THE TRANSFORMATION OF THE OLAP APPLICATIONS INTO THE OLAP MULTIDIMENSIONAL DATABASE

Jana Ficová
Investment and Development Bank

Introduction

Systems designed to provide fast data access, summarization and analytical processing a mass of data are called On Line Analytical Processing (OLAP) systems. OLAP systems give end-users an opportunity to make a data analysis through their analytical needs and categories represented by individual views or dimensions. This kind of view on data is called a multidimensional view. In nowadays we can register a rapid development of OLAP tools and multidimensional databases.

The SAS System provides the OLAP tools mostly through the module SAS/EIS Motore as the extension of the module SAS/EIS. This module offers convenient tools for a rapid development of OLAP applications. In the Investment and Development Bank with OLAP applications strategic bank’s managers can rapidly examine key performance data through unlimited views of summarized data, even on very large data volumes. The data are summarized on various levels and are stored in so called base and summary tables.

The other, rather new possible way of OLAP application development is to use the OLAP Multidimensional Model / Database (MDDB). The structure of the MDDB represents a progressive way of the data storage and offers saving a disk space as well as improving the data access and making it faster.

The paper deals with a short experience with the OLAP MDDB in the Investment and Development Bank and describes the process of movement from SAS/EIS Motore OLAP applications towards OLAP applications built on the MDDB.

The first part of the paper describes the process of building OLAP applications using the SAS/EIS Motore. The second part deals with the basic principles of SAS MDDB and compares them with the base and summary tables used in SAS/EIS Motore. In the subsequent part we mention the practical experience with MDDB in the Investment and Development Bank. In the conclusion we would like to summarize knowledge obtained while using MDDB as well as the reasons that lead to the transformation of the SAS/EIS Motore applications into the applications built on MDDB.
1. SAS/EIS Motore

Module SAS/EIS Motore is a tool for rapid OLAP applications development. It fills a gap between SAS/EIS objects and SAS/AF® coding that makes the application development easier and increases the flexibility during the screen design.

During the application development the Multidimensional Motore (MDM) is set up in order to access and provide data, all details that are unimportant for the rest of the application are hidden. The developer customizes the MDM class by adapting it to the available data values and dimensions according to the needs of the end-user.

The developer specifies the SAS data tables containing the information to locate the native and the calculated data values and present them to the user. For the calculated variables it is necessary to specify formulas and calculation routines.

There are two types of SAS data tables:

- **Base table** contains the most detailed view on the summarized data.
- **Summary tables** contain those pre-summarized data combinations that are assumed to be most frequently used during the running of the application. The number of summarized tables is a result of the tuning of the application concerning available disk space, storage requirements and demanded end-user response.

In a presentation or a screen design layer, the SAS/EIS Motore provides a rich gallery of ready-to-use classes. For example the visualization classes represent data in a graphical form (pie charts, bubble charts, two or three dimensional graphs) or in a tabular form. The navigation classes are representations of defined dimensions and allow to navigate within them. Using them a user sends to the application the need to display a specific set of data.

The developer has the possibility to set various attributes to the objects, such as colors, fonts, legends, font size, sort order, etc.

However, the individual objects are not available to the developer for the customization. In that way the developer cannot change the behavior of the objects. Similarly, it’s not possible to derive new classes from already existing ones.

2. SAS Multidimensional Model / Database

The other way of OLAP application development is to use the OLAP MDDB. SAS Multidimensional Model / Database is a specific multidimensional structure; a group of cells arranged by the data dimensions. In effect, this is a combination of both a logical and physical structure, designed to provide fast, intelligent data access.

The following terms are commonly used in the connection with MDDB structure:

- **Dimension** is a logical structure to navigate the data model. For example the Region dimension, the Time dimension, the Product dimension.
♦ **Member** is a discrete name to identify position and description within a dimension. For example the Region dimension has a member named Bratislava.

♦ **Level** is used to navigate a dimension that consists of several hierarchies. For example the Region dimension may consist of 2 levels - Main Branches and Cost Centers of a bank.

