

Designing Data Warehouses for Geographic OLAP querying by using MDA

Octavio Glorio and Juan Trujillo

University of Alicante, Spain,
Department of Software and Computing Systems
Lucentia Research Group
{oglorio,jtrujillo}@dlsi.ua.es
www.lucentia.es

Abstract. Data aggregation in Geographic Information Systems (GIS) is a desirable feature, spatial data are integrated in OLAP engines for this purpose. However, the development and operation of those systems is still a complex task due to methodologies followed. There are some ad hoc solutions that deal only with isolated aspects and do not provide developer and analyst with an intuitive, integrated and standard framework for designing all relevant parts. To overcome these problems, we have defined a model driven approach to accomplish Geographic Data Warehouse (GDW) development. Then, we have defined a data model required to implement and query spatial data. Its modeling is defined and implemented by using an extension of UML metamodel and it is also formalized by using OCL language. In addition, the proposal has been verified against a example scenario with sample data sets. For this purpose, we have accomplished a developing tool based on Eclipse platform and MDA standard. The great advantage of this solution is that developers can directly include spatial data at conceptual level, while decision makers can also conceptually make geographic queries without being aware of logical details.

Key words: Spatial data warehousing, GIS-OLAP, MDA

1 Introduction

Many years of data collection in very different systems and domains have generated significant volumes of heterogeneous information. Information that not only have to be stored, although it is required in analysis processes. Data warehouses (DW) are systems that stored and explore huge quantities of information from many data sources. Therefore, these systems are suitable to cover actual information requirement scenarios. For this purpose, the data is structured following the multidimensional paradigm. These structures are more intuitive for designers and faster for exploring tools . The most typical tools for interactively explore the data warehouse are the OLAP (On-Line Analytical Processing) systems although there are other important tools as those that implement data mining

or what-if-analysis techniques. The different components of a DW usually are depicted as a multi-layer architecture in which data from one layer is derived from data of the previous layer [10] as seen in Fig. 1.

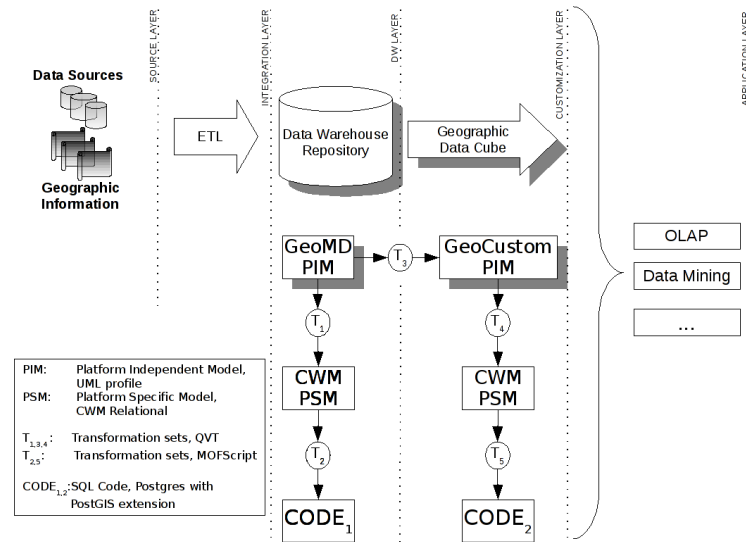


Fig. 1. Model driven development framework able to integrate geographic capabilities in multilayered spatial data warehouses.

Nevertheless, there are data that are not used to their full potential and part of their richness is simply left out that is, the spatial data. This analysis limitation is caused by the typical alphanumeric representation that is used for stored and explored this kind of data. This representation hides the spatial part of the data losing the real correlation between geometric objects that coexist in a reference system. Therefore, to gain better advantage of the spatial data in decision making, appropriate tools must be used. Geographic Information Systems are most typical systems used to capture, store, analyze, manage and present such kind of data but they are not adequate for decision-support applications when used alone. In order to overcome this lack, the Spatial OLAP (SOLAP) [18] were introduced as systems that combine the strengths of GIS, with the strengths of OLAP and provide a unique power for exploring data warehouses with spatial data.

However, the integration and querying of spatial data is still a complex task. The methodologies followed produce ad hoc solutions, extremely platform dependent and non intuitive for developers and decision makers. Also, the actual conceptual modeling of DW and customization layers (see Fig. 1) is used marginally in the development, only some designing guides are provided in order to bridge the logical level.

