PostGeoOlap: an Open-Source Tool for Decision Support

Giovanni Colonese[1], Rodrigo Soares Manhães[1,2], Sahudy Montenegro González[2], Rogério Atem de Carvalho[3], Asterio Kiyoshi Tanaka[4]

[1]Faculdade Salesiana Maria Auxiliadora (FSMA)
[2]Universidade Candido Mendes – Campos (UCAM-Campos)
[3]Centro Federal de Educação Tecnológica de Campos (CEFET-Campos)
[4]Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

colonese@lagosnet.com.br, rmanhaes@cefetcampos.br, sahudy@ucam-campos.br, tanaka@uniriotec.br, ratem@cefetcampos.br

Abstract. *This work describes PostGeoOlap, a decision support tool that integrates OLAP (On-Line Analytical Processing) and GIS (Geographical Information System) technologies in a single application. PostGeoOlap is an open source and a general-purpose tool to be used by application developers to easily develop their decision support applications. This tool works on the PostGreSQL DBMS using its spatial extensions (PostGIS) and performs the analytical and geographical functionalities using data warehouses.*

1. Introduction

Nowadays data warehousing has been proved to be an efficient storage technology for supporting Decision Support Systems (DSS) applications. The capability to analyze aggregated data integrated from several sources makes a Data Warehouse (DW) allied to an On-Line Analytical Processing (OLAP) application, valuable tools for the organizations' decision makers. Another technology historically used for decision-making support is the Geographical Information System (GIS). It deals with spatial functionalities and produces maps to help users to analyze data with geographical reference.

Many researchers are working on the integration of analytical and geographical technologies in a single application. This idea provides better support to the decision-making process allowing analysis under the perspectives of business, time and space. Most of the recent works regarding analytical and geographical integration focuses the merge of already existent GIS and OLAP applications to produce an intersection among their results. This fusion generates a third application involving the desired integration. A few proposals present a spatial OLAP without any modeling technique to design an application from the conceptual level.

This paper presents PostGeoOlap, a decision support tool that integrates OLAP and GIS technologies. PostGeoOlap is part of the GeoOlap project [Colonese 2004], which proposes a technique to model an application from its initial conception where the coexistence of the spatial and time dimensions is essential. The main goal of PostGeoOlap is to be an open source and a general-purpose tool used to easily produce a decision support application.

The remainder of this work is organized as follows. In Section 2 we present a review of the previous DW and GIS integration proposals. Section 3 shortly describes the GeoOlap project, an unified modeling technique to apply geographic stereotypes on

DW modeling using UML (*Unified Modeling Language*). Section 4 explains the architecture and functionalities of PostGeoOlap. Section 5 describes a case study. Section 6 presents conclusions and future work.

2. Related Works

There are several works related to GeoOlap project and to PostGeoOlap tool. These works have different approaches in respect to GIS and DW integration.

GeoMiner [Stefanovic 1997] allows OLAP operations on cubes with georeferenced data and MapCube [Shekar et al 1997] proposes cube operators that can return maps. But both just propose analytical processing without considering the use of a GIS. GOAL (Geographical Information On-Line Analysis) [Kouba, Matoušek and Mikšovský 2000], SIGOLAP [Ferreira, Campos and Tanaka 2001] and GOLAPA [Geographical On-Line Analytical Processing Architecture) [Fidalgo, Times and Souza 2001] do not use a unified model with geographical and analytical concepts. Instead they treat these two technologies separately and propose some kind of integration module that maps requests and data from one side to another. Works in [Han, Stefanovic and Koperski 1998] and [Papadias et al. 2001] are the most similar to the approach in this paper (although we do not consider spatial attributes on measures) but they do not propose any technique for modeling the system as a whole from its conceptual abstraction level, as GeoOlap project proposed. This paper presents a spatial OLAP tool, PostGeoOlap, capable of developing decision support applications integrating spatial and analytical functionalities as a whole.

3. The GeoOlap Project

The GeoOlap project [Colonese 2004] creates new spatial OLAP systems from scratch. It provides a unified method to model multidimensional systems with a geographical component. We define Spatial Data Warehouses as DW where one or more dimensions (in the star schema) have spatial attributes.

