On-Line Analytical Processing

Alberto Abelló and Oscar Romero
Polytechnic University of Catalonia, Barcelona, Spain

March 3, 2017

Synonyms

▶ OLAP

Definition

On-line analytical processing (OLAP) describes an approach to decision support, which aims to extract knowledge from a data warehouse, or more specifically, from data marts. Its main idea is providing navigation through data to non-expert users, so that they are able to interactively generate ad hoc queries without the intervention of IT professionals. This name was introduced in contrast to on-line transactional processing (OLTP), so that it reflected the different requirements and characteristics between these classes of uses. The concept falls in the area of business intelligence.

Historical Background

From the beginning of computerized data management, the possibility of using computers in data analysis has been evident for companies. However, early analysis tools needed the involvement of the IT department to help decision makers to query data. They were not interactive at all and demanded specific knowledge in computer science. By the mid-1980s, executive information systems appeared introducing new graphical, keyboard-free interfaces (like touch screens). However, executives were still tied to IT professionals for the definition of ad hoc queries, and prices of software and hardware requirements where prohibitive for small companies. Eventually, cheaper and easy-to-use spreadsheets became very popular among decision makers, but soon it was clear that they were not appropriate for using and sharing huge amounts of data. Thus, it was in 1993 that Codd et al. [5], coined the term OLAP. In that report, the authors defined 12 rules for a tool to be considered OLAP. These rules caused heated controversy, and they did not succeed as Codd's earlier proposal for relational database management systems (RDBMS). Nevertheless, the name OLAP became very popular and is broadly used.

Although the name OLAP comes from 1993 and the idea behind them goes back to the 1980s, there is not a formal definition for this concept, yet. As proposed by Nigel Pendse [13], OLAP tools should pass the FASMI (fast analysis of shared multidimensional information) test. Thus, they should be fast enough to allow interactive queries; they should help analysis tasks by providing flexibility in the usage of statistical tools and what–if studies; they should provide security (both in the sense of confidentiality and integrity) mechanisms to allow sharing data; they should provide a multidimensional view so that the data cube metaphor can be used by users; and, finally, they should also be able to manage large volumes of data (gigabytes can be considered a lower bound for volumes of data in decision support) and metadata. However, there are not measures and thresholds for all these characteristics in order to be able to establish whether one of them is fulfilled or not, and therefore it is always arguable that a given tool fulfills them. Nevertheless, it is generally agreed that in order to be considered an OLAP tool, it must offer a multidimensional view of data.

Since their first days, OLAP tools have been losing weight and lowering prices, while at the same time, offering more functionality, better user interfaces and easier administration. Thus, time has come for small companies to use OLAP. They can afford it and they are willing to use it in their decision processes. Part of OLAP industry was associated into the OLAP Council (created in January 1995), whose aim was the promotion and standardization of OLAP terminology and technology. However, some major vendors never became members of this council, so eventually it disappeared (last news date from 1999). Nowadays, there is no standardization institution specifically devoted to OLAP. Therefore, it seems difficult to have a standard data model and query language in the near future, despite the fact that it is clearly desirable.

Foundations

OLAP environments have completely different requirements, compared to OLTP. Figure 1 summarizes the main differences. Firstly, their usage is different. While OLTP systems are conceived to solve a concrete problem and are used in the daily work of companies, OLAP systems are used in decision support. Thus, in the first case, since the addressed problem can be completely specified, the workload of the system is clearly predefined. Conversely, a decision support system aims to solve new problems every day. Therefore, ad hoc queries are executed. OLTP systems read as well as write data, while OLAP systems are considered read-only, because decision makers do not directly modify data. Nevertheless, the queries in a decision support system are much more complex, since they usually include big volumes of information processed by joining several tables, grouping data and calculating functions. Queries in OLTP systems do not usually involve volumes of data of the same magnitude, neither as many tables, nor groupings or calculations. The number of records in OLTP operations can be estimated as tens or hundreds at most, while OLAP queries usually involve thousands or even millions of records. Finally, the number of users is also dif-

Figure 1: Comparing OLTP Versus OLAP.

Figure 2: Example of cross-tab or statistical table representation of a 2 × 2 × 2 data cube.

ferent in both kinds of systems. OLTP systems can have thousands or millions of users (like in the case of cash machines), while OLAP systems have tens or maybe hundreds of users.

The main characteristic of OLAP is multidimensionality. The data cube metaphor is used to make user interaction easier and closer to decision makers' way of thinking, who would probably find SQL or any other text-based query language hard to understand and error prone. Thus, it is much easier for them to think in terms of the multidimensional model, where a Fact is a subject of analysis and its Dimensions are the different points of view that analysts could use to study the Fact. In this way, the instances of a Fact are shown in an n-dimensional space usually called Cube or Hypercube.

