GeoOlap: An Integrated Approach for Decision Support

Rodrigo Soares Manhães[1], Sahudy Montenegro González[2], Giovanni Colonese[3], Rogério Atem de Carvalho[4] and Asterio Kiyoshi Tanaka[5]

[1,2]Universidade Candido Mendes, Rua Anita Peçanha, 100, 28040-320, RJ, Brazil
rmanhaes@gmail.com sahudy@ucam-campos.br
[3]Faculdade Salesiana Maria Auxiliadora, Rua Monte Elísio, 27943-180, RJ, Brazil
colonese@gmail.com
[4]Centro Federal de Educação Tecnológica de Campos, 28030-130, RJ, Brazil
ratem@cefetcampos.br
[5]Universidade Federal do Estado do Rio de Janeiro, 22290-240, RJ, Brazil
tanaka@uniriotec.br

Abstract. The integration of information from different sources within the enterprise is one of the basis for implementing successful Decision Support Systems (DSS). Typically a DSS supplies analytical information obtained and transformed from large data warehouses. Recently, the geographical component of information has become more important due to the growing use of this type of information in logistics, marketing, and other applications. This work describes GeoOlap, a decision support approach that proposes a method to develop decision support applications that integrates analytical and georeferenced elements from the ground up - at the design stage - and OLAP (On-Line Analytical Processing) and GIS (Geographical Information System) technologies to visualize the results. Additionally, this paper describes PostGeoOlap, an open source general-purpose tool, which supports GeoOlap and that allows developers to easily build decision support applications. A case study is presented to validate the proposed ideas.

Keywords: *Data warehousing, Open source, decision support system, Data integration, Enterprise modeling and integration*

l. INTRODUCTION

Many applications used for business intelligence are often built by using data warehousing tools. Data integration appears with increasing frequency as the volume and the need to share existing data. The capability to analyze aggregated data integrated from several sources makes a Data Warehouse (DW) allied to an OLAP (On-Line Analytical Processing) application, valuable tools for the decision makers. Another technology historically used for decision-making support is the Geographical Information System (GIS), which deals with spatial data and produces maps to help users to analyze data with geographical references.

Many researchers are working on the integration of analytical and geographical technologies. This idea provides better support to the decision-making process

allowing analysis under business perspectives, time and space. Most of the recent works regarding analytical and geographical data 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 modeling techniques to design an application from the conceptual level.

This paper presents GeoOlap. It proposes an approach to develop decision support applications integrating analytical and georeferenced data. Also describes PostGeoOlap, an open source decision support tool based on GeoOlap proposal. To model an application from its initial conception it's very important to reflect the real world where the coexistence of the spatial and temporal dimensions is essential. The main goal of PostGeoOlap is to be an open source and a general-purpose tool used to easily yield a decision support application. Recently, PostGeoOlap was approved by the Bisgrez initiative to add OLAP functionalities to the BizGres project.

The remainder of this paper is organized as follows. In Section 2 we present a breve review of the previous DW and GIS integration proposals. Section 3 describes the GeoOlap project. Section 4 presents PostGeoOlap tool in junction with case study screens. The case study was useful to validate the proposed ideas. Section 5 presents the conclusions and future directions.

2. RELATED WORK

There are several works related to GeoOlap project. These works have different approaches to develop geographical and analytical data integration.

GOAL in Kouba [1], SIGOLAP in Ferreira [2] and GOLAPA in Fidalgo [3] do not use a unified model with geographic and analytical concepts. Instead they treat these two technologies separately and propose an integration module that maps requests and data. Works in Stefanovic [4] and Papadias [5] are similar to GeoOlap but do not propose any technique for modeling the system as a whole from its conceptual abstraction level. GeoOlap intends to be an open source platform to develop decision support applications integrating OLAP and GIS technologies. Applications are modeled using GeoOlap´s modeling technique and developed using the PostGeoOlap tool as it will be described in the next sections.

3. THE GEOOLAP PROJECT

Medium and small businesses have needs when it comes to data management and analytical processing to take important decisions about reducing billing costs and increase customer satisfaction. In the world of small and medium sized business, finding the right software solutions can be challenging. GeoOlap project in Colonese [6] is considered a solution for this kind of companies. It offers a low cost

environment (using free software solutions) to develop decision support applications from its conceptual level until its visualization.