Spatial DWs are conceptually modeled using a UML diagram with geographical stereotypes [Colonese 2004] to represent the geographical classes. According to [Trujillo, Palomar and Gomez 2001], the use of UML can be explained because it considers the information system’s structural and dynamic properties at the conceptual level more naturally than the classic approaches such as Entity-Relationship Model. Further, UML provides OCL (*Object Constraint Language*) for embedding user requirements and constraints in the conceptual model. In addition, UML also provides support to represent stereotypes, which simplify the representation of extensive hierarchies of objects. A representative icon or symbol associates a class to the whole extensive hierarchy.

The GeoOlap project is meant to easily model applications where the analytical and geographical functionalities are present from its conceptual phase. At the end, the use of the PostGeoOlap implies the correct understanding and development of the application model. The process comprises the following activities: (1) modeling the data warehouse using UML with spatial stereotypes [Colonese 2004] (for the geographical dimensions); (2) mapping the spatial DW schema (dimensional-relational) from the UML model (conceptual level); and (3) using PostGeoOlap to manipulate the data warehouse in order to provide *on-line* capabilities to analytically and geographically query the data and to visualize the results both on a grid and on a map.

4. The PostGeoOlap Tool

PostGeoOlap is a tool for creating spatial OLAP solutions on top of PostGreSQL DBMS [PostGreSQL 2005] and PostGIS [PostGIS 2005], its spatial extension. The name **PostGeoOlap** was assigned because of the integration of geographical properties, OLAP technology and PostGreSQL.

PostGreSQL has PostGIS geographical extension, indispensable for this work. In a feasibility study of PostGreSQL DBMS for data warehousing, [Almeida 2004] concludes that PostGreSQL version 7.4.x is not suitable for this kind of application. It mainly fails in the query optimizer and aggregation features. The current version, 8.0.x, solves many negative aspects pointed by Almeida and has substantial improvements in the query optimizer. Very recently, the BizGres initiative [BizGres 2005] (yield by the PostGreSQL developers) works to make PostGreSQL a robust DBMS for Business Intelligence and Data Warehousing. The PostGeoOlap tool was approved by this initiative to add OLAP functionalities to the BizGres project.

4.1. Design Principles

PostGeoOlap is a general-purpose tool for OLAP analysis of conventional and geographical data, written exclusively in Java. We adopt to be and to use open-source software. This plays an important role because it provides access to small and medium organizations to develop low cost applications using data warehouse and GIS technologies.

PostGeoOlap has adopted ROLAP as its data warehouse storing model to take advantage of the object-relational DBMS capabilities. Both analytical and geographical queries are processed and answered by the PostgreSQL extended by PostGIS, and all data (from the base level to the aggregations) are kept in the relational model.

The main goals of PostGeoOlap are: (1) to provide to applications a mechanism to perform queries with analytical and geographical features on their data warehouses; (2) to provide to application developers an easy-to-use GUI tool to build their decision support applications.

Figure 1 shows the architecture of the current implementation. PostGeoOlap uses classes from the JUMP Unified Mapping Platform (JUMP) Java framework [JUMP 2005] to perform visualization of maps and results of geographical queries.

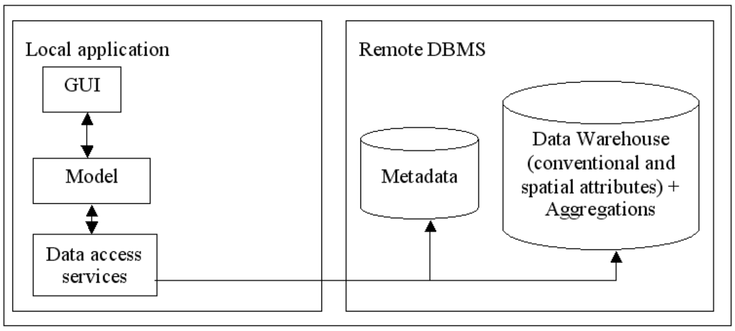

Local application
GUI
Model
Data access services
Remote DBMS
Metadata
Data Warehouse (conventional and spatial attributes) + Aggregations

Figure 1. PostGeoOlap architecture

4.2. PostGeoOlap Functionalities

Table 1 resumes the use cases of the proposed tool.

