Towards a Service Oriented Architecture for Mobile Reporting

Ilias Michalarias[1] **Veit Köppen**[2]

[1]*Institute of Production, Information Systems and Operations Research*
Berlin-Brandenburg Graduate School on Distributed Information Systems
Free University Berlin, Garystr. 21, 14195 Berlin, Germany
Tel: +49-30-83853244, Fax: +49-30-8385405, E-mail: ilmich@wiwiss.fu-berlin.de

[2]*Institute of Production, Information Systems and Operations Research*
Free University Berlin, Garystr. 21, 14195 Berlin, Germany
Tel: +49-30-8385255, Fax: +49-30-8385405, E-mail: koeppen@wiwiss.fu-berlin.de

Abstract

OLAP applications form a necessary part of every major enterprise. Traditional OLAP systems, using a desktop computer as the end user device, have been a major research domain within the database community thoroughly investigated. On the other hand, there is a growing need for ubiquitous access to enterprise data. In this paper we introduce the notion of mobile reporting in order to denote the concurrent need for mobile OLAP combined with advanced operations for business data analysis, such as simulation, deviation analysis and what-if analysis. We identify the issues that arise and need to be readdressed. We propose various optimization techniques at different architectural levels. We finally propose a service oriented system architecture aiming for a mobile reporting solution, which deals with all contradicting aspects stemming from the inherent shortcomings of mobile devices and the advanced user requirements.

Keywords:

Mobile OLAP, Sampling from Multidimensional Data, Service Oriented Architectures, Broadcasting, Simulation

1. Introduction

A wide variety of mobile devices are available to address a broad range of applications and users. Mobile devices have already gained their momentum, being an indispensable assistant for all types of knowledge users. At the same time enterprises with a large number of employees need to be able to provide them with business data regardless of location and time. The mobile market claims to be in the position to provide credible solutions to almost every problem concerning application deployment in mobile devices. In fact the design, implementation and maintenance of an application running on mobile devices remains a non-trivial task.

There already exist some architectures dealing with mobile OLAP. Nevertheless, to the best of our knowledge, there is no proposed architecture available, which gives advanced analysis facilities to mobile end users, and which at the same time minimizes the interoperability problems stemming from the highly heterogeneous environment of the mobile market.

In this paper we propose an architecture, which provides extended facilities in comparison to already proposed OLAP implementations, running on mobile devices. It is rational to expect that a user does not perform exactly the same operations when using a mobile device and a desktop system as a terminal. Instead, what is required is what we call *mobile reporting*. This term extends the notion of mobile OLAP, in order to encompass additional analysis facilities, such as simulation of business data, deviation analysis and what-if analysis. Generally, a mOLAP user is not interested in performing operations that he could well perform when he has access to a desktop system. More important is, that quick responses to the queries or to what-if analysis are guaranteed, without being concerned about "where and when" of the underlying computations or transformations.

Moreover, both the academia and the commercial vendors have delved into the OLAP domain, producing solid and robust solutions for typical OLAP applications. These solutions make use of sophisticated tools, dealing with the whole spectrum of data warehousing. Despite the fact, that data modeling has been extensively investigated, an equivalent effort was not sustained in the context of the presentation or visualization of data. The majority of the commercial tools are missing a concise and autonomous presentation model for OLAP [1].

This has an immediate impact on the design of a mobile OLAP architecture, since it is the presentational part that needs to be addressed in an entirely different way and which radically influences the whole architecture. This can be easily justified by the limited screen size of mobile devices. We believe that the visualization of data should not be considered only at the front end of any architecture, but should be taken into consideration in the middle tiers, too.

In addition to that, the heterogeneity of the mobile market expressed in terms of available devices, operating systems, synchronization modules and wireless protocols, poses extra and new challenges. To overcome this problem, a service-oriented architecture will be proposed. To forestall any possible criticism, note, that such an architecture may exhibit an inferior performance in comparison with a proprietary architecture, however it is more generic, since it allows the reuse of many software components and can be easily deployed using any kind of middleware as well as in every mobile device, with rudimentary hardware requirements.

