BUILDING OLAP TOOLS OVER LARGE DATABASES

Rui Oliveira, Jorge Bernardino
ISEC – Instituto Superior de Engenharia de Coimbra, Polytechnic Institute of Coimbra
Quinta da Nora, Rua Pedro Nunes, P-3030-199 Coimbra, Portugal

ABSTRACT

During the last few years, On-Line Analytical Processing (OLAP) has emerged as a valuable tool for the analysis, navigation and reporting of hierarchically organized data from data warehouses. Still, it remains a challenging task to implement and deploy OLAP tools over large databases, since no standardized architecture exists, which describes the common components and functionality of OLAP tools. Managers all over the world face the challenge of how to analyze their most valuable information. The fundamental nature of the right information analysis is on choosing a tool with the features which best aids a management decision. This paper presents a prototype containing a group of features that are recommended to be implemented in on-line analytical processing (OLAP) tools, for analyzing stored data with quality. The prototype was evaluated on a data warehouse implemented in a relational repository using the star schema but this prototype can be implemented on any database system.

KEYWORDS

OLAP tools, Analytical front-end tools, Data warehouse tools, Metadata and Security, Open source database tools

1. INTRODUCTION

A Data Warehouse (DW) provides information for decision making, which is used by analytical applications (OLAP – On-line Analytical Processing) with different user requirements as referred by Codd et all (1993). Chaudruri, Dayal (1997) agree that the usage of OLAP tools for retrieving DW data is the usual process for obtaining decision making information. Data is extracted from heterogeneous relational databases and other legacy systems, and is stored in DW to be used by organizations to retrieve key information for decision support. This factor leads to the use of complete, high organized, and user friendly OLAP tools to analyze data. DW must have efficient OLAP tools to explore the data and to provide the users with a real insight from the data in a DW. OLAP tools structure data hierarchically, allowing to visualize, it in different ways, enabling analysts to gain insight into data through fast and interactive access to a variety of possible views on information. Two OLAP tools will be analyzed in this paper and, a prototype will be evaluated, which was implemented in an OLAP tool. A critical understanding of the analyzed tools features was a key aspect for proposing the prototype. One of the main disadvantages is the difficulty to implement all of the recommendations, maintaining the practical implementation, also user friendly. All the tools created in the future, who follows this prototype, will have the capability to analyze large databases, implemented in several business areas like, e-commerce, banking, industry, and many other business applications. This paper is organized as follows, in section 2, it is given an overview of related work on OLAP tools. In section 3 we present the most important issues and features that are identified as the most important in an OLAP tool. Key topics like, simplified metadata layers, security measures, typical OLAP operations from the evaluated prototype will also be described, along with the prototype's implementation in section 4. Section 5 concludes the paper with some remarks and future work.

2. RELATED WORK

Data warehousing has become an important strategy to integrate data from heterogeneous sources and to enable online analytic processing (OLAP). One of the most complete reviews on OLAP area was made by Stumme (1998), which uses a methodology oriented by a mathematical approach of all the OLAP processes referring OLAP based operations, like slice and dice, pivoting and other. Body et al (2002) gives a specific approach to the Multidimensional and Multiversion Structure for OLAP Applications, which is one of the most important principles of the OLAP tools, addressed in our prototype. Gorla (2003), refers some features that must considered in a Data Warehouse, specially the importance that OLAP tools has on decision support, it also refers the difference between Multidimensional OLAP and Relational OLAP. There is a very complete paper addressing OLAP security issues, developed by Priebe et al. (2000) which refer how basically in a secure way an OLAP tool must access a database, finishing with a brief comparison in terms of security between some of the more important OLAP tools available at the time, our security suggestions in the prototype are based in these ideas. One of the papers which address some of the important topics that makes an overview of data warehousing and OLAP technology was written by Chaudhuri, Dayal (1997). Mclaren (n.d.) also refers some key topics, on how a database must be structured to enable a query tool, specially the importance of the summary results of a query, provided by a query tool, this paper gave us some topics, that helped to establish how summary data, must be displayed to the end user. Kimball, Ross (2002) in their seminal book introduced the notion of dimensional modeling, which addresses the gap between relational databases and multidimensional databases needed for OLAP tasks. Corcoran (2006), in its paper, provided the motivation for proposing the prototype as a step for a standard for OLAP Tools, making it as complete as compact, avoiding the use of other tools, to complement our implementation, and to maintain These sources and other related research, accompanied by some testing made to existent OLAP tools gives an background that helps the propose of a new approach for building a generic OLAP prototype. Based on this work, we will present in the next section an approach for a complete prototype.

