



HAL
open science

Conceptual design and implementation of spatial data warehouses integrating regular grids of points

S. Bimonte, M. Zaamoune, P. Beaune

► To cite this version:

S. Bimonte, M. Zaamoune, P. Beaune. Conceptual design and implementation of spatial data warehouses integrating regular grids of points. *International Journal of Digital Earth*, 2017, 10 (9), pp.901-922. 10.1080/17538947.2016.1266040 . hal-01872583

HAL Id: hal-01872583

<https://hal.science/hal-01872583>

Submitted on 12 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conceptual design and implementation of spatial data warehouses integrating regular grids of points

Sandro Bimonte (a), Mehdi Zaamoune (a) and Philippe Beaune (b)

a-Technologies and information systems for agricultural systems Departement, Irstea, Aubiere, France;

b- Agaetis, Clermont-Ferrand, France

Abstract. Spatial OLAP (SOLAP) and spatial data warehouse (SDW) systems are geobusiness intelligence technologies that enable the analysis of huge volumes of geographic data. In the last decade, the conceptual design and implementation of SDWs that integrate spatial data, which are represented using the vector model, have been extensively investigated. However, the integration of field data (a continuous representation of spatial data) in SDWs is a recent unresolved research issue. Enhancing SDWs with field data improves the spatio-multidimensional analysis capabilities with continuity and multiresolutions. Motivated by the need for a conceptual design tool and ROLAP implementation, we propose a UML profile for SDWs that integrates a regular grid of points and supports continuity and multiresolutions. We also propose an efficient implementation of a ROLAP architecture.

1. Introduction

Data warehouses (DWs) and online analytical processing (OLAP) systems (such as Microsoft Analysis Services and Oracle BI) are business intelligence technologies that support decision makers in the online analysis of huge volumes of data (Inmon, 2005) (Kimball, 1996).

In the era of spatial big data (Shekar et al., 2012), a greater amount of spatial data is collected using new technologies, such as satellites, sensor networks, and simulation models. To take advantage of important analysis capabilities that are associated with these data, geographic information system (GIS) functionalities have been integrated into DWs and OLAP systems, which have produced the concepts of a spatial data warehouse (SDW) and spatial OLAP (SOLAP) (Stefanovic et al., 2000) (Bédard et al., 2001). Warehoused spatial data are modeled according to a spatio-multidimensional model, which defines the concepts of spatial dimensions (the analysis axes) and spatial measures (the analysis subjects) (Malinowski et al., 2008). Warehoused (spatial) data are analyzed using SOLAP systems, which have been defined by Yvan Bédard as "*Visual platforms built especially to support rapid and easy spatiotemporal analysis and exploration of data, following a multidimensional approach, comprised of aggregation levels, available in cartographic displays as well as in tabular and diagram displays*" (Bédard, 1997). The integration of OLAP and GIS functionalities enables the use of spatial analysis operators and cartographic representations of SOLAP query results (Bimonte, 2010). SOLAP systems have demonstrated their effectiveness in several application domains, such as marketing, health (Bernier et al., 2009), and agriculture (Bimonte, 2015). SOLAP systems and GISs are complementary technologies because GISs are well adapted for operational tasks and complex spatial analysis using small spatial data sets, whereas SOLAP is suitable for spatial analysis based on the exploration and summarization of massive spatial data sets.

Existing spatial OLAP systems (such as Map4Decision and Oracle BI) (Bimonte, 2010) (Golfarelli et al., 2013) (Bimonte, 2014) enable the management of spatial data using a vector representation.

Geographic information can be represented according to two models: the discrete (vector) model and the field model (Tomlin, 1990) (O'Sullivan et al., 2002). The vector model is used to model human geographic concepts, such as regions and buildings. The field model is used for geographic phenomena that continuously change in space, such as temperature and pollution. (Couclelis, 1992). Some analysis operators that differ from the analysis operators of

the vector model (i.e., topological operators) have been proposed for fields because they enable a point-by-point analysis (i.e., map algebra (Tomlin, 1990) (Ledoux et al., 2006)). Note that the distinction between the field and the vector is primarily a distinction of convenience and perspective because movement from one representation to a second representation according to the analysis needs is easy (Camara et al., 1995). To implement the *continuity* of fields, two categories of representations of fields have been proposed in the literature: the incomplete representation and the complete representation (Laurini et al., 1992). The first representation involves assigning a numerical value to each point of the space. For the second representation, the numerical values are assigned to exact locations in space, and additional spatial interpolation functions (O'Sullivan et al., 2002) are used to calculate the values for non-sampled locations. Sometimes, the analysis of a geographical phenomenon, such as territorial changes (Plumejeaud et al., 2011), at different levels of detail (LOD) (resolutions, scales and thematic granularities) is necessary because it explains unknown trends and patterns (Timpf, et al., 1992). Thus, several authors have investigated issues related to the fields and LOD by proposing logical and physical data models and analysis techniques in the context of GIS and spatial database management systems (SDBMSs) (Laurini et al., 2004).

The integration of field data in SOLAP systems offers significant analytical capabilities (Gomez et al., 2010) (Gómez et al., 2013). Indeed, several application domains such as climate change, natural disasters, energy sources, agriculture, and urban planning can be analyzed using field data, which comes from satellite images and/or simulation models.

However, this topic has not been extensively investigated. Few recent studies investigate the extension of the spatio-multidimensional model and SOLAP operators to manage the complete and incomplete representations of fields. In the same manner, the management of LOD spatial vector data in the multidimensional model is proposed in some studies (Bédard et al., 2002) (Concepción et al., 2009).

However, *no work addresses the representation of an incomplete field at different resolutions at the conceptual and logical levels in a Relational OLAP-GIS architecture* for the following main reasons: (i) spatial members usually present only not varying attributes (member properties) (Malinowsky et al., 2008), whereas *continuity* can be considered to be a attribute that change in space, (ii) *multiresolution* structures do not adequately correspond with spatial hierarchies. Indeed spatial hierarchies are defined using the topological inclusion relationship (Malinowsky et al., 2008), while multiresolution hierarchies (Concepción et al., 2009) correspond more to versioning hierarchies (Body et al., 2003). Although they are easily associated with versioning hierarchies (Body et al., 2003), their implementation in existing (S)OLAP systems is difficult.

Encouraged by the need for standards in SDW and SOLAP systems, we present two new SDW logical models in our previous preliminary studies (Zaamoune et al., 2013). The proposed logical models extend the well-known relational DW star-schema (Kimball, 1996) to represent multiresolution regular grids of points and simultaneously provide efficient storage and computation performance. The use of SQL, which is the standard of relational database management systems (DBMS), enables the implementation of our proposal in any relational SOLAP architecture and reduces time and money-consuming efforts related to the acquisition of new technological skills.

In this paper, we extend (Zaamoune et al., 2013) in different ways to effectively integrate regular grids of points in SOLAP.

Contribution i. Motivated by the need for conceptual models (such as UML and ER) for SOLAP applications as effective tools that enable DW experts and decision makers to exchange using a common, non-ambiguous, simple and readable language (Torlone, 2003), we propose an extension of the UML profile for SOLAP applications proposed by (Boulil et al., 2015) for modeling spatio-multidimensional models with multiresolution regular grids of points.

Contribution ii. We propose *FieldMDX* (Zaamoune et al., 2013)—an extension of MDX (the de-facto OLAP standard query language) (MDX, 2015)—which generates the continuity of an incomplete field with spatial interpolation functions.

Contribution iii. We validate our proposal using a real case study concerning odor monitoring, and we present its implementation in the classical UML CASE tool and relational SOLAP architecture.

The paper is organized as follows: Section 2 introduces the main concepts of SOLAP and field data; Section 3 presents the motivations of our study; Section 4 presents our UML profile, and its implementation is presented in Section 5; Section 6 presents the evaluation of our proposals, and Section 7 concludes the paper.

2. Background

2.1 Geographic information

Geographical space can be represented as a field with space- (and time-) numerical varying attributes (Tomlin, 1990). Examples of fields are temperature phenomenon and land use data. Because the number of points in a continuous representation of the space is infinite, a discretization of the analyzed space is needed to be represented in information technology systems. Therefore, some types of discretization (such as raster, TIN, and DEM) have been proposed in the literature to implement fields (Couclelis, 1992), where each type is adapted to a particular geographical phenomenon in terms of data and associated analysis operators. For example, a digital elevation model (DEM) is a digital model of the 3D representation of a terrain's surface; it involves slope analysis. These representations can also be classified in complete and incomplete representations regardless of whether they assign a value to each point of the space. As shown in Figure 1, a raster is an example of a field that employs the complete representation (Figure 1a), whereas a TIN is an incomplete representation (Figure 1b).

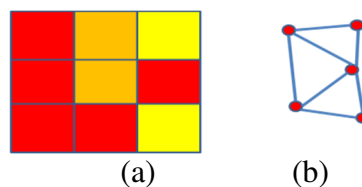


Figure 1. Examples of incomplete and complete representations of continuous fields: a) raster and b) TIN

These representations are well supported by GISs such as ArcGIS and QuantumGIS, which provide ad-hoc storage models and a set of analysis tools. In the context of Spatial DBMS, several academic efforts have been conducted to provide models (Ferreira et al., 2014) and query languages (Laurini et al., 2004) that can handle fields using relational and the object-oriented models. Existing SDBMSs, such as Oracle Spatial and PostGIS (PostGIS, 2015), support only the raster representation by providing DDL/DQL SQL extensions.

