

From Design to Visualization of Spatial OLAP Applications: A First Prototyping Methodology

Sandro Bimonte^{1(✉)}, Ali Hassan¹, and Philippe Beaune²

¹ TSCF, Irstea, 9 Avenue Blaise Pascal, 63178 Aubiere, France
{sandro.bimonte, ali.hassan}@irstea.fr

² Agaetis, 10 Rue Evariste Galois, 63000 Clermont-Ferrand, France
pbeaune@agaetis.fr

Abstract. The design of Spatial OLAP (SOLAP) applications consists of (i) Spatial Data Warehouse (SDW) model design and (ii) SOLAP visualization definition because a specific set of understandable and readable cartographic visualizations corresponds to a particular type of SOLAP query. Unfortunately few works investigate geovisualization issues in SOLAP systems and propose new methodologies to visualize spatio-temporal data, and no works investigate tools for readable SOLAP cartographic displays. Moreover, some works propose ad-hoc methodologies for DWs and SDWs exclusively based on data and user analysis requirements. Therefore, we present in this paper (i) a new geovisualization methodology for SOLAP queries that yields readable maps and (ii) a new prototyping design methodology for SOLAP applications that accounts for geovisualization requirements.

Keywords: Spatial data warehouse · Spatial OLAP · Geovisualization

1 Introduction

Spatial Data Warehouse (SDW) and Spatial OLAP (SOLAP) systems are first citizens of GeoBusiness Intelligence technologies. A SOLAP system has been defined as “A visual platform built especially to support rapid and easy spatio-temporal analysis and exploration of data. It follows a multidimensional approach that is available in cartographic displays, as well as in tabular and diagram displays” [1]. SOLAP systems allow for the analysis of huge volumes of geo-referenced data by simple interactive and online data exploration operators (i.e., SOLAP operators). Decision-makers trigger SOLAP operators through simple interactions with the visual components of SOLAP clients (pivot tables and graphical and cartographic displays). Therefore, they can easily and interactively explore spatial data, looking for unknown and/or unexpected spatial patterns. The success of SOLAP rests on the geovisualization analytic paradigm. Geovisualization “integrates the techniques of scientific visualization, cartography, image analysis, and data mining to provide a theory of methods and tools for the representation and discovery of spatial knowledge” [14]. Semiology rules allow the good readability of (spatial and alphanumeric) information displayed on a map. In the context of GISs, several works provide semiology tools and frameworks for readable maps [4, 19]. These

rules depend on several factors, such as the number and type (i.e., numerical, ordinal, etc.) of variables (i.e., represented information elements), and the type of geometry (points, lines, etc.). An adequate visual variable must be used for each variable, for example, for one numerical variable (e.g., the total products sold), the color visual variable (choropleth map) can be safely used [13]. Contrary to GISs, SOLAP cartographic displays are represented by interactive maps that are created online using SOLAP operators. The choice of the correct visual variable is manually performed by decision-makers during the analysis process, which represents an important limitation on effective visual analysis for two reasons: it can delay the exploration process, and decision-makers have to be geovisualization experts to choose the right visualization for the right data. Therefore, *we present in this paper a framework for the correct (readable) visualization of the results of SOLAP queries.*

Moreover, we have *integrated it in a SOLAP prototyping methodology.* The design of SOLAP applications consists of (i) SDW model design [16] and (ii) SOLAP visualization configuration. Indeed, a set of understandable and readable cartographic visualizations is needed for each particular spatial data set. Unluckily, some works propose ad-hoc methodologies for SDWs based exclusively on data and user analysis requirements [9, 16]. Therefore, we present a new prototyping design and implementation methodology for SOLAP applications that takes into account (i) user analysis requirements and (ii) geovisualization requirements. The proposed design methodology allows decision-makers to rapidly implement their SDW model and deploy it on a web-based SOLAP system [5] with well-suited cartographic visualizations. It extends the ProtOLAP DW methodology [6] that allows for the prototyping of DW models using the ICSOLAP UML profile for SDW [7]. Therefore, motivated by the relevance of conceptual representations of complex data models in prototyping tools [16], we extend ICSOLAP with various new conceptual representations of the geovisualization methods for SOLAP, and we integrate it in the ProtOLAP methodology. Finally, we implement our approach by extending our previous SOLAP tool presented in [5].

