Lily: A Geo-Enhanced Library for Location Intelligence

Matteo Golfarelli[1], Marco Mantovani[2], Federico Ravaldi[2], and Stefano Rizzi[1]

[1] DISI – University of Bologna,
V.le Risorgimento 2, 40136 Bologna, Italy
{matteo.golfarelli,stefano.rizzi}@unibo.it
[2] iConsulting Srl,
Via Bazzanese 32/7, 40033 Casalecchio di Reno (BO), Italy
{m.mantovani,f.ravaldi}@iconsulting.biz

Abstract. Location intelligence is a set of tools and techniques to integrate a geographical dimension into BI platforms, aimed at enhancing their capability of better monitoring and interpreting business events. Though most commercial data warehouse tools have implemented spatial extensions to support GIS integration, the user experience with spatial data is still mostly limited to the visualization of maps labeled with numerical indicators. To overcome this limit we developed *Lily*, a geo-enhanced library that adds true location intelligence capabilities to existing BI platforms. Lily provides end-users with a highly-interactive interface that seamlessly achieves a bidirectional integration between the BI and the geospatial worlds, so as to enable advanced analytical features that truly take into account the spatial dimension. In this paper we describe Lily from a functional and architectural point of view, and show an example where Lily is coupled with the Oracle Suite to be used for location intelligence in the field of telecommunications.

Keywords: Spatial data warehouse, Business intelligence, GIS.

1 Introduction

Over 80% worldwide companies take their business decisions based on data characterized by a spatial component [6]. This fact, together with the ongoing success of online services based on geo-localization such as Google Maps and Facebook Places, and with the widespread diffusion of smartphones and tablets (both capable of detecting the owner's exact location) has contributed to raise the interest of decision-makers in spatial analyses, that allow the relationship between events and their territorial distribution to be precisely established and effectively examined. In response to this need, *spatial data warehouses* have been emerging as an enabling technology for spatial analyses. A considerable research on spatial data warehouses has been done over the last years; the specific topics investigated range from conceptual modeling for spatial data [10,3] to spatial OLAP operators [11,4] and efficient processing of spatial data [17,12].

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2013, LNCS 8057, pp. 72–83, 2013.

In much the same way as business intelligence (BI) tools build on traditional data warehouses to overcome the limits of OLAP analyses and enable more advanced techniques such as data mining and what-if analysis, there is a need for building sophisticated applications on spatial data warehouses to benefit from spatial data in a broader sense. In this direction, we use the term *location intelligence* to denote a set of tools and techniques that integrate a geographical dimension into BI platforms, aimed at enhancing their capability of discovering patterns, opportunities, and risks, and at better monitoring and interpreting business events with specific reference to their territorial distribution. While geographical information systems (GISs) are used by expert users for operational tasks to achieve tactical benefits, location intelligence is meant to be used by business users for decision-making tasks to achieve strategical benefits.

The human mind processes visual patterns 60,000 times faster than tabular lists. So, location intelligence is really useful when huge amounts of data can be aggregated into numerical indicators whose graphical positioning on an interactive map allows spatial patterns to emerge more clearly than a simple tabular representation or a chart. On the other hand, using BI techniques to investigate complex phenomena on a map fed with consistent and integrated data leads to leave behind the purely spatial analyses made possible by GISs. In general, location intelligence can be applied whenever there is a need for discovering meaningful correlations between phenomena described by indicators that share nothing but spatial proximity.

Spatial extensions have been implemented on most commercial data warehouse tools in order to integrate them with GISs [8,9]. However, the user experience with spatial data is still mostly limited to the visualization of maps labeled with numerical indicators of business, which hardly enables decision-makers to take full advantage of the huge information power lying with geography. To overcome this limit we developed *Lily* (Location Intelligence Library), a geo-enhanced library that adds true location intelligence capabilities to existing BI platforms. Lily is written in Javascript and and leverages the AJAX technologies to provide end-users with a highly-interactive interface that seamlessly achieves a bidirectional integration between the BI and the geospatial worlds, so as to enable advanced analytical features (such as what-if analysis and data mining, etc.) taking truly into account the spatial dimension.

In this paper we describe Lily from a functional point of view (Section 4), show an architecture for coupling Lily with the Oracle Suite (Section 5), and discuss an example where Lily is used for location intelligence in the field of telecommunications (Section 6). The paper outline is completed by Section 2, that discusses the related works, Section 3, where the main industrial solutions for location intelligence are classified, and Section 7, that draws the conclusions.

