



HAL
open science

OLAP Analysis Operators for Multi-state Data Warehouses

Franck Ravat, Jiefu Song, Olivier Teste

► **To cite this version:**

Franck Ravat, Jiefu Song, Olivier Teste. OLAP Analysis Operators for Multi-state Data Warehouses. International Journal of Data Warehousing and Mining (IJDWM), 2016, 12 (4), pp.20-53. 10.4018/IJDWM.2016100102 . hal-04303256

HAL Id: hal-04303256

<https://hal.science/hal-04303256v1>

Submitted on 23 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 16959

To link to this article : DOI : 10.4018/IJDWM.2016100102
URL : <http://dx.doi.org/10.4018/IJDWM.2016100102>

To cite this version : Ravat, Franck and Song, Jiefu and Teste, Olivier
OLAP Analysis Operators for Multi-state Data Warehouses. (2016)
International Journal of Data Warehousing and Mining, vol. 12 (n° 4).
pp. 54-74. ISSN 1548-3924

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

OLAP Analysis Operators for Multi-states Data Warehouse

Franck Ravat^[1], Jiefu Song^[1], Olivier Teste^[2]

^[1] IRIT - Université Toulouse I Capitole, 2 Rue du Doyen Gabriel Marty
F-31042 Toulouse Cedex 09, France

^[2] IRIT - Université Toulouse II Jean Jaurès, 1 Place Georges Brassens
F-31703 Blagnac Cedex, France

^{[1] [2]} {ravat|song|teste}@irit.fr

ABSTRACT

Data reduction in *Multidimensional Data Warehouse* (MDW) allows increasing the efficiency of analysis and facilitating decision-makers' tasks. In this paper, we model a MDW containing reduced data through a set of states. Each state is valid for a certain period of time; it contains only useful information according to decision-makers' needs. In order to carry out analyses in a MDW composed of states, an extension of traditional OLAP analysis operators is required. In this paper, we define a set of OLAP operators compatible with reduced MDW. For each operator, we propose a user-oriented definition along with an algorithmic translation. To show the feasibility and the efficiency of the proposed concepts, we implement the analysis operators in a R-OLAP framework.

Keywords: Data reduction, multidimensional design, OLAP analysis operators

INTRODUCTION

Multidimensional Data Warehouse (MDW) organizes data in a multidimensional way in order to support *On-Line Analytical Processing* (OLAP). More specifically, a MDW schema is based on facts (analysis subjects) and dimensions (analysis axes). The facts contain analysis indicators, while a dimensions includes analysis parameters organized in hierarchies from the minimal (most detailed) granularity to the maximal (most general) granularity. In a classical MDW, all data are permanently stored and new data is periodically added. The increasing volume of MDW makes the tasks of decision-makers more difficult since they may be lost during their analyses. On the other hand, information is usually timely sensitive; most of detailed information loses its value over time. Nevertheless, information at high granularity levels is more stable, and it can generally fulfill decision-makers' needs when analyses are carried out over older data (Skyt, Jensen, & Pedersen, 2008). For instance, an analyst may have interest in analyzing sale amounts by product's brand for the last five years. However, as lots of today's brands did not exist before, the brand granularity level may be useless for an older period. As a result, the analyst may have no more interest in analyzing sale amounts by brand over the last ten years but by a higher and more stable granularity level, such as product's category.

Facing large volumes of data among which a great amount of inadequate detailed data are found, our aim is to both increase the analysis efficiency and facilitate the analysts' tasks. To this end, we propose a solution for supporting OLAP analyses over only pertinent data as time goes

on. Firstly, we provide a conceptual MDW model based on data reduction to aggregate and then remove useless detailed data. Secondly, we propose a set of algebraic OLAP operators to support analyses over reduced data. The execution process of each operator is illustrated with the help of an algorithm. At last, we implement the reduced MDW model and the set of analysis operators in a R-OLAP framework. The framework aims at proving the feasibility of proposed concepts and evaluating the efficiency of carrying out analyses over reduced data.

This paper is composed as follows. Section 2 studies related works. Section 3 presents preliminary concepts of reduced MDW illustrated with a case study. Section 4 describes our modeling solution for OLAP analysis operators. Section 5 presents a multi-states analysis framework showing the feasibility of concepts. Section 6 provides the results of experimental assessments on the efficiency of proposed OLAP operators.

RELATED WORK

Model of reduced data

Reducing data allows both decreasing the quantity of irrelevant data in decision making and increasing future analysis quality (Udo & Afolabi, 2011). In the field of MDW, (Skyt et al., 2008) presents a technique for progressive data aggregation of a fact. This study specifies data aggregation criteria of a fact according to granularity levels of dimensions. As mentioned in (Iftikhar & Pedersen, 2011), this work is highly theoretical but it fails to provide us a concrete implementation strategy. In (Iftikhar & Pedersen, 2011), a gradual data aggregation solution based on conception, implementation and evaluation is provided. This solution is based on a table containing different temporal granularities: second, minute, hour, month and year. However, the work of (Iftikhar & Pedersen, 2011; Skyt et al., 2008) is limited to the reduction of a single fact. Moreover, no formal model for reduced multidimensional data has been proposed.

Other related work concerning the management of data evolution in MDW is also proposed. The objective of this work is different from the data reduction. In general, its objective is to manage changes in data sources. (Kimball, 1996) put forwards the concept of Slowly Changing Dimensions (SCDs). SCDs imply that data may change within a dimension, even though it occurs less frequently than in a fact. Three basic modeling solutions have been proposed for managing dimensional data changes, namely overwrite old data, create new record for each change and keep data changes as alternative values in MDW. (Matteo Golfarelli & Rizzi, 2009) point out that not only data but also MDW schema can change over time according to user's requirements. SCDs, however, do not provide solution for handling schema changes.

Two approaches allowing managing both data and schema changes have been proposed, namely schema evolution, and schema versioning (Matteo Golfarelli & Rizzi, 2009). The schema evolution approach supports only the lasted MDW version, while the schema versioning approach allows keeping different MDW versions by stamping each MDW version with a validation period.

