Web services-oriented architectures for mobile SOLAP applications

Thierry Badard*

Centre for Research in Geomatics
Laval University
Quebec, Qc G1K 7P4, Canada
E-mail: thierry.badard@scg.ulaval.ca

*Corresponding author

Yvan Bédard, Frédéric Hubert, Éveline Bernier and Étienne Dubé

Centre for Research in Geomatics
Laval University
Quebec, Qc G1K 7P4, Canada
and
Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support
Laval University
Quebec, Qc G1K 7P4, Canada
E-mail: yvan.bedard@scg.ulaval.ca
E-mail: frederic.hubert.1@ulaval.ca
E-mail: eveline.bernier@scg.ulaval.ca
E-mail: etienne.dube.2@ulaval.ca

Abstract: With the growing popularity of mobile computing and the empowerment of mobile devices by high end-users and decision-makers, the development of architectures based on web services for the deployment of mobile Spatial OLAP (SOLAP) applications is being perceived as a powerful solution. This paper deals with geospatial (web) Services-Oriented Architectures (SOA) for mobile processing of geo-decisional information. The paper first reviews the underlying concepts of typical SOLAP applications as well as the general concepts of mobile computing. The existing SOLAP functionalities are presented and their viability/usefulness in a mobile environment are discussed. Different web services based on an extended geospatial SOA are then proposed in order to tackle the several issues related to mobile SOLAP. Finally, several research challenges and outlooks are discussed.

Keywords: Service-Oriented Architectures; SOA; web services; Spatial OLAP; SOLAP; mobile applications; service composition; Location Based Services; LBS; context-aware services; geospatial requirements; standards.

Reference to this paper should be made as follows: Badard, T., Bédard, Y., Hubert, F., Bernier, É. and Dubé, É. (2008) ‘Web services-oriented architectures for mobile SOLAP applications’, Int. J. Web Engineering and Technology, Vol. 4, No. 4, pp.434–464.

Copyright © 2008 Inderscience Enterprises Ltd.
Web services-oriented architectures for mobile SOLAP applications

Biographical notes: Thierry Badard is Professor in Geomatics and Geoinformatics at Laval University, Quebec City. He is a full-time researcher of the Centre for Research in Geomatics and of the GEOIDE NCE. He is also a Research Collaborator of the Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support. His research interest deals with geospatial (Web) Services Oriented Architectures, location-based and context-aware services and the design of intelligent mobile applications for better decision support. He acts as a reviewer for several international journals and scientific conferences and has already an impressive record of scientific contributions. For further details: http://geosoa.scg.ulaval.ca.

Dr. Yvan Bédard is Professor of GIS and Spatial Databases at Laval University, Quebec City. He holds a Canada NSERC Industrial Research Chair in Geospatial Database. He is an active member of the Centre for Research in Geomatics where he acted as Director for 7 years, and of Canada’s GEOIDE network of centers of excellence. Dr. Bédard has a multi-million dollars record in both fundamental and applied research. He has contributed to over 100 full-refereed papers and 300 non-refereed papers and conferences. His research interest focuses on geospatial databases UML design methods, spatial datacubes and Spatial OLAP.

Frédéric Hubert is a postdoctoral student at the Centre for Research in Geomatics. He received his PhD in Computer Science in 2003 from the University of Caen, France. Currently, he is member of the Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support. On January 2007, he will become assistant professor in the Department of Geomatic Sciences at Laval University in Quebec City, Canada. His main research interests include artificial intelligence, human computer interactions, spatial databases and web services.

Éveline Bernier holds a BSc and an MSc in Geomatics from Laval University. She has been working as a research assistant at the Centre for Research in Geomatics of Laval University since 2002 and she is a member of the Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support. Her main research interests include multiple representation databases, on-demand web mapping, multidimensional spatial databases and Spatial OLAP (SOLAP).

Étienne Dubé is a graduate student in Geomatic Sciences at Laval University. His Master’s degree project is entitled ‘Development of a web service for the real-time delivery of SOLAP mini-cubes to mobile clients’. He received a BEng in Computer Engineering at Polytechnique Montreal, and has been awarded a scholarship from the Geomatics Canada Scholarship Program in 2006.

1 Introduction

Web services have become key elements of several enterprise distributed applications. It is especially the case of web services for mobile devices where spatial information plays a central role (e.g., Location-Based Services). In the meantime, the Business Intelligence (BI) community and the geomatics community have developed Spatial Online Analytical Processing (Spatial OLAP) tools to provide powerful means to exploit and support
the analysis of decisional information having a spatial reference such as geographic position or a street address. Such tools can represent spatial phenomena and hence, fully exploit the spatial component that characterises about 80% of an enterprise’s data (Franklin, 1992). Spatial OLAP or SOLAP, provides efficient means to easily and interactively explore spatio-temporal decisional information at different levels of detail using maps, tables or diagrams (Rivest et al. , 2005). With the growing popularity of mobile computing and the empowerment of mobile devices by high end-users and decision-makers, providing SOLAP solutions in a mobile environment must now be addressed.

However, current OLAP and SOLAP solutions are based on a classic client/server architecture that is not best suited for mobile environments. One of the reasons relates to the fact that a broadband network is required and must always be available for such architecture, a situation which may not be the case in a mobile context. Besides, the development of architectures based on web services for the deployment of mobile SOLAP applications is being perceived as a powerful solution.

This paper deals with geospatial (web) Services-Oriented Architectures (SOA) for mobile processing of geo-decisional information, including the requirement analysis for deploying SOLAP applications on mobile devices using such architectures. The paper first reviews the underlying concepts of typical SOLAP applications, including the requirement analysis for deploying SOLAP applications on mobile devices using such architectures. The paper then presents the viability/usefulness of web services for the deployment of mobile SOLAP. Different web services based on an extended geospatial SOA are then proposed in order to tackle the several issues related to mobile SOLAP. One of these proposed web services is currently under development and is further detailed. Finally, several research challenges and outlooks are discussed.

### 2 Spatial OLAP concepts

In the BI domain, Online Analytical Processing (OLAP) tools are well known for being powerful means to exploit and support the analysis of decisional information. Such tools offer a unique gateway to analytical data usually stored in a data warehouse and provide efficient functionalities to help decision-makers analyse their data in a simple and intuitive manner. The information is presented using tabular or graphic views and may be analysed at different levels of granularity (e.g., sales by year, month or day). However, OLAP tools lack in their capacity to represent spatial phenomena and hence, do not fully exploit the spatial component that may characterise about 80% of an enterprise’s data (Franklin, 1992). “Hidden in most data is a geographical component that can be tied to a place: an address, postal code, global positioning system location, (...) region or country” (ESRI, 2000). Besides, viewing a phenomenon in its geographic reference framework helps the user to better understand it and may bring new insights that would have remained hidden otherwise. For instance, users may discover specific tendencies or patterns in the spatial distribution of a phenomenon. They may also point out spatial relationships among different geographic phenomena. In fact, having a map as a data exploration medium allows one to have a model that is closer to the user’s reality and, accordingly, would require less abstraction effort from the user and consequently increase his efficiency (Bédard et al., 2005).
Geographical Information System (GIS) may be seen as appropriate tools to analyse spatial decisional information. However, “despite their capabilities, it is recognized that today’s GIS are designed neither to support decisional applications nor to support highly interactive navigation through spatial data at different levels of aggregation and through different epochs” (Rivest et al., 2005). GIS are especially suited for databases that have a transactional nature. Such databases are oriented towards data acquisition, data storage, data updates, data integrity checking, not towards analytical needs.