The most common operators for fields are Tomlin's map algebra operators (Tomlin, 1990). Map algebra operators were originally defined for raster data but were subsequently extended with new operators (Cordeiro et al., 2009) for other types of representations, such as Voronoi (Ledoux et al., 2006), and spatio-temporal data (Mennis et al., 2005). Map algebra is a language that enables field calculations using mathematical operators. Usually, three types of functions are defined:

- Local functions: The measure value of a point is calculated from the values of the same point of more input maps.

- Focal functions: The measure value of a point is calculated using the values of its neighborhood points.
- Zonal functions: The measure value of a point is calculated using the values of points in a particular zone.

For example, the use of the local map algebra operator enables the calculation of the monthly average temperature map from the daily temperature maps by applying the average aggregation function point-by-point to all temperature values of each day of the month.

Because fields are used to represent continuous spatial phenomena, their modeling must consider the fact that every phenomenon has several levels of detail that represent different perspectives perceived by decision makers in their analysis (Timpf, et al., 1992). A decision maker may require different levels of detail to obtain the best perspective of a geographic phenomenon because each level of detail reveals different information, which explains why decision makers have to identify the appropriate levels of detail for their spatial data prior to their analysis of the data. The use of several levels of detail implies that the GIS or spatial DBMS should provide a set of methods for navigating among these levels of detail, which enables decision makers to transparently zoom-in or zoom-out in their data. An important academic study addresses storing, querying and visualizing levels of detail (LOD) data (Timpf et al., 1992). These approaches can be classified into two main groups: *i*) the *on-the-fly approach*, in which the different representations are computed in real time from an input map, and *ii*) the *pre-computation approach*, in which the different representations are pre-computed. Generally, the former improves storage costs because the new level of detail map is computed online, whereas the latter does not require online computation despite additional storage costs (Bereuter et al., 2013). In the same manner as (Laurini et al., 1992), Figure 2 shows a common GIS pyramid model, which represents different resolutions. This model divides the study space into regular quadrants at each resolution and recursively divides each quadrant for more detailed resolution.

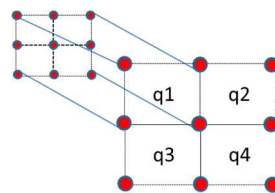


Figure 2 Pyramid model for regular grids of points

2.2 Spatial OLAP

Warehoused data are represented using a multidimensional model. The multidimensional model defines the concepts of facts and dimensions. A fact represents the subject of analysis; it is described by numerical measures, which are analyzed using different granularities that are referred to as levels of hierarchies that compose dimensions. OLAP systems explore warehoused data using OLAP operators (Inmon, 2005) (Kimball, 1996). The classical OLAP operators are as follows: Slice selects a part of the warehoused data; Roll-up aggregates measures using SQL aggregation functions along dimension hierarchies; and Drill-down is the inverse of Roll-up.

OLAP systems can be implemented using relational OLAP architecture (Kimball, 1996). Relational OLAP architecture is a multi-tier architecture, in which data are collected from different heterogeneous sources and transformed using extraction-transform-load (ETL) tools (such as Talend) before they are stored in DWs.

The data warehouse tier enables data to be stored in a DBMS (such as Oracle and PostgreSQL). Warehoused data are stored using two particular logical models: the star schema or the

snowflake schema (Kimball, 1996). The star-schema represents hierarchies of dimensions using denormalized tables that enable time-efficient “join-group by” (i.e., aggregation) queries and a table that contains measures and foreign keys to dimension tables. The snowflake schema is similar to the star schema, but it employs normalized tables for dimension hierarchies.

The OLAP server (such as Mondrian and MS Analysis Services) handles the mapping among multidimensional concepts, such as dimensions and facts, and the relational tables stored in the Data Warehouse tier. It also implements OLAP operators that are implemented using multi-dimensional expression (MDX) language (MDX, 2015), which is the de-facto standard for OLAP servers.

The OLAP client (such as Saiku) provides tabular and graphical displays for the visualization of OLAP query results. These displays enable the triggering of OLAP operators.

The integration of spatial data into OLAP systems produces the concepts of spatial data warehouse (SDW) and spatial OLAP. The SDW has been defined as “A collection of subject-oriented, integrated, non-volatile and time-variant spatial and non-spatial data to support the decision-making process” (Stefanovic et al., 2000). Warehoused spatial data are usually represented using a vector model (Bédard, 1997). Therefore, the spatio-multidimensional model redefines the concepts of spatial level and spatial measure (Malinowski et al., 2008). Typically, a spatial level extends a classical level by adding a geometrical attribute using the vector model. An example of spatial members of a spatial dimension is shown in Figure 3. Departments and regions are represented by polygons, and cities are represented by points.

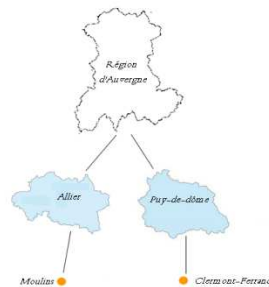


Figure 3 Example of vector spatial dimension

Adding spatial attributes to the elements of the multidimensional model enhances OLAP analysis capabilities: i) the cartographic representation of facts enables decision makers to visually analyze the spatial correlations of facts; ii) decision makers can introduce vector operators (topological, metrical and directional functions) in OLAP operators. SOLAP tools provide a unique framework for exploring and visualizing warehoused spatial data using SOLAP operators and interactive cartographic displays, respectively (Bernier et al., 2009). Common SOLAP operators include (Malinowski et al., 2008) spatial rollup and spatial drilldown, which enable navigation among spatial levels, and Spatial Slice, which enables the selection of a subset of data using vector predicates. For example, the use of a retail SOLAP application enables queries such as “Visualize the average sales per year and all cities 50 km from Paris” to be answered.

SOLAP systems integrate OLAP capacities and GIS functionalities that provide users with an effective spatio-multidimensional analysis (Bimonte, 2010). This solution is referred to as the OLAP-GIS tool.

As described in (Bimonte, 2010), the first tier of a classical relation SOLAP architecture is responsible for managing the storage of spatial data using relational Spatial DBMS with some extensions of the star and snowflake schemas. For example, (Siqueira et al., 2012) denormalizes only the geometrical attributes of the classical star schema to improve the computation performance.

The second tier is an extension of the OLAP server (SOLAP server) (such as GeoMondrian and Map4Decision) with SOLAP operators. The SOLAP server handles spatial measures and spatial dimensions, prevents redundancy when aggregating facts, and enables the calculation of materialized spatial views.

The SOLAP client (such as Map4Decision and TableauSoftware) is responsible for displaying SOLAP query results. It enables the visualization of alphanumeric and spatial data using tabular and graphic displays coupled with interactive maps.

Two other types of architecture have been developed for SOLAP solutions (Bimonte, 2010) (Bédard et al., 2001): an OLAP-dominant solution and a GIS-dominant solution. The OLAP-dominant solution reduces the GIS functionalities to a cartographic visualization, whereas the GIS-dominant solution does not integrate a SOLAP server, which is simulated in the spatial DBMS. Therefore, GIS-dominant solutions do not support navigational and calculation functionalities provided by SOLAP servers. Only OLAP-GIS tools represent effective solutions for SOLAP analysis.

SOLAP systems primarily integrate vector data because field data raise several issues, and data can be transformed from vector to field (and vice versa) for spatial analysis needs while in GISs; this transformation is not possible in SOLAP. The spatial decision-making process in GISs can be defined as a workflow of spatial analysis operators that transform spatial and alphanumeric data (Seffino et al., 1999). Several functionalities are provided to transform a field into vector data (and vice versa) because the similarities among these two models have been well identified (Câmara et al., 1995). Conversely, *in SOLAP systems, spatial data can “only” be explored* using SOLAP operators (Bédard et al., 2001). Once the SDW is designed and loaded with source data, the structures and data in the dimensions cannot be changed for two reasons: (i) immense volumes of data are stored, and ETL operations are complex and long; consequently, any data change cannot be performed online, and (ii) to accelerate SOLAP queries on warehoused data, some external data structures, such as indexes (Siqueira et al., 2012) and materialized views (Stefanovic et al., 2000), are usually employed. Any data change implies the need update data with important computation costs.

From a conceptual point of view, addressing geographic phenomena increases the complexity of a multidimensional model, as shown in (Bimonte, 2010). SOLAP applications usually present complex dimensions and aggregations, which stress the importance of a correct multidimensional design phase. Conceptual spatio-multidimensional modeling becomes a critical issue for SOLAP applications. Therefore, several studies have been proposed for spatio-multidimensional modeling using conceptual formalisms, such as ER (Malinowski et al., 2008), UML (Boulil et al., 2015), etc.), which enable a visual representation of the spatiality of the multidimensional model's elements.

3. Motivation

In this section, we present the modeling and querying requirements related to the integration of grids of points in SDWs (Sec. 3.2) using a real environmental project to investigate the odors in urban areas. The project is developed in collaboration with the French enterprise Agaetis. As stated in the previous section, conceptual models for SOLAP applications are mandatory tools in SOLAP projects. In Section 3.2, we present the main concepts of the ICSOLAP UML profile, which we extend in this paper to integrate multiresolution regular grids of points.