In the schema versioning approach, (Eder, Koncilia, & Morzy, 2002) was the first to address questions on multiversion MDW in order to manage both schema and instance changes. A metamodel is proposed to stamp each element in OLAP cube with a validity interval, so that multiple MDW versions can be managed and queried through the metamodel. However, no discussion about how a new version can be derived from a previous one is made in this work. (Body, Miquel, Bédard, & Tchounikine, 2002) propose a conceptual model allowing building a

multiversion fact table. Several structural evolution operators are proposed in this work in order to manage the mapping between two MDW versions. (Wrembel & Bębel, 2007) distinguishes two types of MDW versions, namely real version and alternative version. The real version is determined by changed in application domain and user's requirements, while the alternative version is created to support business simulations and what-if analyses. 15 elementary schema change operations and 7 instance change operations are available to modify a MDW version (Wrembel, 2009). However, no conceptual model for multiversion MDW is proposed in (Wrembel, 2009; Wrembel & Bębel, 2007). (Ravat, Teste, & Zurfluh, 2006) propose a conceptual definition for multiversion MDW based on a constellation model. However, the implementation framework is not explicitly presented in this work.

Note that, our objective is to propose a solution for OLAP analyses over reduced data. The work on MDW data reduction only partially satisfies our objective since only reduction of the fact is involved. Therefore, our first aim is to propose a solution for handling the complete MDW schema changes after data reduction. By consequence, all of the dimensions as well as the fact should be susceptible to sustain reductions to different granularity levels. The information judged useless is aggregated and then deleted from MDW in order to provide only necessary data for analysis.

On the other hand, inspirations of solutions for complete schema changes can be found within the field of schema versioning. However, the proposed concepts are not directly applicable. Adaptation should be defined especially for the derivation between different schemas in order to calculate aggregated data. Moreover, the conceptual solution should be implemented in an analysis framework in order to show the feasibility of the proposed concepts.

Reduced data analysis

To the best of our knowledge, no solution for OLAP analysis is proposed in the field of data reduction in MDW. Only few querying operators (select, project, aggregate) are formally proposed in (Skyt et al., 2008).

Some mechanisms for querying a multiversion MDW can be found. (Morzy & Wrembel, 2004) propose an extension of SQL vocabulary. Decision-makers can implicitly query multiple versions by specifying a temporal interval or the involved versions in a SQL query. After the execution of a query, a decision-maker can receive several partial results; each partial result is annotated with information concerning the involved MDW version. If possible, several partial results are integrated into a consistent set (Mitrpanont & Fugkeaw, 2006).

However, the work about querying a multiversion MDW is only limited to a special usage of SQL. To support on-line interactive analyses, a decision-maker needs more complex multidimensional analysis operators. To the best of our knowledge, no work in the field of multiversion MDW has involved OLAP analysis operators. Moreover, analysis results are possible to be presented separately, which makes the interaction with data more complex.

Our objective is to define a complete solution for OLAP analyses over reduced data. Neither the work over data reduction nor the work of multiversion MDW proposes OLAP operators; the work in both fields is limited to either query language or formal querying operators.

Facing to these issues, we aim at proposing a modeling solution for OLAP analysis operators. In contrast with the related work, our proposed analysis operators should facilitate decision-makers' tasks by not requiring the different involved schemas. Moreover, after the execution of

analysis operator, only one global result should be presented to decision-makers in order to facilitate the interaction with displayed result.

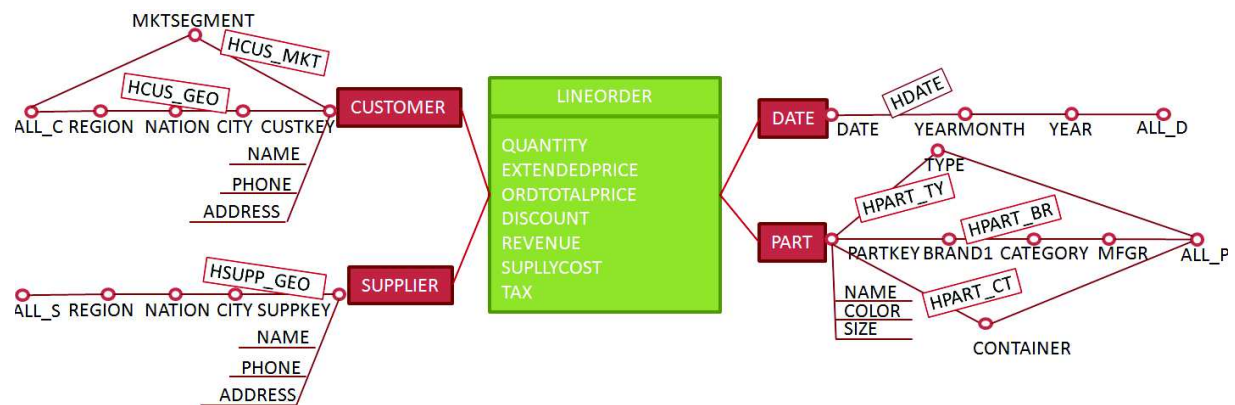
CONCEPTS OF REDUCED MDW

We firstly describe a case study of data reduction in MDW. This case study aims at giving a first glance at selective deletion of data in MDW as well as data model notations. After the case study, we define a formal presentation of MDW concepts adapted to the reduction context. At last we present the basic constraints that a valid reduced MDW should satisfy.

Case Study

Referring to one of the reference points in the field of OLAP, our case study is built upon the *Star Schema Benchmark* proposed in (O’Neil, O’Neil, Chen, & Revilak, 2009). The SSB is designed to measure the performance of OLAP data warehouse systems. Large volume of data can be generated by SSB following a classical star schema model. Our case study consists in an extract from SSB, which includes one fact named *LINEORDERS*. It contains seven measures, namely *QUANTITY*, *EXTENDEDPRICE*, *ORDTOTALPRICE*, *DISCOUNT*, *REVENUE*, *SUPLLYCOST* and *TAX*. Each *LINEORDER* is associated to four dimensions: the time of line order (*DATE*), the customer who takes the order (*CUSTOMER*), the product part (*PART*) and the corresponding supplier (*SUPPLIER*). Each dimension contains attributes organized according to their granularity level in one or several hierarchies. For instance, the dimension *CUSTOMER* contains two hierarchies: *HCUS_MKT* and *HCUS_GEO*. In the dimension *CUSTOMER*, analysis parameters are organized from the lowest granularity level *CUSTKEY* to the highest level *ALL_C* with two hierarchies. Based on the graphical notation introduced in (M. Golfarelli, Maio, & Rizzi, 1998), we propose a conceptual notation of the SSB MDW (cf. figure 1).