To overcome these problems, we have defined a model driven approach to accomplish the developing of systems that combine the strengths of GIS and data warehouses meaning a Geographic Data Warehouse (GDW). This approach has been based on MDA ¹ standard and several UML ² profiles that adapt the metamodel to certain aspects of the GDW design. Then, we have focused on the DW and customization layer, thus, we have developed two different data models for each layer. As any other data models, they are conceived to support query languages. Therefore, they may be formalized as abstract data types: data + operations. Whereas data structures are implemented by UML profiles and operations are defined by an algebra implemented in OCL ³ language.

The main advantages of integrating geographic information into multi-layer data warehouses and particularly in a model driven framework are (i) the enrichment of the decision support process while analyst can conceptually apply geographic operations over data cubes without being aware of logical details, (ii) the decrement of the design effort while developers can directly include spatial data at conceptual level, (iii) the high quality systems obtained while good practices are reused in the modeling process, and (iv) the new platform technologies easily adapt while only new mapping relations have to be developed to obtain different implementations.

The rest of this paper is structured as follows. Next section has briefly detailed some related works. Section 3 defines DW layer data model to represent spatial data following multidimensional modeling. A data model and algebra to represent and apply geographic operations over multidimensional data cubes is presented in section 4. In section 5 we show the automatic derivation of spatial repositories and geographic customized data cubes in a GIS OLAP example application. Finally, in section 6 we present our conclusions and sketch some future works.

Running Example Throughout the paper we will focus on a supermarket scenario. Sales is the required analysis fact and the context of analysis are the clients, the stores, the products and the date. Finally, we have built a GIS-OLAP application using our approach in order to analyze the scenario combining both operations. We have used real data taken from supermarket analysis examples and three hypothetical layers that describe the location of the clients, stores and hospitals. The geographic information is introduced to combine the pharmaceutical product sales with the proximity between clients, stores and hospitals. In this way, the feasibility and power of our approach to combine GIS functionalities with data warehouse analysis techniques are shown.

¹ Object Management Group, Model Driven Architecture (MDA)

² Object Management Group, Unified Modeling Language (UML)

³ Object Management Group, Object Constraint Language (OCL)

2 Related Work

There are many works related to conceptual modeling for DW. Some of the most well-known approaches are reviewed in [9] and a formal and standard model is presented there. Nowadays, the growing of spatial data plus the power that suppose combine GIS and DW functionalities, many approaches have added spatiality to their MD core model. They use the new spatial feature to represent spatial data warehouses (SDW) and SOLAP systems by adding spatial elements (i.e., spatial dimensions, aggregate functions, etc.) to the MD model. Also some algebras have been proposed in order to query the implemented spatial data.

Stefanovic et al. [20] propose one of the first spatial models defining spatial dimensions and spatial measures. Rivest et al. [18] extend the definition of spatial measures by the inclusion of measures represented as spatial objects or calculated using spatial metric or topological operators. Further, they define spatial OLAP including spatial new features grouped in data visualization, data exploration, and data structures.

Bimonte et al. [2] propose a set of requirements to obtain a consistence and powerful spatial MD model. In [19] propose *GeoCube*, a multidimensional model that integrates geographic information and ensure correct aggregation over measures associated to geometric objects. Then presents GeWolap [3], an extended model that support a web implementation of SOLAP tools. Finally in [1] propose the extension of the traditional spatial dimensions to support operations on cartographic representations improving the analysis. Gomez et al. [7] also defines a formal model for representing spatial data and a GIS-OLAP tool named *Piet* [4]. This model identify three parts in spatial dimensions, the OLAP part, the geometric part and the algebraic part. It also supports complex algorithms for improve query performance.

Malinowski et al. [11] integrates spatial dimensions, measures, facts and levels in one concise model supported by a diagram editor named *MADS*. Also introduce into this model topological types that describe spatial hierarchies. The conceptual models that describes the repository and the geographic customized cube are inspired in *MADS*. However we considered that is important to clearly separate the geographic functionalities in customization models and the geographic information implementations on repository models. Then in [13] some guidelines to obtain implementations of spatial data warehouses using *MADS* for modeling are defined. Finally in [12], possible implementations of *MADS* model into different requirement developing methodologies (*Demand-Driven*, *Supply-Driven* and *Demand/Supply-Driven*) have been presented. The framework that implements our proposal corresponds to the *Demand/Supply-Driven* approaches. In [16] we have developed a requirement methodology and in [5] a tool that reconciles the multidimensional modeling with the data sources.