In order to benefit from the spatial component of analytical data, a new category of tool, called Spatial OLAP, has emerged. SOLAP, as firstly termed by Bédard et al. (1997), came from the combination of GIS tools with OLAP tools. SOLAP can be defined as “a type of software that allows rapid and easy navigation within spatial databases and that offers many levels of information granularity, many themes, many epochs and many display modes that are synchronized or not: maps, tables and diagrams” (Rivest et al., 2005). Depending on the level of integration, LGS Group distinguishes three categories of approaches: GIS-dominant, OLAP-dominant or integrated SOLAP (LGS, 2000). The GIS-dominant approach provides solutions in which GIS functions are predominant and OLAP functions are kept to a minimum. The user interface is generally based on a GIS traditional interface. The OLAP-dominant approach is the opposite and gives birth to solutions that are closer to existing OLAP solutions and that include minimal GIS functionalities. Finally, the integrated SOLAP approach offers solutions with a high level of functionalities for both spatial and non-spatial data and views. It allows a sophisticated integration and synchronisation of OLAP and GIS functions. Such solutions can be implemented in several weeks using commercial bridges provided by GIS or OLAP vendors or in hours without programming by using generic stand-alone software built especially to develop SOLAP applications. To our knowledge, the JMap Spatial OLAP developed by our group and commercialised by KHEOPS Technologies still is the only such solution (Rivest et al., 2003; Bédard, 2005).

The architecture of JMap SOLAP (see Figure 1) is composed of:

- spatiotemporal database (or data warehouse) server
- SOLAP/JMap server
- SOLAP administrator
- SOLAP client.

The spatiotemporal database usually stores the geometry associated to dimensions, members and measures. A DBMS such as Oracle 10 g includes functionalities to manage both descriptive and geometric data. There may be some variants to this architecture as the geometry can be stored on geometric (proprietary) files, such as Shape files (ESRI) or Mid/Mif (MapInfo) files.

The SOLAP/JMap server handles, apart from the spatiotemporal database, “the numerical and spatial calculations necessary to compute the measure values associated with possible combinations of dimensions member” (Rivest et al., 2005). However, spatial data are a complex kind of data and performing on-the-fly spatial aggregation is still a research issue.
The SOLAP administrator is composed of web pages and provides functionalities to define the spatiotemporal data cube parameters (dimensions, measures) and spatial data sources, to manage metadata, to create legends (also called ‘thematics’), etc. Users’ profiles and security parameters are also defined by using the SOLAP administrator.

The SOLAP client provides functionalities to efficiently navigate among the data within spatial data cubes. It allows to open any available data cubes and to navigate within the data using operations such as drill-down and roll-up executed from a tabular, a graphical or a cartographic view. SOLAP clients are based on an intuitive and easy-to-use Graphical User Interface (GUI). The interactions are based on mouse clicks and there is absolutely no need to know querying languages such as Structured Query Language (SQL) to perform analyses. Each operation is achieved simply by using the mouse.

The JMap SOLAP architecture has been developed for desktop computers in a wired context. This architecture and all its components must be revisited in order to satisfy mobile issues (small devices and wireless connection).

3 Mobile technologies and SOA

3.1 Mobile technology and applications

Mobile equipments, such as laptops, smartphones and Personal Digital Assistants (PDAs), are among those areas that show the most important growth in the computing business. Nowadays, people, no matter where they are, want to stay connected to the internet and have access to information at anytime. This is even more typical of executives and decision-makers because in a business context, stakes of mobile technology are multiple: productivity gains and time gains, immediate decision making, client satisfaction and costs reductions (Cesmo, 2004).

The terms mobile and wireless are sometimes used to refer to the same concept. However, they mean two different things and are not necessarily dependant. For example, a desktop computer (not mobile) may rely on a wireless network (this is sometimes the case for old buildings where there is no cables) while a laptop (mobile) may be connected to a wired network.

Several limitations are associated to mobile devices. They must be taken into account during the development of mobile solutions (Duda et al., 2005). Small sizes of display, low screen resolutions and few available colours imply the adaptations of
graphical user interfaces. Processors indicate low capacity of computation. Disconnection risks must be managed in order to avoid losses of information and not disturb the user in his/her tasks. Low bandwidth and small storage and memory capacities limit the possibilities of data transfer and storage. By using wireless networks, data security must be improved in order to avoid hacking. Finally, identification of a user and his mobile device conduct to propose solutions adapted to their context of usage.

3.2 Service-oriented architectures and notions of web services

A SOA is a model of architecture that integrates and makes available one or more services, loosely coupled (Kraflizig et al., 2004; Jørstad et al., 2005; Badard, 2006). A service, the central element of a SOA, is a building block offering a contract of use, one or more interfaces of access and an implementation. By focusing on SOA, implemented systems acquire benefits with more efficient developments, reductions of development costs, simplified maintenance, incremental adoption and a greater evolutionary (Newcomer and Lomow, 2004). These benefits are justified by the following characteristics: modular architecture, loosely coupled services, meaningful to the service requester, well-defined service contracts published in a service repository, discoverable, loose process coupling, loose technology coupling, standard based, design for multiple invocation styles, and so on (Newcomer and Lomow, 2004). In SOA, three basic roles can be distinguished: service provider, service requester and service broker. A service provider provides an access to implemented services according to service contracts. A service requester is a client application or another service, which invokes a service implemented by a service provider. A service broker is a repository of service contracts published by service providers. This service is used by the service requester in order to find required or desired services.

A possible implantation of a SOA is a web services platform (Badard and Braun, 2004). A web service is a set of related application methods that can be remotely accessed across a network (such as a corporate intranet or internet itself). As SOA, web services are based on the concept of interoperability. They offer standardised accesses via interfaces; they can be transparently used in local, intranet or internet environments; they can be reused no matter of the technologies; and they can easily be maintained. Moreover, a web service is based on open standards that are independents of a product, a technology and a middleware (Newcomer and Lomow, 2004). So, some basic standards are integrated into the web services architecture in order to increase the concept of interoperability. The Simple Object Access Protocol (SOAP) standard from W3C is a communication protocol used to exchange messages between a client application and a web service through a network (Box et al., 2000). SOAP defines an information encoding based on XML over the HTTP protocol. A second standard from W3C is a Web Service Description Language (WSDL) that is intended to formally describe services over internet (Christensen et al., 2001). This descriptive information is exploited by a service requester (a client or another web service) in order to programatically invoke a web service from a service provider. In order to know which web services are available and how a service requester can exploit them, another W3C standard, named Universal Description, Discovery and Integration (UDDI), is used as a service discovery (OASIS, 2002). This standard defines a web services repository, in which WSDL documents are indexed, allowing developers and applications to locate web services.
3.3 Web services-oriented architectures for mobile applications

Conventional architectures such as client-server architectures are not adapted to mobile contexts. Zeichick (2003) considers that the client/server architecture is not adapted to mobile and wireless contexts because a broadband network is required and must always be available. But, as mentioned in Section 3.1, an architecture must support and manage low bandwidth and possible network disconnections in a mobile context. By using a web service-oriented architecture, specific web services can be deployed in order to avoid some network problems and/or to warn mobile users, like orchestration/choreography service (Ibach and Horbank, 2004) or errors service.