Figure 1. MDW schema before data reduction



After populating the dimensions and the fact, we obtain a MDW saturated with all detailed information over time. The important volume of detailed data slows down the analyses during a decision-making process. Moreover, most of the old detailed data become obsolete as time goes by, they are no more used in analysis process and should be deleted according to user’ needs. For illustrative purposes, we suppose a decision-maker expresses her/his needs for reducing data in the SSB MDW as follows:

- From 1997 to 1998, analyses are carried out with reference to the lowest levels of granularity (*CUSTKEY*, *SUPPKEY*, *DATE* and *PARTKEY*) (cf. figure 1).
- In the previous period from 1995 to 1997, except for the dimension *DATE*, all other dimensions are summarized to higher granularity levels (cf. figure 2). For instance, the dimension *CUSTOMER* keeps only *CITY*, *NATION*, *REGION* and *ALL_C* levels. Useless parameters are aggregated and then removed from the schema within this period.
- Before 1995, only aggregated information by customers' *NATION*, *YEARMONTH* and products' *CATEGORY* makes sense (cf. figure 3).

Figure 2. MDW schema valid from 1995 to 1997

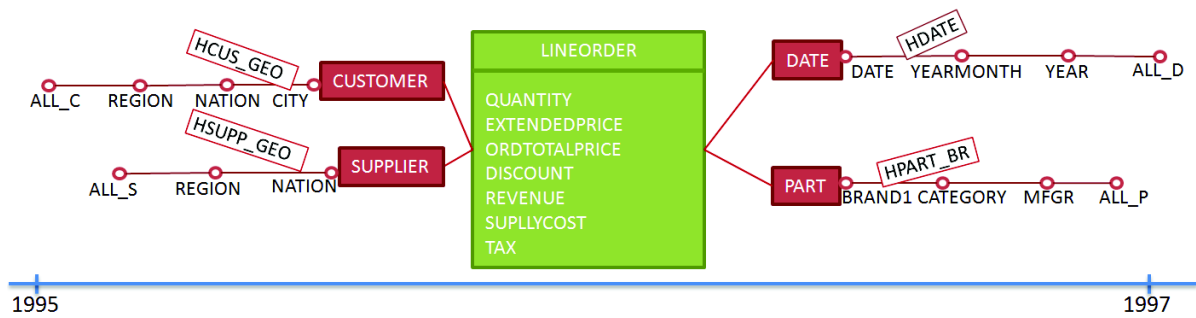
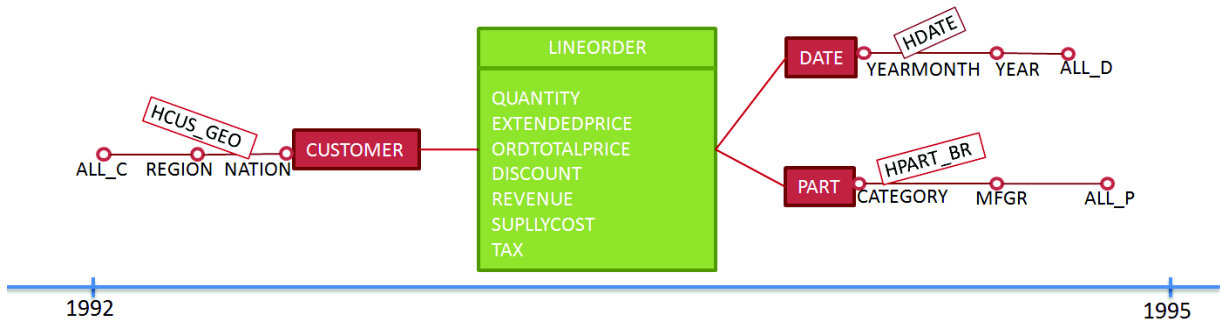


Figure 3. MDW schema valid from 1992 to 1995



Basic concepts of reduced MDW

Classical MDW is composed of one single schema which does not support selective deletion of useless information over time. In order to manage progressive schema changes, we model a reduced MDW with a set of star schemas called **states**. The current state corresponds to the present status of the MDW containing the most detailed information, while past states correspond to a succession of reduced states over time in which information is aggregated.

Let us define \mathcal{N} such as $\mathcal{N} = \{n^1, n^2, \dots\}$ is a finite set of names, $\mathcal{N} \neq \emptyset$; \mathcal{F} such as $\mathcal{F} = \{F_1, \dots, F_n\}$ is a finite set of facts of a reduced MDW, $\mathcal{F} \neq \emptyset$. Each fact F_m ($F_m \in \mathcal{F}$) is associated to one and only one state of a MDW, where the fact F_1 represents the fact in the first state of MDW while the fact F_n corresponds to the fact in the n-th state of a MDW; \mathcal{D} such as $\mathcal{D} = \{D_1^1, \dots, D_k^n\}$ is a finite set of dimensions, $\mathcal{D} \neq \emptyset$. Each dimension D_m^i ($D_m^i \in \mathcal{D}$) is only valid in a unique state, it corresponds to the m-th dimension in the states i ;

Definition 1. A MDW is defined by $S = (n^S ; \mathcal{E} ; Map)$, where :

- $n^S \in \mathcal{N}$ is the name of the MDW;
- $\mathcal{E} = \{E_1 ; \dots ; E_n\}$ is a set of states composing the MDW, $\mathcal{E} \neq \emptyset$; The state denoted E_1 corresponds to the current state of MDW;
- Reduction function $Map \{ \mathcal{E} \rightarrow \mathcal{E} \mid Map(E_k) = E_{k+1} \}$ defines the state named E_{k+1} obtained by the reduction of E_k .