Pinet et al. [17] have implemented a geographical UML profile to represent spatial data and a set of spatial operations in formally defined in OCL language. The algebra that we have proposed to geographically customize data cubes is inspired in this OCL extension. However, this work is very platform specific and only binary results are obtained from topological operations.

We believe that the above-presented approaches: (i) do not define a systematic, well structured and standard approach for the development of every data warehouse layer, (ii) do not provide methods to design the GDW in an integrated manner to avoid interoperability and integration problems between data warehouse layers, and (iii) do not define formal and standard transformations in order to unequivocally and automatically derive every possible logical representation from a conceptual model or help the designer to obtain the most suitable logical representation of the conceptual model, and (iv) final users could not conceptually query spatial data without being aware of complex logical details.

3 A Geographic Multidimensional Model

In previous work [9], we presented a UML profile for MD modeling of the DW repository. The information is elegantly organized into facts and dimensions. These facts and dimensions are modeled by **Fact** (⌘) and **Dimension** (⌘) stereotypes. A fact is composed of measures or attributes corresponding to **FactAttribute** (**FA**) stereotype. With respect to dimensions, each aggregation level of a hierarchy is specified by classes stereotyped as **Base** (⌘). Every **Base** can contain several attributes stereotypes as **DimensionAttribute** (**DA**) and must also contain a description attribute stereotyped as **Descriptor** (**D**). An association stereotyped as **Rolls-upTo** (⌘) between **Base** classes specifies the relationship between two levels of a classification hierarchy. Within it, role **R** represents the direction in which the hierarchy rolls up, whereas role **D** represents the direction in which the hierarchy drills down.

Then, in [6] we introduced the spatial level element as a hierarchy level with a geometry associated. These elements are used for correct aggregation over partially contained hierarchy levels. In this work, we also introduce a spatial measure element for support multidimensional analysis over geometric objects. These spatial level and spatial measure are modeled by **SpatialLevel** (⌘) and **SpatialMeasure** (*) stereotypes. The resultant model is referenced as *SpatialMD* and is able to represent at conceptual level spatial and non spatial data following the multidimensional paradigm for structuring and exploring the information.

In this section, we propose to introduce geographic information in previous spatial model. Geographic information is basically spatial data referenced to the *Earth* and these complex data have been represented in [6] by using spatial levels. Therefore, spatial levels are possible suitable elements for representing geographic information, they are able to represent thematic attributes and a geometric description. However, levels (spatial or not) are related to the multidimensional model by semantics constraints codified in Object Constraint Language (OCL). For example, *levels can only be associated with dimensions or levels*. On the other side, geographic information is not associated with the multidimensional elements by these semantics, these data are just related with all other data that coexist in the same reference system. This is the reason to create another conceptual element for represent the geographic information in the

multidimensional modeling with its own semantic and OCL restrictions. Thus, we have introduced a new element inspired on GIS layers, meaning a implementation of geographic information that describe a phenomenon inside a reference system. We have also introduced in this new element, a name, a geometric attribute and the possibility to add some descriptive attributes. Finally, we have implemented it in our *SpatialMD* profile creating a new stereotype named **Layer**:

Layer Stereotype

Name: Layer


Base metaclasses: Class

Generalizations: None





IsAbstract: No

Description: element inspired on GIS layers whereby geographic information is implemented.

Tag definitions: geometry (GeometricType type)

Icon: 

The **geometry** describes the geometric type of the objects represented by the layer. All the allowed geometric primitives have been grouped in a enumeration element named **GeometricTypes**. These primitives are included on ISO ⁴ and OGC ⁵ spatial standards, in this way we ensure the final mapping to platform code. The resultant data model is referenced as *GeoMD* and is able to implement correctly the geographic information in the multidimensional structure. Furthermore, this model will be used later to generate automatically an initial geographic customization model to take advantage of GIS functionalities over a data cube. In Fig. 2 it is shown the final UML profile implementation.

Based on the example scenario presented in section 1 and the profile accomplished, a conceptual description of the corresponding repository is shown in Fig. 3. There we have a fact named *Sales* (represented as ). It contains three measures named *total*, *cost*, *unities*. They are **FactAttribute** stereotyped and represented as **FA**. Then, the model have three context of analysis, named *Time*, *Product*, *Store* and *Client*. They are **Dimension** stereotyped and represented as . The geographic information modelled are distributed in three layers *Personal Data*, *Store Data* and *Hospitals*. All of them are **Layer** stereotyped and represented as . It is important to notice, that the store and client layers are also stereotyped as **Base** (represented as ) and also have a hierarchy level role. The other layer corresponds to the position of the hospitals, this is a different kind of geographic data cause it is external to the supermarket domain and the data sources of the hypothetic company.