2 Related Work

Integration of spatial data in DW and OLAP systems leads to the concept of Spatial OLAP (SOLAP) [1]. SOLAP introduces the concept of spatial dimension, which is a classical dimension with geometrical attributes. Typically, SOLAP architectures are multi-tier systems composed of a Spatial DBMS (database management system) to store (spatial) data; a SOLAP server, which implements the SOLAP operators; and a SOLAP client, which combines and synchronizes tabular, graphical, and interactive maps. Existing academic and industrial systems propose the use of simple geographic visualization methods, such as choropleth maps, thematic maps, and multimaps [5, 10, 15]. Indeed, only a few works suggest particular geovisualization methods (a survey can be found in [5]). For example, [13] studied a new geovisualization method for trajectory DWs. [3] added multimedia elements, such as photos, videos, etc., to spatial data warehouse data. Finally, [12] studied the usage of chorems to enrich visual variables of SOLAP displays. To best of our knowledge, only [18] has investigated the readability

of SOLAP maps, which it accomplished by providing clustering-based SOLAP visualization methods to avoid visual cluttering.

Several works provide frameworks for the creation of readable thematic maps [1, 4]. However, apart from some simple rules based on “1 or more variables”, the existing SOLAP systems do not implement these frameworks. Therefore, they still provide decision-makers with unreadable maps and leave them to select the correct visualization for each SOLAP query. Moreover, visual variable configuration (i.e., the association of measures with visual variables) is usually conducted manually by the SDW expert using wizards. These tools do not allow for the specification of configurations based on dimensions (for example, using animated maps on the temporal dimension), and their use is quite long and tedious. Therefore, although visualization plays a fundamental role in SOLAP visual analytics and consequently in SOLAP application design, the configuration of the visual variables of SOLAP maps has yet to be included in SDW prototyping methodologies. This represents an important limitation because the success of a SOLAP project is usually related to the definition of a spatio-multidimensional model that fits users’ needs. For that reasons, several works propose DW design methodologies based on data sources and/or user requirements [16]. Among them, some rapid prototyping approaches, based on standard (e.g., ER, UML, etc.) and ad-hoc formalisms, have been developed because they yield time and economic gains [6, 8]. Finally, dashboard design and prototyping have also been investigated in some works, such as [17].

3 Motivation

In this section, using a SOLAP application developed in [12], we describe the motivation of our work. The SDW is loaded using open-data of FAO (Food and Agriculture Organization of the United Nations). It allows for analysis of the total agricultural cultivated surface and the total production per year, country and crop. It presents a spatial hierarchy grouping countries in areas and years by decade. In the remainder of the paper, we use “*nb*” to denote the maximum number of cells (pieces of information) of the resulting pivot table associated with each spatial member. For example, for the pivot table of Fig. 1d, *nb* is 2. Using this SDW, it is possible to answer queries such as “*What was the total surface of wheat per country in 1990?*” (Q1). The SOLAP query is visualized using the choropleth map shown in Fig. 1a.

Consider the following query, “*What was the total production of wheat per country per year (over the last 15 years)?*” (Q2). The result counts 15 variables (the production per year) per country (i.e., *nb* = 15). Therefore, a classical bar chart thematic map as shown in Fig. 1b appears unreadable because it conveys too much information [4]. Therefore, some other geovisualization method, such as a dynamic map [11], should be used. This problem exists because the number of visual variables that can be shown in a map is smaller than the amount of readable information shown in a pivot table. Finally, we consider another query “*What was the total production of wheat and cultivated surface per country in 1990?*” (Q3). The results of this query can be shown using a bar chart with two bars (left side of Fig. 1c). Although the map is readable in terms of visual