Duda et al. (2005) estimates that SOA is predestined for integrating mobile devices into distributed applications because mobile functionalities can be sourced out, and value added services can be made available to end users. By integrating mobile devices into SOA, implemented with web services, some issues can be bypassed like low processing power or low memory (Steele, 2003). Heterogeneity problems between different mobile devices and different application or server platforms can be resolved by defining adequate web services interfaces, which are platform independent. The fact that web services exploit open standards brings the possibility to use mobile devices from different vendors, with different platforms, different versions and different wireless networks (Ojala, 2005).

Nevertheless, a web service in a mobile environment is different from a traditional web service as customised services are offered to the user with any kinds of mobile devices. Accordingly, some constraints must be taken into account: the portability of the device, the association of the device with a particular user identity, the personalisation of the application to the user, and the limitations of devices (Farley and Capp, 2005). The authors also identify some key aspects to take into account in order to define web services for mobile: distributed nature of web services, cooperation between components (usage of XML), sharing possibilities (databases), content adjustments regarding user’s time and location, customisation, etc.

With such an integration of mobile devices into SOA, some drawbacks appear regarding the intrinsic characteristics of XML (Steele, 2003). The parsing of the transferred XML messages from web services may result in a demanding task for mobile devices because XML messages must be processed with limited processing power. Theses messages can be very verbose and cannot be easily transferred because of networks limited bandwidth. Another problem concerns some web services standards, which are not totally mature and widely used. These different disadvantages must be taken into account in our architecture, in order to find and integrate adequate solutions.

4 From a desktop to a mobile SOLAP application

Designing applications for mobile devices requires taking into account the intrinsic characteristics – limitations – of mobile environments. As mentioned in Section 3.1, these include limited bandwidth, limited display size and resolution, limited processing capabilities, limited memory capacity and limited input facilities. To be efficient, an application that has been initially developed for desktop computers cannot just be implemented as such (without any redesign or modifications) into a mobile device (Passani, 2002). It must have been thought right from the start with mobile issues
in mind; “mobile device targeted applications designed without mobile computing paradigms in mind will definitively fail” (Sanna and Demontis, 2004). According to Passani (2002), we should identify the top 20% functionalities when porting an existing application to a mobile environment.

This section focuses on SOLAP functionalities. It is based on the JMap SOLAP solution, designed and developed in Java by our research team and commercialised by KHEOPS Technologies, as it is the only integrated SOLAP solution on the market today. This section is structured as follows: for each SOLAP functionality, it first describes the functionality in the desktop environment (i.e., the classic environment where the client is permanently connected to a wired network), it then discusses its relevance for a mobile environment as well as its associated technological challenges. Finally, the issues related to the connected and disconnected modes that may characterise mobile environments are discussed. The relevance of each functionality for a mobile environment is mostly based on the limitations associated to mobile devices, the utility of having the functionality on the mobile client and its usability.

We consider this functionality analysis as a first step towards a complete revisiting process that must be performed prior to the development of mobile SOLAP applications. Adaptations will surely be required at each layer of the development process and new processes will have to be implemented to efficiently support the mobile environment (data caching mechanisms, remote execution capabilities, new interaction mechanisms, etc.).

Figure 2 shows the user interface of the JMap SOLAP client from KHEOPS Technologies. The available functionalities are presented hereafter followed by a brief discussion on whether it is significant and feasible to have it on a mobile client or not.

Figure 2  The JMap SOLAP client from KHEOPS technologies (the desktop version)
4.1 Main functionalities

4.1.1 The slice and dice panel (A)

Desktop environment

This panel includes the dimensions and measures that are part of a specific data cube. Dimensions preceded by a small earth icon are spatial dimensions; the ones preceded by a small clock icon are temporal dimensions while the other ones are descriptive dimensions. Using this panel, the user identifies the measures and the members (or the level) of each dimension to be taken into account in his analysis. This may be seen as the slice and dice operator as it reduces the dimensionality of the data, i.e., take a projection of the data on a subset of dimensions for selected members of the other dimensions (Chaudhuri and Dayal, 1997).

The lower part of the panel is used to position the selected dimensions and measures on the different axis (columns and rows). The position has an impact on the way the information will be displayed on the different types of view.

Relevance for a mobile environment

This panel offers an essential functionality of a SOLAP client as it allows the user to view the different measures and dimensions that are part of a specific data cube and define the parameters of his analysis. Accordingly, it must be implemented on a mobile SOLAP.

Technological challenges for a mobile environment

The main challenge associated to this panel is in terms of GUI design. For display sizes as limited as the ones offered by mobile devices, the extent of this panel must be reduced. It can also be thought as a sliding panel, appearing only when needed. Besides, the tree approach for displaying dimensions’ members should also be revisited as it is not efficient for dimensions with a large number of members, especially for devices with very limited display areas.

Issues related to connected/disconnected modes

In a connected mode, this panel is automatically populated following a connection to a data cube. To ensure its availability in a disconnected mode, the mobile device must stores the information relating to cube measures and dimensions (hierarchies, levels and members). In the JMap SOLAP solution, this information is stored in the cube schema, defined in a XML format. In a SOA, we could think of a service that gives access to such cube schemas (see Section 5, Service 3). The XML format is however to be reconsidered because as we mentioned in the previous section, XML files may be quite verbose and their transfer over wireless networks with limited bandwidth as well as their ‘parsing’ on mobile devices with limited processing capabilities may cause problems.
4.1.2 The views panel (B)

Desktop environment

A typical SOLAP application offers three types of views: tabular, graphic and cartographic (see Figure 2). Tabular views present the data in a systematic fashion. The dimensions and measures may be positioned either on the column axis or on the row axis. The number of dimensions and measures that can be included in the table is not limited. Graphic views can go from traditional pie charts and histograms to scatter plots or stacked area graphs. There is no limitation as for the number of dimensions that can be viewed on a graph. However, too much dimensions (and especially, too many dimension members) may lead to unreadable and meaningless graphs. Cartographic views present the spatial representation of the data. As opposed to the two preceding views, they are limited by two dimensions with selected members, one spatial and one descriptive. More than that would result in a complex and unreadable map. The user has the possibility to open as many views as he/she wants on the display panel and may navigate among those using different operations.

Relevance for a mobile environment

This display panel is obviously a fundamental element of a SOLAP client and must include the three types of display as they are each appropriated for different analyses. However, several issues must be addressed in order to adapt it to a mobile context.

Technological challenges for a mobile environment

Several limitations associated to mobile devices may affect these display components. These limitations include the limited display size and resolution as well as the reduced colour palette (from 2 to 16 bits). For example, certain types of thematic maps become unreadable on small screens, such as choropleth maps overlaid by pie charts (one for each geographic area) or multimap views resulting from a map pivot (e.g., one map per year in a single map view). Thus, the variety of semiologic rules will have to be adapted.

Furthermore, as opposed to the desktop version, a SOLAP mobile client should probably be limited to only one view at once. The number of dimensions or measures that could be viewed on a tabular display should also be limited to avoid scrolling operations (see Figure 3). Also, graphic displays, such as pie charts, should probably be limited as for the number of members that can be displayed at once. Otherwise, it may leads to unreadable charts (see Figure 4).