♦ **Value** is a value of the data point in the MDDB. For example, the value for the Demand Deposits, for the main branch Bratislava on April, 10, in 1998 is 20 millions Slovak Crones.

♦ **Cell** is similar to a value except that one cell contains the values of all members.

♦ **Sparsity** describes how full the cube is for the possible combinations for all dimensions.

The following terms are used when describing SAS MDDB structure:

♦ **Base table** represents the original source data for creating MDDB. It means that a base table is not a part of MDDB.

♦ **NWAY Crossing** is the same as the Base table in SAS/EIS Motore applications. It provides the most detailed data information as it contains all dimensions and dimensional levels. It is stored within the MDDB.

♦ **Subtable** contains pre-summarized data on the selected summarization levels across individual dimensions and their levels. MDDB may contain several subtables. In comparison with SAS/EIS Motore, each subtable is a part of the MDDB and cannot be stored and accessed separately. It means that the NWAY Crossing represents a specific subtable stored within the MDDB.

It is possible to create MDDB in one of the following ways:

♦ interactively in the SAS/EIS environment using the **SAS/EIS-MDDB**

♦ using the **procedure MDDB**. This allows MDDB to be created in batch.

♦ in the SAS/AF environment with the **Screen Control Language (SCL)** using the available methods of the MDDB Class.

**SAS/EIS-MDDB** enables to create object that makes an MDDB from input pre-registered SAS data table (Base table) and automatically registers the output MDDB. The MDDB and metabase registration are created after the class execution. While interactively defining MDDB there are three predefined collections of subtable definitions:

♦ **Most Paths Covered** - covers most crossings which are supposed to be needed most frequently at run time

♦ **Use Minimal Disk** - creates and stores only the NWAY crossing, all subsequent needed combinations will be calculated at run time
Best Performance - creates and stores all possible crossing definitions and the MDDB contains every possible subtable. This way allows that no subtables are calculated at run time. On the other hand, this will occupy the most disk space.

The procedure MDDB is the easiest way how to create MDDB in batch mode as this is a standard SAS code. The syntax of the procedure MDDB is the following:

```
proc MDDB
  data= dsname
  out= libref.outmddb
  in= libref.inmddb
  label= description
  pw= "password";
  class var1 var2 ... / order-options;
  hierarchy class_var1 class_var2... / name= name|"name"
  display= YES|NO;
  var var1 var2... / stat-options;
run;
```

“class”: defines variables from the SAS dataset used as the classification variables in the MDDB.

“hierarchy”: each hierarchy statement means a creation of one subtable in the MDDB for quicker access to data. We can specify zero or more hierarchy statements. If no one hierarchy statement is specified we will get just the NWAY hierarchy.

“var”: defines variables from the dataset used as the analysis variables in the MDDB. We can specify one or more variable statements. The variables must be numeric. If no statistic is specified the statistic SUM will be used.

Created MDDB can be used (Figure 1):

♦ in SAS/EIS Motore applications where MDDB stands for the base table and summary tables.

♦ in SAS/EIS applications. The module SAS/EIS provides the following classes that are capable of using MDDB:

  Multidimensional Report
3D Business Graphs
Map
Organisational Chart
Graphical Variance Report

Figure 1
3. MDDB in the Investment and Development Bank

During the initial phase of the Management Information System development in the Investment and Development Bank in 1996, we used mostly the module SAS/EIS Motore as the extension of the SAS/EIS module. It provides possibility for a rapid OLAP applications development. From that time the Motore version has been upgraded several times. Currently we are working with the version TS 5009 of the SAS/EIS Motore.

With the consequent development of the OLAP tools and the realization of the idea of storing data in a special multidimensional structure, the SAS Institute has come in the version 6.12 with the support of so-called Multidimensional Model / Database (MDDB). MDDB contains pre-summarized data that allows users faster access to large amounts of data. The module SAS/EIS supports the MDDB as well as the SAS/EIS Metabase supports the registration of MDDB. Similarly, the SAS/EIS Motore is capable of working with MDDB.

