THE INTERROGATION OF THE OLAP CUBE BY USING MULTIDIMENSIONAL ANALYSES

Robert Gabriel DRAGOMIR
The Faculty of Accounting and Finance
Spiru Haret University
robert73d@yahoo.com

Abstract

The present paper proposes to speak about the interrogation of the OLAP cubes, by using the multidimensional analyses. The first part of the paper deals with the dynamics of the business environment, a fact which triggers the necessity of the informatics systems for decision assistance. The second point of the paper is the presentation of the online analytical processing (OLAP). Connected to this point is the declarative interrogation language MDX, which assures the access to the OLAP interrogation, functions, offering us the possibility of defining the calculated members of the dimensions. A few examples show us how they work. In the end, we draw the conclusions, stressing the importance and the benefits of the intelligent solutions.

Keywords: *interrogation, OLAP cube, MDX language, multi-dimensional analysis*

JEL Classification: C81

Introduction

The unprecedented dynamic of the business environment exposes today the decision makers in front of situations where the informatics systems for decision assistance are indispensable. Within the economic financial environment, the decision is a key element, which has to manifest fully, being a rational element of choosing the optimal variant as concern the financing process and the investments.

The importance of the decision will influence the whole informational circuit which derives from the fact that the objective of each decision has a financial character; it mainly consists in increasing the capitalization, the profitability, assuring a permanent availability of cash, the risk avoidance; all these aspects could impose the firm as a leader of the market.

Literature review

The administration decision represents the main instrument for realizing the established objectives for a certain period of time [Ion-Trifoi G., Dinte C.]. Thus,

an increasing attention is given to the instruments the decision makers operate with, because beginning with the projection stage of the informatics systems we have to take into consideration all elements which influence the decisional process.

The data cube represents a data structure which makes the measure aggregation after the levels and the hierarchies of each dimension we want to analyze [http://office.microsoft.com].

The cubes combine more dimensions with synthesized data. The cubes are not cubes in a mathematic sense, because they do not have necessarily the equal sides, but they represent a good way of describing a complex generic concept.

The online analytical processing (OLAP – On-Line Analytical Processing) is comprised in the sphere of the managers' current activity. The results of the OLAP investigations are realized through data multidimensional analyses and are materialized into concrete, consistent, correct and in real time responses to the analyses questions about phenomena, processes and activities. The main OLAP operations which can be done are the following: the specification of the selection criteria, the rotations, the changing of the aggregation level and the specification of the representation manner.

Among analyses and multidimensional presentations, within the OLAP technology [Lungu, I. et al., 2007], some of the authors of the specific literature also included the informatics systems for the executive support (ESS), but also the data mining (Data Mining). OLAP assures multidimensional conceptual views of the data. The data models associated to the OLAP systems comprise two categories [Muntean, M., 2004]: *extensions of the relational data model* (the *Kimball* model, the *Gary* model, the *Gyssens şi Lakshmanan* model, the *Test* model*, the Li and Wang* model, etc.) and *models based on n-dimension cubes* (the *Agrawal* model*, Guptaand Sarawagi* model, the *Cabbibo and Torlone* model, the *Blaschka* model*, the *MAC* model, the *Vassiliadis* model*, the *Guazzo* model, etc.).

The declarative interrogation language within the OLAP cube

The data representation is a maximal importance point of the OLAP systems, because here there are evidenced all the information behind an immense data volume. The ways of representing the multifunctional data are reported to: the data visualization, their navigation and format.

The existence of great volumes of data leads to the implementation of the modern stocking method that is the data warehouse. A solution for passing from the relational data bases to the transactional data bases is with the help of the SQL server. It is the central element and the interrogation source for obtaining the reports and graphics. Each OLAP instrument has as the start point the table and the more performing and efficient it wants to be, the more complex and various graphic representations it will offer. Many modern instruments present the final information in the bi and tri dimension graphs or allow the multi cub visualization. The cube is practically the place where all data which are to be exposed in a board table are calculated and aggregated.

The declarative interrogation language MDX (*MultiDimensional eXpressions)* assures the access to the OLAP interrogation functions, offering us the possibility of defining the calculated members of the dimensions. Microsoft SQL Server 2008 gives a new dimension to the usage of MDX, by manipulating and finding the multidimensional data bases. The language uses expressions compound by identifiers, values, instructions, functions and operators; with their help the Microsoft Analysis Services can make evaluations for finding an object or a scalar value, represented by a row or number. MDX is a language for interrogating the multidimensional data bases, administrated by the OLAP type servers. In its essence, MDX is capable of sending specific interrogations to the server and obtain different results, under the form of data structures well defined.

The multidimensional MDX expressions realize interrogations over the cubes and return multitude of multidimensional cells which contain cube data. For each and every expression, we can use unique names for the calculated members for eliminating all the ambiguities. In order to identify data, the MDX uses the reference system.

It can identify in a unique manner a cell. A cell in a multidimensional cube is the space determined by the junction of a member of the dimension and a member which came from the attributes hierarchy of the respective cube. A member in a dimension can be: the leaf member, a parent member, a data member or the calculated value of all members.