Given the reduced colour palette of mobile devices, a particular attention should be given to the use of graphic variables. As less colour is available in a mobile device, thematics will not be as rich as the ones used for a desktop computer (see Section 3.1). Also, the desktop version allows the user to analyse more than one measure simultaneously on cartographic views by superposing different thematics (one for each measure). Obviously, this practice should not be allowed on mobile devices as it will lead to unreadable and thus useless maps.

Graphic views, in the desktop version, are produced using the free and open source solution JFreeChart. However, in a mobile environment, this component should be adapted or eventually replaced in order to improve mobile performances and provide a better legibility and support of mobile devices diversity. For example, solutions based on
Scalable Vector Graphics (SVG) format could be relevant as they could manage both cartographic and graphic views (SVG Tiny, SVG Basic). SVG is an international standard, specified by the World Wide Web Consortium (W3C), which allow the describing of two-dimensional vector and mixed vector/raster graphics in XML. As a vector format, SVG is scalable: users may zoom in or out on a portion of an image without degradation or loss of detail. The same graphic can appear correctly on any devices without the need to make new images to fit different display dimensions. Such an SVG-oriented solution could be interesting, as long as the data are already stored in a SVG format on the mobile device. XML files transformations (e.g., GML to SVG) are highly time/CPU consuming and should be avoided in a mobile environment (Piras et al., 2004). Such transformations should be performed by a dedicated service, which can be invoked by the mobile application.

**Figure 3** Reducing the number of dimensions that can be displayed on a view: a NXN table (left) versus a 1×1 table (right)

**Figure 4** Reducing the number of members that can be displayed on a view: a pie chart with too many members versus a pie chart with ten members
The way the legend is displayed is also to be reconsidered for mobile devices as it actually takes a lot of space. New solutions, such as the use of transparency or floating boxes, must be considered to more efficiently manage the display space. Also, the view title includes a summary of the active members and measures selections. In a mobile environment, this information should still be presented to the user but probably using different means.

Issues related to connected/disconnected modes

Producing these three types of view may be time/CPU consuming. It should thus be considered, in a connected mode, to produce them on the server side or on a remote computer that has more resources. A possible solution is to remotely produce these views and only exploit (interactive) images on the mobile client. SVG Tiny or Basic may be used in that context (see Section 5, Service 11).

The fact table of the data warehouse, which includes all the unique combinations of dimension members and measures, is the basis for producing any of the three types of display. In a disconnected mode, this table should reside on the client side which may involve storage problems as it is the most voluminous table of the data warehouse. For cartographic displays, spatial data must also be available on the mobile device.

Besides, on-the-fly generalisation of spatial data would be preferable in a mobile environment to reduce the data volume to transport, store and display. In a SOA context, a Geospatial Data Transformation Service could handle such a task (see Section 5, Service 23) and fully harness the power of SGOs and Geometric Patterns (Self-Generalising Objects; (Sabo et al., 2005)) and generalisation operators dedicated to mobile environments (Gimodig, 2004). WFS service could also be used to obtain original spatial data in GML format (OGC, 2004) (see Section 5, Service 21). Moreover, a WMS service could be deployed to return spatial data in raster or vector format (see Section 5, Service 22).

4.1.3 SOLAP operations (C)

Desktop environment

The SOLAP toolbar is composed of specific spatial OLAP functionalities (see Figure 2, Part C). The three first buttons are used to select the display mode; table, chart (with different chart types available) or map. The five following buttons are different drill operations that allow the user to easily navigate among the data. A drill-down operation goes from one Level of Detail (LoD) to a more detailed level within a dimension (see Figure 5). The roll-up (or drill-up) goes from one LoD to a more general level within a dimension. The drill-across operation allows the visualisation of another member or of another dimension at the same LoD. The open and close operations are variants of drill-down and roll-up operations as they keep the context of the other dimension members while drilling (down or up) on a specific member. The sixth button is the selection tool that is used to enlarge columns when more than one measure are displayed in a tabular view and to manipulate the temporal slider. It is followed by the SOLAP information that allows the user to access minimal information about a selected member: the member name, the name of the dimension it belongs to and the corresponding
measure values (calculated using the root member of every other dimension). Finally, the last button is the execute function that is used to refresh the view after having changed the members/measures selection.

**Figure 5** Example of a (default) spatial drill-down on a cartographic display

The default drill operation is applied to a specific member. Two other types of drill are available. The *layer drill* affects every member being part of the selected member’s layer while the *selection drill* affects only the selected members. For example, suppose a spatial dimension composed of the following layers: Country → State → City with an initial cartographic view displaying the boundaries of USA. A default drill applied on the State of California will display the cities being part of this state. A layer drill applied on the same element *(i.e.,* California) will affect all the other states *(i.e.,* other members of the state layer) and will result in a view showing all the cities of USA. Finally, a selection drill applied on California, Florida and North Carolina will display their respective cities.

**Relevance for a mobile environment**

Most of these drill operations must be part of a mobile SOLAP client as they are essential tools to efficiently navigate among the data.

**Technological challenges for a mobile environment**

The drill operations are probably the most essential operations in SOLAP as they allow the user to go from one LoD to another, supporting in-depth or general analyses. However, as presented in the previous paragraph, we believe that drill operations in
mobile devices could be limited to the default ones (i.e., drilling on one selected member). The two other types (layer/selection) may require more CPU (or may take more resources) as they involve more elements, without mentioning the fact that they may also generate densely populated views.

We can think of two different pen-based methods to apply these operations on a mobile device. First, by selecting the operation to apply (by pointing the button with the pen) and then, selecting the member on which the operation must be done (still using the pen). This corresponds to the actual manner to perform SOLAP operations in the desktop version. Second, those operations could also be achieved by stylus gestures. For example, a down-to-up gesture over a member could result in a roll-up operation while an up-to-down gesture could result in a drill-down operation and a left-to-right gesture could result in a drill-across.

For the same reasons that layer/selection drills should be avoided, the open and close functionalities should also be left aside. The SOLAP information should also be not included in a mobile context as it requires querying for new data in the fact table. In a disconnected mode, this querying operation would be done on the client and would require too much CPU. Finally, the execute operator is essential as it allows to apply the selection modifications on the dimensions or measures, while keeping the same view. Also, without such an operator, the view would be updated after each click on the slice and dice panel which is really not optimal.

**Issues related to connected/disconnected modes**

One of the essential characteristics of SOLAP is that it provides quasi instantaneous responses to such operations in order to help users keep their train of thought while navigating among the data. This can only be true if these operations rely on quite powerful processing power, which may be not sufficient on mobile devices. A SOLAP operations service should thus be offered in a SOA to ensure the fluidity throughout the analysis (see Section 5, Service 2).

In a disconnected mode, these SOLAP operations can be achieved only if the fact table as well as the spatial data, reside on the mobile device.

**4.1.4 GIS functionalities (D)**

**Desktop environment**

The traditional minimal GIS functions are available in the JMap SOLAP client. These include zoom in/out/extend/to selection, pan, previous view, measure, label, object information and selection, and attribute queries. Such functions are intended to be used on cartographic displays only.