The transformation of the existing SAS/EIS Motore application into the application built on MDDB required the MDDB creation. According to the SAS specific MDDB terminology, the prebase table used as input for Motore base table creation, turned into the base table for MDDB creation. While creating MDDB we used the second way mentioned above of creating MDDB - the procedure MDDB.

Here is the example of the procedure code for creating MDDB called mddbv1:

```
proc mddb
   data= dirdat.prebase8
   out= dirdat.mddbv1
   label= "MDDB - Deposits";
   class DAY CURRENCY MBRANCH CCENTER SYNT ANAL ;
   hierarchy MBRANCH CCENTER / name= "REGION" DISPLAY= YES;
   hierarchy SYNT ANAL / name= "DEPOSITS" DISPLAY= YES;
   hierarchy DAY / name= "DAY" DISPLAY= YES;
   hierarchy CURRENCY / name= "CURRENCY" DISPLAY= YES;
   var PREP ZUBE / SUM ;
run ;
```

The procedure code was submitted remotely and processed on a server. The MDDB was created and stored on the server according to the existing client/server architecture. The SAS/MDDB Server licence on the server enabled to create and to store MDDB’s permanently. In addition to this, in order to apply the client/server architecture while working with Motore applications built on MDDB, the SAS/EIS licence is required on the server.
After MDDB creation we performed the following steps:

1. we put the MDDB under existing Motore application instead of the Base table. According to the structure of the MDDB it was not needed to define summary tables as the MDDB contains not only NWAY crossing but also the required subtables as well.

2. we developed new SAS/EIS applications built on MDDB.

As we have already mentioned above, during the SAS/EIS application development the developer may use the following objects: Multidimensional Report, 3D Business Graphs, Organisational Chart, Map and Graphical Variance Report.

In comparison with the Motore applications built on base and summary tables, the number of the objects that are able to communicate with MDDB is not high. On the other hand, the developer obtains a possibility to customize the object or to create new objects.

Conclusion

Finally we would like to summarize knowledge and information that we have obtained in the Investment and Development Bank during the development of the SAS/EIS applications built on MDDB.

MDDB represents an effective way of storing the summarized data oriented to the improvement of data access. That leads to the accelerating the running of the application and decreasing of the user response.

Version 6.12 provides the Multidimensional Report, 3D Business Graphs, Organisational Chart, Map and Graphical Variance Report object to support MDDB and their interface has been modified according to the new needs. Using them the end-user can drill down in the selected dimension.

The communication between the objects has been improved in the way that it is not necessary to define a link between objects if they use the same MDDB. Interacting with one MDDB enabled EIS object (e.g. Multidimensional Report) will cause other object to reflect these changes without having to connect two objects explicitly.

During the running of the application after selection of the summarized data it is possible to see the detailed data behind a selected marker or cell. The data are retrieved from the Base table that was used to build the MDDB. The information about the connection to the appropriate Base table is stored in the metabase.

The developer has a chance to customize SAS/EIS MDDB enabled objects using the object-oriented application development environment of the SAS/AF Software and Screen Control Language (SCL). In that way it’s possible to change the behavior of the objects, to extend their existing functionality according to the special needs of the end-user.
References:

1. OLAP MDDB and Viewers, Course Notes, Application Development with The SAS® System, Release 6.12, SAS Institute Inc.


Address for correspondence:
Investičná a rozvojová banka, a.s.
Štúrova 5
813 54 Bratislava
Slovak Republic
Phone: +421 7 389 5229  Fax: +421 7 323 048
E-mail: ficova@irb.sk

1 The registered name of the IRB bank is: Investičná a rozvojová banka, a.s., Slovak Republic

2 SAS, SAS/AF, SAS/EIS, SAS/EIS Motore are registered trademarks of SAS Institute Inc., Cary, NC, USA