In order to show n-dimensional Cubes in two-dimensional interfaces, Cross-tabs or Statistical Tables such as the one in Fig. 2 (its data is entirely fictitious) are used. While in relational tables it is found that fixed columns and different instances are shown in each row, in Cross-tabs both columns and rows are fixed and interchangeable. In this example, you see three dimensions (i.e., Product, Place, and Year) that show the different points of view to analyze the OLAP tools market.

Multidimensionality is based on this fact-dimension dichotomy. A Dimension is considered to contain a hierarchy of aggregation levels representing different granularities (or levels of detail) to study data, and an aggregation level to contain descriptive attributes. On the other hand, a Fact contains quantitative attributes that are called measures. Dimensions of analysis arrange the multidimensional space where the Fact of study is depicted. Each instance of data is identified (i.e., placed in the multidimensional space) by a point in each of its analysis dimensions. Two different instances of data cannot be spotted in the same point of the multidimensional space. Therefore, given a point in each of the analysis dimensions they only determine one, and just one, instance of factual data. Moreover, data summarization that is performed must be correct, i.e., aggregated categories must be a partition (complementary and disjoint) and the kind of measure, aggregation function, and the dimension along which data is aggregated must be compatible. For example, stock, sum and time are not compatible, since stock measures cannot be added along temporal dimensions.

Operations

Unfortunately, there is no consensus on the set of multidimensional operations and how to name them. However, [14] provides a comparison of algebraic proposals in the academic literature, as well as a set of operations subsuming all of

Figure 3: Schema of operations on cubes.

them. A sequence of these operations is known as an OLAP session. An OLAP session allows transformation of a starting query into a new query. Figure 3 draws the transitions generated by each one of these operations (circles and triangles represent different measures for Fact instances):

1. *Selection or dice.* By means of a logic predicate over the dimension attributes, this operation allows users to choose the subset of points of interest out of the whole n-dimensional space (Fig. 3.a).

2. *Roll-up.* Also called “Drill-up”, it groups cells in a Cube based on an aggregation hierarchy. This operation modifies the granularity of data by means of a many-to-one relationship which relates instances of two aggregation levels in the same Dimension, corresponding to a part-whole relationship (Fig. 3.b from left to right). For example, it is possible to roll-up monthly sales into yearly sales moving from “Month” to “Year” aggregation level along the temporal dimension.

3. *Drill-down.* This is the counterpart of Roll-up. Thus, it removes the effect of that operation by going down through an aggregation hierarchy, and showing more detailed data (Fig. 3.b from right to left).

4. *ChangeBase.* This operation reallocates exactly the same instances of a Cube into a new n-dimensional space with exactly the same number of points (Fig. 3.c). Actually, it allows two different kinds of changes in the space: rearranging the multidimensional space by reordering the Dimensions, interchanging rows and columns in the Cross-tab (this is also known as Pivoting), or adding/removing dimensions to/from the space.

5. *Drill-across.* This operation changes the subject of analysis of the Cube, by showing measures regarding a new Fact. The n-dimensional space remains exactly the same, only the data placed in it change so that new measures can be analyzed (Fig. 3.d). For example, if the Cube contains data about sales, this operation can be used to analyze data regarding production using the same Dimensions.

6. *Projection.* It selects a subset of measures from those available in the Cube (Fig. 3.e).

7. *Set operations.* These operations allow users to operate two Cubes defined over the same n-dimensional space. Usually, Union (Fig. 3.f), Difference and Intersection are considered.

This set of algebraic operations is minimal in the sense that none of the operations can be expressed in terms of others, nor can any operation be dropped without affecting functionality (some tools consider that the set of measures of a Fact conform to an artificial analysis dimension, as well; if so, Projection should be removed from the set of operations in order to be considered minimal, since it would be done by Selection over this artificial Dimension). Thus, other operations can be derived by sequences of these. It is the case of Slice (which reduces the dimensionality of the original Cube by fixing a point in a Dimension) by means of Selection and ChangeBase operations. It is also common that OLAP implementations use the term Slice&Dice to refer to the selection of fact instances, and some also introduce Drill-through to refer to directly accessing the data sources in order to lower the aggregation level below that in the OLAP repository or data mart.