Definition 2. Each state $E_i = (F_i ; \mathcal{D}_i ; T_i)$ is a star schema defined for a validation period, where

- $F_i \in \mathcal{F}$ is the fact of the state;
- $\mathcal{D}_i = \{D_1^i ; \dots ; D_m^i\} \subseteq \mathcal{D}$ is a set of dimensions associated to the fact, $|\mathcal{D}_i| \geq 1$. It contains at least one temporal dimension;
- $T_i = [t_{start}^i ; t_{end}^i [$ is a validation period associated to the state E_i , $T_i \neq \emptyset \wedge t_{end}^i > t_{start}^i$. The validation period is semi-open, i.e. the starting time t_{start}^i is included in the interval while the end time t_{end}^i is excluded.

The union of validation periods of all states corresponds to the validation period of the MDW, denoted \mathcal{T}_{TIMES} , such as $\mathcal{T}_{TIMES} = T_1 \cup \dots \cup T_n$. To define the T_i validation period, we adopt a linear and discrete time model approaching time in granular way through time observation units (Wang, Bettini, Brodsky, & Jajodia, 1997). A validation period T_i of state E_i is defined by a couple of instants. These instants can be fixed (temporal grains such as the year of 1992) or dynamic (defined with the instant “ t_{now} ”).

Definition 3. A *fact* denoted F_i is defined by $(n^{F_i} , \mathcal{M}^{F_i})$, where

- $n^{F_i} \in \mathcal{N}$ is the fact name;
- $\mathcal{M}^{F_i} = \{m_1^{F_i} ; \dots ; m_k^{F_i}\}$ is a set of *measures* (also called *indicators*), $\mathcal{M}^{F_i} \neq \emptyset$.

Definition 4. A *dimension* denoted D_k^i is defined by $(n^{D_k^i} , \mathcal{A}^{D_k^i} , \mathcal{H}^{D_k^i})$, where

- $n^{D_k^i} \in \mathcal{N}$ is the dimension name;
- $\mathcal{A}^{D_k^i} = \{a_1^{D_k^i} ; \dots ; a_x^{D_k^i}\}$ is the set of the *attributes* of the *dimension*, $\mathcal{A}^{D_k^i} \neq \emptyset$;
- $\mathcal{H}^{D_k^i} = \{h_1^{D_k^i} ; \dots ; h_m^{D_k^i}\}$ is a set of *hierarchies*, $\mathcal{H}^{D_k^i} \neq \emptyset$.

Definition 5. A *hierarchy*, denoted H_j (abusive notation of $\mathbf{h}_j^{D_k^i}$) is defined by $(n^{H_j} , \mathcal{P}^{H_j} , \prec^{H_j} , \text{Weak}^{\mathcal{H}^{H_j}})$, where

- $n^{H_j} \in \mathcal{N}$ is the hierarchy name;
- $\mathcal{P}^{H_j} = \{p_1^{H_j} ; \dots ; p_k^{H_j}\}$ is a set of attributes called *parameters*, $\mathcal{P}^{H_j} \subseteq \mathcal{A}^{D_k^i} \wedge \mathcal{P}^{H_j} \neq \emptyset$. Each parameter $p_x^{H_j}$ defines a granularity level of a hierarchy;
- \prec^{H_j} is an antisymmetric and transitive binary relation between parameters;

- $Weak_{\mathcal{K}}^{H_j} \subseteq \mathcal{A}^{D_k^i}$ is an application that associates to each parameter a set of dimension attributes, called *weak attributes*, $Weak_{\mathcal{K}}^{H_j} \cap \mathcal{P}^{H_j} = \emptyset$. A weak attribute allows enriching the semantic of the associated parameter, it does not identify any granularity level.

Hierarchies organize the attributes of a dimension, from the finest graduation (root parameter denoted $ID_{D_k^i}$) to the most general graduation (extremity parameter denoted $ALL_{D_k^i}$). Thus, a hierarchy defines the valid navigation paths on a dimension.

Example 1. The E_3 state in the figure 3 is composed of one fact and three dimensions, it is valid from 1992 to 1995. The fact named *LINEORDER* contains seven measures. The first dimension *CUSTOMER* contains a hierarchy named *HCUS_GEO* on which the parameters are organized according to their granularity level: from the lowest level named *NATION* to the highest level *ALL_C*. The second dimension named *PART* contains three parameters *CATEGORY*, *MFGR* and *ALL_P* within one hierarchy *HPART_BR*. The third dimension is a temporal dimension, named *DATE*. This dimension is graduated by the parameter *YEARMONTH*, *YEAR* and *ALL_D* on the hierarchy *HDATE*.

The formal representation of state E_3 is as follows:

$E_3 = (F_3 ; \{ D_1^3 ; D_2^3 ; D_3^3 \} ; [t_{1992} ; t_{1995}[)$ where

- $F_3 = (LINEORDER ; \{ QUANTITY ; EXTENDEDPRICE ; ORDTOTALPRICE ; DISCOUNT ; REVENUE ; SUPPLYCOST ; TAX \})$
- $D_1^3 = (CUSTOMER ; \{ NATION ; REGION ; ALL_C \} ; \{ h_1^{D_1^3} \})$.
- $D_2^3 = (PART ; \{ CATEGORY ; MFGR ; ALL_P \} ; \{ h_1^{D_2^3} \})$.
- $D_3^3 = (DATE ; \{ YEARMONTH ; YEAR ; ALL_D \} ; \{ h_1^{D_3^3} \})$.

We take the hierarchy H_1 on the temporal dimension *DATE* (abusive notation of $h_1^{D_3^3}$) as an example to illustrate the abstract representation for a hierarchy.

$H_1 = (n^{H_1}, P^{H_1}, \prec^{H_1}, Weak^{H_1})$ where

- $n^{H_1} = HDATE$;
- $P^{H_1} = \{ YEARMONTH, YEAR, ALL_D \}$;
- $\prec^{H_1} = \{ (YEARMONTH, YEAR) ; (YEAR, ALL_D) \}$;
- $Weak_{H_1} = \emptyset$.

Derived concepts of reduced MDW

The basic concepts presented in the previous section allow defining the elements appearing in each state of a reduced data model. In order to support analyses over reduced data, it is important to identify the elements present only in certain states but not in others. In this section, we present three derived concepts defining missing elements in reduced MDW.