The remainder of this paper has the following structure: In Section 2, we briefly present the state of the art of the mobile market and explain the motivation for ongoing work in the field of mobile reporting. We enumerate the basic challenges of the domain in comparison with traditional desktop-based systems, and denote how they influence architectural issues. In Section 3 we describe two fundamental mOLAP architectures, whereas in Section 4 we propose optimization techniques. In Section 5, we shortly illustrate the fundamental user requirements for a mobile reporting application. In Section 6 we demonstrate our proposal towards end- to-end (E2E) service oriented architectures. Finally, in Sections 7 and 8 we respectively present related work and conclude with results and future topics.

2. Motivation - Existing Challenges

In this section, we will explain why an end to end architecture, providing advanced business analysis facilities to a mobile user, demands a completely different approach, in comparison to traditional desktop systems. No detailed description of the inherent shortcomings of mobile devices is given, since those are widely known. Instead, we denote how these shortcomings influence or even dictate some architectural design issues.

The most prominent issue, which should primarily be addressed, is the existence of a wireless, generally unreliable, network instead of a high-speed wired network. Despite the continuous technical progress in the telecommunication field, no extraordinary improvement is to be expected in terms of available wireless bandwidth or connection speed in the near future. Additionally, there are quite a few wireless protocols available further increasing the complexity. It seems clear that the communication between the mobile client and the middle tiers should be Internet based for several reasons. Primarily, this enables access (with security constraints, of course) to all possible data sources. Secondly, it does not imply that the client must be a thin client using a common browser, since a browser cannot fully provide the client with the desired functionality. The client can have a sophisticated user interface, using the network only to receive the data, without being enforced to present them in a specific way like HTML pages for example.

On the other hand, using the Internet we can guarantee that the application is not spatial restrained and bound to a specific carrier, who normally provides limited network coverage.

Another limitation in mobile devices is their CPU. It would not be rational to expect that intense computational tasks can take place in a mobile device. Nevertheless, vendors keep releasing devices with enhanced characteristics, so the client will soon be able to perform some computations site, too. This is also reinforced by the user's requirement to retain some functionality, especially when a network is not available. Therefore a load balancing problem arises, which is influenced by many factors such as the type of computations, the type of the shared devices, the availability and the type of the network.

The memory and storage characteristics of mobile devices are equally important. The majority of existing mobile solutions use some proprietary synchronization module, so that the enterprise database can keep a light database system version residing in the mobile space updated. This violates our fundamental design principle of not sticking to specific platforms or operating systems, but it also deprives us from the flexibility to choose what, when and where to transmit. This highly coincides with the available bandwidth. Our fundamental goal is to transmit through the wireless link as less data as possible, since the bandwidth is the most limited and valuable resource.

The limited screen size of mobile devices should also be considered. The end user interface cannot follow the traditional desktop design principles. It is clear that some visualization techniques must be adopted to overcome this shortcoming.

At the middle tiers, we concentrate on the transformation of data. Massive computations should not overload the client. We should also guarantee that computations are not repeated, whenever possible. Equally important is the ability to concurrently serve multiple clients. This can be achieved by broadcasting data that suit more than one client, even if this data lies within a superset of the requested data [7, 8].

Simulation on the other hand, is a powerful method to analyze business models and to find non-conforming data. This method uses stochastic models and shows up in a wide range of analysis [2, 3]. However, since the computational power required for solving non-linear equation models with errors in the variables is very high, the implementation of such business processes should be carefully designed. One option is that the user can choose between already defined simulation reports. These simulations take place in the application server when idle or not receiving many incoming requests. Another, more resource demanding option is what we call on-demand-simulation. To run a simulation the user has to specify the model (he chooses which part of the overall complex business model is used for his simulation) and supplies the information for the data.