Table 1. PostGeoOlap's list of Use Cases

USE CASE	PURPOSE
Create Schema	Creates a connection with a PostGreSQL database.
Create Cube	Creates a cube inside the schema selecting the Fact table and defining its numeric items and the desired operations over these items.
Add Dimension	Creates a perspective for analysis of the data contained in the Fact table, selecting one of the database tables. Defines the dimension hierarchy to allocate a level for each attribute. It deals with conventional data in the Fact table and with conventional or geographical data in the dimensions.
Process Cube	Verifies the mass of the stored data using the metadata and attempts to infer the query performance (execution time). The queries evaluated as *low performance* are optimized by aggregations. This involves the cube analysis under any perspective within reasonable time.
Add Non-Aggregate Dimension	Creates a dimension to data in the Fact table even it is not a perspective. It serves as reference for geographical predicates with the other dimensions that possess spatial attributes.
Data Analysis	Provides an interface that allows the attributes selection for a query using conventional and/or geographical restrictions. It visualizes the query result as tables for analysis of non-spatial data and as maps for spatial data. For more details see the case study in the next section.

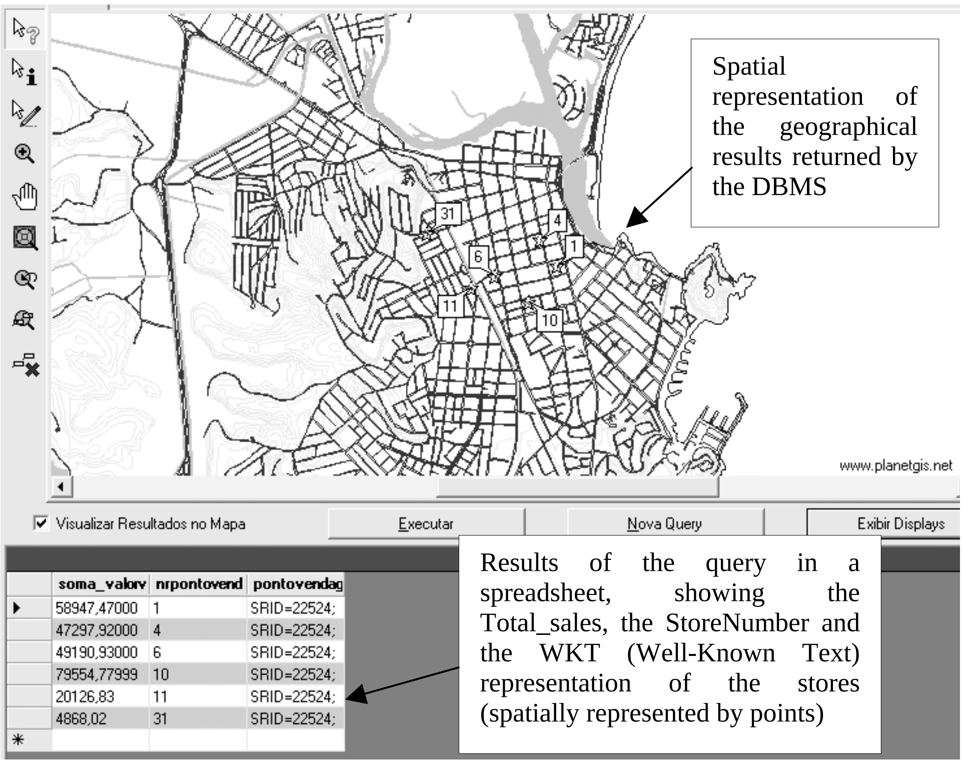
After the definition of the schema and the cube, the tool processes the cube to check for execution performance. If a query performance test falls below the predefined threshold, the OLAP application creates a new aggregation structure represented by a table. The aggregation structure is performed in three steps: (1) creates the table containing the aggregated data, (2) puts the aggregated data into the new table, and (3) creates indexes for the new table (using B-Tree for conventional attributes and GiST - *Generalized Search Tree* - for the spatial ones).

During the cube processing, the tool always starts with the generation of aggregations of the highest perspective. That is, for each dimension the attributes must receive a hierarchy level, from 9 (less aggregated information) to 1 (more aggregated information). Many attributes can share the same hierarchy level. The aggregations in a perspective are generated using the higher perspective above this, improving the performance. A non-aggregate dimension is a dimension of geographical nature. It does not participate in the cube processing and it does not generate aggregations (so the name "non-aggregate"). The only purpose of non-aggregate dimensions is to serve as reference for comparisons with other geographical dimensions. The creation of non-aggregate dimensions is not a mandatory step in cube processing. There is no difference on adding non-aggregate dimensions before or after the cube processing.

Data Analysis (on processed cubes) provides the means to select attributes for analysis and graphically define the constraints (conventional or geographical). Besides, shows the results on maps and spreadsheets.