⁴ International Organization for Standardization, <http://www.iso.org>

⁵ Open Geospatial Consortium, <http://www.opengeospatial.org>

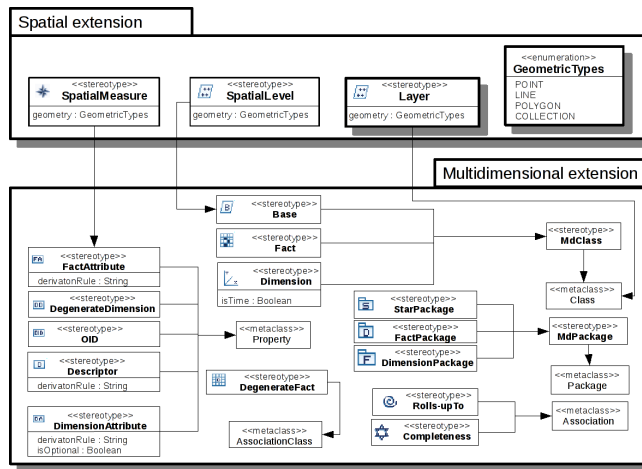


Fig. 2. Spatial multidimensional model implemented in a UML profile extension

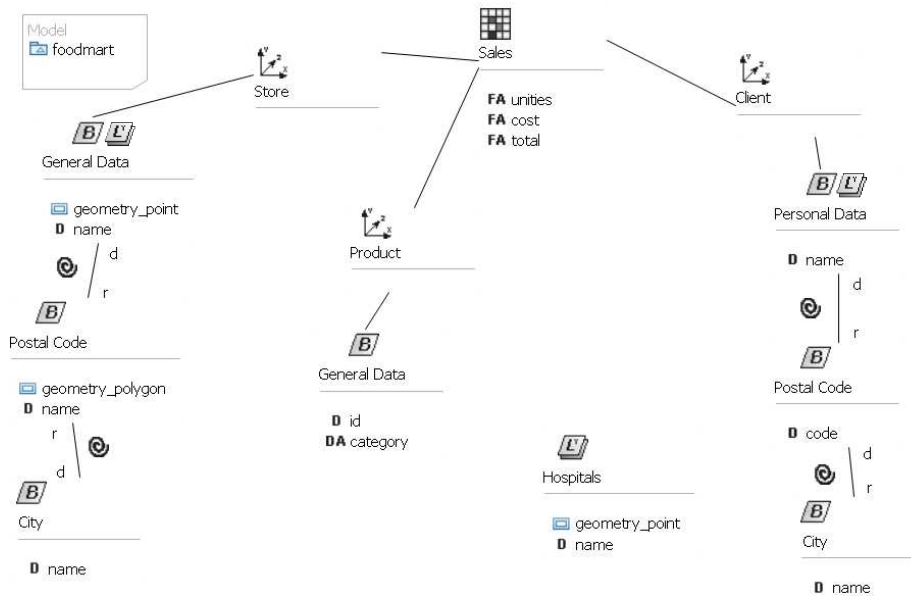




Fig. 3. Conceptual model corresponding to the running example repository

4 A Geographic Customization Model

4.1 Data Modeling

The customization layer encourages the construction and delivering of a pre aggregate data cube for the different analysis tools inside the application layer. We refer as geographic customization to the process in which the user take advantages of GIS functionalities over a data cube by using a visual model. For example, suppose a data cube corresponding to *"monthly total sales of pharmaceutical products in all supermarkets"* and the possible geographic customized data cube corresponding to *"monthly total sales of pharmaceutical products in all supermarkets located less than 500 m of a hospital"*.

First of all, in an initial model we should have the different geographic layers available (i.e., hospital and supermarket locations), this information could be easily taken from the repository model. Once we have the geographic information we should be able to establish conditions and relations. Therefore, we will need topological relations (i.e., intersect, cross, inside, etc.), a distance relation and boolean operators in order to combine the spatial relations. Finally, we should have another geometric object, that is a possible user defined area. The user could define a specific area that could be related with the other geographic objects in order to focus the analysis in this interesting area or just be involved in different geographic predicates.