3.1 ICSOLAP UML Profile

“A profile in the Unified Modeling Language (UML) provides a generic extension mechanism for customizing UML models for particular domains and platforms” (Booch et al., 1998). UML provides users with several tools to design and document object-oriented systems (Booch et al., 1998). Stereotypes, tagged values, and constraints are used to adapt UML elements to a specific

application. These extended elements constitute the UML profile. Stereotypes extend the semantics of existing elements for a specific domain. Tagged values are used to add new properties to existing elements.

Constraints are used to specify rules to check the validity of a stereotype. Object Constraint Language (OCL) provides a standard language to express constraints that can be interpreted by code generators to generate code automatically. Computer-aided software engineering tools facilitate the development of software applications. Computer-aided software engineering (CASE) is the scientific application of a set of tools and methods to a software system to produce high-quality, defect-free, and maintainable software products. Some CASE tools provide an automatic implementation of UML models. The use of UML and automatic implementation for OLAP and spatial OLAP projects has been proven to reduce design time and implementation efforts (Blanco et al., 2009). Therefore, (Boulil et al., 2015) proposed the ICSOLAP UML profile. The ICSOLAP UML profile enables the conceptual representation of complex spatio-multidimensional applications. It contains stereotypes for each spatio-multidimensional element. A <<Fact>> is composed of a <<Measure>> and is associated with dimension levels (<<AggLevel>>) using a <<DimRelationship>>. An <<AggLevel>> is composed of dimensional attributes and can be thematic, spatial or temporal. A <<SpatialAggLevel>> extends the <<AggLevel>> with a geometric attribute (<<LevelGeometry>>). The <<AggRelationship>> is an aggregation relationship between two levels in the same hierarchy. The stereotype <<BasicIndicator>> defines the aggregation rules for a given measure (i.e., "aggregatedAttribute"). It indicates the functions that are used in the aggregation process along dimension hierarchies. ICSOLAP has been implemented in the commercial CASE tool MagicDraw, and a tool for its automatic implementation in Postgres/Oracle and Mondrian has also been developed. An example of ICSOLAP is shown in Section 3.2.

3.2 Requirements

In this section, we introduce spatio-multidimensional modeling and querying requirements (Table 1) that are associated with the integration of a regular grid of points in SDWs.

In the context of the project with Agaetis, from some odor sensors deployed in cities, a 100 *100 resolution (1000 points) regular grid of points is produced using the ADMS 5 simulation model (ADMS5, 2015). A regular grid of points is generated each 15 minutes for different types of odors sources (for example, NO₂).

The developed spatio-multidimensional model for our environmental case study is shown in figure 4. The fact (<<Fact>>) is the class "Odor", which presents a numerical measure (<<NumericalMeasure>>) named "odorValue". It is analyzed using three dimensions: a temporal dimension "Time" with levels "Minute", "Hour", "Day", "Month" and "Year" (<<TemporalAggLevel>>); two thematic dimensions (<<ThematicDimension>> "Pollutant" and "Sources") that represent the type of pollutant (e.g., NO₂) and the source of the odor (e.g., a factory); and a spatial dimension (<<SpatialDimension>> "Location") with a discrete spatial level (<<SpatialAggLevel>> "Grid100") that represent the regular grid of points (<<LevelGeometry>>"Point"). A regular grid of points is an incomplete field representation, as shown in Figure 5a, where only a sample of points is stored. The measure is aggregated using the average for all dimensions. It is represented by a <<BasicIndicator>>, "AvgOdor". This SOLAP model answers SOLAP queries such as "What is the average odor value per hour and point of the grid?" (Q1- RollUp on the temporal dimension), or "What is the average odor value per tracer and city?" (Q2- Spatial RollUp on the spatial dimension) (Figure 6a). These queries enable decision makers to analyze the temporal (Q1) and spatial (Q2) evolution of the odor phenomenon.

Another possible interesting query for decision makers concerns the analysis of an odor around a particular odor producer (e.g., a factory): “What is the odor value per tracer and point in a zone of 2 km around a particular factory (i.e., around a particular point)?” (**Q3- Spatial slice on the spatial dimension**) (Figure 6c). Note that **Q2** can be easily answered because a point is topologically included or not included in the polygon that represents the city (Figure 6b). The previously mentioned model enables only classical SOLAP queries to be answered because continuity is not integrated. Therefore, the first requirement is that this extended spatio-multidimensional model has to support classical SOLAP operators (Spatial RollUp, Spatial DrillDown and Spatial Slice) (e.g., queries **Q1**, **Q2** and **Q3** of Table 1).

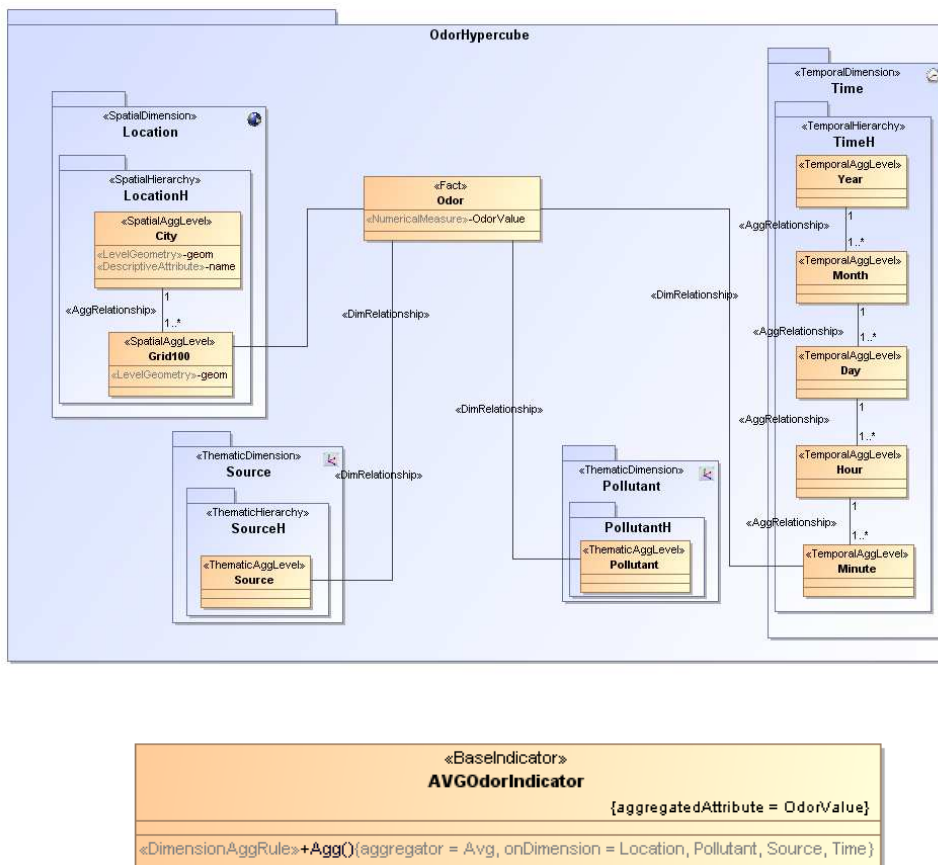


Figure 4. ICSOLAP example: Odor case study

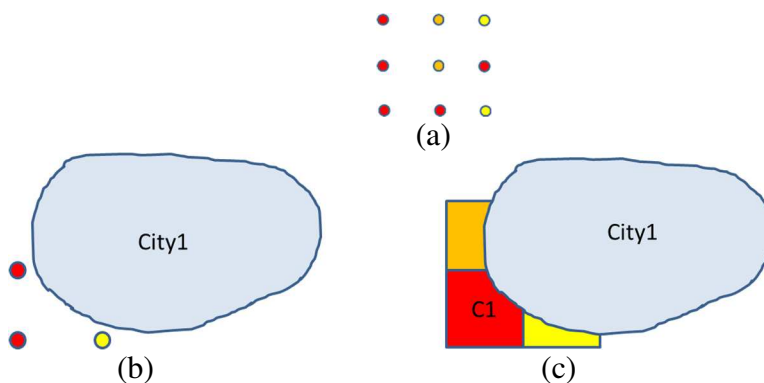


Figure 5. a) Regular grid of points, Spatial hierarchy: b) with Raster and c) Regular grid of points

Assume that the decision makers are interested in knowing the odor value in a non-sampled location (i.e., a point that is not represented by a spatial member), for example, to study the impact of the odor phenomenon on sensible locations, such as a school. A possible SOLAP query is shown in Figure 6b (**Q4** - “*What is the average odor value per year and at the school De Capraris (i.e., at the point x,y)?*”). This query is possible by providing a continuous vision of the spatial level using spatial interpolation functions. We refer to this requirement as *Continuity*.

In our example, we use bilinear and bicubic spatial interpolation functions, which are local deterministic methods. They employ the four nearest neighborhood points and the 16 nearest neighborhood points and calculate a weighted average of the distance from the estimated point. As stated in Section 2.1, multiresolution fields can be useful for spatial analysis. For example, in our case study, the decision makers need to analyze odor value at a more detailed resolution for city zones with an important urban density, such as the downtown area. Therefore, they can provide different resolution queries, such as “*What is the average odor value per month and point of the 200*200 grid?*” (**Q5a**), and “*What is the average odor value per month and point of the 400*400 grid?*” (**Q5b**), where $n*n$ represents a regular grid of n^2 points (Figure 6d). Note that a set of useful and known resolutions exist for each spatial phenomenon; thus, they can be predefined based on data and the needs of decision makers. We define *Multiresolution* as the requirement that enables the regular grid of points at different resolutions to be represented according to the pyramid model (c.f. Figure 2).