2 Location Intelligence in the Literature

Although researches in spatial data warehouse have been pioneered since 1997, when the term *SOLAP* —Spatial OLAP— has been coined [1], only in recent years the research community thoroughly investigated the integration of GIS and OLAP technologies. The main topics that have been investigated are the *architectures*, the *data models*, and the *operators and algorithms* necessary to express and optimize SOLAP queries.

As to the architectures, in [2] the authors classify the solutions according to the type of coupling between the GIS and data warehouse components. *Loosely-coupled* architectures [7] rely either on import-export-reformatting or mapping of data between GIS and OLAP, *semi-tightly coupled* ones are either GIS-dominant or OLAP dominant solutions, while *tightly-coupled* architectures rely on fully integrated spatial OLAP technology. While the last type of architecture provides for larger querying capabilities, the supporters of the first type assert that it favors autonomy, updating, and maintenance of the databases. According to [13] a tight-coupling architecture should be organized on three-levels, characterized by a multidimensional/spatial query engine that translates visual queries to a relational-multidimensional DBMS supporting spatial measures and attributes. In this direction the GeoMDQL architecture is based on a spatial data warehouse, a multidimensional and spatial engine, and a GUI [16].

Research on data models is aimed at enriching the multidimensional model to properly support SOLAP queries. Stefanovic et al. classify spatial dimension hierarchies according to their spatial references in non-geometric, geometric to non-geometric, and fully geometric [17]. Malinowski and Zimányi focus on the conceptual representation of a spatial cube, that is obtained by extending their MultiDimER model with pictograms that represents spatial dimensions, levels, measures, and relationships [10]. In [16], a UML-based formalization for a spatial data warehouse meta-model is proposed that includes a large set of stereotypes modeling the spatial concepts (e.g., points, polygons, and lines).

As to the topics related to querying, many efforts have been done to increase the expressiveness of traditional OLAP operators and languages. In [16] the authors propose GeoMDQL, a query language based on MDX [18] and OGC (Open Geospatial consortium) for querying spatial data cubes. Also GISOLAP-QL [7] extends MDX, but while GeoMDQL relies on a tightly-coupled architecture, GISOLAP-QL relies on a loosely-coupled one. The type of coupling is reflected in the query structure, that is composed of two parts divided by a pipe: the GIS part returns the objects that satisfy a given condition, while the OLAP part performs an aggregation over the resulting objects. In [3,4] the authors propose the GeoCube algebra that extends SOLAP with five new operators, i.e., classify, specialize, permute, OLAP-buffer, and OLAP-overlay. In addition to classical drill and slice OLAP operators, GeoCube provides two operators for navigating the measure hierarchy and two spatial analysis operators that dynamically modify the structure of the spatial cube.

Though several Spatial OLAP Visualization and Analysis prototypes have been developed, they have been mostly used for verifying the expressiveness of the models and languages discussed so far and do not provide advanced GUIs. The GeoMDQL GUI [18] extends JPivot and is capable of displaying results in charts, tables, and maps but it is not described in detail, making it hard to understand how advanced the interaction with the system is. SOVAT [15] combines OLAP and GIS capabilities; its interface supports spatial drill-down and roll-up operations as well as *drill-out*, that allows users to submit queries based on both numerical and spatial aggregation. Finally, in [14] a two-level prototype (JMap) is proposed where the GUI and the query engine are collapsed in the same module. JMap supports both tabular and diagrammatic views; different types of maps can be shown, in particular, multimaps, complex thematic maps, and maps with superimposed diagrams. JMap also allows synchronization of operations and symbols from one display to another, so the same information can be visualized from different perspectives.

3 Location Intelligence in the Industry

In most BI implementations, location information is present in the form of a coarse-grained (and often static) dimension [5]. Of course, this is not sufficient for location intelligence. So, during the last few years, the main vendors of BI systems and GISs have been progressively moving towards location intelligence. The basic solutions consists of a set of static maps (usually based on pre-defined SVG images), graphically annotated with simple indicators and used together with traditional reports for a more effective visualization of information. They are specifically built for specific analyses, poorly reusable, and with limited or no interaction. Examples of this kind of solutions can be found in many reporting tools such as Hyperion Web Analysis, Microsoft Reporting Services, and Business Objects Xcelsius.

Some more advanced solutions provide a communication interface between legacy BI platforms and GISs by enabling them to share the same analysis context. These solutions have started emerging recently, as a result of joint collaborations between BI and GIS vendors. This is the case of APOS LIS (APOS Location Intelligence Solution), that integrates SAP Business Objects and ESRI ArcGIS; of Business Geografic GeoQlik and GeoBI, that integrate a custom GIS with QlikView and SAP Business Objects respectively; of the bridges by Galigeo, that make their GIS communicate with BI plaftorms such as SAP Business Objects, IBM Cognos, and Microsoft; of ESRI Maps for IBM Cognos. Though bridges offer advanced capabilities for interacting with maps (they use fully-featured, state-of-the-art GIS tools), they do not achieve a true integration between spatial and business data, because users still operate on two separate systems: the BI tool contains all the business data, while spatial information resides on the GIS side.

