOLAP modeling approaches and generating the list of essential requirements for an effective spatial data warehouse model, we now provide a broad insight into how a spatial data warehouse 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 be provided by a conceptual design view that fully abstracts over the underlying implementation details. To allow users to interact with the conceptual cube an 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 be single objects such as point, line or region, or a combination of these in terms of spatial partitions or map objects. Figure 2 illustrates such a meta-framework that we propose for spatial data warehouse design.

The conceptual model should provide built-in support for spatial objects by using abstract data types (ADTs) or by extending multidimensional data types such as perspectives (data dimensions) and subjects (facts) to include spatial values. Later, additive, semi-additive and holistic classes of aggregation operations can be defined over them [3, 4, 47]. For example, in Table 1, we show examples of possible spatial aggregation operators (the second line in each of the three categories). Such a conceptual design view can be an extension of

Table 1. Examples of non-spatial and spatial aggregation operators

Type	$\mathcal{B}ig\mathcal{C}ube$ 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, Center of Gravity, Center of Mass*
Non-Additive	*Median, MostFrequent, Rank, LastNonNullValue, FirstNonNullValue, Minimum Bounding Box, Nearest Neighbor, Equi-Partition*

the $\mathcal{B}ig\mathcal{C}ube$ model [17] which provides ADTs arranged over different levels to create the conceptual cube or one of the other conceptual modeling approaches supporting spatial data analysis [26, 24, 34, 28].

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
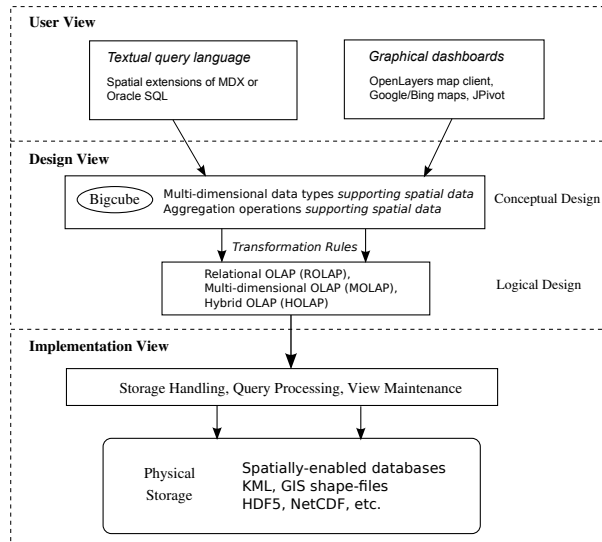
Fig. 2. *A meta-framework for spatial data warehouse design illustrating the distinct conceptual and logical design levels and the user view for OLAP analysis.*

multidimensional design strategies. For example, in-memory multidimensional arrays can be used for constructing the materialized 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 [48], Google or Bing maps or tabular representations using tools such as JPivot. A combination of these tools is often required for effective data visualization and user-friendly analysis to design multiple levels of queries.

This meta-framework can be used to represent the exact semantics of spatial aggregation operations on different view levels. For example, a query such as "*find all adjacent states where more than 5000 iPhone units where sold in 2010*" leads to a selection on thematic attributes followed by a test for the *meet* topological relation on the spatial partitions to generate the required results.

5   Conclusions and Future Work

In this paper, we present a broad overview of existing conceptual, logical and implementation strategies for spatial data warehouses (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 modeling spatial data warehouses. 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 materialized 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 visualizations. Overall, this provides a comprehensive view of the existing state-of-the-art in spatial data warehouse modeling and lays the foundation for describing the exact semantics of SOLAP operations on data cubes and to develop a complete spatial data warehousing solution.

References

1. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition). (2002)
2. Inmon, W.: Building the data warehouse. Wiley (2005)
3. Han, J., Kamber, M.: Data mining: concepts and techniques
4. Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Int. Conf. on Data Engineering **0** (1996) 152
5. Pedersen, T., Jensen, C.: Multidimensional database technology. Computer **34**(12) (2002) 40–46
6. Franconi, E., Kamble, A.: A data warehouse conceptual data model. In: Proc. of Scientific and Statistical Database Management. (2004) 435–436

7. Kamble, A.: A conceptual model for multidimensional data. In: 5th Asia-Pacific Conf. on Conceptual Modelling. Volume 79. (2008) 29–38

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

9. Malinowski, E., Zimányi, E.: Hierarchies in a multidimensional model: from conceptual modeling to logical representation. Data Knowledge Engineering **59**(2) (2006) 348–377

10. Tryfona, N., Busborg, F., Christiansen, J.: starER: A conceptual model for data warehouse design. In: Proc. of ACM 2nd Int. Workshop on Data Warehousing and OLAP. (1999) 3–8

11. Abelló, A., Samos, J., Saltor, F.: YAM^2: a multidimensional conceptual model extending UML. Information Systems **31** (2006)

12. Luján-Mora, S., Trujillo, J., Song, I.: A UML profile for multidimensional modeling in data warehouses. Data Knowledge Engineering **59**(3) (2006) 725–769

13. Prat, N., Akoka, J., Wattiau, I.: A UML-based data warehouse design method. Decision Support Systems **42**(3) (2006) 1449–1473

14. Golfarelli, M., Maio, D., Rizzi, S.: The Dimensional Fact Model: a Conceptual Model for Data Warehouses. Int. Journal of Cooperative Information Systems **7** (1998) 215–247