Declarative Languages

There are some research proposals of declarative query languages for OLAP. Cabibbo and Torlone [4] propose a graphical query language, while Gyssens and Lakshmanan [9] propose a calculus. From the industry point of view, MDX (standing for multidimensional expressions) [12] is the de facto standard. It was introduced in 1997, and in spite of the specification being owned by Microsoft, it has been widely adopted. Its syntax resembles that of SQL:

```
[WITH <MeasureDefinition>+]
SELECT <DimensionSpecification>+
FROM <CubeName>
[WHERE <SlicerClause>]
```

However, its semantics are completely different. Roughly speaking, an MDX query gets the instances of a given Cube stated in the FROM clause and places them in the space defined by the SELECT clause. Moreover, complex calculations can be defined in the WITH clause, and the dimensions not used in the SELECT clause can be sliced in the WHERE clause (if not explicitly sliced, it is assumed that dimensions that do not appear in the SELECT are sliced at the highest aggregation level: All).

```
WITH MEMBER [Measures].[pending] AS '[Measures].[Units Ordered]-[Measures].[Units
Shipped]'
SELECT [Time].[2006].children ON COLUMNS,
[Warehouse].[Warehouse Name].members ON ROWS
FROM Inventory
WHERE ([Measures].[pending],[Trademark].[Acme]);
```

In the previous MDX query, an ad hoc measure “pending” is first defined as the difference between units ordered and shipped. Then, the children of the instance representing year 2006 (i.e., the 12 months of that year) are placed on columns, and the different members of the aggregation level “Warehouse Name” on rows. Now, this matrix is filled with the data in “Inventory” cube, showing the previously defined measure “pending” and slicing “Acme” trademark.

Key Applications

Managers are usually not trained to query databases by means of SQL. Moreover, if the query is relatively complex (several joins and subqueries, grouping, and functions) and the database schema is not small (with maybe hundreds of tables), using interactive SQL could be a nightmare even for SQL experts. Thus, OLAP is used to ease the tasks of these managers in extracting knowledge from the data warehouse by means of Drag&Drop, instead of typing SQL queries by hand. The primary idea behind OLAP is to be used to gain quick insight into data, whereas data mining is meant to thoroughly explore the correlations and hidden patterns in the data. Indeed, one naturally follows the other in most cases. In some tools, OLAP functionalities are intertwined with data mining functionalities (so called OLAM).

Some existing alternatives follow the same spirit as OLAP (i.e., quick analysis of data) and are sometimes incorrectly categorized as OLAP tools. This is the case of, for example, QlikView[1], which is based in associative rules. Thus, data is not arranged in a multidimensional fashion (most importantly, the concept of dimension hierarchies is not considered) and the potential analysis tasks enabled by QlikView substantially differ from those empowered by an OLAP tool (and viceversa).

Future Directions

Traditionally, operational data have been collected in the DW of the company by means of ETL flows, and deployed in Data Marts for later analysis with OLAP tools. However, not only real-time analytics, but also situational BI has been recognized as a real need in today world (see [11]). This entails the need of a much faster BI cycle, reducing the intervention of IT specialist at the same time that we integrate more heterogeneous (potentially providing lower data quality) sources.

Indeed, more and more data is available every day. Some come from public institutions (e.g., Open Data Portal[2] offered by the European Commission), and others from private companies like Facebook, Tweeter, etc. This phenomenon is fueling the Big Data business, which is directly related to analytics.

Some proposals, like [1], already appeared to fuse internal data cubes in the companies with external data in the Web. As explained in [2], to enable such possibility, semantics and reasoning are a must. Thus, we need to define the meaning of the data being offered to others. W3C already defined a vocabulary for the exchange of statistical data in [15]. Nevertheless, as outlined in [6], this is not enough and it must be enriched with OLAP metadata.

The role played by external data in current OLAP systems and the need to assist the user to explore these data repositories is addressed in [3]. There, the authors discuss how to capture the semantics of the queries posed by the users

[1] http://www.qlik.com/es

[2] https://open-data.europa.eu

and exploit them to assist the user in her future analysis. Nevertheless, query recommendation should not be the only support provided by OLAP tools but also visualization support and self-tuning techniques according to the usage of the system (e.g., most used fact tables).

Also, OLAP has been traditionally related to the analysis of numerical data (e.g., sales, income, revenue, etc.), whereas new approaches are extending the multidimensional concept to any kind of data. For example, in [8] the authors propose to exploit the cube metaphor to analyze spatiotemporal data and highlight the relevance of designing dynamic dimensions (see for example [7]) and hierarchies (based on the available data) instead of design-time-based dimensions.

Other research directions in OLAP can be the improvement of user interaction and flexibility in the calculation of statistics (see Visual OLAP definitional entry), and the integration of what-if analysis (see What-if Analysis definitional entry). As proposed in [10], OLAP tools need to be extended with writing capabilities in order to provide planning functionalities.