The MDX interrogations look like the SQL interrogations and require a data request (clause SELECT), a starting point (clause FROM) and a filter (clause WHERE). These instructions will facilitate the extraction of specific parts in an analyzed cube. MDX also offer a great set of functions for manipulating the data, as well as the ability to extend the MDX with the functions defined by the user. There are specific commands for creating or erasing cubes, dimensions, measures or other subordinated objects. The scope of the MDX expressions is that of facilitating the data accessing from many dimensions. The SQL language uses two dimensions: column dimension and row dimension. Within every table, at the row and column junction, there is a single data element, called the field. Within the multidimensional data comprised in a cube, the structure presents multiple dimensions. The junction of the dimensions in a cube determines the measure, which has one or more elements of data type. Each dimension can be detailed on certain levels, which, in their turn, contain members. The common components of the MDX include: tuple, axes and multitude. In order to identify and extract data, the MDX language uses a reference system, called tuple. It represents a collection of members, each of them being selected from a dimension.

The syntax for a tuple is:
– *[Dimension].[Member]* – for a single member and a single dimension;
– *([Dimension_1].[Member], [Dimension_2].[Member]... [Dimension_n].[Member])* – for many members and dimensions.

The general syntax of the MDX SELECT instruction is the following:

[WITH <SELECT WITH clause> [,<SELECT WITH clause> …]] SELECT [*] (<SELECT interrogation axe clause> [,<SELECT clause interrogation> …] (FROM <SELECT clause cube name> [< SELECT clause axe section>] [< SELECT clause cell properties>].

The key word WITH is optional. In order to define a value domain (Range), one can use the ":" sign; it has the role to unite the extremities. The first 5 axes have been standardized, having the following predefined values: X axis – ON COLUMNS, Y axis – ON ROWS, Z axis – ON PAGES, V axis – ON CHAPTERS and W axis – ON SECTIONS.

The junction of multiple tuples is another tuple. Unlike the relational model, where the tuples of a table have the same structure, within the multidimensional model, it is possible for us to obtain tuple with a different structure after an interrogation.

The usage of the MDX language

Example no. 1

 – MDX interrogation to find out which are the selling in the Muntenia and Transylvania regions for the MAN, VOLVO and IVECO trucks.

SELECT{[Trucks].[Name Class].&[MAN],[Trucks].[Name Class].&[VOLVO],[Trucks].[Name Class].&[IVECO]}

ON ROWS, {[Shops].[Region].&[Muntenia],[Shops].[Region].&[Transylvania]}
ON COLUMNS FROM CARS_CUBE

Any axis nominates a collection of members derived from one or more dimensions. The filtering of the specific cube values along the members with associated values represents the way of constructing the axes. During an interrogation, the syntax of an axis will use the axis specification after the key word SELECT. A set is comprised between accolade parentheses, within the same instruction SELECT and nominates those cube cells whose values are to be returned under the form of a response to the interrogation.

The sets are tuples collections both on rows and on columns that is we have valid tuples on the both axes. On the basis of all MDX interrogations is the SELECT instruction. An MDX interrogation has as definite: the number of axes, the members of every dimension, the name of the interrogated cube and the members of the sectioned axis.

The general structure [Spofford, G., Harinath S., Webb C., 2006] of an MDX interrogation consists in:

✓ *The key word SELECT*: it is the word which initializes any interrogation;
✓ *Axes where the results of the interrogation are obtained* (maximum 128): the words ON designate the columns axis (COLUMNS) and the line axis (ROWS). Usually, the results of the MDX interrogations are represented in a bi dimensional form, but there can be also the third dimension (PAGES), or even further dimensions (CHAPTERS, SECTIONS).

✓ *Accolades {,}*, which designate the definition of a set of elements in a dimension or a set of dimensions. The elements in a set are separated by commas and the names within a hierarchy are included between square parentheses [,].

✓ *Clause FROM*, which specifies the name of the data cube, from where we can obtain the interrogation results. It is important to mention that MDX is not a language of a CASE-sensitive type, which is of no importance if we write the names in uppercase or lowercase letters.

✓ *Clause WHERE*, is the one which allow the user specify the conditions referring to the data set which is to be introduced into the result. This clause is an optional one.

Example no. 2

– an MDX interrogation which presents the number of cars with the acquisition value between 180000 and 250000 Euros, sold between the years 2009-2010: SELECT{[Data].[time].[an].&[2009]:[Data].[time].[an].&[2010]}

ON COLUMNS, {[Cars].[Value].&[180000]:[Cars].[Value].&[250000]}

ON ROWS FROM CARS_CUBE WHERE [Measures].[Selling Count]

Example no. 3

In order to determine the value of the annual expenses for every shop in every locality, we can apply the following MDX interrogation.

SELECT {[Data].[time].[an].MEMBERS}ON COLUMNS,

{[Shops].[Locality].MEMBERS}ON ROWS

FROM CARS_CUBE WHERE {[Measures].[Expenses]}

On the column axis, a list with the values of the level [year] are to be generated and on the line axis will appear a list of the level [Shops][Locality].