When dealing with such a problem, especially when data needs to be gathered from different data sources, many intermediate layers are involved and the performance is a crucial factor, normally one can think one of the following three approaches:

- *Ad-hoc*: No caching or storing mechanisms are assumed. Nothing beyond the requested is pre-fetched or pre-computed.
- *Materialized*: Every possible request from the end user has to be pre-computed.
- *Partially materialized*: Only one part of the pertinent data is pre-computed.

We select a hybrid approach for our architecture. In most cases, we want to send as less data as possible through the wireless network since this is costly. Meanwhile, in order to provide some offline functionality, highly aggregated data can be sent to the client, for later offline functionality.

3. mOLAP Architectures

One could think of two fundamental architectures, where mOLAP functionality can be provided. For the time being we refer to OLAP operations exclusively, without including any kind of simulation functionality, or any notion of service oriented architecture elements. Let us describe these briefly without, for the time being, going into thorough details. Naturally we do not cite the case where a mobile client merely uses a web browser to query OLAP data. A fundamental requirement for mOLAP is offline operation functionality, which cannot be provided in this case. We assume that clients store incoming data locally, not only to allow offline operations, but also to accelerate subsequent queries.

The first architecture consists of a central server facility, which typically resides in an application server, and several mobile clients. This is shown in Fig.1a. Mobile clients pose queries to one or more data cubes from a Data Warehouse. We assume that the server is able to respond to any incoming query referring to a data cube, mainly by retrieving them from the backend data warehouse, when necessary. In other words, there is no direct connection from the client to the data warehouse. The server is responsible for answering all incoming queries. No communication between clients is assumed. We assume that there is a single broadcast channel that is monitored by all clients and that the channel is fully dedicated to the data broadcast (i.e., the data server can use the entire bandwidth), and of course a uplink channel for posing requests. Clients continuously monitor the downlink channel after making a request, to check for requested data.

A second architecture extends the previous one, by enabling clients to directly communicate with each other, in order to minimize connections with the central server. This is quite rational, since this would not only be more efficient, in terms of access time, but it could also be favorable in economical terms, as many wireless networks are volume based. This architecture is depicted in Fig.1b.

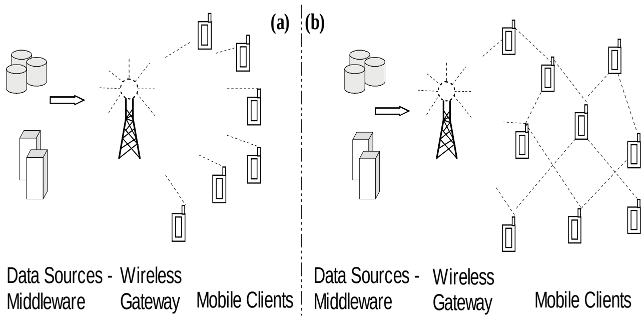

Figure. 1. Two fundamental mOLAP architectures

4.Enhancing mOLAP

The mobile and wireless industry has already surpassed its infancy, having matured to a point, where wide-scale adoption is not unrealistic. Nevertheless, when it comes to highly demanding applications, as in the case of mOLAP, optimization techniques are necessary. In this section, we describe how this can be achieved, by means of broadcasting, simulation and sampling.

Broadcasting / Multicasting

The broadcast mechanism [4, 5], also known as broadcast disk, is adopted to disseminate the data items periodically and continuously in order to conserve the energy and the bandwidth.

In general, there are two modes of broadcasting, the push model and the pull model. Using push, data items are sent out to the clients, without explicit requests, whereas using pull data items come in response to explicit requests. Pull can be used either for unicast or for broadcast and is also referred as on-demand broadcast. The transmission of data is initiated by client request and not based on profiles or subscriptions [6].