The most advanced solutions rely on fully-integrated spatial OLAP technology to provide larger querying capabilities and better scalability and efficiency. The basic idea is that spatial data are stored together with business data in a spatial data warehouse; the main solutions in this category are Oracle Database (with its Spatial option), Microsoft SQL Server (the recent 2012 release introduces many new spatial features), and PostgreSQL (with its PostGIS extension). A unified front-end may then be available to produce fully-integrated OLAP-spatial reporting, like done by Oracle Business Intelligence in the Oracle suite.

4 Lily: A Functional View

With reference to the design cycle of a location intelligence system, Lily comes into play during the phase where user applications are developed. The functionalities provided by Lily can be classified as follows:

1. **Geo-enhanced query formulation**.
   - *Spatial drill*, that provides guided geography-based navigation of a domain of interest from a high-level perspective to the deepest detailed information available. This enables an analysis of a given numerical indicator either by means of a standard territorial hierarchy —beginning with the country level to proceed with Nielsen areas, regions, provinces, municipalities, and so on— or by implementing a custom-based hierarchy (custom sales areas, districts, etc.).
   - *SOLAP queries.* This function supports the formulation of queries by actively involving the geographical dimension. In particular, proximity-based slice&dice enables proximity analyses of business phenomena, such as the selection of a set of points of sales, real estate properties, or crime locations. End-users may use prefixed shapes or draw custom areas directly onto a map and use this selection tool as a way to filter the subjects of interest or as a benchmark for comparison purposes. In this way the system can answer questions such as: *Show the locations of the crimes that were suffered from victims aged less than 18 within a 10-mile radius from a school (or a river, a car park, a sports centre, a tube station, etc.).*
   - *Geo-coding.* The geocoding process is necessary to handle points as spatial objects, and consists in translating an unstructured alphanumerical address into a couple of latitude-longitude coordinates to store them into a spatial point format, and vice versa, i.e., inferring the address of a place from its geographic coordinates. Lily can leverage different geocoding mechanisms: the most commonly used is through the public web-services exposed by vendors such as Nokia, Tomtom, and Google (the web service accepts alphanumerical addresses as input and returns latitude/longitude coordinates together with quality information, e.g., exact match, street-level match, city-level match); another option, more efficient for large amounts of data, is to exploit the native geocoding capabilities of spatial-enabled DBMSs (in this case, address-level information must be loaded into the database).
2. **Geo-enhanced processing**.
   - *Spatial triggering.* This feature enables end-users to actively interact with maps by triggering custom scripts, which allows developers to associate complex processing (such as data mining, forecasting, what-if analysis, etc.) to any map area. Custom complex algorithms can be implemented to leverage mathematical models or data mining features, so it is

possible to simulate scenarios and gain forecast insights for questions like: *What are the distinguishing features of the areas with highest crime rate? Which areas show these features to some extent? Are the schools in a specific district ready for plan enrollment and student services for next year? Where is the best location for a new store, based on the market territory potential?*. In general, it is possible to trigger any custom action (an algorithm or just a specific visualization) in response to the selection of any feature of interest on the map.

– *Spatial clustering.* This function enables the detection and graphical representation of clusters of features of interest to provide an holistic view of a phenomenon, such as the distribution of residents, pollution, politicians, or any custom function that combines elementary data onto a wide geographic area. The same concept can be applied to *in-doors analysis*: *How do customers move inside a shopping centre, an airport, a museum, or a casino? How do customers visit my stores —i.e., where do they go, how long do they remain in a specific section, which areas are rarely visited, etc.?*

3. **Geo-enhanced data visualization**.

   – *Integration with external maps.* External maps are typically used as background imagery, or to visually identity correlations between different data sets. They are integrated at run-time as raster images (tiles), in the form of layers that can be stacked below (or on top of) the internal vector (interactive) or raster (static) themes; to achieve good performances, all vector themes must be based on spatial data residing on the spatial data warehouse. Lily is compatible with several map providers such as Google Maps, Bing Maps, OpenStreetMap, Ordinance Survey and, in general, with all providers using the Web Map Service (WMS) and Tile Map Service (TMS) standards. Using the same mechanism, it is also possible to interact with custom base-maps hosted on in-house GIS systems.