3. OLAP TOOL PROTOTYPE

In this section we will present a couple of common concerns and features that are implemented in our OLAP tool. We've analyzed two OLAP tools, Discoverer Desktop from Oracle 9i, and JPivot 1.3.0 (one commercial tool and one open-source tool). From this study and further work, we recommend the most important features and concerns that one generic OLAP tool must include, which will be presented in the next subsections.

3.1 The Metadata Layer

Metadata is "all the information in the data warehouse environment that is not the actual data itself" as stated by Kimball, Ross (2002). The metadata function is an abstraction layer of the physical model, to address several issues. Chaudhuri, Dayal (1997) presents three types of metadatas. An *Administrative* metadata layer, which includes all of the information necessary, for setting up and using a DW (predefined queries and reports, data mart locations and contents, user profiles, along with other features). The second metadata layer proposed is the *Business* layer which includes business terms and definitions, ownership of the data, and charging policies. Finally the *Operational* layer, contains information that is collected during the operation of the warehouse as for instance, the lineage of migrated and transformed data, usage statistics, error reports, and audit trails. In our prototype's implementation we've followed a simplified approach of the previous idea. The Business layer, doesn't matter for an OLAP tool, since usually OLAP based operations are over specific measures (i.e. average sales, product count, and so on), instead of business terms or definitions. We've also noticed, that the creation of a metadata layer, for the use of statistical measures, like number of errors occurred over the time, and access stats, isn't mandatory, because this information can be stored and retrieved from the database. Based on the described *Administrative* metadata and on software implementation designated by DWS Query & Report tool, an OLAP tool based on the paper from Bernardino (2002), from Critical Software® Software Architecture Specification (2002) the broad evolution of those two metadata architectures, is proposed in figure 1.

This metadata schema is sub-divided into three sections. The group named "Application Access", is the sub-section of the metadata responsible for the users access to the OLAP tool. This group has the table USERS for information about who in an enterprise has access to the OLAP tool.

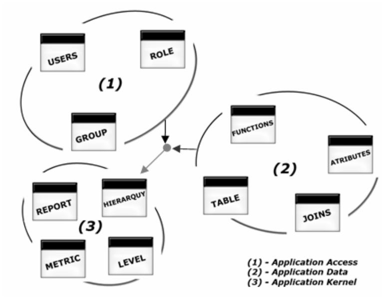

Figure 1. Lightweight Metadata for the proposed OLAP tool prototype

Besides that, the user must have a role (i.e. a Chief Executive must have a higher access level than a Chief of Marketing) this is set by table ROLE. Finally, on the GROUP table, it will be assigned a group of access. The target data of the OLAP tool is defined in "Application data". The table "TABLE" is responsible for providing to the OLAP tool, all tables names that will participate in the OLAP operation, and the table ATRIBUTE is responsible for defining all the attributes related to the specified tables, and its value types. The JOINS table has all the information responsible for defining the joins of the tables defined previous. Only after (1) & (2) configurations finished, it is possible to proceed to, the configuration of the applications kernel. At this point the user can make his personal report templates, and save it to latter retrieving (table REPORT). In the METRIC table is defined all the metric calculations used to create custom measures to analyze a specific business area. The information stored into the HIERARCHY table is responsible for defining a custom tree of related attributes. Within this hierarchies are different levels (branches of the tree), that are configured in the table LEVELS. These two tables are responsible for the ability of the OLAP tool to do drill-down and roll-up based operations (features referred in section 3.3).

3.2 Security

Priebe et al. (2000) agree with, three mechanisms to enforce secure access by using OLAP based operations. The first mechanism consists on the security measures included in any relational DBMS, where the access is possible by creating multiple physical databases, or using SQL Views. On the second mechanism the authentication is made on the OLAP server (when available). On the third mechanism restrictions are made to the data seen by the OLAP tool itself. The two analyzed OLAP tools follow the two first security mechanisms. While Discoverer uses a user created in the Oracle, and its user authentication for retrieving data for analysis, JPivot uses the web OLAP Server Mondrian, to do its authentication. The problem in JPivot 1.3.0 is that it doesn't include a mechanism for the input of username and password of the user data, feature that is required in any OLAP tool. Additionally attributing roles to a user can be a method very effective of security. This prevents, for instance in Oracle, a user from accessing data that its not intended for him to analyze, and it allows that the right individuals can "refine" data and prepare it to be well displayed to the end user client. In JPivot roles are defined in an XML document that contains all the schema of the database. The user Role's are verified by the Mondrian OLAP server at the time of the client connection, by parsing the correct XML schema file. The proposed prototype has the two previous mechanisms. The OLAP tool implemented by the prototype connects to a generic OLAP server, with a username/password from a list of usernames allowed by this server. The OLAP server associates the received *userOLAP/passOLAP* to the generic *userDB/passDB* used to access the data repository. The roles are checked by the OLAP Server based on the user provided credentials, who implements the Application Access section (1) of the prototype metadata.