In order to introduce this customization model in our approach, we have implemented it in a UML profile. To represent the geographic information that the user could relate, we have defined a new element, it has been implemented by the **ThematicLayer** stereotype and represented by . Similar to spatial levels, it has a geometric attribute that describe the spatiality of the phenomenon. The allowed geometric primitives have been grouped in a enumeration element named **GeometricTypes** as we did it in *GeoMD* model of previous section. These primitives are included on ISO ⁶ and OGC ⁷ spatial standards, in this way we ensure the final mapping to platform code. To be able to represent user defined areas we introduce an element implemented by the **UserDefinedArea** stereotype and represented by . It also has a geometric description of interesting area and the possibility to add some descriptive attributes.

In order to relate the represented geographic information, we define three kind of relations. The first one corresponds to a distance relation and it is implemented by an stereotype named **DistanceRelation**, it have an attribute (named **distance**) to define a boolean expression (i.e., *less than 500 meters* or *< 500m.*). The second one relation is introduced in order to group all the others available topological operations, it is implemented in a stereotype named **TopologicalRelation** and it have an attribute (**operator**) that define the specific topological operator. The most common operations are grouped in an enumeration named **TopologicalOperators** and more could be added easily. Finally, an **association** stereotyped as **BooleanRelation** is introduced in order

⁶ International Organization for Standardization, <http://www.iso.org>

⁷ Open Geospatial Consortium, <http://www.opengeospatial.org>

to link the different geographic relations. The stereotype has an `operator` attribute that describe a precise boolean operator (*and*, *or*, *not*). In Fig 4, it is shown a complete diagram of the UML profile extension presented.

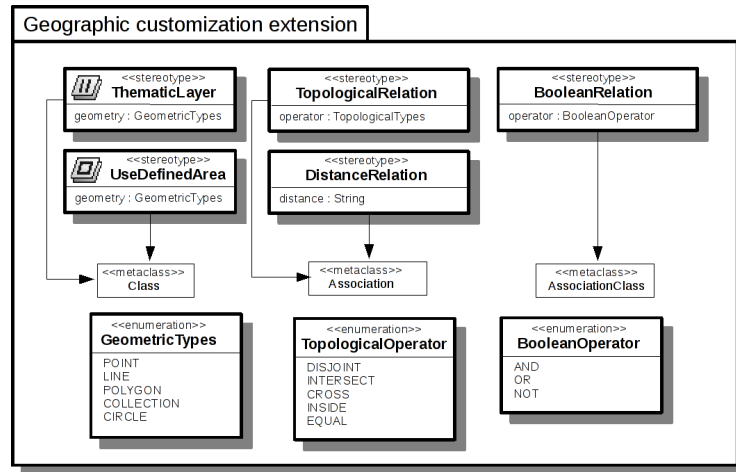


Fig. 4. Geographic customization model implemented in a UML profile extension

The initial geographic customization model is obtained using a trivial horizontal transformation of the repository model. Based on the example scenario presented on section 1, the corresponding initial model is shown in Fig. 5(a), this model represents the geographic analysis potential. There we have three geographic layers represented by *Personal Data*, *Store Data* and *Hospitals*. All of them are **ThematicLayer** stereotyped and represented as . These elements represent the geographic information which can be related by different operators and/or user-defined areas in order to perform a geographic customized cube. In Fig. 5(b), it is shown a geographic user-customized model, it restricts the data cube to “stores located less than 500 m of a hospital and inside the user defined area Area1”.

4.2 Query Algebra

In order to integrate, the previous geographic operations in development processes at the conceptual level, a customization algebra may be codified in a given programming language. Several properties may be desired for this language, some of them:

- Understandable for software engineers
- Declarative but executable
- Expressive enough for allocating geographic operations



Fig. 5. Initial and user-customized geographic models based in the running example.

- Capable of querying conceptual models

Due to the usage of UML class diagrams, OCL is an excellent candidate. It is closely integrated with UML and it has semantics that allows to query multidimensional models according to geographic restrictions. Every geographic customization starts by manipulating some data cube. Then, geographic operations are successively applied over it. This process is elegantly formalised as a closed algebra, thus, an initial data cube is needed in order to start a geographic analyses. Therefore in [14], we have defined a convenience data cubes called **grain** and a complete OLAP algebra suitable to operate over them. In present work, we have defined a geographic algebra over data cubes in order to combine GIS functionalities with OLAP analyses. Next, we have cited the **grain** definition and we refer reader to [14] for a further explanation of developed OLAP algebra.

Data cube grain. *It is a data cube containing all data cell for every fact class of the database (represented with class diagrams) at the finest granularity for all involved dimension of the fact.*

Given the **grain**, a geographic customization algebra can be applied in order to answer information needs or gain insight into data. The translation of this spatial operations into OCL is described next. Its OCL definition is shown together with an example occurrence based on running example.