Example no. 4

If there is necessary the obtaining of a centralized situation of the previous values, we can use the indicator of supplementary selection SELF_AND_BEFORE, which presents the required values. If in our example, we want information about Dobrogea region,

SELECT{[Trucks].[Name Class].MEMBERS}

ON COLUMNS, {DESCENDANTS ([Clients].[Clients income].[Ad Region].&[Dobrogea],[Clients].[Annual Income],

SELF_AND_BEFORE)} ON ROWS FROM CARS_CUBE WHERE [Measures].[Selling Count]

Thus, we will obtain the synthetic value of the sellings for Dobrogea "parent".

The calculating operations in the MDX language make recurrence on members' functions which access every member of a dimension and return a member or a zero result. The strong point of the MDX language is represented by the generation of the calculated members grouped into permanent and virtual members. The used operators are: "accolade parentheses", ":" and "." From the member functions, we mention: *.Parent* which returns the parent of the specified

member; *.Children* which returns the child of the specified source member; *Ancestor ( )* finds out a member on a level or on a distance: the parent, the grandparent, for the specified source member; *Cousin ()* begins with the members order and position on the levels of a dimension in a hierarchical structure and returns the child member which take the same relative position under a parent member; *FirstChild* returns the first child of the specified member; *.LastChild* returns the last child of the specified member; *.FirstSibling* operates with the concept of horizontal order of the hierarchical data structure; *.LastSibling* operates with the concept of horizontal order of the hierarchical data structure, thus the member which is the last child on the horizontal level will be the last brother for any other member at the same level.

Any interrogation addressed to the data cube has as a result a cells collection. The role of the calculated members is to offer a place where to be put the calculating results; the simplest form of a calculated member is represented by a member of a dimension and a calculating formulae which refers to that member. A calculated member has three components:

- ✓ *Identifier calculated member,* which supposes the establishing of the name, the dimension and of the hierarchical position for the specified member;
- ✓ *The calculating formulae of the calculated member,* with its help we can calculate;
- ✓ *The optional properties*, which can refer to the supplementary calculation and other types of information.

A calculated member is introduced into the interrogation before the SELECT clause, through the indicative WITH. The member has to be associated with a cube dimension, which is the *name member* will also contain the name of the dimension to which the member belongs.

The general syntax is the following:

MEMBER <name_member>   AS   <formulae_member> [,<supplementary_properties>]

The defining of a member by using the WITH clause within an MDX interrogation makes it available inside the interrogation, but it is not visible from any other interrogations. The calculated members focus on three types of domains or levels: the domain of the current interrogation (*query scope*); the domain of the user session (*session scope*); and the global domain of the cube (*global scope*).

Conclusions

The actual decisional process bases on the OLAP models. The OLAP instruments have a greater and greater ability to define the way of defining a variable. Ideally, an OLAP instrument should allow an implicit format of a variable display; it should be associated to the variable.

The interrogation will precede the results and the way of presentation has to take into account the user's needs. The decision makers have some expectations connected to the final format of the presentations. Once the calculated members

have been defined and calculated, the used software programs have to export the data for visualizing. Many programs developers create applications which are able to import these types of resulted data and also display the final data in more and more attractive and intuitive formats. The design of the final reports assures a great part of the intelligent solution. Such a solution is offered by the Microsoft Excel 2010, through PivotTable and PivotChart.

REFERENCE

Lungu, Ion et al. *Sisteme informatice executive*, Bucureşti: Editura ASE, 2007.

Muntean, Mihaela. *Iniţiere în tehnologia OLAP. Teorie şi practică*, Bucureşti: Editura ASE, 2004.

Niemi, Tapio, Niinimäki Marko, Nummenmaa Jyrki, and Thanisch Peter. *Constructing an OLAP cube from distributed XML data*, Eleventh International Conference on Information and Knowledge Management, November 4-9, 2002, McLean, VA (near Washington, DC), http://www.cikm.org/2002/#program.

Niemi, Tapio, Nummenmaa Jyrki, Thanisch Peter. *Constructing OLAP cubes based on queries*, Proceedings of the 4th Association for Computing Machinery international workshop on Data warehousing and OLAP, November 5-10, 2001, Atlanta, Georgia.

Pardillo, Jesús, Mazón Jose-Norberto, Trujillo Juan. *Bridging the semantic gap in OLAP models: platform-independent queries,* Proceedings of the ACM 17th Conference on Information and Knowledge Management, California, October 26-30, 2008.

Spofford, G., Harinath S., Webb C. *MDX Solutions*, Second Edition, Indianapolis: Wiley Publishing Inc., 2006.

Ion-Trifoi, Gigi, Dinte Constantin. *Decizia şi rolul său în managementul financiar al întreprinderii*, Annales Universitatis Apulensis Series Oeconomica, Universitatea "1 Decembrie 1918" Alba Iulia, http://www.oeconomica.uab.ro/upload/lucrari/820062/55.pdf.

http://office.microsoft.com/ro-ro/excel-help/prezentare-generala-a-olap-online-analytical-processing-hp010177437.aspx.