**Relevance for a mobile environment**

The zooms and the pan functionalities should definitively be part of a mobile SOLAP client as they support the navigation within cartographic views. They are even more important for mobile devices given that most of the time the cartographic view will not display the whole extent of the map (data). The previous view functionality of the GIS toolbar could not be included as the SOLAP toolbar already
proposes this functionality (and it can be applied to graphic and tabular displays as well), idem for the object info (that can be obtained by the SOLAP report tool). The measure tool is probably not significant as mobile SOLAP applications (as well as their desktop counterparts) are not meant to provide precise cartographic data and perform complex GIS analyses. Thematic characteristics are much more important that geometric characteristics.

**Technological challenges for a mobile environment**

Given that no complex GIS functionalities are offered on the client, the main challenge is to provide fluidity and rapidity while performing such operations. Available plug-ins for displaying SVG documents already offer lightweight zoom and pan functionalities on mobile devices. They could thus been used to help in implementing GIS functionalities.

**Issues related to connected/disconnected modes**

These functionalities may be available either in a connected or a disconnected mode as they do not require to access new information that has not already been downloaded on the device.

### 4.1.5 Others (E)

**Desktop environment**

The ‘Other’ toolbar (see Figure 2, Part E) is composed of the following functionalities: backward (or undo), forward (or redo), selection, report, print and e-mail. Backward and forward act as view navigation tools as they allow the navigation between already displayed views. The selection tool is used to select specific members on which we want to apply SOLAP operations (e.g., selection drills) or to get a report. Four selection modes are currently offered: punctual, rectangular, circular and user-defined polygon. The report functionality can only be used for cartographic members and allows the user to access their associated descriptive information. The report can be saved or e-mailed. The print functionality allows the user to print an existing view and offer customisation possibilities (layout, text edition, etc.).

**Relevance for a mobile environment**

The backward and forward functions must be provided on SOLAP mobile clients while the remaining functions could be offered.

**Technological challenges for a mobile environment**

The backward/forward operations do not necessarily involve great challenges. We must be able to keep the navigation history and the operations that have been involved.

The selection tool could be limited to the punctual selection. Interaction principles must be revisit in a mobile context. Obviously, selecting more than one element leads to more voluminous data sets that must be downloaded. More bandwidth and CPU is then required.
Issues related to connected/disconnected modes

The print and e-mail function are possible in a disconnected mode: e-mails and printing sent while disconnected are processed when the device connects again to the network. For the backward and forward operations, it is necessary to have the data (spatial and descriptive) that have already been downloaded and displayed on the client. This may lead to storage space issues. Also, solutions to locally search and fetch data must be provided on the mobile device. The selection in a disconnected mode is not a problem a priori, as long as the resulting information is available on the client. Finally, the report option could be handled by a specific web service that could provide the mobile user with structured information about dimensions members (see Section 5, Service 12).

4.2 Other functionalities

4.2.1 Connection to data cubes

Desktop environment

Prior to data analysis, the user has to select and log on to a specific data cube. This involves two fundamental actions: authentication on the SOLAP client and accessing the list of the available data cubes. The first action basically relies on a username and password validation. Once logged on, a list is prompted to the user containing all the data cubes that the user has access to. The multidimensional data (dimensions, measures, etc.) of the selected cube is then downloaded from the server to the SOLAP client.

Relevance for a mobile environment

This functionality must obviously be included in a SOLAP mobile, though the authentication process could be integrated in the mobile SOLAP client and hidden to the user.

Technological challenges for a mobile environment

The main challenge associated to this operation is the storage space that would be needed for storing the cube data on the mobile device. Innovative solutions would have to be developed to offer a right balance between what it is stored on the mobile versus what is accessed on the server. Besides, security issues would have to be taken into account in order to ensure data transfer reliability, protect data integrity and confidentiality. These issues could be managed by specific security web services based on recent XML extensions, such as SAML, XACML or XKMS, which enables secure authentication and high encryption of data during client/server exchanges.

Issues related to connected/disconnected modes

In a connected mode, the authentication process could be supported by a web service (see Section 5, Service 26). Another web service may be offered to provide the mobile SOLAP user with a list of all the available data cubes he/she may access (see Section 5, Service 5). To preserve confidentiality, data encryption must be performed by the service which delivers the data.
In a disconnected mode, there is no longer authentication notion. All the cube information must be stored on the mobile device. Compression techniques are to be taken into account to minimise the associated data volume.

4.2.2 Swap operation

Desktop environment

The swap (or pivot) operation consists in interchanging visible dimensions or visible background dimensions in order to modify the content of displayed axes. Though this action may be done by manually interchanging the dimensions/measures on row and column boxes in the slice and dice panel (see Section 4.1.1), it is usually done directly on the displayed view, via a right click contextual menu.

Relevance for a mobile environment

The swap operation must be part of the functionalities offered on the mobile client as it perceived as one of the basic SOLAP operations and is frequently used.

Technological challenges for a mobile environment

There is no particular challenge associated to this functionality. However, a swap operation executed on a cartographic display may result in several views. In the desktop version, these views are displayed simultaneously while in the mobile version, they should be accessed one at a time. It may then become necessary to restrict on-the-fly this function for certain views or combination of dimensions.

Issues related to connected/disconnected modes

It can be used either in a connected or a disconnected mode as it displays the same data, but with a different perspective.

4.2.3 Synchronisation between different views

Desktop environment

More than one view (of the same type or of a different type) may be opened conjointly. Views can be grouped in collection and synchronised with each other. View synchronisation results in an operation performed in one view automatically replicated on the other views of the same collection. The graphical legend used from one display to another may also be synchronised to facilitate the identification and the interpretation of the data.

Relevance for a mobile environment

Even if mobile SOLAP clients should be limited to only one view at once, this functionality is of interest in such a context so that when one switches from one view to another one, they are synchronised.
Technological challenges for a mobile environment

Implementing this functionality in a mobile context does not address any specific technological challenge if all required data are stored on the mobile device. Synchronisation will be performed behind the scene and thus will not be visible on the mobile SOLAP graphical interface. The user will be able to switch between the synchronised views on data in a similar manner to the classical (i.e., not synchronised) views.

4.2.4 Drillable timeline

Desktop environment

The drillable interactive timeline allows the user to control the display of the time dimensions and supports the drill-down/up operations (directly on the time cursor). This timeline allows for the display of animated maps and hence, facilitate the visualisation of the evolution of phenomena (Rivest et al., 2005).

Relevance for a mobile environment

We think that this functionality is not essential in a mobile context as it can be simulated using slice and dice or drills operations in parts A or B of the display (see Figure 2). However, this widget could be implemented as it simplifies the drill up and down operations on the temporal dimensions. As display capabilities are limited on mobile devices, it involves new GUI considerations.

Technological challenges for a mobile environment

Implementing this functionality in a mobile context does not address any specific technological challenge if data required for temporal computations are all stored on the mobile device.

4.2.5 Interactive legend

Desktop environment

The interactive legend can be seen as a graphical view specific to the semantics of the analysed data. It proposes a new type of SOLAP operation: the classification drill operation. This new type of drill is intended to be applied on the data classification used to represent measures on the different displays. This operation allows for the visualisation of different levels of detail of the data classification (Rivest et al., 2005).

Relevance for a mobile environment

This widget is not relevant for a mobile environment and would require too much processing resources as it is based on an on-the-fly reclassification. Furthermore, it is rarely used on desktop applications.

4.2.6 Creation of new measures to be calculated on-the-fly

Desktop environment

New measures, called calculated measures, may be defined by users. Such measures are usually based on existing ones and may be defined using numeric functions.
Relevance for a mobile environment

We believe this is something important in a mobile context as it may fulfil an important personalisation requirement.