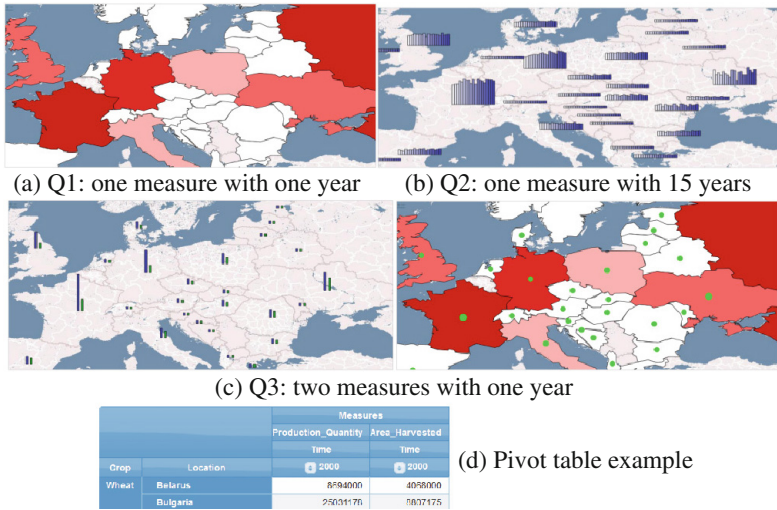


Fig. 1. SOLAP maps (Color figure online)

variable number, it is not adequate for representing two different measures with different numerical domains (hectares and tons). Therefore, the use of different visual variables for different measures is recommended [12], as shown on the right side of Fig. 1c. This means that decision-makers should be able to configure their cartographic displays according to the semantics of the warehoused data [12].

4 Framework SOLAP Visualization

To avoid the manual configuration of SOLAP cartographic displays and to avoid readability issues related to the number of visual variables (nb), we propose a new geovisualization framework based on the “display rules”. First, it is necessary to define the maximum number of pieces of information to display for one spatial member (nb_{max}). Then, a set of rules can be specified. For each rule, we must define:

Preference: determines what rule must be used if several rules are applicable.

Conditions: determines when the rule should be utilized depending on nb. Several conditions could specify:

- Range of nb: $nb = [x_1, x_2]$,
- Range of the number of measures used in the SOLAP query: $n_{Measure} = [x_{1M}, x_{2M}]$,
- Range of the number of members of each dimension (except the spatial dimension) to which the rule will be applied: $n_{di} = [x_{1di}, x_{2di}]$

Actions: determines how information will be visualized on the map (cloropeth, bar chart, dynamic map, etc.) if all conditions are achieved.

Example: In the context of the FAO example, to avoid unreadable (see Fig. 1b) or unsuitable (see the left side of Fig. 1c) geovisualization methods, we can define the rules of Table 1. R1 imposes the use of a choropleth map when only one piece of information is to be displayed in a spatial member, which corresponds to Fig. 1a ($nb = 1$). If there are two pieces of information, each for a different measure, then R2 imposes the display of the ‘Production’ measure by choropleth and the ‘Surface’ measure by circles, which corresponds to the right side of Fig. 1c. If nb is between two and six, the information can be displayed by bars (R3). R2 is preferred to R3 to prevent the geovisualization method of the left side of Fig. 1c. If one measure is to be displayed for one crop over several years, a dynamic map is used (R4); if several measures and/or several crops are used, then multi-dynamic maps should be used (R5). Using ‘*’ in the condition means that there is no limit. Let us note that these rules have been defined by SDW experts in collaboration with decision-makers.

Table 1. Display rules for the FAO example

R1	R2	R3	R4	R5
Preference = 1	Preference = 2	Preference = 3	Preference = 2	Preference = 4
Conditions	Conditions	Conditions	Conditions	Conditions
$nb = [1]$	$nb = [2]$	$nb = [2, 6]$	$nb = [2, *]$	$nb = [2, *]$
Actions:	$n_{Measure} = [2]$	Actions:	$n_{Measure} = [1]$	$n_{Time} = [2, *]$
Choropleth map	Actions:	Bars	$n_{Time} = [2, *]$	Actions:
	Production: choropleth		$n_{Crops} = [1]$	Multi-dynamic maps
	Surface: circles		Actions:	
			Dynamic map	

Once the rules have been defined, taking into account the number of visual variables to displays, the system has to guarantee that one visual display exists for each possible pivot table display. In other words, we have to define a way to verify that the defined rules allow for displaying all possible pivot tables. Therefore, for example, considering that an nb_{max} of 15 is chosen, display rules should cover all possible analyses that correspond to this number. By using a multidimensional matrix (Fig. 2a), we can present the number of pieces of information to display (nb) for each analysis according to the number of measures and members of each non-spatial dimension. In our example, $nb = n_{Measure} \times n_{Time} \times n_{Crops}$. In Fig. 2a, red cells correspond to the number of pieces of information greater than nb_{max} , and therefore the display rules do not concern these cases of analysis; contrariwise, they should cover all green cells that have a number smaller than nb_{max} . To verify this, we use the algorithm below.

