FRAMEWORK AND JUSTIFICATION FOR STANDARD OLAP QUERY LANGUAGE

Andi Wahju Rahardjo Emanuel

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

Jl. Ir. H. Juanda 96, Bandung 40132

Abstract

Bahasa query untuk OLAP adalah bahasa yang dibutuhkan untuk memanfaatkan keunggulan dari aplikasi OLAP dalam mengambil informasi dari data-data belum terolah dari perusahaan. Pengguna-pengguna utama dari bahasa query dari OLAP – analis, manajer, dan eksekutif – yang mempunyai pengetahuan terbatas mengenai pemrograman basis data, menjadikan sebuah tantangan dalam pengembangan bahasa query untuk OLAP. Kerangka kerja untuk menentukan kemungkinan standard dalam bahasa ini dikemukakan dalam paper ini. Berdasarkan kerangka kerja yang dibangun, beberapa bahasa query OLAP seperti JOLAP, OLE DB, nD-SQL (hasil riset), TOLAP (hasil riset), SQL_M (hasil riset) dan XML for Analysis dapat dievaluasi. Kandidat yang patut diperhitungkan adalah JOLAP (Java OLAP).

1. General Overview of OLAP

OLAP is defined as a class of technologies that uses multidimensional view of aggregate data to provide a quick access to strategic information for further analysis (OLAP Council, 2001). The purpose of OLAP is to enable analysts, managers, and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information. It is done by transforming raw data into information so that it reflects the real dimensionality of the enterprise as understood by the user. As a decision making tool for the user, the main capability of OLAP is not just answering "*who?*" and "*what?*", but to answer "*what if?*" and "*why?*".

Unlike the standardization of OLTP query language into SQL, there are many competing commercial and research technologies in the area of query language for OLAP. OLAP query language is unique since it is intended for the non-technical users (analyst, managers, and executives), and must inherently support many different kinds of tools, operations, and environments.

The next subsection will briefly describe OLAP technologies, including its characteristics, key features, operations, tools, and available query languages. The market share for OLAP is also described in section 1.6. The following sections will attempt to develop framework for the justification for OLAP query languages and evaluation of the available query languages. The last section will attempt to predict the future of OLAP query languages.

1.1 OLAP Characteristics

To define the characteristics of OLAP, the FASMI test is proposed by OLAP Report homepage (www.olapreport.com). FASMI is stand of Fast Analysis of Shared Multidimensional Information.

- *Fast*: the OLAP system has to response in fast manner with current expectation less than 20 seconds.
- *Analysis*: the OLAP system is able to cope with any business logic and statistical analysis, and yet easy enough for the target user.
- *Shared*: the OLAP system implements all the security requirements for confidentiality with concurrent update locking at an appropriate level is required if multiple write access is required.
- *Multidimensional*: the OLAP system must support multidimensional conceptual view of data, including hierarchy and multiple hierarchies in the most logical way for business analysis.
- *Information*: the data and information derived must be relevant for the application.

1.2 OLAP Key Features

There are three key features of OLAP as defined by OLAP Council (2001) in "OLAP Council White Paper":

- *Multidimensional Views of Data*: Data is viewed in many aspects (dimensions) more than one-to-one relation in Relational Database. For example financial data is looked by scenario, organization, line items, and time. Sales data is looked by product, geography, channel, and time (OLAP Council, 2001).
- *Calculation-intensive Capabilities*: Ability to do more than simple aggregation but complex calculation, such as share calculation (percentage of total) and allocations (which use hierarchies from a top-down perspective), and algebraic equations such as moving averages and percentage growth (OLAP Council, 2001).
- *Time Intelligence*: Understanding of the sequential nature of time. It also understands the concept of balances over time (OLAP Council, 2001).

- *Other features*: the following is the added features found in OLAP systems but it is specific to particular OLAP applications: *Multiple hierarchies within a dimension*, *Classes within a dimension*, *Derived variables*, *Independently dimensioned variables, Aliases*, *Consolidation Speed*, etc.