The first derived concepts is called *missing attribute*. A *missing attribute* refers to an attribute which is present in the first state (i.e. states E_1) and absent in at least one older state. We associate the *missing attributes* with certain states in order to indicate the unavailable period.

Let us define $\mathcal{A}^{\mathcal{D}_k}$ is the set of attributes on all the dimensions \mathcal{D}_k in the state E_k , $\mathcal{A}^{\mathcal{D}_k} \neq \emptyset$; $\mathcal{E}' \subseteq \mathcal{E}$, is a non-empty subset of states, $\mathcal{E}' = \{E_m; \dots; E_n\}$, $n \geq m > 1$.

Definition 6: The set of *Missing Attributes* for a subset of states \mathcal{E}' is denoted $MA^{\mathcal{E}'}$, $MA^{\mathcal{E}'} = \mathcal{A}^{\mathcal{D}_1} \setminus \{\mathcal{A}^{\mathcal{D}_m}; \dots; \mathcal{A}^{\mathcal{D}_n}\}$, which is the difference between the set of attributes in the first state E_1 and those in the subset of states \mathcal{E}' such as $\mathcal{E}' \subset \mathcal{E} \wedge E_1 \notin \mathcal{E}'$.

Example 2. After removing the *CUSTKEY* in the hierarchy *HCUS_GEO* of the dimension *CUSTOMER* before 1997, *CUSTKEY* becomes a *missing attribute* in the states E_2 and E_3 , denoted $CUSTKEY \in MA^{\mathcal{E}'}$, $\mathcal{E}' = \{E_2; E_3\}$. In the same way, after deleting the parameter named *CITY* for the period from 1992 to 1995, *CITY* becomes a *missing attribute* in the state E_3 , denoted $CITY \in MA^{\mathcal{E}'}$, $\mathcal{E}' = \{E_3\}$. By consequence, the unaffected parameters *NATION* and *REGION* are accessible throughout the seven year (from 1992 to 1998); the parameter named *CITY* can be found within states during the last four years (from 1995 to 1998); the parameter *CUSTKEY* is only available from 1997 to 1998.

The second derived concept refers to *missing hierarchy*. A *hierarchy* becomes *missing* in a subset of states \mathcal{E}' ($\mathcal{E}' \subset \mathcal{E}$), when all the parameters in this hierarchy are *missing attributes*.

Definition 7: The set of *Missing Hierarchies* in a subset of states \mathcal{E}' is denoted $MH^{\mathcal{E}'}$, where $\forall H_k \in MH^{\mathcal{E}'}, \forall P_k \in \mathcal{P}^{H_k} | P_k \in MA^{\mathcal{E}'}$.

The last derived concept is *missing dimension*, which refers to a *dimension* where all its hierarchies are of type *missing hierarchy* in a subset of states \mathcal{E}' ($\mathcal{E}' \subset \mathcal{E}$).

Definition 8: The set of *Missing Dimensions* in a subset of states \mathcal{E}' is denoted $MD^{\mathcal{E}'}$, where $\forall D_k \in MD^{\mathcal{E}'}, \forall H_j \in \mathcal{H}^{D_k} | H_j \in MH^{\mathcal{E}'}$.

Constraints

In order to ensure the coherence of multi-state model, we propose a set of integrity constraints. As announced in the papers (Demuth & Hussmann, 1999; Elmasri, 2011), the common integrity constraints fall into three types:

- Explicit constraints, also called business rules, are defined by the designer, such as attribute domain in the relational model and constraints between hierarchy components in the multidimensional model (Ghozzi, Ravat, Teste, & Zurfluh, 2003). This type of constraints falls out of the scope of this paper.
- Implicit constraints represent integrity rules which are part of the data model. They are specified on individual static components, such as primary key and foreign key in the relational model. This type of constraints can be found within the definitions presented in the section *Basic concepts of reduced MDW*.
- Inherent constraints are not specified in a schema; they are supported by the definition of the corresponding data model. This type of constraints are presented in this section.

Constraints over states.

- A reduced state must contain at most the same number of dimensions than the previous one: \forall two states E_i and E_j with $j > i \rightarrow |\mathcal{D}_i| \geq |\mathcal{D}_j| \wedge (\forall D_x^j \in \mathcal{D}_j \rightarrow D_x^i \in \mathcal{D}_i)$.
- Validation periods of the states must be contiguous: \forall a state E_i defined with $(F_i; \mathcal{D}_i; T_i)$ and a state E_{i+1} reduced from $E_i \rightarrow t_{end}^{i+1} = t_{start}^i$

Constraints over dimension and hierarchy.

- A reduced dimension should contain at most the same attributes as the original dimension: $\forall a_x^{D_k^{i+1}}$ attribute of the dimension D_k^{i+1} obtained by the reduction of the dimension $D_k^i \rightarrow |\mathcal{A}^{D_k^i}| \geq |\mathcal{A}^{D_k^{i+1}}| \wedge (\forall a_x^{D_k^{i+1}} \in \mathcal{A}^{D_k^{i+1}} \rightarrow a_x^{D_k^i} \in \mathcal{A}^{D_k^i})$
- A reduced dimension should contain at most the same hierarchies as the original dimension: $\forall h_x^{D_k^{i+1}}$ of the dimension D_k^{i+1} obtained by the reduction of the dimension $D_k^i \rightarrow |\mathcal{H}^{D_k^i}| \geq |\mathcal{H}^{D_k^{i+1}}| \wedge (\forall h_x^{D_k^{i+1}} \in \mathcal{H}^{D_k^{i+1}} \rightarrow h_x^{D_k^i} \in \mathcal{H}^{D_k^i})$
- A reduced hierarchy contains at most the same parameters as the original hierarchy, at least two parameters ($ALL_{D_k^i}$ and another one): \forall a hierarchy $h_x^{D_k^{i+1}}$ obtained by the reduction of the hierarchy $h_x^{D_k^i}, \Rightarrow \left| Ph_x^{D_k^i} \right| \geq \left| Ph_x^{D_k^{i+1}} \right| \geq 2$

MULTI-STATES ANALYSIS OPERATORS

A reduced MDW facilitates decision-makers' tasks by keeping only useful data over time. In order to carry out analyses over reduced data, a decision-maker needs OLAP operators applicable to MDW composed of multiple states.