   – *Real-time refresh.* Lily supports real-time visualization of spatial data, not only through periodical refresh of the whole map, but also by triggering updates of colors or by smoothly reproducing the movements of objects (e.g., for vehicle tracking purposes), using the asynchronous AJAX technology. This is useful for instance to monitor territory sensors in real-time, or when monitoring road traffic accidents and congestions.

   – *Multi-layer representation.* Several layers can be overlapped onto the same map to represent different but complementary phenomena such as the distribution of residents, customers, roads, stores, and competitors. These layers can be related to different dimensions or different hierarchical levels.

   – *Spatial KPI visualization.* KPIs (*key performance indicators*) can be represented by means of complex visualizations, such as bar charts or pie charts properly positioned in the areas of a map. Moreover, it is possible to view the distribution of specific phenomena exploting the visual power of choropleth maps or heat maps.

- *Temporal slider.* It is possible to animate a map, by showing the evolution of a phenomenon over time and seeing the trend based on the variation of a layer. For example, a dashboard can monitor crimes over time in a specific area (e.g., to check the effects of a preventive action), or the amount of investments made for each branch can be analysed over time.

In Table 1 we show which of these functionalities can be supported by each architecture for location intelligence. Of course, not all the commercial tools that adopt a given architecture actually support all the indicated functionalities; for instance, Oracle with Spatial option adopts a tightly-coupled architecture but does not support real-time refresh and multi-layer representation.

Table 1. Functional support on different architectures for location intelligence

	Spatial drill	SOLAP queries	Geo-coding	Spatial triggering	Spatial clustering	Integr. with ext. maps	Real-time refresh	Multi-layer repr.	Spatial KPI visual.	Temporal slider
Loosely-coupled	✓					✓		✓*	✓	
OLAP-dom. semi-tightly coupled						✓		✓*	✓	
GIS-dom. semi-tightly coupled	✓		✓			✓	✓	✓*	✓	
Tightly-coupled	✓	✓	✓			✓	✓	✓	✓	✓

* Only at a single aggregation level.

5 Lily: An Architectural View

Though Lily is inserted into a tightly-coupled architecture to achieve great performances and advanced functionalities (e.g., active spatial queries and spatial drill), it follows a mash-up approach, whose main advantages are a fast deployment and the possibility of reusing existing services by transparently integrating them within a single user interface. In particular, Figure 1 shows how, within the current Oracle architecture for location intelligence, the Lily component can be interposed between Oracle Business Intelligence (OBI, it includes a presentation front-end and a multidimensional engine called Oracle BI Server) and MapViewer (the Oracle component in charge of map rendering and of supporting the connection between spatial and business data). Lily is currently implemented to be specifically connected with MapViewer; however, any other presentation service (e.g., Business Objects or Microstrategy) can be used instead of OBI provided that it supports web-based reporting.

Figure 2 shows a functional view of the process though which queries are answered in a Lily-based architecture. Assuming that in all the reports shown to end-users we can separate *non-spatial items* (e.g., cross-tables and diagrams) from *spatial items* (e.g., a heat map), two relevant scenarios can be distinguished:

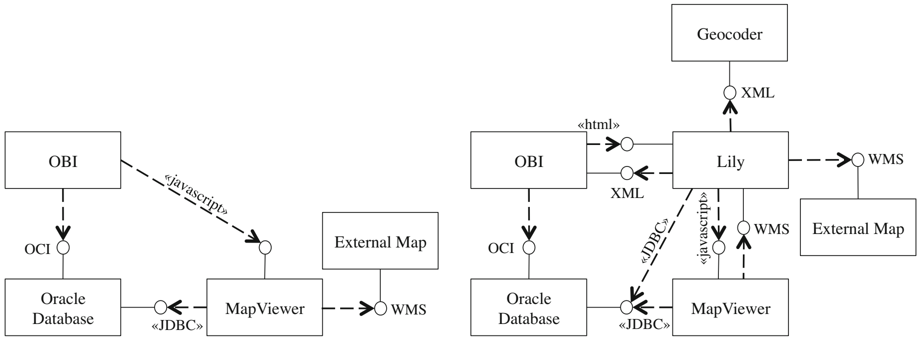


Fig. 1. Oracle-based architecture for location intelligence with —right— and without —left— Lily (OCI: Oracle Call Interface; WMS: Web Map Service)