To add a constraint to a non-geographical attribute (WHERE clause in SQL SELECT command), the application provides a group of comparison operators (equal, smaller than, larger than, etc) and exhibits a list with the current values in the database for the selected attribute for user's choice. If the selected attribute is a geographical one, the tool presents a list of geographical functions (implemented by PostGIS on PostGreSQL), that can be applied to the attribute related to some other geographical attribute belonging to a dimension here denominated "non-aggregate". The constraints on geographical attributes can be concatenated with non-geographical restrictions.

With the attributes collection selected by the user, and the constraints specified through conventional or geographical predicates, the tool starts the search for aggregations. This is done in the order from the most aggregated to the less aggregated (the less aggregated structure of all is the fact table with its dimensions, that is, the level base). Consequently, the result aggregation will be the one with the smallest computational cost for queries, having all the desired attributes, thus obtaining the best aggregation for submitting the requested query. Once the spatial OLAP tool knows which is the best aggregation (that is, a table), it submits the query with the constraints to the already mentioned aggregation in the DBMS and receives the results from it.



Spatial representation of the geographical results returned by the DBMS

	soma_valorv	nrpontovend	pontovendag
▶	58947,47000	1	SRID=22524;
	47297,92000	4	SRID=22524;
	49190,93000	6	SRID=22524;
	79554,77999	10	SRID=22524;
	20126,83	11	SRID=22524;
	4868,02	31	SRID=22524;
*			

Results of the query in a spreadsheet, showing the Total_sales, the StoreNumber and the WKT (Well-Known Text) representation of the stores (spatially represented by points)

Figure 2. Query Results both in a grid and in a map

The results returned by the spatial DBMS are then presented in a spreadsheet frame, and those geographical ones (if any) are passed to a visualization component for displaying the objects in a map (Figure 2).

5. Case Study: Magazine Retailer

A magazine retailer distributes its products (newspapers and magazines) for sale to many stores geographically distributed along regions of Rio de Janeiro State. The DSS provides analysis about the amount of products sold in a period of time from suppliers and grouped by the stores. The application is conceptually modeled using UML with the geographical stereotypes proposed at GeoOlap project [Colonese 2004] (see Figure 3).

It’s very important for the DSS to answer questions like: - *How much products are sold during the year of 2002 in ‘Scientific Research’ category and near to schools?* (for example: 100 meters maximum).

This kind of question integrates geographical features to the DW. In order to answer it, the model uses a *Point* stereotype associated to *Store*, a *Polygon* stereotype for *Quarter* and associates *ReferencePoint* with the *Point* stereotype. The *ReferencePoint* class serves as a geographical reference to the Store (schools at 100 meters maximum). It is also important to mention that the *ReferencePoint* class is not associated with any other class in the schema.