In the scientific literature, we can find some classical analysis operators allowing exploiting OLAP cubes. (Ciferri et al., 2013; Ravat, Teste, Tournier, & Zurfluh, 2008) propose a set of user-oriented OLAP operators supporting displaying, drilling, rotating and selecting operations. Even though these operators sustain flexible and adaptable to different classical modeling solutions (star and constellation models), they become incompatible with reduced MDW due to the following reasons:

- In a reduced data model, a set of schemas (states) is defined for MDW over time. As classical algebraic analysis operators are conceived to manipulate one unique schema, they cannot be directly applied to reduced MDW.
- Regarding to reduced data instances, a reduced MDW does not contain all detailed information in older states. Unfortunately, classical analysis operators do not provide solutions for handling the selective deletion of detailed information in different states.

To carry out analyses in reduced MDW, decision-makers need extensions of classical OLAP operators that can deal with both schema and instance changes over time. Facing to this issue, we extend five commonly-used OLAP operators in order to support basic multi-states analysis in reduced MDW. The first one is denoted $Display^{multi-states}$, it allows building the first analysis

results; the second and third operators are denoted Drilldown^{multi-states} and Rollup^{multi-states}, they allow increasing and decreasing analysis granularity respectively; the fourth operator is Rotate^{multi-states} operator, it allows changing the content of a currently displayed analysis axis; the last operator, denoted Select^{multi-states}, allows adding restriction predicates to analysis results.

In this section, we firstly introduce the principles that the multi-states analysis operators follow. Secondly we propose an extension of classical representation solution for analysis results. At last we present the five multi-states analysis operators as well as their corresponding execution algorithms. The operators and algorithms are represented in algebraic and technology-neutral form, so that they can be easily implemented in different environments (e.g. R-OLAP, M-OLAP and H-OLAP).

Characteristics of multi-states analysis operators

User-oriented algebraic operators

Each analysis operator is user-oriented, it defines an elementary analysis operation in the point of view of a decision-maker. To facilitate decision-makers' tasks, the multi-states analysis operators manipulate conceptual concepts such as *fact* and *dimensions* in the context of reduced MDW. Decision-makers just need to specify "what-to-do" without worrying about "how-to-do". Moreover, decision-makers do not necessarily need to explicitly specify the involved states. They simply choose a validation period, the multi-states analysis operators automatically search the involved states.

Bi-directional adaptation

Analysis cannot be carried out with missing elements (i.e. *missing attributes*, *missing hierarchies* and *missing dimensions*). Adaptations should be taken if missing elements are involved during an analysis. We distinguish two different adaptation approaches, namely *Breadth-first* and *Depth-first*.

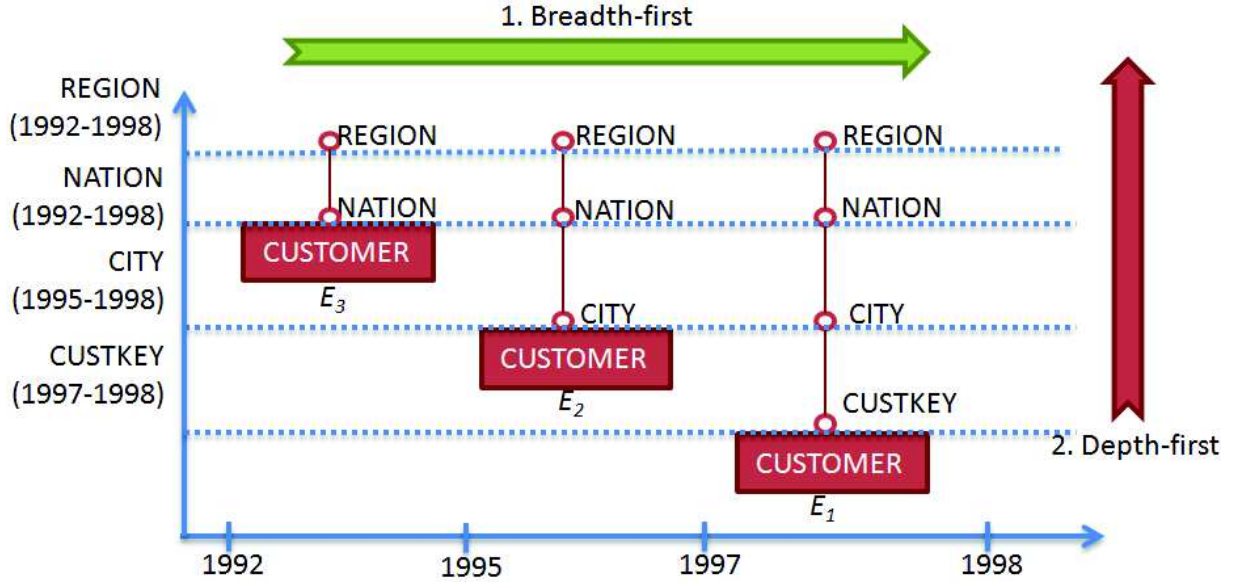
- The *Breadth-first* approach gives priority to the span of temporal intervals involved in analysis. More specifically, it consists in (a) keeping as much as possible the span of involved temporal intervals and (b) increasing the granularity level until we find the first common parameter among all involved states.
- The *Depth-first* approach attaches importance to the granularity levels of analysis. It allows (a) maintaining the chosen granularity levels at the price of (b) reducing the temporal intervals to find the subset of states that contains all involved parameters.

Example 3. If the parameter named *CITY* (available from 1995 to 1998) is involved in an analysis during the period from 1992 to 1998, adaptations can be taken as follows (cf. figure 4):

- According to the *Breadth-first* approach, the temporal interval $[t_{1992}, t_{1998}[$ of analysis remains unchanged. We look for the first common parameter among all involved states. In this case, the parameter *NATION* is the first common parameter in the states E_1 , E_2 and E_3 . Therefore the granularity level of analysis increases to *NATION* level in order to keep the same breadth of temporal interval.
- According to the *Depth-first* approach, we adjust the span of temporal interval while keeping the chosen parameter named *CITY* unchanged. In this case, we reduce the

temporal interval to $[t_{1995}, t_{1998}[$ which is the validation period of the parameter *CITY*. Finally the analysis is carried out by *CITY* with the adjusted temporal interval from 1995 to 1998.