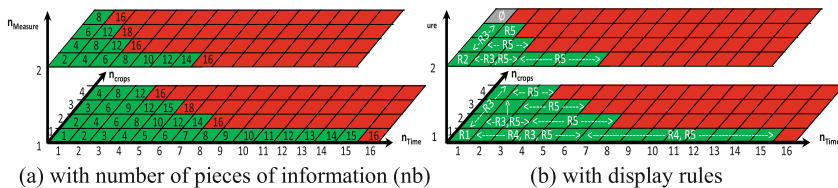


Fig. 2. Multidimensional matrix. (Color figure online)

This algorithm takes as input the display rules (Table 1) and the multidimensional matrix (Fig. 2a). It gives as results (i) a Boolean value indicating if the rules cover all required cases or not and (ii) the multidimensional matrix with the display rules (Fig. 2b). To accomplish this, the algorithm checks if each green cell (line 2) attains all conditions (lines 5–7) for each rule (line 4). If so, the rule is added to the result table (T) (lines 8–9). If the algorithm cannot find a rule to apply for a cell, the resulting Boolean value becomes ‘false’ (line 10), indicating that the rules do not cover all required cases.

The result of the application of this algorithm to our example (i.e., the display rules of Table 1 and the multidimensional matrix of Fig. 2a) is shown in Fig. 2b. This result shows that there is a case of analysis (grey cell) with $n_{\text{Measure}} = 2$, $n_{\text{Time}} = 1$ and $n_{\text{Crops}} = 4$ that is not covered by the rules. This requires either the addition of a new rule to cover this case or a modification of the conditions of an existing rule to cover it.

Algorithm verification

```

Input: Display Rules, the multidimensional matrix T
Output: T with rules, result
1: result = true
2: for each green cell  $C_i$  in T:
3:   found_rule = false
4:   for each Rule  $R_j$  in Display Rules:
5:     if ( $C_i.nb < nb_{\max}$  and  $C_i.nb \geq R_j.nb$ ) then applicable_rule = true;
6:     for each dimension  $D_k$  of T:
7:       if ( $C_i.n_{Dk} < R_j.n_{Dk}$ ) then applicable_rule = false;
8:       found_rule = (found_rule OR applicable_rule)
9:       if applicable_rule then  $C_i.add(R_j)$ ;
10: result = (result and found_rule);

```

5 Prototyping Methodology

5.1 Background

ProtOLAP has already been successfully applied in some real DW projects. ProtOLAP allows for rapid DW prototyping [6]. With ProtOLAP, decision-makers’ analysis requirements are translated by DW experts into a UML model presented in [7]. Then, this model is automatically translated into a relational model and its corresponding OLAP server model. Then, ProtOLAP allows decision-makers to feed the DW some sample data. Finally, decision-makers interact with a real OLAP client to validate the multidimensional model.

Encouraged by time and important economic gains associated with the usage of conceptual models (UML, ER, etc.), ProtOLAP is based on the ICSOLAP UML profile. ICSOLAP allows a conceptual representation of spatio-multidimensional models using UML stereotypes [7]. Indeed, a profile in the Unified Modeling Language (UML) provides a generic extension mechanism for customizing UML models for particular domains and platforms. Stereotypes, tagged values, and constraints are used to adapt UML elements to a specific application. Finally, Object Constraint Language (OCL) constraints are used to specify rules to verify the validity of a stereotype. UML profiles can be easily implemented in computer-aided software engineering (CASE) tools, such

as MagicDraw or Eclipse. ICSOLAP contains stereotypes for each spatio-multidimensional element. A «Fact» is composed of «Measure». An «AggLevel» is composed of dimensional attributes and can be thematic, spatial or temporal. Moreover, the stereotype «BasicIndicator» defines aggregation rules for a given measure (i.e., “aggregatedAttribute”). It indicates the functions used in the aggregation process along dimensional 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 for our case study is shown in Fig. 3.