GeoOlap proposes a unified method to model multidimensional systems with geographical components. We define *spatial data warehouses* as DW where one or more dimensions (using a star schema) have spatial attributes. Spatial DW is conceptually modeled using a UML diagram with geographical stereotypes to represent the geographical classes. According to Trujillo [7], 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 (for the geographical dimensions); (2) mapping the spatial DW schema (dimensional-relational) from the UML model (conceptual level); (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.

3.1 Spatial Stereotypes

The OpenGIS Consortium (OGC) defines specifications to allow interoperability in the processing of geographical data. It provides the OpenGIS Geometric Object Model for the geographical universe. Any real geographical object can be modeled and represented. The same data type defined in the conceptual model will be used in the logical model and also in the database implementation with no transformation nor conversion of concepts.

We propose to model a DW integrating the dimensional and spatial concepts in a unique diagram. We use geographical stereotypes to represent the geographical classes and no stereotypes to represent conventional or non-geographical classes (such as the fact class).

The stereotype representation (1) simplifies diagrams, (2) keeps semantic wealth and (3) facilitates the coexistence of different domains concepts (in this case, analytical and geographical concepts). Figure 2 uses a stereotype to represent the association between the `River` dimension and the `LineString` hierarchy as an abbreviation of `LineString` extended hierarchy shown in Figure 1.

4. THE POSTGEOOLAP TOOL

As part of the GeoOlap project was proposed PostGeoOlap. It is a tool for creating spatial OLAP solutions on top of PostGreSQL DBMS and PostGIS, its spatial extension. The name PostGeoOlap was assigned because of the integration of geographical properties, OLAP technology and PostGreSQL.

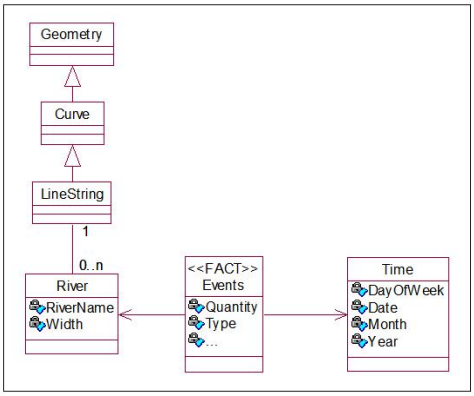

Figure 1. Association with an Extensive Hierarchy for the Dimension River

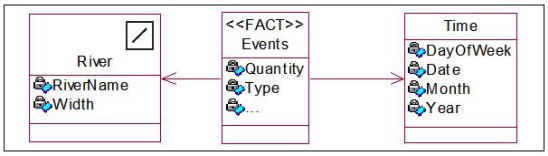

Figure 2. Stereotype to Represent the Association of River to LineString

PostGreSQL has PostGIS geographical extension, indispensable for this work. A feasibility study of PostGreSQL DBMS for data warehousing in Cunha [8] 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 versions, 8.x, solves many of these negative aspects and has substantial improvements in the query optimizer. Recently, the BizGres initiative (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 provide OLAP functionalities to the BizGres project.

4.1 Design Principles

PostGeoOlap means to be a general-purpose tool for OLAP analysis of conventional and geographic data. We adopt the open-source and free software paradigm as project definition. The visualization classes and all APIs, frameworks

and database software used in this project are open source. This plays an important role because it provides access to small and medium organizations to develop applications and to make use of data warehouse and GIS technologies, which have forbidding costs in their proprietary incarnations.

PostGeoOlap has adopted ROLAP as its data warehouse storing model to take advantage of the object-relational DBMS capabilities to store conventional and geographic information to use spatial and aggregate functions and to define new ones. Thus, both analytical and geographical queries are processed and answered by the PostgreSQL, and all data (from the base level to the aggregations) are kept in the relational model.

4.2 Main Goals

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

4.3 PostGeoOlap Metadata and Architecture

The metamodel in Figure 3 represents the metadata used by PostGeoOlap for the manipulation of the data maintained in the data warehouse. The Schema class represents the PostGreSQL database containing the DW. The `Cube` class represents each business perspective of data to be analyzed. The `Table` class represents all relations existing on data warehouse. The `Dimension` class is a subclass of `Table` and refers to all components of a cube (both the fact and dimension tables). The `Attribute` and `Field` class represents the data existing, respectively, in each dimension and table. The self-association indicates a conventional attribute as a label for a geographical one. The `Aggregation` class refers to the data aggregations implemented to improve the system performance. The `Hierarchy` class represents the several hierarchies for the attributes of a dimension. The `HierarchyItem` class allocates each attribute to a hierarchy.