1.3 OLAP Inherent Operations

OLAP as an application has several inherent operations related to its purpose in converting raw data into valuable information to its users:

- *Roll Up*: Obtaining higher-level summary from the lower level summary of a dimension hierarchy. For example, having computed total sales per city, the total sales per country can be summarized on the Location dimension, or having computed total sales per product, the total sales per category can be summarized on the Product dimension (Poulovassilis, 2001).
- *Drill Down*: Obtaining lower-level summary from the higher-level summary of a dimension hierarchy. For example: Having asked for total sales per country, total sales per city can also be asked, or having asked total sales per category, total sales per product can also be asked (Poulovassilis, 2001).
- *Pivoting*: Obtaining summary on a subset of a dimension from the summary of a number of dimensions. For example, from 3-dimensional Sales array, pivoting on the Location and Time dimensions to get 2-dimensional array of total sales per locationId and timeId can be obtained (Poulovassilis, 2001).
- *Cube Operation*: Given a measure with k dimensions, there are 2k possible combinations of these dimensions. Recognizing the commonalities between queries over these combinations allow more efficient computation of them.
- *Slicing*: Choosing a sub-cube (choosing a "slice") of a cube by specifying specific values for one or more the dimensions (Poulovassilis, 2001).
- *Dicing*: Choosing a sub-cube of a cube by specifying selection criteria on one or more of the dimensions (Poulovassilis, 2001).

1.4 OLAP Storage Tools

The purpose of OLAP Tools is to provide the users who have little or no experience in database programming with an easy-to-use, powerful, scripting and visual interface for common business-oriented analysis tasks (Poulovassilis, 2001).

- *ROLAP (Relational OLAP)*: This OLAP Tools store all the data in the conventional Relational Database System (RDBMS). The end user queries are translated into one or more SQL Queries and then submitted to the RDBMS (Poulovassilis, 2001). Examples of ROLAP are MicroStrategy, Intelligence Server, and Informix Meta Cube.
- *MOLAP (Multidimensional OLAP)*: This OLAP Tools store all the data in a proprietary, multi-dimensional array data structure (Poulovassilis, 2001). Examples of MOLAP Tools are Hyperion Essbase (OLAP Server), Microsoft PivotTable (desktop OLAP tool).
- *HOLAP (Hybrid OLAP)*: This OLAP Tools use both kinds of data storage: relational database and multi-dimensional arrays for better query performance on aggregated data (Poulovassilis, 2001). Examples of HOLAP Tools are Oracle Express and Microsoft SQL Server Analysis Services.

1.5 OLAP Market Share

The market value of OLAP has reached US$3.7b in 2003 and expected exceeding US$ 4b in the year 2006. The following table shows the market share of OLAP in year 2003. The data is based on OLAP Report homepage (Pendse, 2004a).

Table 1.6.1. OLAP Market Share 2003

Company	Major Product	Market Share
Microsoft Corporation	SQL Server 5.0 Analysis Services	26.1%
Hyperion Solutions Corporation	Hyperion Intelligence, Enterprise and Pillar, Essbase	21.9%
Cognos	e-Planning, Powerplay	14.2%
Business Object	BusinessObjects and WebIntelligence, Crystal Analysis and Report	7.7%
Microstrategy	Microstrategy 7i	6.2%
SAP Institute	SAP BW	5.8%
Oracle	Express and Oracle OLAP Options	4.0%
TOTAL		***85.9%***

1.6 OLAP Query Languages

There are many kinds of OLAP Query Languages, some of them are developed by commercial provider and others are still under research.

- *XMLA (XML for Analysis)*: XML for Analysis is communication API developed by Microsoft with the support from Hyperion and SAS Institute. Many of the OLAP vendors are the member of the council to develop XML for Analysis. This language is the development from OLE DB with no tie to specific platforms. (Pendse 2002).
- *JOLAP (Java OLAP)*: JOLAP is Java-based OLAP Query Language designed by Hyperion Solutions, IBM, Oracle and Sun. This API is described as complete API for J2EE platform. This API only gain little success in gaining support from many vendors.
- *OLE DB for OLAP*: OLE DB for OLAP is Application Program Interface for Microsoft's OLAP product. The expression syntax for OLE DB is called MDX (Multi Dimensional Expression).
- *SQL_M*: SQL_M is an OLAP data model, formal algebra and query language for OLAP application. This technology is developed by Dennis Pedersen, Karsten Riis, and Torben Bach Pedersen from Aalborg University in Denmark.

The standardization of OLAP query language similar to SQL in OLTP is not yet reached. Some framework in justification and analysis for the OLAP Query Language will be presented to provide evaluation for the possible standard OLAP Query Language. The above OLAP query languages are then evaluated based on the identified framework.