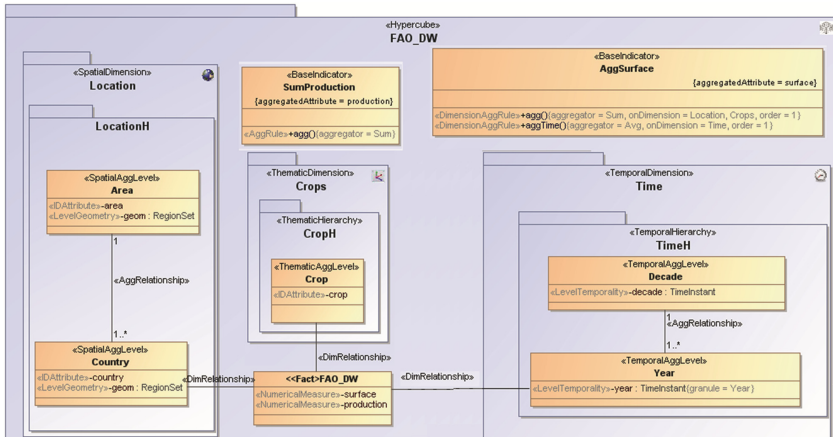


Fig. 3. FAO Spatial Data Warehouse

5.2 SOLAP Prototyping Methodology

To take SOLAP geovisualization issues into account in a prototyping methodology, we propose the following methodology *extending ProtOLAP with steps 3, 4, 6 and 9*.

1. *Informal definition of indicators:* Decision-makers define the analysis needs in terms of dimensions and measures.
2. *Conceptual design:* DW experts translate decision-makers’ spatio-multidimensional needs from step 1 using ICSOLAP [7] (e.g., Fig. 3).
3. *Informal definition of geovisualization methods:* Decision-makers define how to visualize their data using cartographic displays.
4. *Extending the conceptual design:* DW experts translate decision-makers’ geovisualization needs from step 3 into a UML model extending [7] (e.g., Figs. 4 and 5).
5. *SDW implementation:* The system automatically creates the DBMS and the OLAP server schemata.
6. *SOLAP visualization implementation:* The system automatically creates the geovisualization schemata.
7. *Domain feeding data:* Decision-makers feed the SDW with sample data.

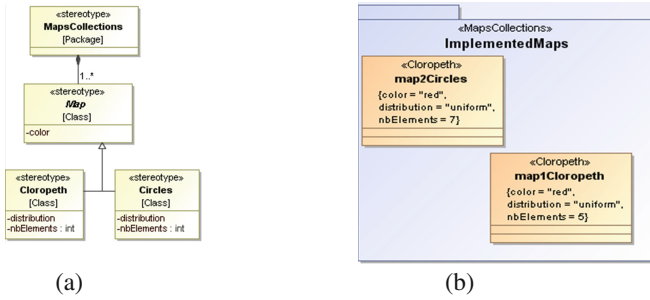


Fig. 4. UML profile for map representation: (a) meta-model, (b) example

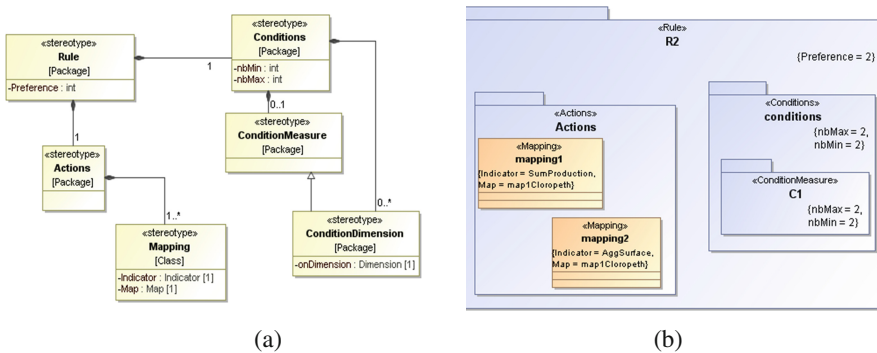


Fig. 5. UML profile for display rules: (a) meta-model, (b) example of R2

8. *OLAP-based indicator validation*: Decision-makers validate the dimensions and measures of the SDW. If the spatio-multidimensional model has not been validated, go to step 1.
9. *SOLAP-based geovisualization validation*: Decision-makers validate the geovisualization methods for SOLAP query results. If the geovisualization methods have not been validated, go to step 3.
10. *ETL implementation*: During this last step, the ETL is implemented to load the SDW defined in step 5.