Operation Definition: Geographic Customization

```
def Cube::SpatialOperation(c: Cell | gd: SpatialDice) : Cube =
  self ->select(c | gd)
```

This operation filter data cells by considering a geographic restriction over spatila data present. It is fairly implemented in OCL, by just remaining the

`select` operation of OCL `Collection` data type (over which `Cube`'s are supported).

Parameter `gd` may be bound to unrestricted OCL predicates in order to represent powerful topological and boolean operators over spatial data. The general syntax of the proposed OCL topological functions is:

`TopologicalOperator(A,B) : C`

Where `A`, `B`, `C` are the geometries of the involved spatial objects. In the case of `distance` operation, `C` is a numeric value. The available topological relations are the most common operations and more could be added easily. Those are `disjoint`, `inside`, `equal`, `cross`, `intersect` and `distance`. Turning back to customized model of Fig. 5(b), we have following presented an example:

Example Query: $C_{geo} = \text{"}C_o \text{ only in stores located less than 500 m of a hospital and inside the user defined area Area1"}$

```
c_geo->SpatialOperation (c | distance(c.store.geometry, hospital.geometry)
< 500 and inside(c.store.geometry, areal.geometry)) =
c_geo->select(c | distance(c.store.geometry, hospital.geometry) < 500 and
inside (c.store.geometry , areal.geometry) )
```

In the constraint definition, every geographic elements (`UserDefinedArea` or `ThematicLayer`), as in any class diagrams, are referenced directly by names. Nevertheless, in order to make a reference to the geometry of the element the `geometry` attribute has to be instantiated (p.e., `hospital.geometry` refers to a point objects described on hospitals layer).

5 Practical Application

In this section, we have presented an application example demonstrating how our model driven framework can be applied in practice. We have focus on three layers of our architecture, the DW layer, the customization layer and the application layer (see Fig.1). The mapping *model-to-model* and *model-to-code* have been implemented using QVT and MOFScript, since it can be applied on any MOF-compliant modeling and is part of the MDA standard. All the designing and transformation process have been done using our Eclipse⁸ based developing tool. In Fig. 6, it is shown an snapshot of the tool. There we have three windows, at left side it is shown a DW platform independent model (PIM) in a diagram editor, in the centered window there is some QVT relations code, and at right side, it is shown the resultant platform specific model (PSM) obtained in a CWM tree editor.

⁸ The Eclipse Foundation, Eclipse Ganymedes

5.1 DW layer derivation

The PIM used on the DW layer is the UML metamodel extended by the profile defined in section 3. Then, the most common representation for MD models is relational [8], thus, we assume that our PSM is a relational specific one. The PSM is modeled using the relational metamodel from CWM⁹, since it is a standard to represent the structure of data resources in a relational database, so we will be able to obtain SQL code in an easy and straightforward way from relational CWM models. Once the MDA elements of DW layer have been defined, we designed a geographic multidimensional schema based on our running example. Then, we have executed the mapping QVT relations in order to obtain a CWM model and finally, a trivial mapping is applied to obtain repository specific code using MOFScript language. The specific spatial data base (SDB) technology chosen was PostgreSQL¹⁰ with the spatial extension PostGIS.

5.2 Geographic customization layer derivation

In the customization layer, we have implemented a GIS client. Therefore, we have focused on the spatial functionalities over a multidimensional data cube. The PIM used is the UML metamodel extended by the profile defined in section 4. Using this visual query language the user clearly and intuitively defined geographic operations over the datacube. The PSM and code platforms are the same used in the DW layer, since the geographic customization model is implemented as different spatial views on the CWM metamodel or SQL language.

5.3 Application layer derivation

First, we have implemented an OLAP client. The OLAP derivation was done in [15], where metadata is generated from multidimensional modeling by using MDA. This metadata automatically configured an open source OLAP tool named Mondrian¹¹. Then, we have integrated the geographic customization derivation in a diagram editor which basically generates the new SQL spatial views by using the conceptual schema. Finally, a map viewer visualizes the geographic part of the datacube and Mondrian updates the OLAP client tool with the new customized data cube.

The GIS OLAP application developed is based on plugins for Eclipse, the same technique used for the development tool. Together with Mondrian, we have also used uDig¹² which is a GIS framework for Eclipse. In Fig. 7 it is shown a snapshot of the GIS OLAP prototype. The OLAP client is at left side of the window, on the right side it is shown a geographic customization diagram editor and the corresponding map view.