In the context of mOLAP, broadcasting or multicasting can prove very beneficial, since they can, in conjunction with cube *subsumption* property, improve the system performance [7, 8]. In the architectures described in the previous section, there is a wireless gateway, which acts as the connection point bridging the wired and wireless world. The gateway can use broadcasting to answer queries posed by several clients and thus reduce consumption of system resources. Instead of establishing two separate connections with two clients, which have requested sub-cubes, which are connected in the *data cube lattice*, which is a directed graph depicting the relationships between all 2^N sub-cubes in a given N-dimensional space. The gateway broadcasts the bigger sub-cube and both clients are being served. One could think of this procedure as a multicast, since we assume that the mobile clients that are not interested in the transmitted sub-cube simply deny the incoming packets.

The Role of Simulation

A data warehouse typically consists of terabytes of data. Naturally, transferring such amounts of data to a mobile client is neither practical nor feasible, since the available resources (wireless bandwidth, client energy level, client processing power) are limited. In this context, including a simulation layer in the architecture of a mOLAP-server is quite rational. Frequently, through simulation of data, a mobile client can in a conceptual level, obtain the knowledge, he asks for, avoiding costly transfer of voluminous data. But beyond the technical perspective, the ability to perform sophisticated simulation analysis enhances the systems functionality.

Simulation analyses

Business Intelligence plays an important role in modern enterprises. One essential part is the analysis of old data and resolving the new state of the business. What-If analysis and Forecast-analyses play an important role in decision processes [3]. To improve results simulation is the tool of choice. Markov Chain Monte Carlo (MCMC) methods are a key device in simulation of business figures, because they are easy to implement and a powerful tool [9].

For simulation of consistent data two ways are possible. On the one hand the complete distribution function of all related data can be approximated. On demand the simulation of the interested information is done. The drawback of this procedure is the exclusion of errors in data. On the other hand an underlying model is needed to include error finding. If a model to the data is given, the distribution functions are simplified because the multidimensionality is reduced. On demand the figures can be simulated and compute with the model.

Sampling from the Data Cube

Sampling of data is an additional option, in order to reduce the amount of transferred data. However, sampling from a data cube is not a trivial task because of the multidimensionality of data. When more than one dimension is involved, the process encounters many obstacles.

Primarily, data should be examined and the dependencies have to be discovered. This task can be done from time to time. The result should be an approximated multidimensional distribution function for all data. Because of the expenditure this process should be done if the server is in idle mode.

The next step is done whenever a query (client) needs data from the Data Warehouse. In the context of the distribution function from step one the sampling for the client is determined. Here Monte Carlo methods are again used because sampling from multidimensional distribution functions is in general impossible with standard methods.

5. Fundamental User Requirements

In this section, the basic user requirements of a mobile reporting application are briefly described. Of course, this is not exhaustive. These requirements play a significant role in the proposed architecture as will become clear in the following section.

Primarily, the goal of a mobile reporting application is to provide ubiquitous access to business data. The end user is only interested in receiving answers to his requests as fast and as accurate as possible. The same applies not only to OLAP data, but to simulation data, too. These results can be presented as histograms, tables, moments of the distribution or as other figures.

Another fundamental user requirement is the seamless and uninterrupted analysis. A possible network interrupt should not prevent the user from performing all kinds of analysis. Some elementary functionality needs to be spare in case the user's device is disconnected.

The end user must also have a user interface, which will balance the effect of the inevitably small size screen, the limited screen resolution and the absence of mouse or keyboard. OLAP by its definition supports the user to make business decision. In this context visualisation of data becomes a crucial factor for the usability of the application. Therefore, we include some advanced visualisation techniques, for example, Table Lens Technique [10, 11].

6. Architecture

We now present the architecture of mobile reporting, a pilot academic prototype enabling OLAP visualisation and business figures analyses for mobile devices. Details about each tier of the architecture are presented.