As described in step 4, our methodology uses an extension of ICSOLAP to conceptually represent geovisualization methods and rules. This extension is described in the remainder of the section.

A map (≪Map≫) is defined as an abstract class, and its implementations represent the geovisualization methods implemented in the SOLAP system, as shown in Fig. 4a. Here, the two stereotypes ≪Cloropeth≫ and ≪Circles≫ have been defined to represent cloropeth and thematic maps with circles in a generic manner. A cloropeth map is described by a color range (“color”), a number of color classes (“nbElements”) and a distribution function. An example is shown in Fig. 4b, where a cloropeth map (i.e., *map1Cloropeth*) that uses 5 classes of the color red with a uniform distribution (e.g., Fig. 1a) is implemented.

To define display rules as defined in Sect. 4, we define some new stereotypes as shown in Fig. 5a. A rule (defined as a «Rule» package) is composed of a set of conditions («Conditions») and actions («Actions»). A condition is defined as a package and presents the minimum and maximum number of elements (“nbMin” and “nbMax”); a condition can be defined on measures («ConditionMeasures») and on dimensions («ConditionDimension» presents a tagged value with the ICSOLAP «Dimension» type). An example for rule R2 of Table 1 is shown in Fig. 5b. An action is defined as a package containing a set of «mapping» classes (Fig. 5b). A mapping is a class that contains two tagged values representing an «Indicator» ICSOLAP and a «Map», and it is used to define what geovisualization method is used for each indicator. For example Fig. 5b indicates that, when the conditions are verified, the two indicators *SumProduction* and *AggSurface* of Fig. 3 are visualized with choropleth maps. Our profile has been implemented in MagicDraw.

6 Implementation

In this section, we present the implementation of our proposal.

Based on our previous work [5], the architecture of our prototype consists of three tiers: the SDW tier, SOLAP Server and SOLAP Client. The SDW tier is implemented using PostGIS, which is a spatial DBMS. This tier is responsible for storing alphanumeric and spatial multidimensional data. The OLAP server used is Mondrian. The SOLAP client tier is composed of the Pivot4J OLAP client and OpenLayers GIS client. OpenLayers has been integrated in the ProtOLAP architecture [6]. For the implementation of the geovisualization methods represented by the «Map» stereotype, we have defined a set of SLD (Styled Layer Descriptor) and GML (Geography Markup Language) templates (more details in [5]). GML and SLD are XML-based representations of spatial data with visualization. SLD and GML templates provide an implementation of the geovisualization methods that is not dependent on the SOLAP client used. Therefore, according to our prototyping methodology, the SLD and GML templates can be automatically generated from UML diagrams.

Because [5] does not support display rules, we extend it by adding an XML representation of displays rules as described in Sect. 4. These rules have been integrated in the SOLAP client and are automatically triggered during each SOLAP query. These XML files correspond to the «Rule» packages previously described.

A video example of a SOLAP application with display rules is shown at <https://www.youtube.com/watch?v=ZHUTqVRtKu8>. Decision-makers visualize their pivot table and their associated bar thematic maps. Then, changing the pivot table by 2 measures and one year, the cartographic visualization is automatically adapted using rule R2 as previously described.

7 Conclusion

Motivated by the importance of geovisualization tools in SOLAP analysis, we present a methodology for prototyping SOLAP applications. Our current work involves the automatic implementation of the UML profile and the evaluation of the methodology using the Goal Question Metric framework. Moreover, because decision-makers are not always GIS experts, we will define a methodology to automatically derive maps and display rules from the SDW schema.

Acknowledgement. This work is supported by the CAPTIVEN project of the ANR 11-CNRT-0003 program “investissements d’Avenir - valorisation PME”.