⁹ Object Management Group, Common Warehouse Metamodel (CWM)

¹⁰ PostgreSQL Global Development Group, PostgreSQL

¹¹ Pentaho Corporation, Mondrian, Pentaho analysis services

¹² Refrations Research, User-friendly Desktop Internet GIS

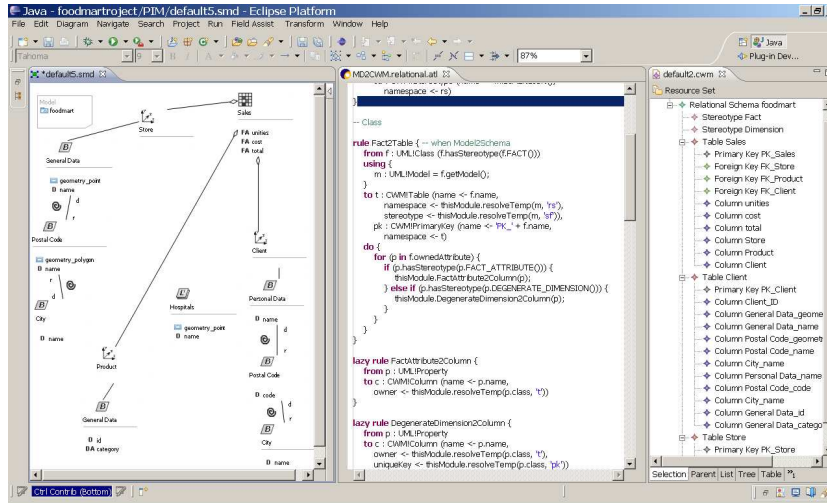


Fig. 6. Development environment for geographic data warehouses based on the Eclipse platform.

6 Conclusions

In this paper, we have presented a data model for represent and query geographic information tailored on multidimensional structure of the data and the OLAP analysis technique. Thus, we have defined formally with OCL and modelled with UML profiles the customization and the DW layers (see Fig. 1). Both models have been also integrated in in a model-driven framework to accomplish the development and based on previous work [6]. This framework addresses the design of the whole GDW system by aligning every component with the different MDA viewpoints. Thanks to the use of MDA and QVT, the development of GDWs is simplified in just two tasks: (i) the development of a conceptual models for each component; and (ii) the development of the corresponding QVT transformations to automatically generate the GDW implementation from every conceptual model developed. The great advantage of this solution is that developers can directly include spatial data at conceptual level, while decision makers can also conceptually query them without being aware of logical details.

Finally, a practical application has been presented in order to show the benefits of our proposal. We have implemented our methodology on Eclipse platform by using plugin extensions. With this developing tool we derivate a GIS OLAP application example by designing conceptual models. This application is also implemented in Eclipse platform and also use uDig as map interface (a GIS framework for Eclipse) and Mondrian as OLAP Server. In order to obtain a data implementation, we have used PostgreSQL with the spatial extension PostGIS.

Our short-term intentions include improving our MDA approach for the development of GDWs by adding other applications metadata generation. Currently,

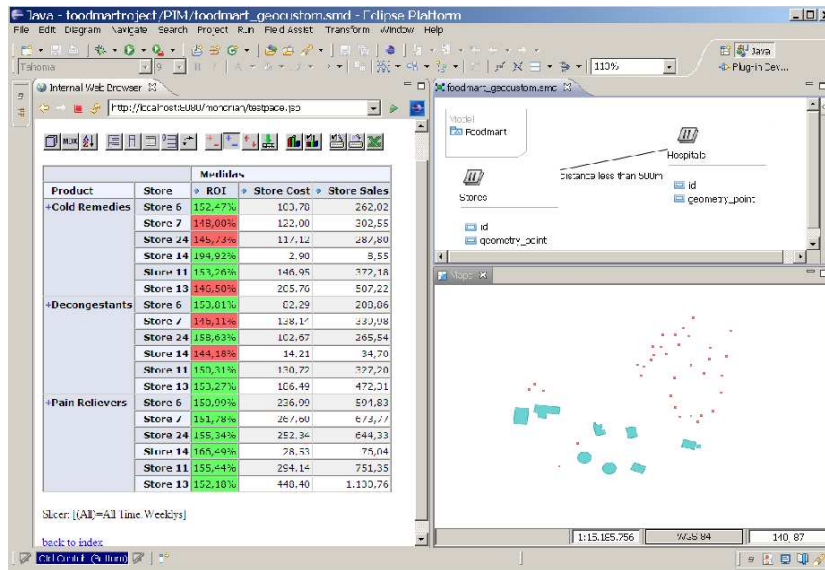


Fig. 7. GIS OLAP application example.

we are working on data mining and what-if-analysis techniques. We are also planning to merge application specific operators on a common cartographic canvas in order accomplished more intuitive and 'powerfull analysis sessions.