The Figure 4 shows the architecture of the current implementation. PostGeoOlap is implemented in Java and it uses classes from the JUMP Unified Mapping Platform (JUMP) Java framework to perform visualization of maps and results of geographical queries. JUMP is a GUI application for presenting and processing spatial data. It has a number of functions for analysis and handling geospatial data. PostGeoOlap implementation utilizes the fact that JUMP exposes all its functionality for full programmatic access, including its map visualization classes. Other advantage of the use of JUMP is to provide a spatial object model compliant to OpenGIS Consortium specifications. This eliminates the need to map between them. The PostgreSQL representation of geographic data can be directly used as JUMP objects.

PostGeoOlap uses PostgreSQL to store the metadata, to make use of the standard spatial data types and to perform any standard SQL aggregation functions on the data

(i.e. sum, max, min, avg, count) and all geographical functions defined by the OpenGIS Consortium (i.e. `touches`, `overlaps`, `crosses`, `distance`, `within`).

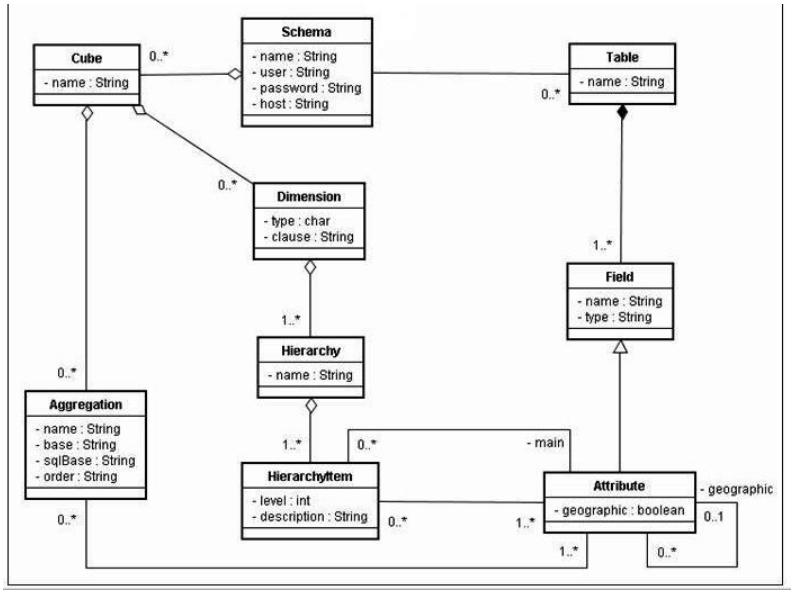

Figure 3. PostGeoOlap Metamodel

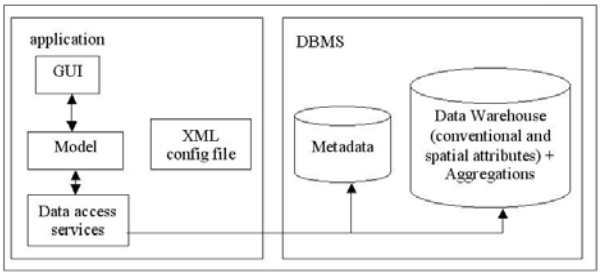

Figure 4. PostGeoOlap Architecture

4.4 PostGeoOlap Functionalities

Use cases are resumed as follows.

- *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 geographic 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. 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 (most-aggregated information). Many attributes can share the same hierarchy level.
- *Add Non-Aggregate Dimension*: 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*: provides an interface that allows the attributes selection for a query using conventional and/or geographic restrictions. It visualizes the query result as tables for analysis of non-spatial data and as maps for spatial data.

The `Create Schema`, `Create Cube` and `Add Dimension` functions build the structure of the cubes that will be verified and processed by PostGeoOlap.