- *OLAP queries.* These are queries that end-users formulate by interacting with a non-spatial item, so that spatial items are only involved in visualizing the results. An OLAP query q is sent to Oracle BI Server, that translates it into SQL. The resulting SQL query is sent to Oracle Database that returns the corresponding data to Oracle BI Server, that in turn sends these data to the front-end to feed non-spatial items and also routes them to MapViewer. At the same time, the rendering statements associated with q are sent to Lily that translates them into a SQL spatial query q_{spa}. Oracle Spatial & Graph answers q_{spa} and returns the corresponding spatial data to MapViewer that, based on the intensional mappings provided by Lily, integrates them with the non-spatial data received by Oracle Database to build an integrated rendering for spatial items.
- *SOLAP queries.* These queries are formulated by interacting with a spatial item. Lily translates SOLAP queries into SQL and sends them to MapViewer,

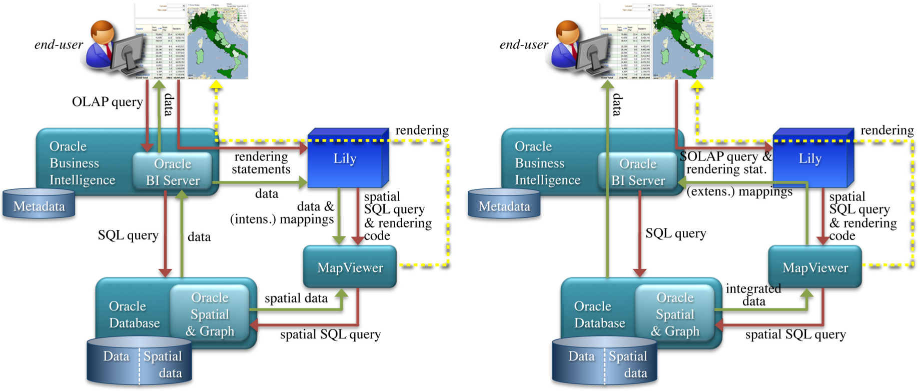


Fig. 2. Functional view of how OLAP (left) and SOLAP (right) queries are answered using Lily

that in turn routes them to Oracle Spatial & Graph for answering. The integrated (spatial and non-spatial) data returned are sent to MapViewer that builds the integrated rendering for spatial items and, based on the instructions provided by Lily, creates the extensional mappings necessary for synchronizing the non-spatial with the spatial component of the results. Oracle BI Server uses these mappings to send a non-spatial query to Oracle Database, that returns the data for updating the non-spatial items in the end-user report.

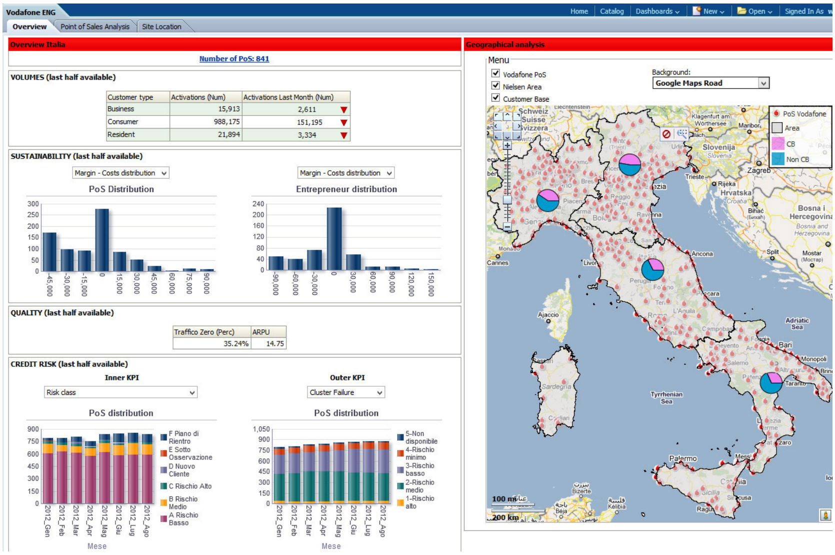
6 An Example

In this section we show an example in which Lily is coupled with the Oracle Business Intelligence Foundation Suite to achieve location intelligence functionalities for a major Italian telecommunication company.

The main functions provided can be summarized as follows:

1. *Geo-dashboard*: Area-based comparison of the main KPIs (number of activations, profitability, etc.) with guided navigation of detailed area-based reports.
2. *Volumes*: Area-based analysis of selling volumes by customer type, business channel, manager, etc.
3. *Sustainability*: Area-based analysis of economical sustainability by business format, business channel, manager, etc., with visualization of costs and revenues, influence areas of points-of-sales, etc.
4. *Points-of-sales Analysis*: Proximity-based analysis of a point-of-sales with reference to a benchmark set and analysis of correlations between points-of-sales; geographical representation of competitive pressure in terms of distance from competitor points-of-sales and identification of uncovered areas.
5. *Analysis of Territory Potential*: Identification of areas with high selling-potential (in terms of population, wealth KPIs, competitor presence, etc.) aimed at identifying the best positions for new points-of-sales; what-if simulation of the opening of a new point-of-sales with real-time computation and representation of its catchment area.