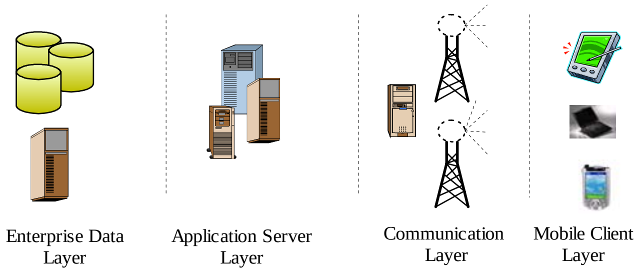

Figure 2. System Architecture

In our architecture there are four separate layers to be distinguished, as shown in Fig. 2: The *enterprise data layer, the application server layer, the communication layer* and *the mobile client layer*. Each one of these carries separate application logic. The software architecture is depicted in Fig. 3.

Enterprise data layer

The enterprise data layer comprises of all available data sources, which normally consists of a data warehouse with the corresponding OLAP server. Generally, the mobile reporting solution does not interfere with this layer. Nevertheless, it is possible to define some operations in the OLAP server, in order to achieve results, relevant with business deviation analysis. For example, a classical business analysis is a comparison between planned and actual data. The corresponding computations can be performed either by the OLAP server (if feasible) or by the application server. Due to performance reasons, it is rational to allocate those computations to the OLAP server. Generally speaking, this layer should remain as transparent as possible to the rest of the architecture.

Application server layer

The *Request/Receive module* is responsible for requesting and receiving data from the data source, i.e. from the OLAP server. The requested data can be the actual data stemming from the data warehouse or data that represent the deviations of some measures and which result from some predefined functions of the OLAP server. Observe that we can have multiple data sources due to various physical locations.

The *Transformation module* transforms the format of the received data into a XML representation. CPM [1], a discrete and autonomous presentation model, enables the clear separation between the modelling and the presentation

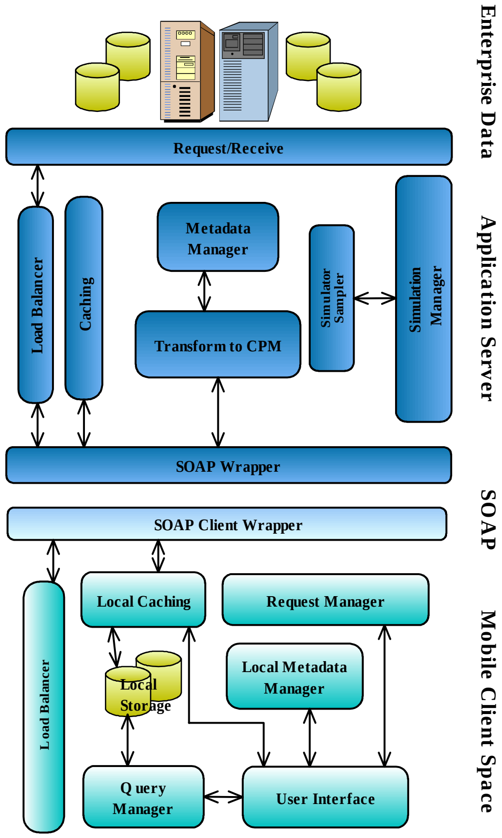

Figure 3. Software Architecture

of data. In this context the transformation module prepares a CPM form. To be more specific normally the CPM entity *Crossjoin* is created. Crossjoins are two-dimensional views of OLAP data, as shown in Fig. 4. They represent what is actually to be transmitted to the mobile client. Nevertheless, we can also transmit other CPM entities, such as a *Multicube*, which represents a presentational form of a hyper cube, so that the client receives more data than needed at his disposal for off-line usage.

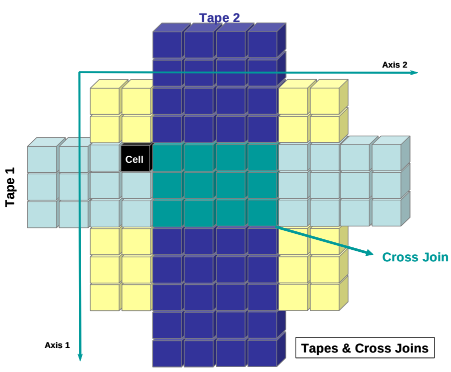

Figure 4. CPM Crossjoin

Naturally, an alternative presentation model can be used. In this case, we can just replace the existing transformation module or even have two different modules, encapsulating functionality in different web services. By performing different transformations, or even by not performing transformations in the application server, the load can be transferred to the client and provides an additional load balancing approach.