2. Framework to Justify OLAP Query Languages

The primary users of the OLAP are analysts, managers, and executives who do not have technical knowledge about database programming. The language comparable to SQL in OLTP (On-Line Transaction Processing) is not suitable for OLAP Query Language since it is designed for IT Personnel with significant programming experience. Some basic key requirements should be established for the query language.

User Orientation

OLAP is intended as analytical tool for decision making for analysts, managers, and executives. So, the main users who use OLAP query language have little or no knowledge about programming language like SQL. The main characteristics for OLAP query language based on user orientation are:

- *Simple*: the OLAP Query Language should be simpler than SQL. This is necessary since the language is intended for non-technical users.
- *Easy to Learn*: the learning curve to study the language should be steep to allow almost immediate usefulness of the language to a new user.

Technology

The technology for OLAP query language should have the following characteristics:

- *Open Architecture or Open Technology*: the base of the technology for OLAP query language should be open or the architecture should be open to allow the continuous improvement and development of the language.
- *Powerful Statistical Support*: The OLAP Query Language should support a wide range of statistical functions such as standard deviation, mean, median, average, moving average, etc.
- *Powerful Mathematical Support*: The OLAP Query Language should support a wide range of mathematical function such as Relational Algebra, Relational Calculus, etc.
- *Powerful Graphical Support*: one way to extract information from a data is to graph the data. So, the powerful graphical support in OLAP query language is required.
- *Inherent Understanding for Time-Series Data*: one of the main features of OLAP dimension is time dimension. The OLAP query language should have inherent understanding for time, time conversion, etc.
- *Inherent Understanding for basic OLAP operations*: the OLAP query language should inherently understand the basic OLAP operations such as Roll Up, Drill Down, Slicing, Dicing, etc. For example, the built in ROLLUP, ROLLDOWN, and DRILLDOWN function within the language.
- *Cross Platform Support*: cross platform support is needed to ensure wide acceptance of the language.

Wide Acceptance

The OLAP query language should have wide acceptance and support from OLAP communities. This can be based on the political reason such as backing from the key players in the market, or the technological reason such as simplicity and powerful technological solutions, etc. Some aspects need to be considered are:

- *Independent or not controlled by particular company*: the dependency of the language to a particular company will make the reluctance of accepting the language by competing companies.
- *Interoperability with many existing OLAP servers and applications*: there are already many OLAP servers and applications being used by enterprises. The ability to operate with many different kinds of servers and applications will ensure the wide acceptance of the language.

3. Justifications for Existing OLAP Query Languages

By using the framework established in previous section. The existing OLAP query languages can be evaluated. There are three primary query languages that are being evaluated for in this paper: JOLAP (Java OLAP), XMLA (XML for Analysis) and research query languages SQL_M. The evaluation is based on the key features, advantages, and disadvantages of the languages.

JOLAP (Java OLAP)

JOLAP is the standard Java-based Application-Programming Interface (Java API) for creating, storing, accessing, and managing data and metadata in OLAP Servers. The JOLAP was initiated in August 2000 by Hyperion, IBM, Oracle, and Sun Microsystem but not supported by Microsoft (Callaghan, 2000).

JOLAP is intended as the counterpart of JDBC (Java Database Connectivity) in the Java 2 Enterprise Edition Platform (J2EE). The final specification for JOLAP is described in JSR-069 from Java Community Process (www.jcp.org). The final version is drafted in September 2003 and final approval in June 2004.

Key features:

- Proposed not only for query but also for creating, storing, accessing and maintaining data and metadata in OLAP server and multidimensional database.
- Based on Java 2 Platform, Enterprise Edition.

- Open standard architecture.
- Foundation Technologies: Common Warehouse Model (CWM), Meta Object Facility (MOF), XML Metadata Interchange (XMI), Java Metadata Interface (JMI) (JOLAP Expert Group, 2001).

Advantages:

- Suported by Hyperion Solutions and Oracle as the primary vendors for OLAP applications.
- Based on Java that is the cross platform. Java has powerful graphical and mathematical capabilities that will enable the JOLAP to have a good graphical support as well as statistical support.
- Strong support from a large section of OLAP communities (excluding Microsoft Corporation) and Java communities.

Disadvantages:

- Not supported by Microsoft Corporation, which has 26.1% market share in 2003 (Pendse, 2004a).
- Java is a language is proprietary to Sun Microsystems.