Url to Code

Some OLAP vendors:

1. Microsoft Analysis Services: `http://www.microsoft.com/en-us/server-cloud/products/analytics-platform-system`

2. IBM Cognos: `www.ibm.com/software/analytics/cognos`

3. Oracle Business Intelligence: `http://www.oracle.com/us/solutions/business-analytics/business-intelligence/overview/index.html`

4. SAP Business Objects: `http://www.sap.com/pc/analytics/business-intelligence/software/overview/bi-platform.html`

5. MicroStrategy: `http://www.microstrategy.com/us/platforms/analytics/self-service-analytics`

6. Tableau: `http://www.tableausoftware.com`

Some open source OLAP tools:

1. Mondrian: `http://mondrian.pentaho.org`

2. Palo: `http://www.palo.net`

Cross-references

▶ Business Intelligence
▶ Cube
▶ Data Mart

▶ Data Mining *(to be replaced by new entry "OLAM")*
▶ Data Warehouse
▶ Dimension
▶ Hierarchy
▶ Hierarchical Data Summarization
▶ Measure
▶ Multidimensional Modeling
▶ OLAP Personalization and Recomendation *(new entry)*
▶ Star Schema
▶ Summarizability
▶ Visual On-Line Analytical Processing (OLAP)
▶ What–If Analysis

Recommended Reading

[1] Alberto Abelló, Jérôme Darmont, Lorena Etcheverry, Matteo Golfarelli, Jose-Norberto Mazón, Felix Naumann, Torben Bach Pedersen, Stefano Rizzi, Juan Trujillo, Panos Vassiliadis, and Gottfried Vossen. Fusion Cubes: Towards Self-Service Business Intelligence. *Int. J. on Data Warehousing and Mining*, 9(2):66–88, 2013.

[2] Alberto Abelló, Oscar Romero, Torben Bach Pedersen, Rafa Berlanga, Victoria Nebot, M. José Aramburu, and Alkis Simitsis. Using Semantic Web Technologies for Exploratory OLAP: A Survey . *IEEE T. on Data and Knowledge Engineering*, PP(99):1, 2014. DOI 10.1109/TKDE.2014.2330822.

[3] Marie-Aude Aufaure, Alfredo Cuzzocrea, Cécile Favre, Patrick Marcel, and Rokia Missaoui. An envisioned approach for modeling and supporting user-centric query activities on data warehouses. *IJDWM*, 9(2):89–109, 2013.

[4] Cabibbo L. and Torlone R. From a procedural to a visual query language for OLAP. In Proc. 10th Int. Conf. on Scientific and Statistical Database Management. 1998, pp. 74–83.

[5] Codd E.F., Codd S.B., and Salley C.T. Providing OLAP to user-analysts: An IT mandate. Technical Report, E. F. Codd & Associates, 1993.

[6] Lorena Etcheverry, Alejandro Vaisman, and Esteban Zimanyi. Modeling and Querying Data Warehouses on the Semantic Web using QB4OLAP. In *Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[7] Matteo Golfarelli, Simone Graziani and Stefano Rizzi. Shrink: An OLAP operation for balancing precision and size of pivot tables. *Data & Knowledge Engineering*, 93:19–41, 2014.

[8] Leticia I. Gómez, Silvia A. Gómez, and Alejandro A. Vaisman. A generic data model and query language for spatiotemporal olap cube analysis. In *15th Int. Conf. on Extending Database Technology (EDBT)*, pages 300–311. ACM, 2012.

[9] Gyssens M. and Lakshmanan L.V.S. A foundation for multi-dimensional databases. In Proc. 23rd Int. Conf. on Very Large Data Bases, 1997, pp. 106–115.

[10] Bernhard Jaecksch and Wolfgang Lehner. The Planning OLAP Model - A Multidimensional Model with Planning Support. *T. Large-Scale Data- and Knowledge-Centered Systems*, 8:32–52, 2013.

[11] Volker Markl. Situational Business Intelligence. In *Int. Workshop on Business Intelligence for the Real Time Enterprise (BIRTE), in conjunction with VLDB*, 2008. Informal Proceedings.

[12] Microsoft. Multidimensional Expressions (MDX) Reference. Available at `http://msdn2.microsoft.com/en-us/library/ms145506.aspx`, 2007. SQL Server books online.

[13] Nigel Pendse. The OLAP Report - What is OLAP?, 2007. Business Application Research Center.

[14] Romero O. and Abelló A. On the need of a reference algebra for OLAP. In Proc. Int. Conf. on Data Warehousing and Knowledge Discovery, 2007, pp. 99–110.

[15] W3C. The RDF Data Cube Vocabulary. Available at `http://www.w3.org/TR/vocab-data-cube`, 2014. Recommendation