The *Caching module* guarantees that frequently requested data are not requested from the OLAP server every time, thus deteriorating the systems performance. Specifically, it contains the most frequently requested crossjoins.

The *OLAP metadata module* is responsible for maintaining the metadata of the corresponding real or simulated data. This is very important in regard to semantics of usage and systems performance, since this module works together with the caching and the broadcasting modules.

The *Simulation module* plays a fundamental role in the application server layer, and decisively influences its performance. Simulation is a very CPU demanding process.

The simulation module adapts its functionality according to the system load. OLAP analysis is supported by simulation with MCMC methods.

Sampling from the fact table is an efficient technique for reducing the volume of the transferred data and consequently, for improving access time.

Finally, we optimize the simulation by estimating the necessary distribution functions for MCMC. This time however, this procedure is done on plan.

Additionally, the simulated model is checked and the simulation is computed. The over-all simulation model describes business figures like Balanced Scorecards, Process Chains or production functions, but most of the time only parts of this model will be requested. The data produced by a simulation run is modelled in such a way, that the communication layer can handle them. Histograms, moments and other business measures can be produced from the simulation and stored into the database, represented by the *Simulation Storage Manager*. This allows to rapidly requests simulation reports that are preliminarily generated.

The *Load-balancing module* can dynamically distribute fractions of computational tasks to the mobile client. Today this might be the obvious choice; nevertheless this module will gain more importance with the enhancement of the mobile devices capabilities. Instead of transforming the data from its initial form to a CPM form, the load-balancing unit sends the data directly to the client. This can be helpful when the server is over-loaded.

Communication layer

The communication layer plays an important role in the proposed architecture. This layer represents the communication aspects both on the server and client side. We make this separation to emphasis the importance of communication aspects, as the operations of this layer could have been well described within the application server and mobile client layers.

A service-oriented architecture based on Web Services allows us to use different platforms, programming languages and operating systems. At the same time, however, the way can be crucial how the data to be transmitted are organised and prepared for client usage. The objective of this layer is to expose such methods with which the client can exploit all the proposed functionality.

The communication protocol used in Web Services is SOAP. *The SOAP Wrapper* is responsible for formatting OLAP or simulated data in such a way, that the data can be transmitted by this protocol, since specific rules must be followed. We have defined some XML schemas to specify the way that our data are encoded inside the body of the SOAP messages. All responses are encoded in SOAP messages. The client interprets them using the corresponding WSDL files, [12].

The client side communication module ensures that when more data than requested arrive, they are not rejected. It acts as a filtering module, though without discarding what was not requested. The SOAP Client Wrapper represents this. This data is stored locally and can be used for example for off-line usage or simply to avoid a new communication session with the application server.

The client is free to use every platform, which supports the deserialization of soap messages, namely the mapping from XML structures to the supported object model.

Mobile Client layer

The mobile client layer encapsulates all the functionality that the end user requires. We propose two different alternatives in order to make our architecture applicable to as broader spectrum of devices as possible. Primarily we want to use the exposed by the web services methods, only to receive the data, in an appropriate for visualisation format. We further want to apply advanced visualisation techniques to provide a sophisticated user interface. When this is not feasible, for example due to insufficient device or platform capabilities, we can resort to a more lightweight solution, making a plain rendering with XSLT transformations, so that the requested data are shown in the web browser of the device.

When broadcasted data is received, the *Storage Manager* stores data and metadata locally to enable off-line operations. This data can be either plain crossjoins, ready for direct visualisation, or aggregated data contained in a multi cube. This module works in conjunction with the communication module to be able to distinguish between the actually requested data, and data that might not have been requested, but were received as a result of a broadcast.