XML for Analysis (XMLA)

XML for Analysis is the SOAP (Simple Object Access Protocol) - based XML API initiated by Microsoft Corporation in late 2000. The system is evolved around OLE DB but intended to be client and server platform independent. The scripting language of XML for Analysis is called mdXML, which is the encapsulation of MDX in XML statements (Microsoft Corp. & Hyperion Solutions Corp., 2001).

The final specification of the XMLA was released in November 2002 and currently is supported by Microsoft Corporation, Hyperion and SAS Institute. The further development of the XMLA specification is done by XMLA Advisory Council with 20 members. The organization for XMLA is www.xmla.org . The main vendor not supporting XMLA is Oracle.

Key features:

- Can be used for web queries as well as for server-to-server communication (Pendse, 2002).
- Two generally accessible methods: **discover** and **execute** (Microsoft Corp. & Hyperion Solutions Corp., 2001).
- *Discover*: to obtain information and meta data from Web Service for example the list of available data sources and data about the provider of particular data source.
- *Execute*: to execute MDX statement or other provider specific commands against a particular XML for Analysis data source.
- Optimised for web application, especially to minimize the data interaction between client and server by using XML statement as data sharing mechanism.

Advantages:

- Based on Internet standard HTTP, XML, and SOAP from World Wide Web Consortium.
- Support from two main market leaders in OLAP: Microsoft Corporation and Hyperion Solutions Corporation.
- Not only developed for OLAP, but also for Data Mining.

Disadvantages:

- Requirement for data provider to support mdXML language and MDX.
- No support from other key player: Oracle Corporation.

Research: SQL$_M$

SQL$_M$ is an OLAP Data Model, formal algebra, and query language that support irregular dimension hierarchies, automatic aggregation of data, and SQL compatible. This technology is developed by Dennis Pedersen, Karsten Riis, and Torben Bach Pedersen from Aalborg University in Denmark. (Pedersen, 2002).

Key Features:

- Support irregular dimension hierarchies (not just tree hierarchies).
- SQL-Compatible that makes the OLAP can be queried using standard SQL in OLTP.

Advantages:

- SQL-Compatible allows the technology to be easily adapted with current relational database technology.
- The technology is not just a query language for OLAP, but also data model, and formal algebra.
- Allow integration to external XML data.

Disadvantages:

- The technology is still in the research stage with no OLAP technology is currently adapting it.
- SQL query language is already difficult to the non-technical professional who currently uses OLAP such as managers, executive, etc. So making SQL-Compatible language will not support the notion of ease of use for non-technical user.

4. Comparing Query Languages based on Framework

By knowing the currently developed or currently used OLAP Query Languages and the framework, we can put together a table as shown in below:

OLAP Query Language	**Simple**	**Easy to Learn**	**Open Architecture / Technology**	**Statistical Support**	**Math Support**	**Inherent Graphical Support**	**Inherent Time Series**	**Inherent Basic OLAP Operations**	**Cross Platform Support**	**Independent**	**Interoparability of OLAP Servers**	**SCORE**
JOLAP	*	*	*	*	*	*	o	*	*	o	*	**9**
XMLA	*	*	*	o	o	o	o	*	*	*	*	**7**
SQL$_M$	*	*	o	o	*	o	o	*	o	o	o	**4**

Table 4.1. Evaluation of OLAP Query Languages based on the Framework

Note:

* = compy, o = not comply

Based on the table above, the better candidates for the standard OLAP Query Langugae is JOLAP. Being a programming language other that scripting language, JOLAP has the advantage of having strong mathematical support and graphics compared to XMLA. In other aspects JOLAP and XMLA are very similar. The primary pitfalls of the research OLAP Query Languages such as SQL_M, are the inabilities to address all aspects of the framework, and lack of support from OLAP commercial communities.

5. Conclusion

OLAP is an analytical tool used by analysts, managers, and executives to derive information from aggregate data. The uniqueness of users who are no or little knowledge about programming database requires special requirements for OLAP query languages. This will require a language that is simple but powerful in dealing with aggregate data.

Based on the OLAP users, characteristics, key features, and tools, the framework for possible standard OLAP query language is established. The framework is divided into three categories: user orientation, technology, and wide acceptance.

- *User orientation*: simple and easy to learn.
- *Technology*: open architecture or open technology, powerful statistical support, powerful mathematical support, powerful graphical support, inherent understanding for time-series data, inherent understanding for basic OLAP operations, and cross-platform support.
- *Wide acceptance*: independent or not controlled by a particular company, and interoperability with many existing OLAP servers and applications.