Technological challenges for a mobile environment

There is no specific technological challenge associated to calculated measures as it is calculated on-the-fly on the server side. However, a difficulty may arise when defining the measure formula as it must be expressed using the Multidimensional Expressions (MDX) syntax, which may be compared to SQL for relational databases. MDX syntax is rather verbose and editing measure formula with a pointing device as a stylus may result in a painful task.

Issues related to connected/disconnected modes

In a disconnected mode, the major problem is that we no longer have access to the server to perform on-the-fly processes. The mobile should thus be able to calculate the measure on-the-fly. It requires more CPU, time and storing capabilities as all the relevant data must be available on the mobile.

In a connected mode, a calculated measures service could be used in order to compute remotely new data values (see Section 5, Service 8).

4.2.7 Context management

Desktop environment

It is possible to save the user’s context, i.e., opened views, selected members, collections, etc. When the user opens the application, he/she only has to load his/her predefined context to start his/her analysis exactly where he/she ended the last time.

Relevance for a mobile environment

The context management is relevant for a mobile environment as it allows the user to save any interesting analyses.

Technological challenges for a mobile environment

The main challenge would be to offer the possibility to access contexts in both connected and disconnected modes. This would require sophisticated solutions to ensure that the data that are included in a previously saved context are still available while opening that context in a disconnected mode.

Issues related to connected/disconnected modes

The possibility to save a specific context could be offered in both modes. In a connected mode, parameters may be stored on the server side while in a disconnected mode, all the information must reside on the mobile device. As mentioned above, the possibility to open an existing context should only be offered in a connected mode to avoid any inconsistencies. In a connected mode, we may think of a web service that would manage
contexts (parameters, right access, etc.) and offer the possibility to share them among SOLAP users (see Section 5, Service 15). It may also provide means to adapt a desktop SOLAP context for a mobile SOLAP client.

4.2.8 Metadata

Desktop environment

Metadata can be associated to cubes, dimensions, members and measures. Their management is done in the SOLAP administrator but they can be accessed via the client. Metadata are usually stored in databases. When the user asks for metadata, a query is sent to the server, which search for relevant information and send it back to the client, which in turn display it using a dialog box.

Relevance for a mobile environment

Metadata are important as they provide information about the data that are being analysed. In some cases, they may inform the user about data quality issues and help him/her to evaluate his/her result.

Technological challenges for a mobile environment

There is no specific technological challenge to access metadata on the mobile client. The size of a metadata database is usually small and does not involve complex queries.

Issues related to connected/disconnected modes

To be accessed in a disconnected mode, metadata must be stored on the mobile device. In a connected mode, metadata may be managed by a web service that could offer mechanisms to efficiently access, query and deliver (only a subset of metadata or generalised metadata) them (see Section 5, Service 10).

4.2.9 Thematics

Desktop environment

The SOLAP client offers the possibility to manage the thematics associated to data cubes. Thematics are associated to measure and define the visual variables, such as colour, to be applied when displaying those measures. The user may edit existing thematics (for example, change colours or symbols) or create new ones.

Relevance for a mobile environment

Thematics selection or edition could be offered on a mobile client but the creation of new ones is not necessarily relevant in such a context and should only be offered in the desktop version.

Technological challenges for a mobile environment

A first challenge would be to propose a graphic interface for the creation of new thematics that would be adapted for mobile devices. A second challenge is associated to the visual variables that compose thematics such as colour, texture, line weight and line
style. Due to colours limitations of mobile devices, transformation or adaptations must be made to desktop thematics. Besides, when thematics are based on different textures or line contours, they are CPU-consuming and hence not necessarily adequate for mobiles.

Thematics are associated to cubes, more specifically to cube measures. In a mobile context, they must then be downloaded along with cubes. Accordingly, we must consider additional space storage on the mobile client.

New thematics especially created for mobile devices should be offered with the application.

Issues related to connected/disconnected modes

Thematics management can be performed in both modes and can be supported by a web service (see Section 5, Service 9) that would give access to existing thematics and facilitate the creation of new ones.

4.2.10 SOLAP help and about

Desktop environment

A SOLAP online help is proposed on the desktop version. It is based on Macromedia RoboHelp and it is essentially composed of HTML pages. It includes text, images and hyperlinks.

Relevance for a mobile environment

This is important, though a lighten version which focuses on mobile SOLAP functionalities should be produced.

Technological challenges for a mobile environment

This could be achieved using web pages adapted for mobile devices.

Issues related to connected/disconnected modes

In a disconnected mode, the mobile client would have to store the information, or the web pages. This could be included with the application, or downloaded upon request.

4.2.11 Export/Save

Desktop environment

It is possible, on the desktop version, to save and export displayed views. Tabular views can be directly exported in MS Excel or as images. Graphic and cartographic views are exported as JPEG image files.

Relevance for a mobile environment

This is something interesting but not a key functionality for a mobile SOLAP client. Besides, it would require extra processing power, especially when producing compressed JPEG images.
4.3 Overview

Mobile platforms have some inherent limitations related to network bandwidth, storage space, display size and input devices. Due to these limitations, we have demonstrated that direct access to full-fledged SOLAP data warehouses is not well suited in the mobile context. Web services architectures appear to be better adapted to the deployment of mobile SOLAP applications. Definition of some concept services has then been initiated in the previous sections.

In addition to limitations related to devices, mobile users often have very specific requirements about data, mostly depending on their location and current work context. Most often, only a subset of the available data is needed to answer mobile decision-making queries. For this purpose, we define the notion of SOLAP mini-cube as a limited-size, context-specific geospatial decisional data mart. A data mart is defined by Kimball and Ross (2002) as “a logical and physical subset of the data warehouse’s presentation area”. The limited size implies a reduced number of dimensions, members, hierarchy levels and measures when compared to typical desktop-oriented SOLAP applications. The context will determine which subset of available data is extracted for each of these elements. For example, a forest fire-fighting crew could want the latest information available on the average tree age in forests stands located in a 15 km radius. The mini-cube needed would contain the following elements:

- ‘Forest division’ spatial dimension – only the members at ‘Forest stand’ hierarchical level that intersect a 15 km radius circle around the geographic location of the mobile user
- Time dimension – the members representing the most recent information available for this theme, e.g., the current year
- Measures – only the ‘tree age’ measure. To handle the creation of these data marts from available data sources, ideally in an automated way, the definition of dedicated and modular web services is required.

5 Towards a web services-oriented architecture for mobile SOLAP applications

Section 4 has pointed out that many different web services are required in the context of a mobile SOLAP application. According to this information, the current section identifies and presents a set of web services which could be implemented in order to design a consistent web SOA for the deployment of efficient mobile SOLAP applications. Three different categories of services can be distinguished: SOLAP services, geospatial services and basic services. SOLAP services are services that allow the production of different kinds of geo-decisional data. We have divided this category into two sets of services following the connected/disconnected usage performed by a mobile device. To our mind, SOLAP services in connected mode could be:

1. **Data cubes updating service** – Cubes can undergo several updates such as data insertions, modifications or deletions. Such updates may be applied on measures, dimensions or dimensions members. Mechanisms should be developed in order to inform SOLAP clients about such updates and transfer adequate data or changes on mobile devices.
2 SOLAP operations service – A remote SOLAP server is available and accessible. Operators such as drill-down, roll-up, drill-across, open and close can be applied on dimension members in a tabular view, a graphic or a cartographic view. The resulting data can be provided by a web service in a XML format or a pre-processed format (i.e., images of charts or maps). From a mobile SOLAP client, this service can be invoked each time a drill, open or close operation is applied (see Section 4.1.3).