The *Query Manager* is responsible for locally querying stored data The user can define when and how the stored data are synchronized with the enterprise data. Since we refer to OLAP applications, where data are less frequently updated, the synchronization is not of the utmost concern, within this architecture. Nevertheless the user can perform ad-hoc queries and analysis. In this case, the caching module in the application server and the local storage in the mobile device are ignored.

The *Local Caching module* implements a tiny cache within the client's space to contribute to the systems performance.

The *Request Manager* is responsible for passing the requests initiated from the user interface to the communication layer.

The *User Interface* module enables the user to control the application, namely to request OLAP data, navigate through the dimensions, make comparisons between planned and actual data and order the execution of simulations.

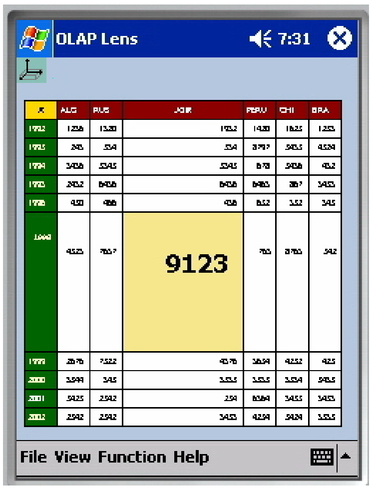

Figure 4. User Interface Screenshot

In Figure 4, an example of what the user sees is depicted. The magnified cell indicates a value with bigger presentational importance, abiding to the Table Lens Technique [10, 11].

The client *Load Balancing module* is optional, since the computations within the mobile client require processing power and memory, beyond the current standards. But since this is certain to change in the near future, when some of the data received in the client are raw, namely they have not been yet transformed into a presentational form, then an integration module is necessary to provide transparency to the user interface module.

7. Related Work

There already exist some proposals, dealing both with the mobile OLAP domain and with the implementation of OLAP operations using web services.

At the data visualization area, in [15, 16] parallel co-ordinates approaches are presented, that scale well for high multidimensional data. Cube View, [1], is an academic prototype system, which provides a generic approach, towards the visualization of OLAP data, both on desktop systems and mobile devices. The focus is on the efficient presentation of data, using non traditional visualization techniques.

Hand-OLAP, [13], is a proposed system for delivering OLAP functionality to mobile clients. In this approach, issues of compression and summarization of data have a leading role. The main purpose of this system is to allow a handheld device to request a bulk of information coming from an OLAP server distributed on a wired network, and store the received (compressed) data locally, in order to query the received information off-line.

On-demand scheduling algorithm, called *STOBS and FCLOS* can be found in [7, 8], which exploit the derivation semantics among OLAP summary tables. They maximize the aggregated data sharing between mobile users and reduces the broadcast length for satisfying a set of requests compared to the already existing techniques. In [17] a novel caching framework for clusters of mobile devices is proposed.

In regard to OLAP service oriented related work, there exist the XML for Analysis specification [14], originally co-sponsored by Hyperion and Microsoft. This is a proposed standard for exchanging multidimensional data. The functionality is exposed to the clients via web services. The simplified interface model has two web methods. Its main drawback, concerning our architecture, is that it does not allow us to make data transformation, before transmitting them to the client. This approach is suitable for plain OLAP visualization since it essentially dictates a rendering with XSL transformations. Furthermore this standard has not yet succeeded in gaining wide adoption from the database vendors.

8. Conclusions and Future Work

In this paper, we have introduced the notion of mobile reporting, in order to provide ubiquitous access to business data and advanced operations. Plain visualization of OLAP data is generally not quite adequate. We claim that additional analysis, such as simulation reports, deviation analysis or what-if analyses are strongly needed by middle to top management. We explained why building application targeting mobile devices needs a completely different approach, in comparison with traditional desktop OLAP applications. We proposed optimization techniques, by means of broadcasting, simulation and sampling. On this basis, and taking into consideration the basic user requirements, we proposed a service oriented architecture, based on the Web Services technology. Thus we managed to provide a generic approach, which alleviates the problems stemming from the currently highly heterogeneous domain of mobile applications, but which at the same time also takes into consideration the constantly evolving characteristics of mobile devices.