There are already many OLAP query languages, both in research stage such as SQL_M; and commercial development such as JOLAP and XMLA. By using the established framework, it can be concluded that the research languages still lack many areas including the needed support from OLAP communities. The noteworthy candidate for unified OLAP query language is JOLAP.

6. Future of OLAP Query Language

The future of the standardization of OLAP Query Language is still uncertain. The commercial interest is still overwhelmingly more dominant than the research interest in the area. The two main approach are still noteworthy: JOLAP (Java OLAP) and XML for Analysis.

There are also tendency of merging between OLAP and Data Mining into a new area called OLAP Mining. It is still unknown whether the new area will need a new query language or the currently developed language such as JOLAP and XMLA capable of handling the new requirement in OLAP Mining area.

References

1. Accrue Software, Inc. 2001. White Paper: An Introduction to OLAP Multidimensional Terminology and Technology. Accrue Software, Inc. White Paper. Available: http://www.accrue.com/olap/wp_intro_olap.pdf. Accessed: 22 October 2001.
2. Callaghan, D. 2000. Microsoft Dishes JOLAP Initiatives. EWeek online magazine. 23 August 2000. Available: http://www.zdnet.com/zdnn/stories/news/0,4586,2618858,00.html. Accessed: 11 November 2001.
3. Everett, D. 2004. Compare and Contrast JOLAP and XML for Analysis. Hyperion Developer Network homepage. Available: http://dev.hyperion.com/resource_library/articles/jolap_xmla.cfm. Accessed: 7 December 2004.
4. Hyperion. 2001. Essbase Enterprise Services Release 6.1 FAQ. Hyperion Essbase homepage. Available: http://www.essbase.com/main.asp. Accessed: 11 November 2001.
5. JOLAP Expert Group. 2001. BOF Session 1078: Java OLAP Interface. Sun Microsystem homepage. 4 June 2001. Available: http://java.sun.com/products/jmi/pres/JOLAPOverview.pdf. Accessed: 11 November 2001.
6. Microsoft Corp. & Hyperion Solutions Corp. 2002. XML for Analysis Specification version 1.1. XMLA organization homepage. 20 November 2001. Available: http://www.xmla.org/spec/1.1/xmla1.1.doc. Accessed: 7 December 2004.
7. Microsoft Corporation. 1999. OLE DB for OLAP 2.0 Beta Specifications. Microsoft OLE DB homepage. July 1999. Available: http://www.microsoft.com/data/download/olap.exe. Accessed: 11 November 2001.
8. Microsoft Corporation. 2000. OLE DB for OLAP Frequently Asked Questions. Microsoft OLE DB homepage. 12 May 2000. Available: http://www.microsoft.com/data/oledb/olap/faq.htm. Accessed: 11 November 2001.

9. OLAP Council. 2001. OLAP Council Whitepaper. OLAP Council homepage. The homepage is no longer available.
10. Pedersen, D *et al.* 2002. A Powerful and SQL-Compatible Data Model and Query Language for OLAP. Department of Computer Science Aalborg University Denmark. Conferences in Research and Practice in Information Technology homepage. Available: http:// crpit.com/confpapers/CRPITV5Pedersen.pdf . Accessed: 1 December 2004.
11. Pendse, N. 2002. Commentary: OLAP API Wars. OLAP Report homepage. 26 April 2002. Available: http://www.olapreport.com/Comment_APIs.htm . Accessed: 29 November 2004
12. Pendse, N. 2004a. Market Share Analysis. OLAP Report homepage. 8 April 2004. Available: http://www.olapreport.com/market.htm. Accessed: 29 November 2004
13. Pendse, N. 2004b. What is OLAP? OLAP Report homepage. 26 May 2004. Available: http://www.olapreport.com/fasmi.htm. Accessed: 7 December 2004
14. Poole, J.D. 2000. Java Specification Request: JSR 69 Java OLAP Interface (JOLAP). Java Community Process homepage. 2 June 2000. Available: http://jcp.org/jsr/detail/069.jsp. Accessed: 11 November 2001.
15. Poulovassilis, A. 2001. PG Dip. in E-Commerce: Database Systems and Data Warehousing. Lecture 6: OLAP. Postgraduate Diploma in E-Commerce, Spring Term 2001 Lecture Notes. January 2001. Available: http://www.dcs.bbk.ac.uk/~ap/teaching/pdecDW2001/. Accessed: 23 October 2001.