As shown in Figure 3, the geo-dashboard provides a summarized view of the main KPIs in table/chart format as well as on a map using the integration with external maps, multi-layer representation, geo-coding, and spatial KPI visualization functionalities of Lily. In particular, three themes are graphically overlapped: a bucket theme showing the boundaries of areas, a point theme showing all selected points-of-sales, and a pie chart theme indicating the customer base KPI on each area. The reference functional scenario is the one shown in Figure 2, left. To give a flavor of how query and data flow are processed by the different components, in Figure 4 we report examples for the main flows involved in the drawing of the pie charts; the rendering code and the intensional mappings are not shown because they come in form of complex Javascript code generated by Lily starting from rendering statements.



VOLUMES (last half available)

Customer type	Activations (Num)	Activations Last Month (Num)
Business	15,913	2,611 ▼
Consumer	988,175	151,195 ▼
Resident	21,894	3,334 ▼

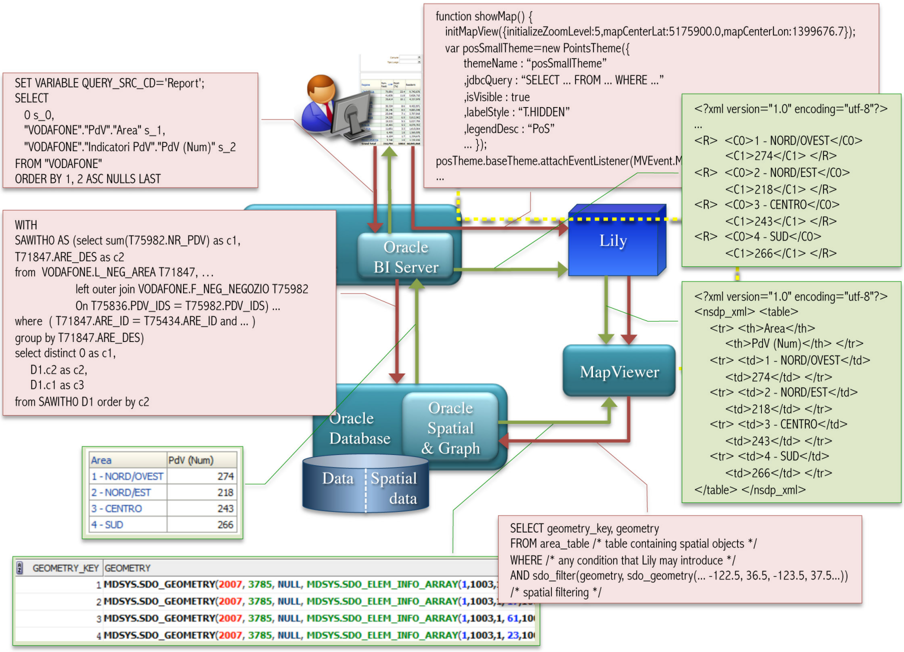
SUSTAINABILITY (last half available)

QUALITY (last half available)

Traffico Zero (Perc)	ARPU
35.24%	14.75

CREDIT RISK (last half available)

Fig. 3. Geo-dashboard



```
SET VARIABLE QUERY_SRC_CD='Report';
SELECT
  0 s_0,
  "VODAFONE"."PdV"."Area" s_1,
  "VODAFONE"."Indicatori PdV"."PdV (Num)" s_2
FROM "VODAFONE"
ORDER BY 1, 2 ASC NULLS LAST
```

```
function showMap() {
  initMapView({initializeZoomLevel:5,mapCenterLat:5175900.0,mapCenterLon:1399676.7});
  var posSmallTheme=new PointsTheme({
      themeName : “posSmallTheme”
      ,jdbcQuery : “SELECT ... FROM ... WHERE ...”
      ,isVisible : true
      ,labelStyle : “T.HIDDEN”
      ,legendDesc : “PoS”
      ... });
posTheme.baseTheme.attachEventListener(MVEvent.M
...
```

```
<?xml version="1.0" encoding="utf-8"?>
...
<R>  <C0>1 - NORD/OVEST</C0>
     <C1>274</C1> </R>
<R>  <C0>2 - NORD/EST</C0>
     <C1>218</C1> </R>
<R>  <C0>3 - CENTRO</C0>
     <C1>243</C1> </R>
<R>  <C0>4 - SUD</C0>
     <C1>266</C1> </R>
```

```
WITH
SAWITH0 AS (select sum(T75982.NR_PDV) as c1,
T71847.ARE_DES as c2
from  VODAFONE.L_NEG_AREA T71847, . . .
            left outer join VODAFONE.F_NEG_NEGOZIO T75982
            On T75836.PDV_IDS = T75982.PDV_IDS) ...
where  ( T71847.ARE_ID = T75434.ARE_ID and ... )
group by T71847.ARE_DES)
select distinct 0 as c1,
   D1.c2 as c2,
   D1.c1 as c3
from SAWITH0 D1 order by c2
```

```
<?xml version="1.0" encoding="utf-8"?>
<nsdp_xml> <table>
   <tr> <th>Area</th>
      <th>PdV (Num)</th> </tr>
   <tr> <td>1 - NORD/OVEST</td>
      <td>274</td> </tr>
   <tr> <td>2 - NORD/EST</td>
      <td>218</td> </tr>
   <tr> <td>3 - CENTRO</td>
      <td>243</td> </tr>
   <tr> <td>4 - SUD</td>
      <td>266</td> </tr>
</table> </nsdp_xml>
```