The usage of map algebra operators in the spatio-multidimensional analysis process (*Map Algebra* requirement) is mandatory because, as detailed in Section 2.1, map algebra is an effective tool for analyzing fields.

The *local map algebra* operator corresponds to a classical RollUp operator. A RollUp operator aggregate measures values using a mathematical aggregation function for each spatial member (a point). An example is the previously described **Q1** query, which corresponds to a local map algebra operator that employs all points of the grid as inputs with the odor value at the daily granularity and outputs a grid with the average odor value at the monthly granularity.

The *focal and zonal map algebra* operators can be associated with spatial data warehouse window queries (Papadias et al., 2001) because they consist of aggregate measure values for a spatial area that is not predefined as a spatial member. An example is “*What is the average odor value per tracer and a zone of 2 km around a particular factory (i.e., around a particular point)?*” (**Q6**). This query is similar to the previously mentioned spatial slice query (**Q3**) with the difference that it aggregates the measure values in the circular 2 km zone.

Requirement		Query example
SOLAP operators	Spatial DrillDown/Roll Up	Q2: “ <i>What is the average odor value per tracer and city?</i> ”
	Spatial Slice	Q3: “ <i>What is the odor value per tracer and point in a zone of 2 km around a particular factory (i.e., around a particular point)?</i> ” (Figure 6c)
	OLAP operators	Q1: <i>What is the average odor value per month and point of the grid?</i> ” (Figure 6a)
Continuity		Q4: “ <i>What is the average odor value per year and at the school De Capraris (i.e., at the point x,y)?</i> ”
Multiresolution		Q5a: “ <i>What is the average odor value per month and point of the 200*200 grid?</i> ” (Figure 6d) Q5b: “ <i>What is the average odor value per month and point of the 400*400 grid?</i> ” (Figure 6d)

Map algebra	Local (Spatial-RollUp)	Q1: <i>What is the average odor value per month and point of the grid?</i> (Figure 6a)
	Focal Zonal (Window queries)	Q6: <i>“What is the average odor value per tracer and a zone of 2 km around a particular factory (i.e., around a particular point)?”</i> (Figure 6e)

Table 1. Requirements for a spatio-multidimensional model that integrates the regular grid of points

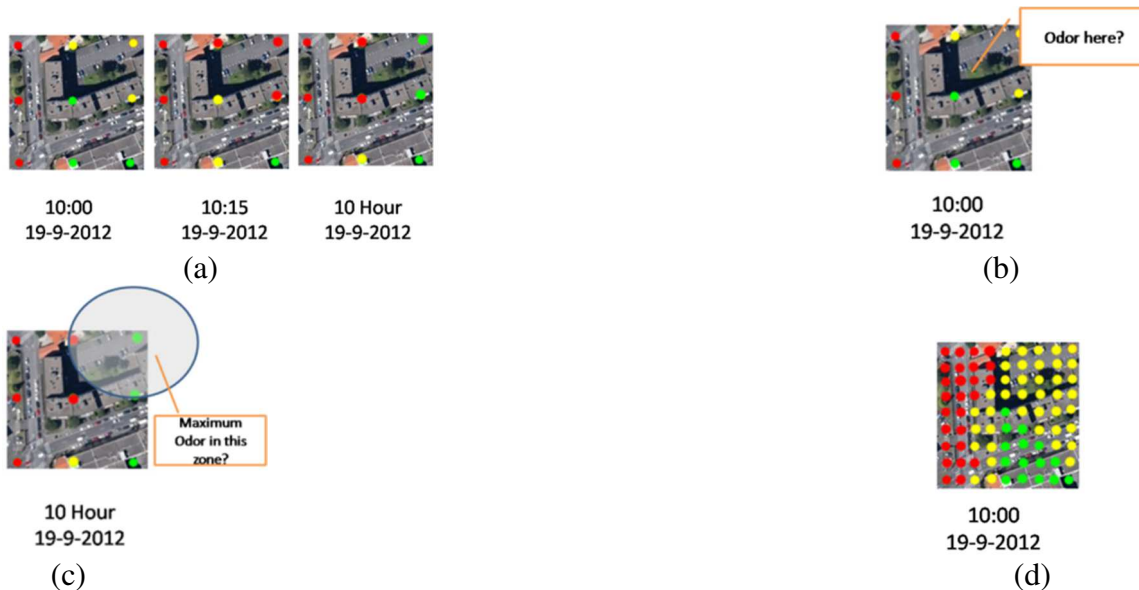


Figure 6 Cartography representation of a) map algebra b) continuity c) spatial slice d) multiresolution

4. Related work

Many studies have been proposed to handle vector data and OLAP (Boulil et al., 2015). In the last decade, few studies investigate the integration of fields in OLAP. In the following section, we review these studies.

(McHugh, 2008) (Li et al., 2014) and (Kasprzyk, 2014) investigate the integration of raster data in OLAP. The seminal work of (McHugh, 2008) defines four new types of dimensions that integrate raster data. They also define the "matrix cube", in which the cells of the raster contain the facts. They implement their model in a commercial SOLAP tool based on a relational SOLAP architecture. (Li et al., 2014) present the "tile cube". The "tile cube" implements a raster pyramid as a spatial dimension. The authors propose an implementation using the Map-Reduce framework to improve performance. (Kasprzyk, 2014) also investigates the integration of raster data as spatial measures. The author presents an implementation in a GIS dominant tool (without using a SOLAP server).

The introduction of incomplete representations of fields has been investigated in (Ahmed et al., 2005) (Bimonte et al., 2014) (Gómez et al., 2013) and (Gomez et al., 2010). (Ahmed et al., 2005) propose a multidimensional model for handling continuous incomplete fields. The approach stores a sample of points as vector spatial members and simulates the spatial continuity using a set of spatio-temporal interpolation functions implemented in the SOLAP client. They employ an OLAP server that does not support the spatial slice operator. (Gomez et al., 2010) propose a conceptual model that extends MultiDimER to integrate continuous fields.

They define continuous measures and continuous levels but do not define hierarchies between two continuous levels. A GIS-dominant implementation is also provided. In (Bimonte et al., 2014), the authors propose a multi-dimensional model to integrate fields in dimensions and measures. They propose a formal representation of field hierarchy, field measure and field cube. They also describe a theoretical implementation based on the ROLAP architecture extended with field hierarchies. (Gómez et al., 2013) define the map algebra operators on the top of a SDW. They integrate these operators to classical SOLAP operators.

The representation of multidimensional data at different levels of resolution can be considered as multirepresentation. Few studies (Bédard et al., 2002) (Concepción et al., 2009) (McGuire et al., 2008) have investigated this issue. (Bédard et al., 2002) define a conceptual model based on UML that includes several geometric and semantic properties of spatial level representations. This model supports only vector data; no detail about the implementation is provided. In (Concepción et al., 2009), the authors propose a conceptual model for the multirepresentation of spatial dimension using vector data and provide a logical model to implement the data. (McGuire et al., 2008) define a snowflake schema for environmental applications, in which the same spatial elements in each spatial dimension are represented with different representations.

Table 2 shows a comparison of existing studies according to our three previously defined requirements: “continuity”, “map algebra” and “multiresolution” of Table 2. Their implementation using an OLAP-GIS solution is also analyzed: the logical models for the SDW and SOLAP server tiers and the SOLAP client. The possibility of a conceptual representation is detailed (conceptual model).

As listed in Table 2, only (Ahmed et al., 2005) (Bimonte et al., 2014) and (Gómez et al., 2013) support “continuity” because all studies of multirepresentation integrate vector data, whereas other studies focus on raster data, which does not require methods for querying any point of the space because a raster is a complete representation of the space. The use of an OLAP-GIS implementation is not achieved for incomplete field data. In the same manner, (Gomez et al., 2010) propose a conceptual model for the field without considering multiresolution, and (Concepción et al., 2009) proposes a conceptual model for multiresolution for vector data.

		(McHugh, 2008)	Li et al., 2014)	(Gomez et al., 2010)	(Kasprzyk, 2014)	(Ahmed et al., 2005)	(Bimonte et al., 2014)	(Gomez et al., 2013)	(Bédard et al., 2002)	(Concepción et al., 2009)	(McGuire et al., 2008)	Our approach
Spatial data type		R	R	I	R	I	I	I	V	V	R	I
continuity	conceptual model	No	No	Yes	No	P	No	No	No	No	No	Yes
	sdw tier's logical model	No	No	Yes	Yes	No	No	No	No	No	No	Yes
	solap server model	No	No	No	No	No	No	No	No	No	No	Yes
Multiresolution	conceptual model	Yes	No	No	No	No	No	No	Yes	Yes	No	Yes
	sdw tier's logical model	Yes	No	No	Yes	No	No	No	No	Yes	Yes	Yes
	solap server model	Yes	No	No	No	No	No	No	No	No	No	Yes

Map algebra	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	No	Yes
-------------	-----	-----	-----	-----	----	-----	-----	----	----	----	-----

Table 2. Comparison of existing studies (P denotes partially and V, I, and R denote vector, incomplete field, and raster, respectively)

5. UML profile for regular grid of points

In this section, we extend the ICSOLAP UML profile proposed by (Boulil et al., 2015) to address continuity (Sec 5.1) and multiresolution (Sec 5.2).