7 Acknowledgements

This work has been partially supported by the ESPIA project (TIN2007-67078) from the Spanish Ministry of Education and Science and by the QUASIMODO project (PAC08-0157-0668) from the Castilla-La Mancha Ministry of Education and Science (Spain). Octavio Glorio is funded by the University of Alicante under the 11th Latin American grant program.

References

1. S. Bimonte, A. Tchounikine, and M. Bertolotto. Integration of geographic information into multidimensional models. In *ICCSA (1)*, pages 316–329, 2008.
2. S. Bimonte, A. Tchounikine, and M. Miquel. Towards a spatial multidimensional model. In *DOLAP '05: Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, pages 39–46, New York, NY, USA, 2005. ACM.
3. S. Bimonte, P. Wehrle, A. Tchounikine, and M. Miquel. Gewolap: A web based spatial olap proposal. In *OTM Workshops (2)*, pages 1596–1605, 2006.
4. A. Escribano, L. Gomez, B. Kuijpers, and A. A. Vaisman. Piet: a gis-olap implementation. In *DOLAP '07: Proceedings of the ACM tenth international workshop on Data warehousing and OLAP*, pages 73–80, New York, NY, USA, 2007. ACM.

5. O. Glorio, J. Pardillo, J.-N. Mazón, and J. Trujillo. Dawara: An eclipse plugin for using i* on data warehouse requirement analysis. *Requirements Engineering, IEEE International Conference on*, 0:317–318, 2008.
6. O. Glorio and J. Trujillo. An MDA Approach for the Development of Spatial Data Warehouses. In *DaWaK*, pages 23–32, Turin, Italy, 2008.
7. L. Gomez, S. Haesevoets, B. Kuijpers, and A. A. Vaisman. Spatial aggregation: Data model and implementation. *CoRR*, abs/0707.4304, 2007.
8. R. Kimball and M. Ross. *The Data Warehouse Toolkit*. Wiley, second edition, 2002.
9. S. Luján-Mora, J. Trujillo, and I.-Y. Song. A uml profile for multidimensional modeling in data warehouses. *Data Knowl. Eng.*, 59(3):725–769, 2006.
10. Y. V. M. Jarke, M. Lenzerini and P. Vassiliadis. *Fundamentals of Data Warehouses*. Springer, 2000.
11. E. Malinowski and E. Zimányi. Representing spatiality in a conceptual multidimensional model. In *GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pages 12–22, New York, NY, USA, 2004. ACM.
12. E. Malinowski and E. Zimányi. Requirements specification and conceptual modeling for spatial data warehouses. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4278/2006 of *Lecture Notes in Computer Science*, pages 1616–1625. Springer Berlin / Heidelberg, 2006.
13. E. Malinowski and E. Zimányi. Implementing spatial datawarehouse hierarchies in object-relational dbmss. In *ICEIS (1)*, pages 186–191, 2007.
14. J. Pardillo, J.-N. Mazón, and J. Trujillo. Bridging the semantic gap in olap models: platform-independent queries. In *DOLAP*, pages 89–96, 2008.
15. J. Pardillo, J.-N. Mazón, and J. Trujillo. Model-driven metadata for olap cubes from the conceptual modelling of data warehouses. In *DaWaK*, pages 13–22, 2008.
16. J. Pardillo and J. Trujillo. Integrated model-driven development of goal-oriented data warehouses and data marts. In *ER*, pages 426–439, 2008.
17. F. Pinet, M. Duboisset, and V. Soullignac. Using uml and ocl to maintain the consistency of spatial data in environmental information systems. *Environmental Modelling and Software*, 22(8):1217–1220, 2007.
18. Y. B. S. Rivest and P. Marchand. Toward better support for spatial decision making: Defining the characteristics of spatial on-line analytical processing. *Geomatica*, 55(4):539–555, 2001.
19. Sandro Bimonte, Anne Tchounikine and Maryvonne Miquel. Geocube, a multidimensional model and navigation operators handling complex measures: Application in spatial olap. In *Advances in Information Systems (ADVIS)*, pages 100–109, Berlin / Heidelberg, Germany, 2006. Springer.
20. N. Stefanovic, J. Han, and K. Koperski. Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Trans. on Knowl. and Data Eng.*, 12(6):938–958, 2000.