We do not claim that the proposed architecture provides all possible functionality to the user. Future research must focus on the so-called what-if or why analysis, a term which refers to the discovery of the reasons or factors affecting the deviation between planned and actual data. Additionally, we aim at the inclusion of business analysis based on balanced scorecards and its relationship to further measures of a firm's behaviour. Another important point of interest is security. Many security aspects must be built into the architecture in order to be accepted on large scale.

9. References

[1] A.Maniatis, P.Vassiliadis, S.Skiadopoulos, Y.Vassiliou, G. Mavrogonatos, I. Michalarias: A Presentation Model and Non- Traditional Visualization for OLAP, *International Journal of Data Warehousing and Mining, 2005.*

[2] B. Morgan: Elements of Simulation, *Chapman & Hall, 1984.*

[3] V. Köppen, A. Hausmann, H.-J. Lenz: Simulation – A support for Controllers Decision process. *In the Proceedings of ICTM 2005, 1155-1170, Melaka; Malaysia.*

[4] S. Acharya, R. Alonso, M. J. Franklin, S. B.Zdonik: Broadcast disks: Data management for asymmetric communications environments. *In Proceedings of the 1995 ACM International Conference on Management of Data, pages 199—210*

[5] J.-L. Huang and M.-S. Chen: Dependent data broadcasting for unordered queries in a multiple channel mobile environment. *IEEE Trans. On Knowledge and Data Engineering, 16(6), Jun. 2004*

[6] M. Franklin and S. Zdonik, "A framework for scalable dissemination based systems," in *Proc. ACM OOPSLA Conf.*, Oct. 1997, pp. 94–105.

[7] M. A. Sharaf and P. K. Chrysanthis, On-Demand Data Broadcasting for Mobile Decision Making, *Mobile Networks and Applications 9, 703–714, 2004 Kluwer Academic Publishers.*

[8] I. Michalarias, H.-J. Lenz: Dissemination of Multidimensional Data in Wireless Networks using Broadcast Clusters. *In Proceedings of the the International Conference on Distributed Computing and Internet Technology, 2005, Bhubaneswar, India*

[9] W.R. Gilks, S. Richardson, D. Spiegelhalter: Markov Chain Monte Carlo in Practice, Chapman & Hall, 2001.

[10] P. Pirollo, R. Rao: Table Lens as a Tool for Making Sense of Data. *In Proceedings of the AVI 1996 Workshop, pp. 67 – 80, Gubbio, Italy.*

[11] R. Rao, S.K. Card: The Table Lens: Merging Graphical and Symbolic Representations in an effective Focus + Context Visualization for Tabular Information. *In Proceedings of ACM SIGCHI 1994, Boston, USA.*

[12] W3C, 2002, Web services Activity, http://www.w3.org/2002/ws/.

[13] A.Cuzzocrea, F.Furfaro, D.Sacca. Hand-OLAP: a System for Delivering OLAP Services on Handheld Devices, *ISADS 2003, pp 213 – 224, Pisa, Italy, April 9 – 11, 2003.*

[14] XML for Analysis URL http://www.xmla.org/.

[15] Inselberg. A: The plane with parallel coordinates. *The Visual Computer. 1, 69–97, 1985.*

[16] Y. Fua, M. O. Ward, and E. Rundensteiner: Navigating Hierarchies with Structure-Based Brushes. *In Proc. of Information Visualization, October 1999, pp. 58-64.*

[17] B. Liu, W. C. Lee, D. Lun Lee: Distributed Caching of Multi-dimensional Data in Mobile Environments. *In Proceedings of MDM* 2005 05 Ayia Napa Cyprus