5.1 Continuity

The main idea to introduce continuity in SDW is to enhance a spatial level with a spatial interpolation function that can be employed to estimate the measure values at each point of the study space. Our extension of the ICSOLAP profile is shown in figure 7. It defines a <<RegularGridPoints>> level as a type of a spatial level with a spatial interpolation function to represent the continuity ("ContinuityInterpolation" is a tagged value that assumes "FieldInterpolation" as values, where "FieldInterpolation" are operations that represent spatial interpolation functions) and the resolution level to which it belongs (the tagged value "resolution"). In addition, the geometry of the spatial level has to be a point (as stated by the OCL constraint defined in the <<LevelGeometry>>).

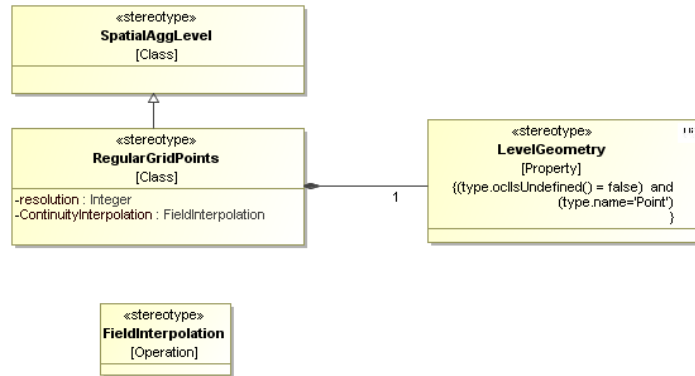


Figure 7 Extension of ICSOLAP with continuity for regular grid of points

An example of an OCL constraint is expressed by the following statement, which states that one spatial interpolation function per <<RegularGridPoints>> exists.

```

Context RegularGridPoints inv:
    self.ContinuityInterpolation->size()=1
    
```

An instance of the profile for our case study is illustrated in figure 8. The SDW presents a spatial dimension, in which the spatial level (“Grid100”) represents the regular grid of points. “Grid100” is composed of the geometric attribute "Point", its resolution level (100*100), and the bilinear interpolation function. This model supports continuity queries, such as Q4, due to the bilinear interpolation function. We note that the regular grid of points is associated with a less detailed classical spatial level that represents cities.

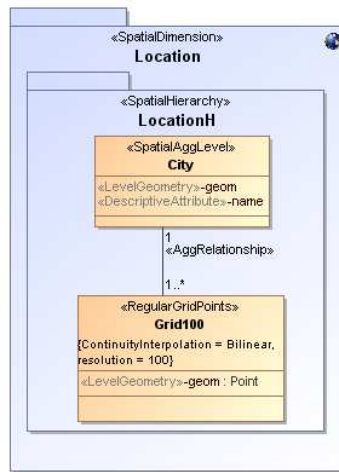


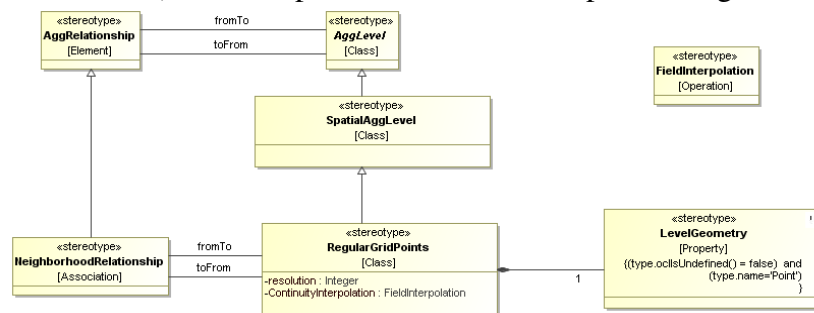
Figure 8 Continuity: example of <<RegularGridPoints>> level

5.2 Multiresolution

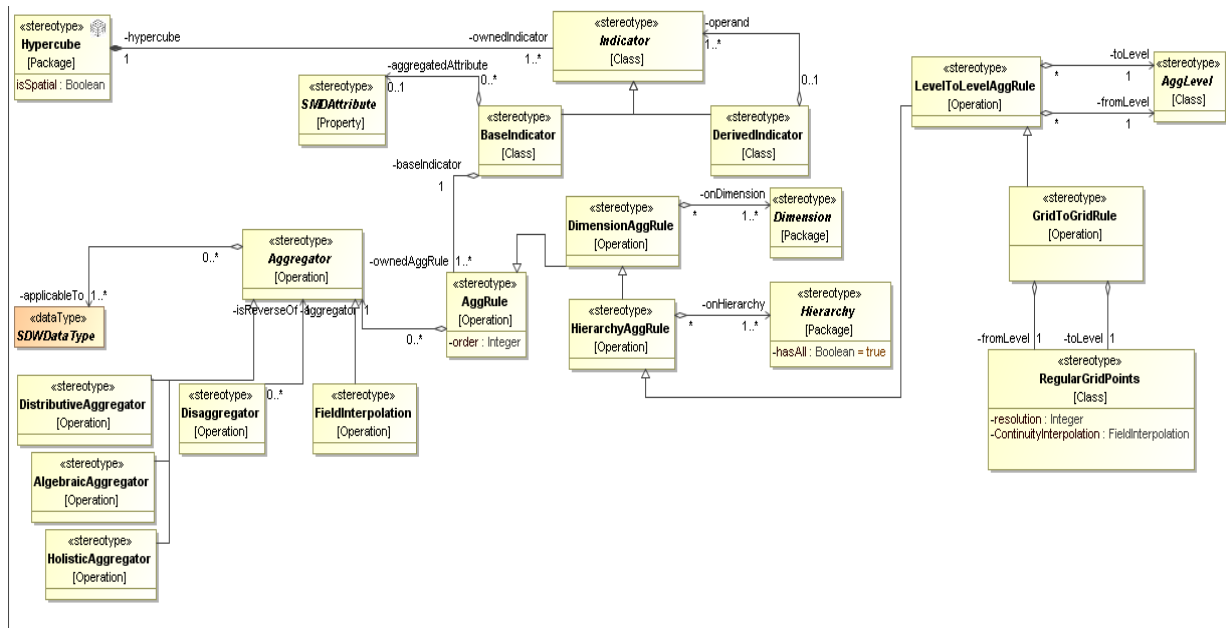
In (Zaamoune et al., 2013), to integrate regular grids of points at different resolutions in a spatio-multidimensional model, we propose two approaches: the "Field Aggregation Star-Schema" (FASS) and the "Field Interpolation Star-Schema" approach (FISS).

The Field Aggregation Star-Schema introduces a spatial level for each resolution, in which the most detailed resolution (e.g., contains more points) presents a foreign key to the fact table. This finding implies that factual data are stored at the most detailed resolution spatial level. In this manner, a decision maker can analyze warehoused spatial data that are represented as a regular grid of points at different resolutions in the same OLAP analysis session. Therefore, changing the resolution is simply achieved by changing the spatial level in the MDX query.

The main idea of the Field Interpolation Star Schema approach (FISS) is to use spatial interpolation functions to estimate measure values at the most detailed resolutions while retaining facts at a less detailed resolution. In this approach, navigation among the regular grids of points levels is performed from the less detailed grid to the more detailed grid using spatial interpolation functions instead of aggregation functions. This approach is only possible when continuous data, such as temporal and spatial data, are addressed. According to the Tobler geography law "Everything is related to everything else, but near things are more related than distant things" (Tobler, 1970), non-sampled values can be computed using near-space positions.



(a)



(b)

Figure 9 Multiresolution: extension of ICSOLAP for the FISS approach

Thus, we redefined the rules of aggregation between the levels proposed by (Boulil et al., 2015) by defining the stereotype `<<GridToGridRule>>` (Figure 9b) and extending the `<<LevelToLevelAggRule>>` (c.f. Sec 3.1), which is associated to `<<FieldInterpolation>>` (a spatial interpolation function), which serves as a rule during navigation between two `<<RegularGridPoints>>` levels. The relation between two grids of points is not an aggregation relation (such as for classical level with `<<AggRelationship>>`) but is employed by the spatial interpolation functions to obtain the neighborhood points. Therefore, we define a new association `<<NeighborhoodRelationship>>` by extending `<<AggRelationship>>`, which associates two `<<RegularGridPoints>>` levels (Figure 9a).

We describe an instance of the profile using a case study (Figure 10). The spatial dimension is extended with three `<<RegularGridPoints>>` levels of different resolutions (“Grid100”, “Grid200” and “Grid400”). The fact is associated with the less detailed `<<RegularGridPoints>>` level “Grid100”. “Grid200” and “Grid400” are associated with “Grid100” using a `<<NeighborhoodRelationship>>`.

The indicator “AVGOdorIndicator” defines a classical aggregation function (the average) for navigating from “Grid100” to “City”, whereas two spatial interpolation functions are used to navigate from “Grid100” to “Grid200” (the bilinear function) and from “Grid100” to “Grid400” (the bicubic function).