3.3 Data Manipulation and Reporting

Regular OLAP tools are capable of implementing several useful features. For instance by using slice and dice in an OLAP Tool, the user can retrieve multiple data (attributes from tables), from different hierarchies and join them in a unique spreadsheet. For example it can be retrieved data from the location hierarchy (all->Continent->country->city->address), and from a product hierarchy (all->hardware->Physical Storage-

>Flash disk) by using Drill-Down/Roll-Up OLAP functions, and by cross-joining this data with the sales fact table, the analyst (from an department for manufacturing of hardware) can analyze in the spreadsheet, in which country a particular pen disk is best sold. It is this aggregated data, presented to the user, which is the target of OLAP based functions. The user can also exchange the rows of the spreadsheet with the columns to serve better his needs or preferences, this process is known as pivoting.

Additionally OLAP tools, provide ways for retrieving data which will be displayed in the spreadsheet layout. The first and most user friendly approach, is to have an user interface tree of the database tables, and the spreadsheet is built based on the right selection of the various attributes from these tables, and on measure definitions in a graphical way. Discoverer uses the SQL language to build custom queries directly to the database physical model, which are abstracted using a graphical approach. The second approach requires a deeper level of knowledge, forcing the user to understand how communication between a client and a database server works. JPivot by itself uses an approach for querying and building its “OLAP Table”, known as MDX. This query language was originally created by Microsoft® for OLAP based functions, with the objective of being, an abstraction dialect for communicating easily with the business. It has a defined set of expressions that can be used, which are documented in the MDX Overview (2000) at Microsoft’s site. This MDX language doesn’t replace SQL, because the databases only “speaks” SQL, so what JPivot does, is to implement its own MDX parser (based on Mondrian OLAP Server), which translates all the input MDX expressions, in one or more SQL expressions.

After retrieving the data, Report generation is needed. Our prototype's approach, implements all the referred OLAP operations, data retrieving methods, printing capabilities and generation of file reports in the *.xls Microsoft Excel format, and the Adobe Acrobat *.pdf.

4. PROTOTYPE IMPLEMENTATION

Based in the features described in the previous sections we have implemented an OLAP tool prototype. The prototype is based on the web OLAP client JPivot, having some key features included on Oracle Discoverer. As Mondrian, JPivot uses MDX as its native query language and it is written in Java and JSP. The proposed prototype uses SQL as native query language. A graphical tree interface for selecting the table’s attributes as proposed earlier was developed, enabling a user friendly way of retrieving and selecting the attributes, described in the XML metadata document. It was implemented in the tool the ability of running simultaneously, SQL and MDX query’s. For the implementation of the SQL feature, a short circuit, for accessing the database directly, was required in JPivot+. The two new features are shown in figure 2.