Figure 4. Breadth-first and depth-first approaches in the hierarchy *HCUS_GEO*



Analysis results

As n-dimensional cube is difficult to be exploited by decision-makers (Gyssens & Lakshmanan, 1997; Maniatis et al., 2005; Ravat et al., 2008), we propose to model analysis results in tabular form, called *Multidimensional Table*. In the context of reduced MDW, analysis result presented in *Multidimensional Table* may come from one or several states of MDW. Since the results of each analysis are only valid for a certain period of time (i.e. the union of validation periods of all involved states), we extend the classical *Multidimensional Table* by adding a validation period to analysis results.

Definition 9. A *Multidimensional Table*, denoted MT_i , is defined by $(S^i; Ax^i; R^i; I^i)$, where:

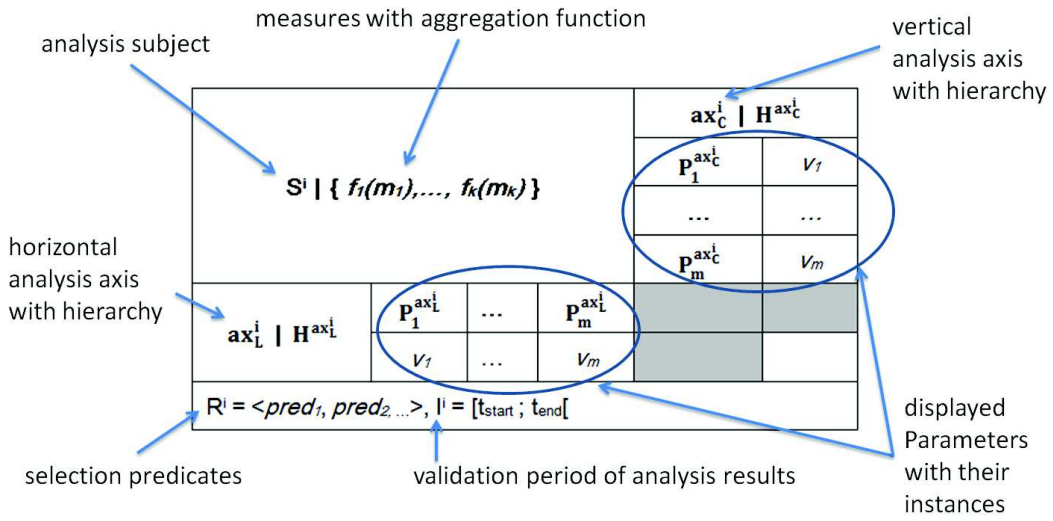
- $S^i \in \mathcal{F}$ is the displayed fact (analysis subject);
- $Ax^i = \{ax_L^i, ax_C^i\}$ is the set of analysis axes currently presented, among which ax_L^i and ax_C^i are the horizontal and vertical displayed axes respectively. Each analysis axis $ax^i \in Ax^i$ is composed of a set of dimensions D^{ax^i} , hierarchies H^{ax^i} and parameters P^{ax^i} , where $ax^i = (D^{ax^i}, H^{ax^i}, P^{ax^i})$ such as $D^{ax^i} \in \mathcal{D}$, $H^{ax^i} \in \mathcal{H}^{D^{ax^i}}$ and $P^{ax^i} = \langle ALL_H^{ax^i}, p_n^{H^{ax^i}}, \dots, p_1^{H^{ax^i}} \rangle$.
- $R^i = \langle pred_1, pred_2, \dots \rangle$ is a set of selection predicates on non-temporal dimensions and/or fact.

- $I^i \subseteq \mathcal{T}_{TIMES}$ is a set of selection predicates on the temporal dimension, it represents the validation period of MT_i .

Remark. Besides eliminating some displayed instances, the temporal criteria also represents the validation period of a MT. For this reason, the selection predicates on the temporal dimension are not included in R^i but managed separately by I^i .

Analysis results may come from several states. Partial results from each state are integrated into one MT before being presented to decision-makers. No more information about state of MDW is shown in MT (cf. figure 5).

Figure 5. Structure of a Multidimensional Table



It is important to note that the bi-dimensional representation of MT does not implicate the analyses should be limited to two dimensions. If the set of displayed parameters (i.e. p^{ax^i}) includes parameters coming from several dimensions, the corresponding axis in MT (i.e. ax^i) may correspond to multiple dimensions in MDW schema.

Building the first multidimensional table

The first MT is obtained by using the $Display^{multi-state}$ operator. This operator define the structure of the MT and display the highest granularity levels of the chosen hierarchies of dimension.

Table 1. Algebraic multi-states display operator

$Display^{multi-states} ([I]; n_{display}; S_{display}; M_{display}; D_L; H_L; D_C; H_C; [Adaptation]) = MT_0 ;$	
Input	<ul style="list-style-type: none"> • $I \subseteq \mathcal{T}_{TIMES}$: a set of optional temporal intervals. • $n_{display} \in \mathcal{N}$: the name of MDW, $n_{display} \neq \emptyset$ • $S_{display} \in \mathcal{F}$: the fact (analysis subject) of MDW to be displayed. • $M_{display} = \{f_1(m_1), \dots, f_k(m_k)\}$, $M_{display} \geq 1$: a non-empty set of indicators to be displayed along with their associated aggregation functions