15. Hüsemann, B., J.Lechtenbörger, G.Vossen: Conceptual Data Warehouse Design. In: Workshop on Design and Management of Data Warehouses. (2000) 3–9

16. Zepeda, L., Celma, M., Zatarain, R.: A Mixed Approach for Data Warehouse Conceptual Design with MDA. In: Int. Conf. on Computational Science and Its Applications. (2008) 1204–1217

17. Viswanathan, G., Schneider, M.: BigCube: A MetaModel for Managing Multidimensional Data. In: Proceedings of the 19th Int. Conf. on Software Engineering and Data Engineering (SEDE). (2010) 237–242

18. Blaschka, M., Sapia, C., Höflng, G., Dinter, B.: Finding Your Way through Multidimensional Data Models. In: 9th Int. Workshop on Database and Expert Systems Applications. (1998) 198

19. Vassiliadis, P., Sellis, T.: A survey of logical models for OLAP databases. SIGMOD Record **28**(4) (1999) 64–69

20. Pedersen, T., Jensen, C., Dyreson, C.: A foundation for capturing and querying complex multidimensional data. Information Systems **26**(5) (2001) 383–423

21. Kimball, R.: A dimensional modeling manifesto. DBMS Magazine **10**(9) (1997) 58–70

22. Agrawal, R., Gupta, A., Sarawagi, S.: Modeling Multidimensional Databases. In: Proceedings of the 13th Int. Conf. on Data Engineering. (1997) 232–243

23. Rivest, S., Bedard, Y., Marchand, P.: Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing (SOLAP). Geomatica-Ottawa **55**(4) (2001) 539–555

24. Malinowski, E., Zimányi, E.: Representing spatiality in a conceptual multidimensional model. In: Proceedings of the 12th annual ACM Int. workshop on Geographic information systems, ACM (2004) 12–22

25. Ferri, F., Pourabbas, E., Rafanelli, M., Ricci, F.: Extending geographic databases for a query language to support queries involving statistical data. In: Int. Conf. on Scientific and Statistical Database Management, IEEE (2002) 220–230

26. Jensen, C., Kligys, A., Pedersen, T., Timko, I.: Multidimensional data modeling for location-based services. The Int. Journal on Very Large Data Bases (VLDBJ) **13**(1) (2004) 1–21

27. Bimonte, S., Tchounikine, A., Miquel, M.: Geocube, a multidimensional model and navigation operators handling complex measures: Application in spatial olap. Advances in Information Systems (2006) 100–109

28. Bimonte, S., Miquel, M.: When spatial analysis meets olap: Multidimensional model and operators. IJDWM **6**(4) (2010) 33–60

29. Scotch, M., Parmanto, B.: SOVAT: Spatial OLAP visualization and analysis tool. In: Proceedings of the 38th Annual Hawaii Int. Conf. on System Sciences (HICSS), IEEE (2005) 142b

30. Han, J., Koperski, K., Stefanovic, N.: GeoMiner: a system prototype for spatial data mining. In: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, ACM (1997) 553–556

31. Marchand, P., Brisebois, A., Bédard, Y., Edwards, G.: Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis. ISPRS journal of photogrammetry and remote sensing **59**(1-2) (2004) 6–20

32. 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) 73

33. Rivest, S., Bédard, Y., Proulx, M., Nadeau, M., Hubert, F., Pastor, J.: SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. ISPRS journal of photogrammetry and remote sensing **60**(1) (2005) 17–33

34. Gomez, L., Haesevoets, S., Kuijpers, B., Vaisman, A.: Spatial aggregation: Data model and implementation. Information Systems **34**(6) (2009) 551–576

35. Stefanovic, N., Han, J., Koperski, K.: Object-based selective materialization for efficient implementation of spatial data cubes. Knowledge and Data Engineering, IEEE Transactions on **12**(6) (2002) 938–958

36. Han, J., Stefanovic, N., Koperski, K.: Selective materialization: An efficient method for spatial data cube construction. In: In Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'98. (1998) 144–158

37. Rigaux, P., Scholl, M., Voisard, A.: Introduction to spatial databases: with application to GIS. Morgan Kaufmann (2002)

38. Malinowski, E., Zimányi, E.: Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER Model. In: BNCOD. (2005) 17–28

39. GeoMondrian Project: (Dec 2010) http://www.spatialytics.org/projects/geomondrian/.

40. Pentaho Analysis Services: Mondrian Project: (Dec 2010) http://mondrian.pentaho.org/.

41. Java Topology Suite (JTS): (Dec 2010) http://www.vividsolutions.com/jts/.

42. Shekhar, S., Chawla, S.: Spatial databases: a tour. Prentice Hall (2003)

43. Guting, R., Schneider, M.: Realm-based spatial data types: the ROSE algebra. The VLDB Journal **4**(2) (1995) 243–286

44. Guting, R., De Ridder, T., Schneider, M.: Implementation of the ROSE algebra: Efficient algorithms for realm-based spatial data types. In: Advances in Spatial Databases, Springer (1995) 216–239

45. Open GIS Consortium: Reference Model: (Dec 2010) http://openlayers.org.

46. Schneider, M., Behr, T.: Topological relationships between complex spatial objects. ACM Transactions on Database Systems (TODS) **31**(1) (2006) 39–81

47. Ruiz, C., Times, V.: A taxonomy of solap operators. XXIV Simpósio Brasileiro de Banco de Dados, Fortaleza, CE (2009)

48. OpenLayers mapping client: (Dec 2010) http://openlayers.org.