3 Cubes schema transfer service – This service provides access to the schema defining a cube in order to access, query and manage it on a mobile device (see Section 4.1.1). The transferred data include cube measures, dimensions (with their hierarchies) levels and members.

4 Data cubes administration service – This service can be called when a mobile user with an administrator profile wants to create the structure of the cube by defining new measures and new dimensions. So, this structure and several links on the remote data warehouse or database can be saved on a remote SOLAP server.

5 Data cubes listing service – The listing of cubes is necessary for a user in order to know which cubes are available before selecting a cube to load in the SOLAP client on the mobile device (see Section 4.2.1).

6 Dimensions listing service – This listing is useful to create new cubes with existing dimensions. It can be used together with the cubes management service.

7 Spatial and descriptive data listing service – In order to create a new cube, a mobile user must access to information on available data sources. Fact table in a specific database must be identified and linked to the new cube. Other tables, web services or files can be linked to dimensions information such as WFS (see Item 21) or WMS (see Item 22) for spatial data. This service is more relevant for a SOLAP administrator.

8 Calculated measures service – This service has to be called to introduce a new measure in a cube by using the MDX language, which has been extended for geospatial information. It produces new statistical data, according to the current data of a cube, in order to be viewed in a mobile SOLAP client (see Section 4.2.6).

9 Thematics Management Service – On a mobile device, graphical aspects of the different views (table, map, graphs) can be modified by using this service (see Section 4.2.9). These thematics must be transformed and adapted to the mobile device, according to screen resolutions and display constraints like the limited number of colours.

10 SOLAP Metadata Service – An administrator can associate metadata with several cubes, dimensions, hierarchies or levels. This service allows the retrieval of metadata information on the selected element of the cube. A mobile user can thus access metadata via this service (see Section 4.2.8).

11 Images and Charts Production Service – This service is used in order to store or transfer data between mobile devices and produce graphical representations of data (see Section 4.1.2). Images are created on a remote server in order to reduce the computational process of the mobile devices. As an example, verbose GML data
could be transformed in SVG for display purposes on mobile devices by this service or produce charts that allow analyses of data. This service may rely on a web map service (see Item 22).

12 **Reports Production Service** – This service produces a report which contains information about dimension members from a mobile SOLAP application (see Section 4.1.5).

13 **Dashboards Creation Service** – This is a high-level service combining different services in order to create on-the-fly dashboards for delivery to a mobile user according to his/her needs. This service could eventually invoke the images and charts production service.

14 **Alerts Delivery Service** – According to the quality of data, their relevance for an identified user or their updating, alerts can be delivered in order to warn the end-user on his mobile device. This service can rely on another service, named notification service, standardised by OASIS organisation in order to disseminate information (i.e., alerts) to one another.

15 **Context Management Service** – A user context corresponds to information about the different views opened on the SOLAP client, the members selected by the user, the different credentials needed for authentication to the server, and different other parameters related to the user session. This service manages the different contexts of users connected via a mobile device or a desktop computer in order to allow them to easily and rapidly start a SOLAP session according a specific and previously stored context (see Section 4.2.7). Even if a user changes his/her mobile device or chooses to use a desktop computer, this service allows him/her to restore the same context. He/She can also share his/her context information with other users.

SOLAP web services in a disconnected mode are based on the notion of mini-cubes (see Section 4.3). Theses services compose what we name the SOLAP Mini-Cube Creation Service:

16 **ETL-oriented SOLAP Mini-Cube Creation Service** – Most organisational data is present in Online Transactional Processing (OLTP) systems, i.e., operational systems that support transactions such as insertion, update and retrieval of data records. Such systems are not geared towards analysis purposes, mandating the use of Extract, Transform and Load (ETL) processes to populate a decisional data warehouse (Kimball and Ross, 2002; Kimball and Caserta, 2004). The ETL process handles, among other tasks, the mapping between the data sources, which are often normalised, and the de-normalised multidimensional structure of the data warehouse. A similar subsystem could be used and deployed as a web service to create and populate the SOLAP mini-cubes directly from operational data sources. Such data sources can be Relational Database Management Systems (RDBMS), GIS data sets, flat text files, spreadsheets, or even data from other online web services.

17 **ETL System Management Service** – The ETL system itself could either be a custom-made solution using queries and scripts, or utilise an integrated solution from a commercial or open-source provider. In either case, an ETL script repository attached to the system would handle the storage and retrieval of specific data transformation scripts tailored to each data source. The database administrator or
data architect would manage these scripts by the way of a back-end web service. We named this service ‘ETL system management service’. On the front-end, the ETL-oriented SOLAP mini-cube creation service (see Item 16) provides a way for accessing and driving the ETL system, independently of the sub-lying implementation. This is performed by a well-defined service contract that effectively abstracts away the implementation-specific details. In theory, the ETL solution could entirely be replaced by another without breaking the service contract, and thus the interface exposed to the clients.

18 **OLAP-oriented SOLAP Mini-Cube Creation Service** – Organisations already relying on a multidimensional geospatial data warehouse for SOLAP-based analysis may want to extend their investment to mobile clients. In this case, the information available in the data warehouse could be made available to mobile users in the form of mini-cubes. A corresponding OLAP-oriented SOLAP Mini-Cube Creation web service would provide this capability. The mini-cubes can be requested with typical OLAP operators, such as slice, dice, drill-down and roll-up, to select a subcube from one or many existing cubes in the data warehouse. This subcube effectively constitutes a mini-cube, which is transmitted to the web service’s client. The implementation of this web service would rely on an existing SOLAP server, or in other words an OLAP server with spatial capabilities. The requests could be transmitted to the server by the means of a language like MDX, extended by spatial operators and data types. Once again, the service contract exposed to the clients is independent from the actual implementation, meaning that the OLAP server could come from any vendor.

19 **SOLAP Mini-Cube Chaining and Delivery Service** – Once mini-cubes are created, they need to be transmitted to the mobile clients that requested them. This task is conferred to the SOLAP Mini-Cube Chaining and Delivery Service, which constitutes the main entry point to the architecture for all mobile clients. The service will provide the following methods:

a Get information about data available for mini-cube creation, i.e., metadata on the dimensions, hierarchies and measures

b Request the creation of a mini-cube, according to criteria defining the included data. Depending on the time required to perform this operation, the mini-cube could either be immediately delivered to the client, or a notification could be sent later if it takes more time to complete

c Get an existing mini-cube, for example following a notification received by the client after invocation of the previous method.

In order to conform to the web services standards proposed by the W3C and promote interoperability across multiple platforms, the mini-cubes will be stored and transferred in an XML format. This data format could leverage the OpenGIS Geography Markup Language (GML) standard from the Open Geospatial Consortium (OGC) for the representation of geographic entities (OGC, 2004). Compression could be used to reduce the storage footprint and speed the transfer of mini-cubes across wireless network links. Once the mobile
client has retrieved the mini-cube, it can locally and independently perform analysis on the contained data using a SOLAP interface, without needing further connections to the service.