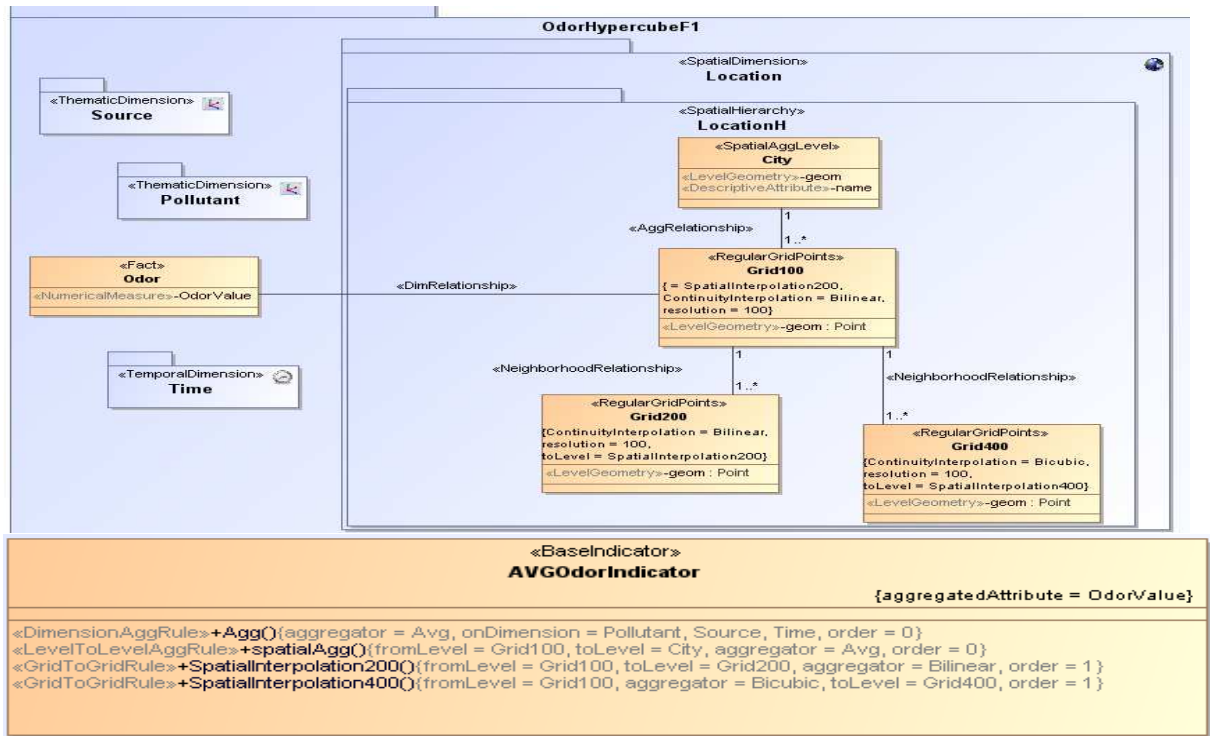


Figure. 10 Multiresolution: example of the FISS approach

6. Implementation

In this section, we present the relational OLAP-GIS architecture for implementing our approach (Sec 6.1) and the associated implementation of continuity and multiresolution (Sec 6.2). The UML profile has been implemented in the CASE tool MagicDraw. MagicDraw supports the definition of UML profiles with new stereotypes and tagged values. Moreover, it supports the definition of OCL constraints for the UML profile (c.f. Sec 5.1). These constraints are automatically verified during the design of the SOLAP model. It allows defining SOLAP models respecting the UML profile defined.

6.1 Architecture

The architecture is shown in Figure 11. The main idea is to use standards to provide generality to our proposal.

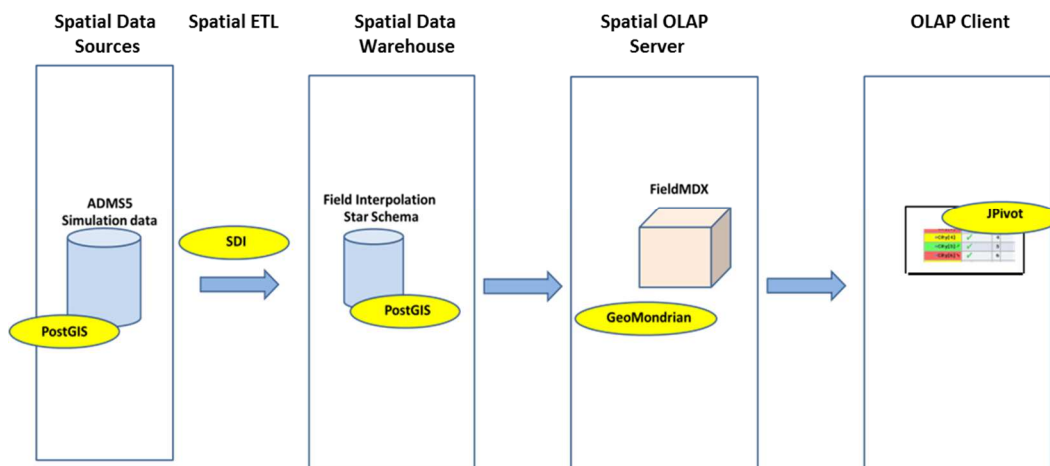


Figure 11 FieldMDX Architecture

Spatial and alphanumeric data are stored according to the logical models presented in the previous sections in the Spatial DW tier. It is implemented using PostGIS (PostGIS, 2015). PostGIS is an extension of the DBMS PostgreSQL for spatial data. The SOLAP server is GeoMondrian (GeoMondrian, 2015). GeoMondrian is a SOLAP server, which extends the OLAP server Mondrian, with support for topological operators on vector data (GeoMDX). We extend GeoMDX to implement our FieldMDX using MDX UDFs with spatial interpolation functions provided by the existing Java API "javax.media.jai api" (JAI, 2015). The OLAP client is based on JPivot (Bimonte, 2014).

6.2 Implementation

In the following section, we describe the logical models that are implemented in the Spatial Data Warehouse tier and the MDX extensions of the Spatial OLAP server tier.

The logical model implementing continuity is based on a traditional star schema with a geometric "point" attribute. The logical model for our case study is shown in figure 12a. The logical model for our case study is shown in Figure 12b.

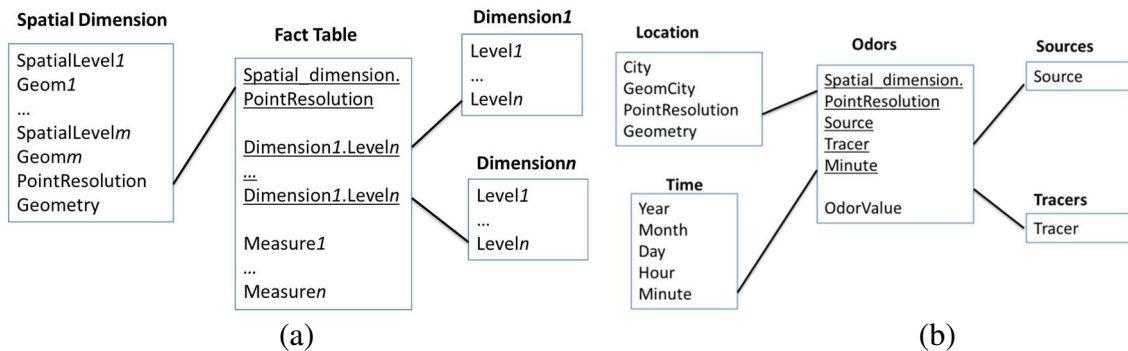


Figure 12 Continuity: a) Logical model and b) example

The implementation of the spatial interpolation function is achieved by the SOLAP server as follows:

MDX is the de-facto standard for OLAP Servers. Some extensions of MDX (MDX, 2015) for handling vector data have been proposed (Zaamoune et al., 2013) (such as GeoMondrian (GeoMondrian, 2015) and PietQL).

To support continuity, we propose FieldMDX, which extends MDX extension with spatial interpolation functions defined as MDX User-Defined Functions (UDFs) (MDX, 2015). In particular, we define spatial interpolation functions as a set of UDFs with the following signature: *NumericType InterpolatePoint (geometry)*

They employ a geometry (point(x, y)) as input and return a numeric value, which is calculated using the measured values of the neighborhood points of the point x,y.

An example is the *NumericType InterpolatePointBilinear (geometry)* function that implements the bilinear spatial interpolation function.

Using this new UDF, a continuity query, such as Q4 (Figure 13), can be answered.

```
With member [Measures].[odorValue] as
'InterpolatePointBilinear(ST\ _GeomFromText("POINT (72.1235
423521)"))'
SELECT [Measures].[odorValue] ON ROWS, [time].[Year].members ON
COLUMNS FROM [odorHypercube]
```

Figure 13 Continuity: example of FieldMDX for Query Q4

We have also implemented the bicubic spatial interpolation function (`NumericType InterpolatePointBicubic (geometry)`) in the same manner.

To implement the FISS approach, we propose a variant of the previously proposed schema, in which the fact is associated with the less detailed regular grid of points level (Figure 14a). For each regular grid of points level that is not associated with the fact table, an attribute named `Neighborhoodi` is introduced. It represents the neighborhood points that are employed by the spatial interpolation function defined by the `<<GridToGridRule>>`.