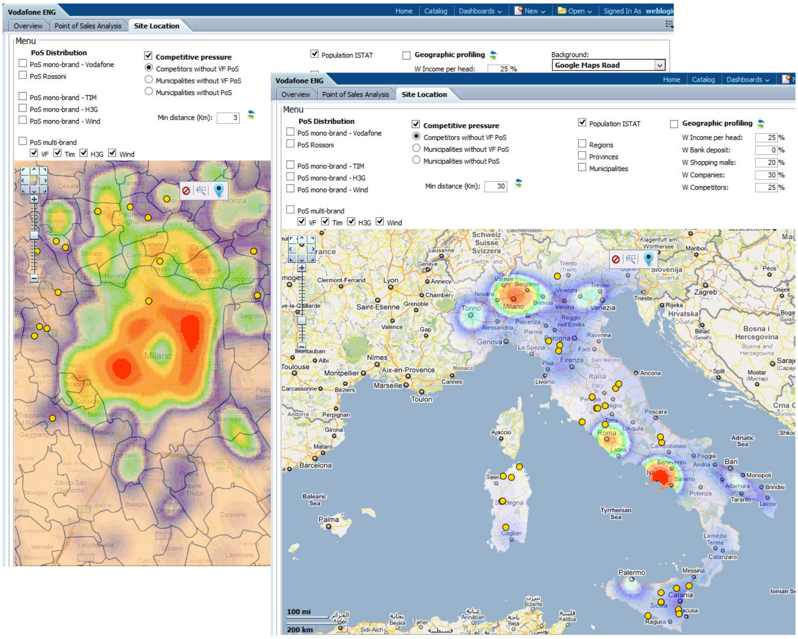
Area	PdV (Num)
1 - NORD/OVEST	274
2 - NORD/EST	218
3 - CENTRO	243
4 - SUD	266

```
SELECT geometry_key, geometry
FROM area_table /* table containing spatial objects */
WHERE /* any condition that Lily may introduce */
AND sdo_filter(geometry, sdo_geometry(... -122.5, 36.5, -123.5, 37.5...))
/* spatial filtering */
```

GEOMETRY_KEY	GEOMETRY
1	MDSYS.SDO_GEOMETRY(2007, 3785, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,
2	MDSYS.SDO_GEOMETRY(2007, 3785, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,
3	MDSYS.SDO_GEOMETRY(2007, 3785, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1, 61,10
4	MDSYS.SDO_GEOMETRY(2007, 3785, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1, 23,10

Fig. 4. Main query and data flows involved in the drawing of pie charts

Figure 5 shows an example of an analysis of territory potential. Besides all the Lily functionalities used for the geo-dashboard, also SOLAP queries are involved in this case for proximity analysis.


Vodafone ENG
Home | Catalog | Dashboards ⌄ | New ⌄ | Open ⌄ | Signed In As weblogi
Overview | Point of Sales Analysis | Site Location
Menu

PoS Distribution
□ PoS mono-brand - Vodafone
□ PoS Rossoni
□ PoS mono-brand - TIM
□ PoS mono-brand - H3G
□ PoS mono-brand - Wind
□ PoS multi-brand
☑ VF ☑ Tim ☑ H3G ☑ Wind

☑ Competitive pressure
● Competitors without VF PoS
○ Municipalities without VF PoS
○ Municipalities without PoS
Min distance (Km): 3