20 **SOLAP Mini-Cube Repository Service** – Such a service can be included in the architecture. Its purpose is to index and store a copy of each created mini-cube. If an existing mini-cube is requested to the Chaining and Delivery Service (see Item 19), it can be sent immediately to the client. In other words, the repository acts as a data cache in the system. Of course, in the case where cached mini-cubes cannot satisfy a query, the Chaining and Delivery service will automatically forward the request to an ETL or OLAP-oriented Mini-Cube Creation Service. In a complex and large-scale environment, we can even envision a combination of multiple data sources, mini-cube creation services and repositories, where the chaining and delivery service uses the most appropriate source to satisfy the requests from its clients (notions of fitness for use and autonomic computing).

Most of the services above will be common and serve both fixed/mobile and wired/wireless SOLAP clients. A prototype implementation of the ETL-oriented SOLAP Mini-Cube Creation Service, along with support services (see Items 16 and 17) is under development (see Figure 6). It will be able to extract data from various sources, while applying a selection (with descriptive, numeric and spatial criteria) to the dimensional members and measures in order to create a mini-cube, representing a subset of the data warehouse. The resulting mini-cubes are then fed to the SOLAP Mini-Cube Repository Service for storage and further delivery (Item 20). The front-end SOLAP Mini-Cube Chaining and Delivery Service has the responsibility of invoking all the previous services when needed, in order to finally deliver the requested mini-cubes to the clients, in an XML format (Item 19).

These web services are being implemented in the Java programming language and in compliance with JMap SOLAP. They rely on the following open-source technologies:

- Pentaho\(^1\) Data Integration (Kettle Project) as an ETL tool
- GeOxygene\(^2\) (Badard and Braun, 2004) as a GIS data framework
- PostgreSQL as a Database Management System (DBMS), along with the Post-GIS\(^3\) extension which enables the support of geospatial data types and operators
- Some tools from the Apache Software Foundation\(^4\) (Axis and Tomcat) as a platform for deploying the different web services.

As mentioned previously in the section, some geospatial web services have to be part of the architecture:

21 **Web Feature Service (WFS)** – This service is standardised by the Open Geospatial Consortium (OGC).\(^5\) It defines interfaces for data access and manipulation operations on geographic features using HTTP as the distributed computing platform. Results are returned in GML format (OGC, 2004), which describes an encoding specification for geospatial data in XML that enables the storage, transport, processing, and transformation of geographic information. This service may also be used to display maps in a vector format on a mobile SOLAP application (see Section 4.1.2) after the geospatial data delivered by such a service are processed by a Geospatial Data Transformation Service (see Item 23).
22 **Web Map Service (WMS)** – This service produces maps of spatially referenced data dynamically from geographic information (spatial databases or GIS files). A map is rendered in a raster format such as JPEG, GIF or PNG, or in a vector format such as SVG. This service can be used in SOLAP mobile applications in order to deliver geospatial data in an easily displayable format (see Section 4.1.2).

23 **Geospatial Data Transformation Service** – This service allows the transformation of geospatial data into new data usable in a mobile context (see Section 4.1.2). As an example of transformation, a compression algorithm using short tags technique (Piras *et al.*, 2004) could be applied to XML documents in order to decrease the data volume exchanged on the network. This service could also invoke the Images and Charts Service in order to produce images or SVG files from a GML document.

24 **Geospatial Data Updating Service** – This service allows for the rapid and consistent propagation of updates on geospatial data stored in mobile devices, databases or data warehouses.

25 **User Positioning Service** – This service allows the identification of the location of a user with his/her mobile device. It can be used by the SOLAP mobile application in order to deliver geospatial or descriptive data adapted to the location of a user and his/her vicinity (notion of context-aware service).

Currently, Digital Rights Management (DRM) must be considered during the dissemination of data through the internet or wireless networks, in a commercial or privacy context. So, specific web services must be integrated in the architecture in order to identify users on mobile devices and transfer secure data over networks:

26 **Authentication web service** – This service is used to authenticate a specific user on a mobile device in order to grant the access to data cubes, with regards to his/her DRM. Before opening a data cube from a mobile SOLAP application, this service can be called in order to authenticate and authorise a user to access specific cubes or services (see Section 4.1.2).

27 **Security web service** – Delivered data on wireless networks can be intercepted by anyone. Security functions like encryption must be developed as services in order to protect integrity and confidentiality of data. WS-Security is the name given to the set of security standards defined by OASIS organisation. All data transfers between a mobile SOLAP application and web services in the SOLAP SOA imply the use of this service about security management.

28 **Billing web service** – In commercial infrastructures, the use of specific data or services must be paid by users. This service is based on an authentication web service in order to grant access to information and store details about the user, his/her payment mode and his/her current balance.

Some services such as WFS or WMS constitute essential building blocks for the design of the SOLAP SOA, other services are of higher level. They provide such an infrastructure with capabilities of intelligence and autonomy. They define the chaining and automate the remote call (with the right parameters) of the relevant and distributed basic services which are needed to achieve a specific task. Such a capability is named intelligent orchestration or choreography of web services (Martens, 2003; Alameh, 2003; Ibach and Horbank, 2004).
Figure 6  The architecture of the SOLAP mini-cube creation services

6 Conclusion

The paper has dealt with a geospatial (web) SOA for mobile processing of geo-decisional information. The underlying concepts of typical SOLAP applications as well as the general concepts of mobile computing have firstly been reviewed. The existing SOLAP functionalities have been presented and their viability/usefulness in a mobile environment has been discussed. Different web services based on an extended geospatial SOA have been then proposed in order to tackle the several issues related to mobile SOLAP.

Future works will first deal with the definition, the compatibility with our desktop/wired SOLAP technology, the priorisation and the implementation of several services detailed in the paper in order to design a consistent web SOA for the deployment of efficient mobile and fixed/wired SOLAP applications. A part of this work has already started. Services dealing with SOLAP mini-cubes creation (see Items 16 to 20) are currently under development. Challenges associated to the design of a SOLAP graphical user interface adapted to mobile devices will also be addressed. Other research works will investigate the issues related to mobility. At present the architecture mainly focuses on the delivery of geo-decisional information to mobile devices. The delivery of information adapted to the profile, the location and the context of the user is not fully addressed yet. Future developments will then deal with the definition of Location Based Services which
will turn the mobile SOLAP applications in truly context-aware geo-decisional applications. Finally, some extensions to the classical SOLAP architectures like mobile hypermedia SOLAP applications are also considered (Bédard et al., 2006).

Acknowledgements

We recognise the financial support of Canada NSERC Industrial Research Chair in Geospatial Databases for Decision-Support and its partners. Please, visit the website: http://mdspatialdb.chair.scg.ulaval.ca.

References


Notes
1 http://www.kheops-tech.com/en/jmap/solap.jsp
2 http://www.w3.org
3 http://www.jfree.org/jfreechart/
4 http://www.w3.org/TR/SVGMobile12/
5 http://www.w3.org/TR/SVGMobile/
6 http://www.w3.org/TR/SVG11/
9 XML Key Management Specification (XKMS): http://www.w3.org/TR/xkms/.
10 http://www.adobe.com/products/robohelp/
11 http://www.pentaho.com
12 http://oxygene-project.sourceforge.net
13 http://postgis.refractions.net
14 http://ws.apache.org
15 http://www.opengeospatial.org/specs/?page=specs