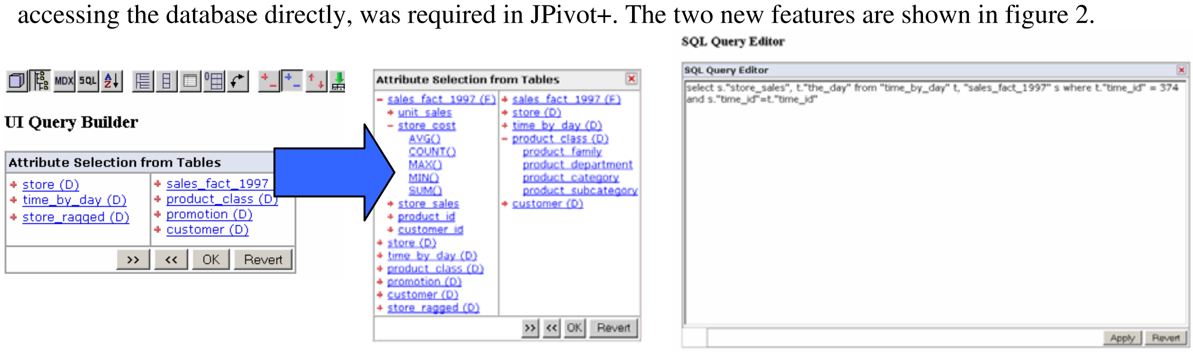

Figure. 2.UI Query Builder and SQL instruction execution input

Previous referred features like drill-down, roll up, slice and dice, pivoting and reporting, are native implementations of JPivot. Access roles to the application it’s actually supported by the Mondrian OLAP Server. Actually the user interface retrieves database elements, from the MDX structure, stored in an administrator defined XML file, and selects them, but further development is needed. When our objective is reached, the OLAP tool will be capable of building ad-hoc SQL statements from the selected fields of this interface. This final important stage of development, will add another approach for data retrieving, meaning that this OLAP tool will have three different approaches, being as most adequate as possible to the various user profiles. Other features like authentication and metric definition for instance, are planned for implementation and it will make this tool very complete comparing with the referred prototype. By enabling user interface interaction with the attributes and, direct SQL injection we've managed to add two features, using at the same time the flexibility that JPivot has for connecting with other DBMS, which enables the manipulation of large databases using the adequate JDBC connection driver. Although JPivot+ is not complete we have verified that it respects all the major principles of our prototype.

5. CONCLUSIONS AND FUTURE WORK

We have described a prototype for developing an OLAP tool. First, it was proposed a generic metadata layer for the prototype tool that communicates with the database physical model. Some security measures that are addressed were also described. Important operational features of the prototype like drill down, roll up, pivoting and slice & dice, as some concerns on building an OLAP tool, were approached in our prototype. We have highlighted, the most common ways an OLAP tool must have for retrieving data, from graphical selection of attributes and measures, to their definition manually by MDX (having the SQL generated in a low level). It was also referred the need of generating reports on an OLAP tool. Finally we've discussed features of the prototype that we've implemented. An important issue that we focus is that the more a tool is easy to use, and the more important features it has, results in a high quality OLAP tool, which will positively be differenced from other existing OLAP solutions. This is the main advantage of the proposed prototype. Because of the various proposed features and key concepts in this prototype, the implementation of all of them in one tool makes it a very difficult and challenging task. We propose for future work, practical implementations based on this prototype (i.e. building OLAP tools using this prototype as standard), Solutions who respect this prototype, can be used and tested for performance in different DBMS for measuring its level of compatibility. In order to follow all the principles of the prototype, further development is needed on our modified version of JPivot.

REFERENCES

Bernardino J., et al. (2002) Approximate Query Answering Using Data Warehouse Striping. Journal of Intelligent Information Systems, 19:2, pp. 145-167.

Body M., et al. (2002) A Multidimensional and Multiversion Structure for OLAP Applications DOLAP'02, November 8, McLean, Virginia, USA.

Chaudruri. S., Dayal U. (1997) An Overview of Data Warehousing and OLAP technology. Sigmod Record

Codd, E.F., et al. (1993) C.T.: Providing OLAP to User Analysts: an it Mandate. Technical Report, E.F. Codd & Associates.

Corcoran M. ,(2006) Business Intelligence goes operational, a white paper available: http://businessintelligence.ittoolbox.com/white-papers

Critical Software®: Software Architecture Specification of the Data Warehouse Query & Report tool, http://www.criticalsoftware.com

Gorla N., (2003) Features to Consider Data warehousing System, Communications ACM, Vol. 46, No. 11

Hasan H., Hyland P. (2001) Using OLAP and Multidimensional Data for Decision Making IT Pro, pp. 1-7.

Mclaren I., (n. d.) Designing the Data Warehouse for Effective Data Mining, http://home.clara.net/imclaren/dmpaper.html

JPivot Open Source Project Available: http://jpivot.sourceforge.net

Kimball, R., Ross, M. (2002) The Data Warehouse Toolkit, Second Edition, John Wiley & sons, Inc

Mondrian Open Source Project Available: http://mondrian.sourceforge.net

MDX Overview available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxbasics_04qg.asp

Oracle Business Intelligence Discoverer, available: http://www.oracle.com/technology/products/discoverer/

Priebe T., et al. (2000) Towards OLAP Security Design – Survey and Research Issues, pp. 33-41

Putting Metadata to work in the warehouse, q white paper (2000), available: http://www.cai.com/products/decisionbase/platbook2.pdf

Security and the Data Warehouse an Oracle, a white paper (2005), available: http://www.oracle.com/technology/products/bi/db/10g/pdf/twp_bi_dw_security_10gr1_0405.pdf

Stumme G. (1998) Online Analytical Processing with Conceptual Information Systems, pp. 1-10