An example using our case study is shown in Figure 14b.

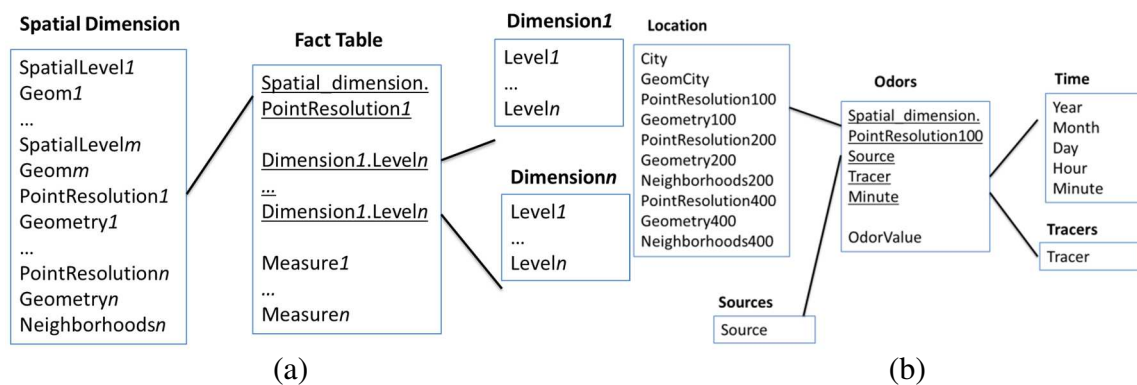


Figure 14 FISS: a) logical model and b) example

Regarding the continuity, to trigger queries on the top of the FISS logical model, we extend MDX with a set of UDFs with the following signature:

```
Numerictype InterpolateResolution (RegularGridPoints Member).
```

These functions are based on the logical model that was previously described. "InterpolateResolution" is a spatial interpolation function (we have implemented bilinear and bicubic functions) that uses a member of a `<<RegularGridPoints>>` level as the input and returns a numerical value. This spatial interpolation function uses the measured values that are associated with the less detailed `RegularGridPoints` level (`PointResolution1`) and the neighborhoods points (`Neighborhoods2`, ..., `Neighborhoodsn`) to calculate the measured values associated with more detailed levels (`PointResolution2`, ..., `PointResolutionn`).

To use these UDFs, we define a calculated measure "EstimatedValue" as follows:

```
"InterpolateBilinear ([SpatialDimension].[PointResolutioni].
```

```
CurrentMember) ". A calculated measure is a measure that is not stored in the fact table but is calculated online by the (S)OLAP Server.
```

In this manner, the navigation among the regular grid of points at different resolutions using the spatial interpolation functions is transparent to the users.

An example of a query is shown in Figure 15. This query enables the calculation of the odor measure for “Grid200” from “Grid100” using a bilinear function (Query 5a).

```
SELECT {[Location].[Grid200].Members} ON ROWS,
      {[Time].[Year].members } ON COLUMNS FROM [odorHypercube]
Where [Measures].[EstimatedValue200]
```

Figure 15 Multiresolution: example of FieldMDX for Query Q5a

7. Experiments

In this section, we detail the storage and time performance of our approaches. The maximum size of the SDWs was 1.176 GB. The computer had the following configuration: Intel Core i3 2.20 GHz, 4 GB RAM, Windows 7 Professional 64-bit OS. Note that we do not provide any validation about the usability of the UML profile, which we employ for the design of the SOLAP applications because ICSOLAP has been successfully validated in several previous real agri-environmental projects (Boulil et al., 2015).

7.1 Continuity

We have implemented two spatial interpolation functions: bilinear interpolation and bicubic interpolation. To evaluate the impact of the number of neighborhood points in the spatial interpolation functions on time performance, we have implemented the FISS approach using a bilinear interpolation (four neighborhood points) and a bicubic interpolation (16 neighborhoods points). As shown in figure 16, the performance time, which is exclusively associated with the spatial interpolation function, is a constant value. Therefore, the bicubic function requires 500 ms more than the bilinear function to retrieve its neighborhoods and estimate the measure values. The total time for a SOLAP query using the FISS approach can be divided into i) the calculation of the OLAP query and ii) the spatial interpolation function. Therefore, the spatial interpolation is selected in the design phase of the SDW according to the needs of the decision makers and its time computation cost.

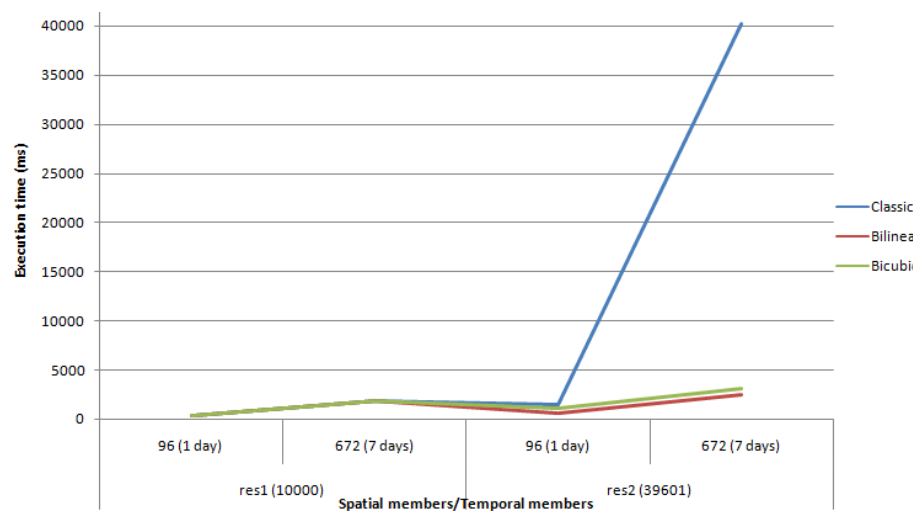


Figure 16 FISS spatial interpolation function performance

7.2 Multiresolution

7.2.1 Storage

Figure 17 shows the size of the fact table (i.e., number of tuples) by the number of spatial and temporal members for the FASS and FISS approaches. In particular, we vary the number of points (spatial members) with different resolutions and the number of temporal members.

Note two important differences: i) the FASS approach is more expensive than the FISS approach in terms of storage because the latter only stores the facts at the less detailed spatial resolution (100*100), and ii) in the FISS approach, the size of the fact table only varies according to the size of the non-spatial dimensions. Thus, even by increasing the size of the spatial dimension (using a very detailed regular grid of points), the fact table does not change because it always contains the measures for the less detailed resolution. Then, the FISS approach enables scaling on the spatial dimension without consequence.

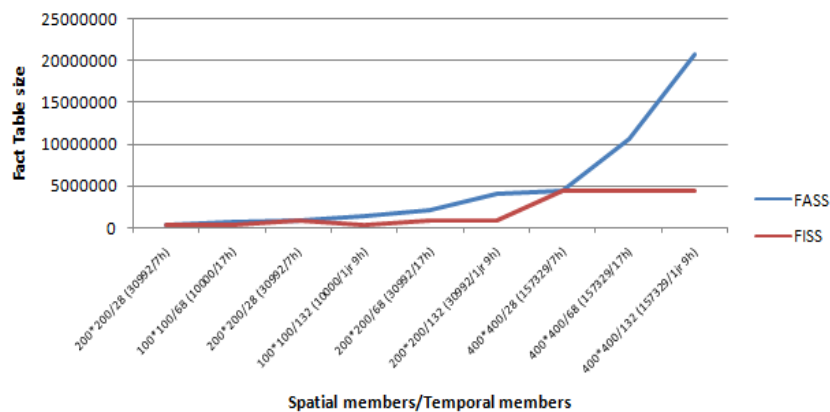


Figure 17 Fact table size with FASS and FISS

7.2.2 Execution time

To evaluate the execution time performance for the resolution change, we executed the previously described queries. Among these queries, we have selected the query with the worst time for our approaches. This query combines an aggregation operation (roll-up) on non-spatial dimensions and a spatial operator (spatial slice) on the grids at different resolutions (Queries Q5+Q1+Q3). The results are described in (Zaamoune et al., 2013). When the number of spatial and temporal members is large, the gap between the FASS approach and the FISS approach becomes important because minimizing the storage and the number of joins between the fact table and the dimensions enables the FISS method, which has a better execution time. Experiments indicate that the execution time increases with the number of spatial and temporal members in the FASS approach, whereas the execution time increases with the number of temporal members in the FISS approach.

8. Conclusions and future work

In this paper, we present a multidimensional model for incomplete fields that are represented as regular grid of points at several resolutions; this model is implemented in an OLAP-GIS architecture based on standards (SQL and MDX). We propose an approach that is based on interpolation to generate continuity over incomplete continuous fields and present two approaches based on the star schema model—FASS and FISS—to generate multiresolution over the incomplete field dimension. The objectives of the FISS approach are to improve the level of details of a continuous incomplete field and to optimize performance based on data modeling. We validate our proposal with experiments that demonstrate the feasibility and performance related to the two approaches (FASS and FISS) using two different interpolation functions.

Our current work is the extension of our conceptual model and its implementation to other types of fields representations such as raster, TIN, etc. Moreover, we plan to define an ad-hoc benchmark to evaluate the performance of SDWs in handling fields.