☑ Population ISTAT

□ Geographic profiling
W Income per head: 25 %

Background: Google Maps Road

Vodafone ENG
Overview | Point of Sales Analysis | Site Location
Menu

PoS Distribution
□ PoS mono-brand - Vodafone
□ PoS Rossoni
□ PoS mono-brand - TIM
□ PoS mono-brand - H3G
□ PoS mono-brand - Wind
□ PoS multi-brand
☑ VF ☑ Tim ☑ H3G ☑ Wind

☑ Competitive pressure
● Competitors without VF PoS
○ Municipalities without VF PoS
○ Municipalities without PoS
Min distance (Km): 30

☑ Population ISTAT
□ Regions
□ Provinces
□ Municipalities

□ Geographic profiling
W Income per head: 25 %
W Bank deposit: 0 %
W Shopping malls: 20 %
W Companies: 30 %
W Competitors: 25 %

Fig. 5. Analysis of territory potential at national and local scale

7 Final Remarks

Integration of true location intelligence functionalities within BI platforms is still an ongoing process. Lily, the library we proposed in this paper, represents a step forward in this direction since it offers a set of geo-enhanced functionalities in the context of a tightly-coupled architecture, so as to ensure top performances coupled with transparent usage and programming. Although Lily has been designed to work with a generic BI platform, in its current implementation it only works with OBI. We are now working to remove this limitation by implementing the interfaces with other platforms that match the tightly-coupled paradigm. Achieving this goal is a bi-directional process that also requires the standardization of the Lily interfaces and the extension of the native functionalities provided by the BI platforms.

References

1. Bédard, Y., Larrivée, S., Proulx, M., Létourneau, F.: Étude de l'etat actuel et des besoins de R&D relativement aux architectures et technologies des data warehouses appliquées aux données spatiales. Tech. rep., Laval University (1997)
2. Bédard, Y., Han, J.: Fundamentals of spatial data warehousing for geographic knowledge discovery. In: Miller, H.J., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, pp. 45–66. Chapman & Hall (2009)
3. Bimonte, S., Bertolotto, M., Gensel, J., Boussaid, O.: Spatial OLAP and map generalization: Model and algebra. IJDWM 8(1), 24–51 (2012)
4. Bimonte, S., Miquel, M.: When spatial analysis meets OLAP: Multidimensional model and operators. IJDWM 6(4), 33–60 (2010)
5. Bitterer, A.: Location intelligence is expanding the scope of BI. Tech. Rep. G00239089, Gartner Research (2012)
6. BusinessWeek Research Services, Pitney Bowes Business Insight: Location intelligence: The new geography of business. White paper (2006)
7. Gómez, L.I., Haesevoets, S., Kuijpers, B., Vaisman, A.A.: Spatial aggregation: Data model and implementation. Inf. Syst. 34(6), 551–576 (2009)
8. Ihm, J., Lopez, X., Ravada, S.: Advanced spatial data management for enterprise applications. Oracle White Paper (2010)
9. Katibah, E., Stojic, M.: New spatial features in SQL Server Code. Microsoft White Paper (2011)
10. Malinowski, E., Zimányi, E.: Requirements specification and conceptual modeling for spatial data warehouses. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM Workshops 2006. LNCS, vol. 4278, pp. 1616–1625. Springer, Heidelberg (2006)
11. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP operations in spatial data warehouses. In: Proc. SSTD, Redondo Beach, CA, pp. 443–459 (2001)
12. Rao, F., Zhang, L., Yu, X., Li, Y., Chen, Y.: Spatial hierarchy and OLAP-favored search in spatial data warehouse. In: Proc. DOLAP, New Orleans, LA, pp. 48–55 (2003)
13. Rivest, S., Bédard, Y., Proulx, M.J., Nadeau, M.: SOLAP: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis. In: Proc. ISPRS Workshop, Québec, Canada (2003)
14. Rivest, S., Bédard, Y., Proulx, M.J., Nadeau, M., Hubert, F., Pastor, J.: SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. ISPRS Journal of Photogrammetry & Remote Sensing 60, 17–33 (2005)
15. Scotch, M., Parmanto, B.: SOVAT: Spatial OLAP visualization and analysis tool. In: Proc. HICSS, Big Island, HI (2005)
16. da Silva, J., de Oliveira, A.G., do Nascimento Fidalgo, R., Salgado, A.C., Times, V.C.: Modelling and querying geographical data warehouses. Inf. Syst. 35(5), 592–614 (2010)
17. Stefanovic, N., Han, J., Koperski, K.: Object-based selective materialization for efficient implementation of spatial data cubes. IEEE Trans. Knowl. Data Eng. 12(6), 938–958 (2000)
18. Whitehorn, M., Zare, R., Pasumansky, M.: Fast Track to MDX. Springer, Berlin (2005)