References

1. Andrienko, G., Andrienko, N.: Interactive maps for visual data exploration. *Int. J. Geogr. Inf. Sci.* **13**(4), 355–374 (1999)
2. Bédard, Y., Han, J.: Fundamentals of spatial data warehousing for geographic knowledge discovery. In: *Geographic Data Mining and Knowledge Discovery* (2001)
3. Bédard, Y., Proulx, M., Rivest, S., Badard, T.: Merging hypermedia gis with spatial on-line analytical processing: towards hypermedia SOLAP. In: Stefanakis, E. (ed.) *Geographic Hypermedia*, pp. 167–187. Springer, Heidelberg (2006)
4. Bertin, J.: *The Semiology of Graphics*. University of Wisconsin Press, New York (1983)
5. Bimonte, S.: A generic geovisualization model for spatial OLAP and its implementation in a standards-based architecture. *ISI* **19**(5), 97–118 (2014)
6. Bimonte, S., Edoh-Alove, E., Nazih, H., Kang, M., Rizzi, S.: ProtOLAP: rapid OLAP prototyping with on-demand data supply. *DOLAP* **2013**, 61–66 (2013)
7. Bouليل, K., Bimonte, S., Pinet, F.: Conceptual model for spatial data cubes: A UML profile and its automatic implementation. *Comput. Stand. Interfaces* **38**, 113–132 (2015)
8. Corr, L., Stagnitto, J.: *Agile Data Warehouse Design: Collaborative Dimensional Modeling, from Whiteboard to Star Schema*. DecisionOne Press, Leeds (2011). ISBN: 9780956817204
9. Ezzedine, S., Turki, Y., Faïz, S.: An approach based on the clustering of spatial requirements’ models and MDA to design spatial data warehouses. *IJDMMM* **7**(4), 276–292 (2015)
10. Golfarelli, M., Mantovani, M., Ravaldi, F., Rizzi, S.: Lily: a geo-enhanced library for location intelligence. In: Bellatreche, L., Mohania, M.K. (eds.) *DaWaK 2013*. LNCS, vol. 8057, pp. 72–83. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-40131-2_7](https://doi.org/10.1007/978-3-642-40131-2_7)
11. Harrower, M., Fabrikant, S.: The role of map animation in geographic visualization. *Geographic Visualization: Concepts, Tools and Applications*, 49–65 (2008)
12. Johany, F., Bimonte, S.: A framework for spatio-multidimensional analysis improved by chorems: application to agricultural data. In: Helfert, M., et al. (eds.) *DATA 2015*. CCIS, vol. 584, pp. 59–80. Springer, Heidelberg (2016). doi:[10.1007/978-3-319-30162-4_5](https://doi.org/10.1007/978-3-319-30162-4_5)
13. Leonardi, L., Orlando, S., Raffaetà, A., Roncato, A., Silvestri, C., Andrienko, G., Andrienko, N.: A general framework for trajectory data warehousing and visual OLAP. *GeoInformatica* **18**(2), 273–312 (2014)
14. MacEachren, A., Gahegan, M., Pike, W.: Geovisualization for knowledge construction and decision support. *IEEE Comput. Graph. Appl.* **24**(1), 13–17 (2004)
15. Malinowski, E.: *GeoBI Architecture Based on Free Software*. Geographical Information Systems Trends and Technologies. Elahé Pourabbas CRC Press, New York (2014)

16. Malinowski, E., Zimányi, E.: *Advanced Data Warehouse Design From Conventional to Spatial and Temporal Applications*. Springer, Heidelberg (2008)
17. Palpanas, T., Chowdhary, P., Mihaila, G., Pinel, F.: Integrated model-driven dashboard development. *Inf. Syst. Front.* **9**(2–3), 195–208 (2007)
18. Silva, R., Moura-Pires, J., Santos, M.: Spatial clustering in SOLAP systems to enhance map visualization. *IJDWM* **8**(2), 23–43 (2012)
19. Southall, H.: Visualization, data sharing and metadata, in *geographic visualization: concepts, tools and applications*. In: Dodge, M., McDerby, M., Turner, M. (eds.) (2008)