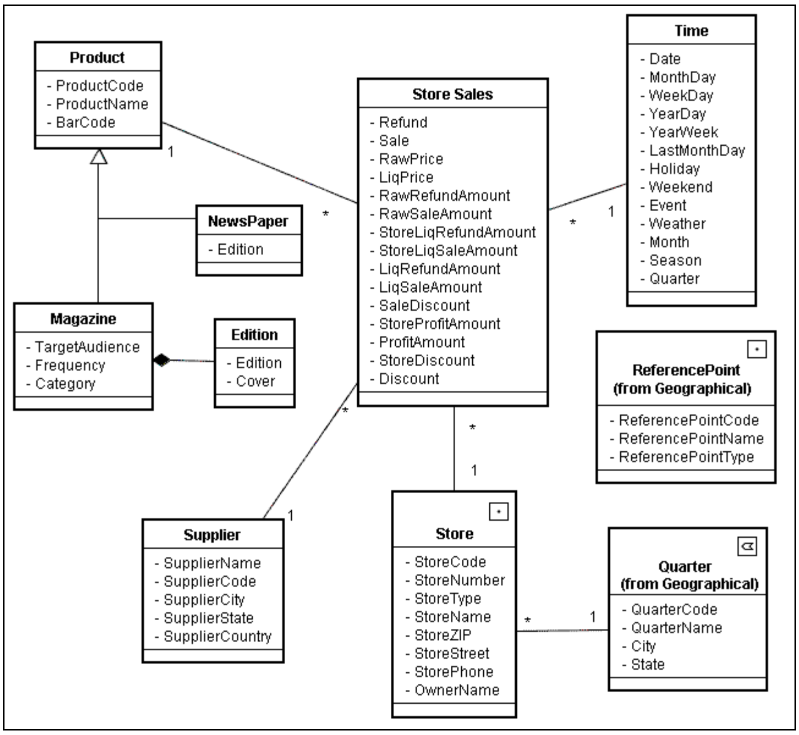


Figure 3. UML Class Diagram for the Magazine Retailer

The class diagram is mapped into the dimensional model. All classes that were part of a dimension are mapped into a single *Dimension* table and the spatial stereotypes are mapped into columns of the tables. Once the ETL (Extract, Transform and Load) phase is done, we build our application using PostGeoOlap to manipulate the data and visualize the results.

5.2 Creating the Cube

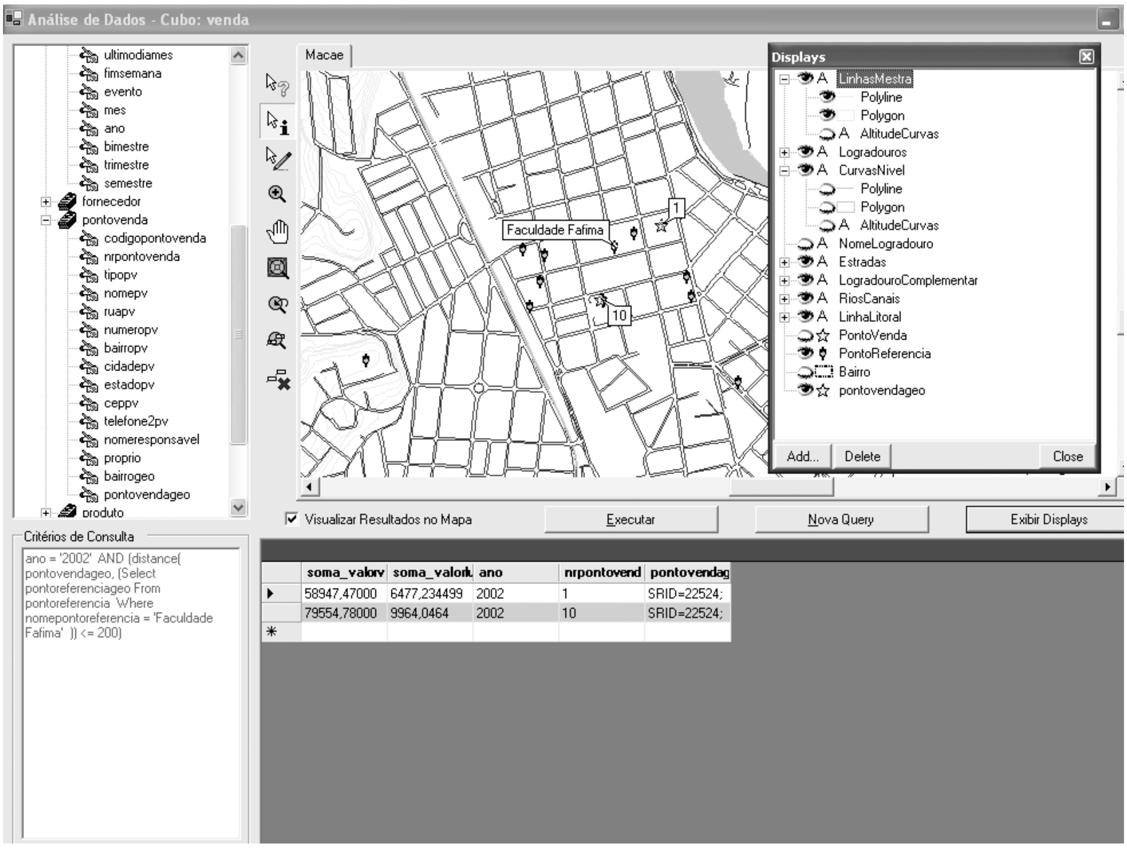
After the definition of the connection properties (server name, user, password, map file and SRID) to PostGeoOlap, a Cube must be created by selecting a table and defining its numeric measures.

The next step is the definition of the Cube dimensions. For each dimension, the attributes must receive a hierarchy level from 9 (less aggregated information) to 1 (more aggregated information). For the *Product* dimension, attributes like *ProductCode*, *BarCode*, *Edition* and *Cover* were left in level 9. *ProductName* and *TargetAudience* received level 8 (because a *Product* can have one or more *Editions*) and *Category*, which encompasses a lot of *Products*, received level 7. The *ReferencePoint* table is a non-aggregable dimension. Once all definitions for the cube are done, the next step is the cube processing, where aggregations for the data will be generated in order to speed up the queries.