Nowadays, some new studies investigate the usage of cloud-based solutions for this kind of data (Li et al., 2014). Then, motivated by the important computation and storage capabilities offered by cloud-based solutions and NoSQL DBMSs (such as MongoDB and HBase) we plan to study new data models and query languages for handling field data.

References

ADMS 5. <http://www.cerc.co.uk/environmental-software/ADMS-model.html>

T. Ahmed, and M. Miquel, 2005. Multidimensional structures dedicated to continuous spatiotemporal phenomena. *22nd British National conference on Databases: enterprise, Skills and Innovation*. Springer-Verlag, Berlin, Heidelberg, p. 29-40.

Y. Bédard, 1997. Spatial OLAP. *Vidéoconférence. 2e Forum annuel sur la R-D, Géomatique VI: Un monde accessible*, Montréal, Canada.

Y. Bédard, T. Merrett and J. Han, 2001. Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery, *Geographic Data Mining and Knowledge Discovery*, Taylor & Francis, p. 53-73

Bédard, Y., M-J. Proulx, S. Larrivée, and E. Bernier, 2002. Modeling multirepresentations into spatial data warehouses: A uml-based approach. *Symposium on Geospatial Theory, Processing and Applications*, Ottawa.

P. Bereuter, R. Weibel, and D. Burghardt, 2013. Content zooming and information exploration for web and mobile maps. Adaptation of real-time map generalisation to the information seeking strategies of web and mobile users. *Revue Internationale de Géomatique* 23(3-4), p. 295-321

E., Bernier, P. Gosselin, T. Badard, and Y. Bédard, 2009. Easier Surveillance Of Climate-Related Health Vulnerabilities Through A Web-Based Spatial Olap Application. *International Journal of Health Geographics*, 8(18)

E. Bertino, E. Camossi, and M. Bertolotto, 2009. Multi-granular Spatio-temporal Object Models: Concepts and Research Directions. *ICOODB 2009*, p. 132-14

S. Bimonte, 2010. A Web-Based Tool for Spatio-Multidimensional Analysis of Geographic and Complex Data. *International Journal of Agricultural and Environmental Information Systems* 1(2), p. 42-67.

S. Bimonte, 2014. A generic geovisualization model for spatial OLAP and its implementation in a standards-based architecture. *Ingénierie des Systèmes d'Information* 19(5), p. 97-118

S. Bimonte, M. Kang, L. Paolino, M. Sebillio, M. Zaamoune, and G. Vitiello, 2014. OLAPing Field Data: a Theoretical and Implementation Framework. *Fundamenta Informaticae*, 132(2), p. 267-290.

S. Bimonte, 2015. Spatial OLAP for agri-environmental data and analysis: Lessons learned. *MIPRO 2015*

C. Blanco, I. Guzmán, D. Rosado, E. Fernández-Medina, and J. Trujillo, 2009. Applying QVT in Order to Implement Secure Data Warehouses in SQL Server Analysis Services. *Journal of Research and Practice in Information Technology* 41(2), p. 135-154

M. Body, M. Miquel, Y. Bédard, A. Tchounikine: Handling Evolutions in Multidimensional Structures. *ICDE 2003*: 581-591

G. Booch, J. Rumbaugh, and I. Jacobson, 1998. *The Unified Modeling Language User Guide, the original developers of the UML*. Addison Wesley.

K. Boulil, S. Bimonte, and F. Pinet, 2015. Conceptual model for spatial data cubes: A UML profile and its automatic implementation. *Computer Standards & Interfaces* 38, p. 113-132

G. Câmara, U. Freitas and M.A. Casanova, Fields and Objects Algebras for GIS Operations. *Proceedings of III Brazilian Symposium on GIS*, 407-424 (1995).

G. Concepción M., and R. Guadalupe, 2009. A Multidimensional Methodology with Support for Spatio-Temporal Multigranularity in the Conceptual and Logical Phases. *Progressive Methods in Data Warehousing and Business Intelligence: Concepts and Competitive Analytics*. IGI Global, p. 194-230.

J. Cordeiro, G. Câmara, U. Freitas, and F. Almeida, 2009. Yet Another Map Algebra. *GeoInformatica* 13(2), p. 183-202

H. Couclelis, 1992. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. *Theories and methods of spatio-temporal reasoning in geographic space*, p. 65-77

K. Ferreira, G. Câmara, and A. Monteiro, 2014. An Algebra for Spatiotemporal Data: From Observations to Events. *T. GIS* 18(2), p. 253-269

GeoMondrian. <http://www.spatialytics.org/fr/projets/geomondrian/>

M., Golfarelli, M., Mantovani, F. Ravaldi, and S., Rizzi, 2013. Lily: A Geo-Enhanced Library for Location Intelligence. *DaWaK 2013*, p. 72-83

L. Gomez, A. Vaisman and E. Zimanyi, 2010. Physical Design and Implementation of Spatial Data Warehouses Supporting Continuous Fields. *12th international conference on Data warehousing and knowledge discovery*, Springer-Verlag, Berlin, Heidelberg, p. 25-39

L. Gómez, S. Gómez, and A. Vaisman, 2013. Modeling and Querying Continuous Fields with OLAP Cubes. *IJDWM* 9(3), p. 22-45

W. H. Inmon, 2005. *Building the Data Warehouse*. John Wiley & Sons, Inc. New York, NY, USA.

R. LAURINI, D. THOMPSON 1992. *Fundamentals of Spatial Information Systems*. Academic Press

R. Laurini, L. Paolino, M. Sebillio, G. Tortora, and G. Vitiello, 2004. A Spatial SQL Extension for Continuous Field Querying. *COMPSAC Workshops 2004*, p.78-81

H., Ledoux, and C.M Gold, 2006. A Voronoi-based Map Algebra. *12th International Symp. On Spatial Data Handling*, Springer, Heidelberg, p. 117-131.

J. Li, L. Menga, F. Wang, W. Zhanga, and Y. Cai, 2014. A Map-Reduce-enabled SOLAP cube for large-scale remotely sensed data aggregation. *Computers & Geosciences*, 70, p. 110–119

E. Malinowski, and E. Zimányi, 2008. *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Data-Centric Systems and Applications, Springer.

MDX. <https://msdn.microsoft.com/en-us/library/ms145595.aspx>

M. McGuire, A. Gangopadhyay, A. Komlodi and C. Swan, 2008. A user-centered design for a spatial data warehouse for data exploration in environmental research, *Ecological Informatics*, 3(4–5), p. 273–285

R. McHugh, 2008. *Intégration de la structure matricielle dans les cubes spatiaux*. Master's Thesis, Université Laval, Canada.

J. Mennis, R. Viger, and C.D. Tomlin, 2005. Cubic map algebra functions for spatio-temporal analysis, *Cartography and Geographic Information Systems*, 30, p. 1:17–30

Mondrian. <http://mondrian.pentaho.com/documentation/olap.php>

D. O'Sullivan, and D. Unwin, 2002. *Geographic information analysis*. John Wiley and Sons

D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, 2001. Efficient OLAP Operations in Spatial Data Warehouses. *SSTD 2001*, p. 443-459

C. Plumejeaud, H. Mathian, J. Gensel, and C. Grasland, 2011. Spatio-temporal analysis of territorial changes from a multi-scale perspective. *International Journal of Geographical Information Science* 25(10), p. 1597-1612

J-P. Kasprzyk. 2014. Prototype SOLAP appliqué sur des champs continus en mode raster ; Analyse de hot spots de criminalité. *SAGEO2014*

R. Kimball, 1996. *The data warehouse toolkit: practical techniques for building dimensional data warehouses*. John Wiley & Sons

JAI. Java Advanced Imaging API.
<http://www.oracle.com/technetwork/java/javase/tech/jai-142803.html>

PostGIS. <http://postgis.net/>

Seffino, L., Medeiros, C., Jansle, V. Rocha, Yi, Y. 1999. WOODSS - a spatial decision support system based on workflows. *Decision Support Systems* 27(1-2), p. 105-123.

S. Shekhar, V. Gunturi, M. Evans, K. Yang: Spatial big-data challenges intersecting mobility and cloud computing. *MobiDE 2012*: 1-6

T. Siqueira, C. Ciferri, V. Times, and R. Ciferri, 2012. The SB-index and the HSB-index: efficient indices for spatial data warehouses. *GeoInformatica* 16(1), p. 165-205

N. Stefanovic, J. Han, and K. Koperski, 2000. Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes. *IEEE Transactions On Knowledge and Data Engineering*, 12(6), p. 938-958.

S., Timpf, A. U., Frank, 1992. Using hierarchical spatial data structures for hierarchical spatial reasoning. *Spatial Information Theory - A Theoretical Basis for GIS*. Springer, Heidelberg p. 69-83.

W., Tobler, 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46(2), p. 234-240.

C. Tomlin, 1990. *Geographic information systems and cartographic modeling*. Prentice-Hall, Englewood-Cliffs.

R. Torlone, 2003. Conceptual multidimensional models, *Multidimensional databases*, IGI Publishing, p. 69-90.

M. Zaamoune, S. Bimonte, F. Pinet, and P. Beaune, 2013. A New Relational Spatial OLAP Approach for Multi-resolution and Spatio-multidimensional Analysis of Incomplete Field Data. *ICEIS*, p. 145-15