4.5 Implementation Details: Cube Processing

Cube processing means the evaluation of the need of generation of aggregations to improve the performance of the queries. The big challenge faced by an OLAP tool is to obtain an acceptable performance of queries when the fact and dimension tables have a great amount of records. The cube-processing algorithm proposed in PostGeoOlap is:

```
Algorithm ProcessCube
  Input: FactAttributesCol: fact table attributes
       DimensionCol: collection of dimensions to be processed
  Begin
   DimCount=DimensionCol.Size  // number of dimensions in aggregation
   Dimension1=DimensionCol[1]  // Dimension instance
   LevelCount= minimum hierarchical level from Dimension1
   Dim1AttribCol= collection containing all attributes from Dimension1
   // create new stacks, adding to each one a level with the value from hierarchy
   Push 0 to NewStack //each stack is a sequence of elements at a level
   Add NewStack to StackCol          // StackCol: collection of stacks
   For i = LevelCount To 9       // LevelCount: number of levels
    NewStack = {}
```

```
 Push i to NewStack
 Add NewStack to StackCol
End For
While StackCol.Size <> 0
 Assigns last element from StackCol to ActiveStack and removes it from
    collection
 If ActiveStack.Size = DimCount  //9 refers to the basic level of the DW
  If not (all levels are equal to 9) //executes the aggregation
   ProcessDimension(ActiveStack) //index, if cost-benefit is not worthless
  Else  ActiveDimension = DimensionCol[ActiveStack.Size + 1]
 LevelCount = minimum hierarchical level from ActiveDimension
 Reset NewStack
 Copy to NewStack the contents of ActiveStack
 Push 0 to NewStack
 For j = LevelCount To 9
  Copy to NewStack the contents of ActiveStack
  Push j to NewStack
  Add NewStack to StackCol
 End For
End While
End
```

After the definition of the schema and the cube, the tool processes the cube to check the execution performance. If the performance of a query falls below the predefined threshold, the OLAP tool 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 Generalized Search Trees - GiST - for the spatial ones).

A complex query can delay very seconds, minutes or even some hours to be executed, depending on the volume of data. To execute a query in an arbitrary acceptable lapse of time, it is necessary to create aggregations useful for the query. Cube processing analyzes the cost/benefit relation of creating new aggregation structures for each combination of levels in each hierarchy.

Considering a cube with four dimensions and supposing each letter is a hierarchical level for each dimension, we have a structure as shown in Figure 5. The first step is the creation of a stack structure whose elements are lists. Initially, for each level of the first dimension, one list containing the level is put on stack, as shown in Figure 6(a). While the stack is not empty, it pops the list from the top and checks if the number of elements in the list is equal to the number of dimensions (in this case, 4). In negative case, it is performed the combination of this list with each level in the next dimensions. It puts on the stack one list for each level in the (next) dimensions.

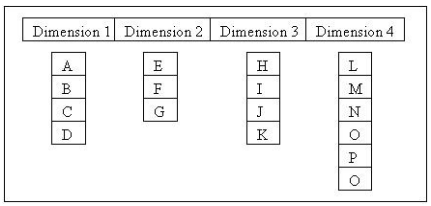

Figure 5. Dimensions to be Processed and Their hierarchic Levels

Figure 6(b) shows the second and third steps of the algorithm. The loop iterations guarantee all possible combinations for each list with the dimension levels. At the end, the number of elements in the list is equal to the number of dimensions and the element is ready to be submitted to cost/benefit analysis to generate (or not) a new aggregation structure.

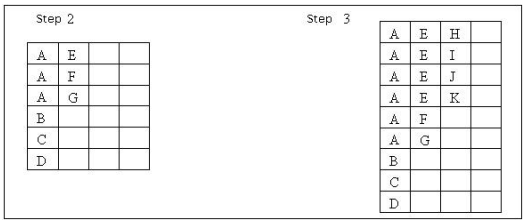

Figure 6. (a) Initial Stack (b) The Second and Third Steps of the Cube-processing Algorithm

4.6 Case Study: Magazine Retailer

As a case study is presented a magazine retailer that distributes its products (newspapers and magazines) for sale to many stores geographically distributed along regions of Rio de Janeiro State. It's very important 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)*. In order to answer it, the model uses a point stereotype associated to store, a polygon stereotype for quarter. The reference point serves as a geographical reference to the store (schools at 100 meters maximum). The results are then presented in a spreadsheet frame, and the geographical ones are passed to a visualization component for displaying the objects in a map (as shown in Figure 7).