5.3. Experimental Results

In order to answer the query: *How much products are sold during 2002 in 'Scientific Research' category and* near to 'Faculdade Fafima'? (set 100 meters maximum). The attributes of interest are picked from a tree on the left-upper panel of the screen. The conventional constraint is applied to the attribute "year" using the Conventional Constraint Screen in the left-bottom panel (see Figure 4).

The geographical constraint is applied on the *StoreGeo* attribute, selecting the geographical function *distance*, with value = 100 meters, to the desired reference point (*ReferencePointName = 'Faculdade Fafima'*). The results of the query are then shown both in a grid and in a map, as shown in Figure 4.



ano = '2002' AND (distance(pontovendageo, (Select pontoreferenciageo From pontoreferencia Where nomepontoreferencia = 'Faculdade Fafima')) <= 200)

	soma_valorv	soma_valorlu	ano	nrpontovend	pontovendag
▶	58947,47000	6477,234499	2002	1	SRID=22524;
	79554,78000	9964,0464	2002	10	SRID=22524;
*					

Figure 4. Results for the query example

6. Conclusions and Future Work

The motivation of this project was the lack of decision support tools that allow to model a data warehouse integrating different concepts (business, time and space) since the conceptual level. In this work, the unified conceptual model can be directly mapped into an application where GIS and OLAP functionalities are native.

This paper presented PostGeoOlap, an open source spatial OLAP tool that aims to facilitate application implementors in the development of decision support applications. It takes advantage of the PostgreSQL DBMS (extended by PostGIS) to utilize its conventional and geographical query engine.

This decision support tool gathers and presents the geographic, analytical and aggregated information graphically, demonstrating to be an easy-to-work tool to build DSS. Future work includes optimization issues about the cube processing. The PostGeoOlap project is available at http://pgfoundry.org/projects/postgeoolap.

References

Almeida, E. (2004) "Estudo de Viabilidade de uma Plataforma de Baixo Custo para Data Warehouse". Master Thesis. Curitiba: Universidade Federal do Paraná.

BizGres (2005) "BizGres". http://www.bizgres.org

Colonese, G. (2004) "Uma Ferramenta Aberta de Desenvolvimento Integrado de Sistemas de Informação para Processamento Analítico e Geográfico". Master Thesis. Campos dos Goytacazes: Universidade Candido Mendes.

Ferreira, A.; Campos, M.; Tanaka, A. (2001) "An Architecture for Spatial and Dimensional Analysis Integration". In: Proceedings of World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001). Vol. XIV Computer Science Engineering. Part II. p.392-395. Orlando, Florida, EUA: SCI.

Fidalgo, R.; Times, V.; Souza, F. (2001) "GOLAPA: Uma Arquitetura Aberta e Extensível para Integração entre SIG e OLAP". GeoInfo 2001, III Workshop Brasileiro de Geoinformática, p. 111-118. Rio de Janeiro: IME.

Han, J.; Stefanovic, N.; Koperski, K. (1998) "Selective Materialization: An Efficient Method for Spatial Data Cube Construction". In: Proceedings of PAKDD'98.

JUMP (2005) "The JUMP Project" http://www.jump-project.org/

Kouba, Z.; Matoušek, K.; Mikšovský, P. (2000) "On Data Warehouse and GIS Integration". In: Proceedings of DEXA2000. Greenwich: DEXA2000.

Papadias, D.; Kalnis, P; Zhang, J; Tao, Y. (2001) "Efficient OLAP Operations in Spatial Data Warehouses". In: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases. p.443-459. ACM Records.

PostGreSQL. (2005) "PostGreSQL". http://www.postgresql.org/

PostGIS. (2005) "PostGIS" http://postgis.refractions.net/

Shekar, S. *et al*. (2001) "Map Cube: A Visualization Tool for Spatial Data Warehouse". http://www.cs.umn.edu/research/shashi-group/mapcube.htm

Stefanovic, N. (1997) "Design and Implementation of On-Line Analytical Processing (OLAP) of Spatial Data". Master Thesis. Simon Fraser University.

Trujillo, J. *et al*. (2001) "Designing Data Warehouses with OO Conceptual Models". P.66-75. IEEE Computer.