Once specified the attributes collection and constraints even conventional or geographical predicates, the tool starts the search for aggregations from the most to the less aggregated level. Consequently, the result aggregation will be the one with the smallest computational cost. Next, it submits the query and returns the results.

5. CONCLUSION

The motivation of this project is the lack of decision support tools that allow to model data integration natively. In GeoOlap, the unified model can be directly mapped into an application where GIS and OLAP functionalities are native. It also described PostGeoOlap, an open source general-purpose tool. The purpose is to ease the work of application implementors in the development of decision support applications. A real study case about a magazine retailer was described in order to validate the tool. PostGeoOlap shows its general purpose character. It demonstrates to be an easy-to-work tool to build DSS. At the current time, tests are being prepared to evaluate the performance potential of PostGeoOlap and future work includes optimization issues at cube processing. The PostGeoOlap project is available at http://pgfoundry.org/projects/postgeoolap.

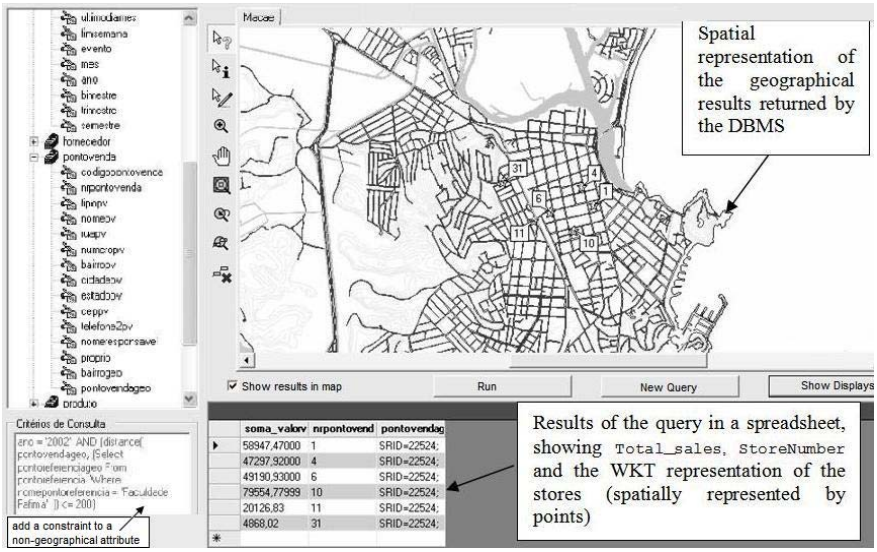

Figure 7. Query Results

REFERENCES

1. Z. Kouba, K. Matouek, and P. Mikovsk, On Data Warehouse and GIS integration, in *Proc. of the IEEE International Conference on DEXA'2000* (Greenwich, London, UK, 2000).
2. A. Ferreira, M. Campos, and A. Tanaka, *An architecture for spatial and dimensional analysis integration*, in *Proc. of World Multiconference on Systemics, Cybernetics and Informatics (SCI), volume XIV Computer Science Engineering. Part II* (2001).

3. R. Fidalgo, V. Times, and F. Souza, Golapa: Uma arquitetura aberta e extensível para integração entre SIG e OLAP, in *Proc. of III Brasilian Workshop of Geoinformática,* Instituto Militar de Engenharia, Rio de Janeiro (2001).
4. N. Stefanovic, J. Han, and K. Koperski, Selective materialization: An efficient method for spatial data cube construction, in *Proc. of the 7th International Symposium on Advances in Spatial and Temporal Databases* (ACM Records, 2001).
5. D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, Efficient OLAP operations in spatial data warehouses, *Lecture Notes in Computer Science*. Volume 21, (2001).
6. G. Colonese,*Uma ferramenta aberta de desenvolvimento integrado de sistemas de informação para processamento analítico e geográfico,* Master's Thesis, Universidade Candido Mendes, Campos dos Goytacazes/RJ, Brazil (2004).
7. J. Trujillo, M. Palomar, J. Gomez, and I.-Y. Song, Designing data warehouses with OO conceptual models, *IEEE Computer*. Volume 34, Number 12, pp.66-75, (2001).
8. E. Cunha and M.S. Sunye, Benchmarking PostgreSQL for data warehousing, in *Proc. of the IADIS International Conference on Applied Computing*, *IADIS* (Algarve, Portugal, 2005), pp.185-192.