	<ul style="list-style-type: none"> • $D_L \in \mathcal{D}, D_C \in \mathcal{D}$: the dimensions to be displayed in the horizontal and vertical axis respectively. • $H_L \in \mathcal{H}^{D_L}, H_C \in \mathcal{H}^{D_C}$: the hierarchies to be displayed in the horizontal and vertical axes respectively. • <i>Adaptation</i> = {<i>BREADTH, DEPTH</i>}: an optional parameter to specify the adaptation method. If no method is specified, the analysis operator takes <i>breadth-first</i> adaptation method by default.
Output	<p>$MT_0 = (S^0; Ax^0; R^0; I^0)$ analysis results MT such as</p> <ul style="list-style-type: none"> • Analysis subject $S^0 = S_{display}$ • Analysis axes $Ax^0, Ax^0 = \{ax_L^0, ax_C^0\}$ such as <ul style="list-style-type: none"> ○ $ax_L^0 = (D_L^0, H_L^0, P_L^0)$ where <ul style="list-style-type: none"> ▪ $D_L^0 = D_L \vee D_L^0 = \emptyset,$ ▪ $H_L^0 = H_L \vee H_L^0 = \emptyset,$ ▪ $P_L^0 = \langle ALL^{D_L} \rangle \vee P_L^0 = \emptyset.$ ○ $ax_C^0 = (D_C^0, H_C^0, P_C^0)$ where <ul style="list-style-type: none"> ▪ $D_C^0 = D_C \vee D_C^0 = \emptyset,$ ▪ $H_C^0 = H_C \vee H_C^0 = \emptyset,$ ▪ $P_C^0 = \langle ALL^{D_C} \rangle \vee P_C^0 = \emptyset.$ • Selection predicates on dimensions and/or fact $R^0 = \emptyset$ • Validation period $I^0 = I \vee I^0 = T_1 \cup \dots \cup T_n \vee I^0 = I_\lambda^*$

*Adjusted temporal interval $I_\lambda \subseteq I \wedge I_\lambda = T_m \cup \dots \cup T_k (1 \leq m < k < n)$

By using the optional input parameter I (temporal interval of analysis specified by decision-makers), the $Display^{multi-states}$ searches only relevant states involved in the analysis. In the absence of the parameter I , the $Display^{multi-states}$ operator is applied to all the states of MDW before constructing the first *multidimensional table* MT_0 . By default, after the execution of $Display^{multi-states}$, the displayed parameters in horizontal or vertical axes are the extreme parameter ALL_D in the hierarchy H_j^D of the dimension D .

However, in some cases, not all parameters provided by decision-makers can be satisfied. The $Display^{multi-states}$ has to adapt the input parameters according to the chosen adaptation method. Since a multidimensional table can only contain information available in all involved states, in certain cases only one analysis axis can be displayed (i.e. $D_L^0 = \emptyset$ or $D_C^0 = \emptyset$). In the case where none of the chosen analysis axes can be displayed, we disable the execution of $Display^{multi-states}$ operator. To be more specific, we propose the following execution algorithm for the $Display^{multi-states}$ operator to deal with all possible circumstances. Note that the subset of states E' in the algorithms is defined as $E' = \{E_x ; \dots ; E_y\}, n \geq y > x \geq 1$.

Algorithm 1: $\text{Display}^{\text{multi-states}} ([I]; n_{\text{display}}; S_{\text{display}}; M_{\text{display}}; D_L; H_L; D_C; H_C; [Adaptation]) = MT_0$;

Input: Set of temporal intervals I ; MDW's name n_{display} ; analysis subject to be displayed S_{display} ; set of indicators with aggregated function to be displayed M_{display} ; horizontal and vertical analysis axes D_L and D_C ; horizontal and vertical analysis hierarchies H_L and H_C ; specified adaptation method $Adaptation$.

Output: the first multidimensional table MT_0

Declare Procedure $AdaptationInputsVariables(States; Dim, Hier, [Adaptation]);$

Input : A set of states $States$; a dimension Dim ; a hierarchy $Hier$,
an optional adaptation method $Adaptation$

Begin procedure

1. If $Dim \in MD^{States}$ then
2. If $Adaptation = DEPTH$ then
3. Find the maximal subset of states $States' \subseteq States \mid Dim \notin MD^{States'}$
4. $States = States'$
5. Else
6. $Dim = \emptyset, Hier = \emptyset$
7. End if
8. Else
9. If $Dim \neq \emptyset \wedge Hier \in MH^{States}$
10. If $Adaptation = DEPTH$ then
11. Find the maximal subset of states $States' \subseteq States \mid Hier \notin MH^{States'}$
12. $States = States'$
13. Else
14. Find a hierarchy $H_k^{Dim} \in \mathcal{H}^{Dim} \wedge H_k^{Dim} \notin MH^{States}$
15. $Hier = H_k^{Dim}$
16. End if
17. End if
18. End if

End procedure

Begin

1. If $I \neq \emptyset$ then,
2. Find the maximal subset of states $E' \subseteq \mathcal{E}, \forall E_j \in E' \mid T_j \cap I \neq \emptyset$
3. Else
4. $E' = \mathcal{E}$
5. End if
6. Call $AdaptationInputsVariables(E'; D_L, H_L, Adaptation);$
7. Call $AdaptationInputsVariables(E'; D_C, H_C, Adaptation);$
8. If $D_L = \emptyset \wedge D_C = \emptyset$
9. Impossible operation, display a warning message.
10. Else
11. For $E_j \in E'$
12. Translate $\text{Display}(n_{\text{display}}; S_{\text{display}}; M_{\text{display}}; D_L; H_L; D_C; H_C)$ into query Q_j
13. Global query $Q = Q \cup Q_j$
14. $I_{TM} = I_{TM} \cup (T_k \cap I)$
15. End for

16. $MT_0 = \text{Result of global query } Q \text{ with } I_0 = I_{TM}, R_0 = \emptyset$

17. End if

End

Example 4. A decision-maker starts the analysis by displaying the total *REVENUE* according to *HCUS_GEO* hierarchy on the dimension *CUSTOMER* and *HPART_BR* hierarchy on the dimension *PART* from 1995 to 1998. The execution algorithm of $\text{Display}^{\text{multi-states}}([t_{1995}, t_{1998}[; \text{SSB}, \text{LINEORDER}, \text{SUM}(\text{REVENUE}), \text{CUSTOMER}, \text{HCUS_GEO}, \text{PART}, \text{HPART_BR})$ is processed as follows:

- As $I = [t_{1995}, t_{1998}[$ is not an empty set, the line 2 is executed to find the subset of states $E' = \{E_1, E_2\}$ involved in the analysis.
- Since $\text{CUSTOMER}, \text{PART} \notin \text{MD}^{E'}$ and $\text{HCUS_GEO}, \text{HPART_BR} \notin \text{MH}^{E'}$, the procedure named *AdaptationInputsVariables* has not adapted the analysis axes and hierarchies.
- For E_1 and E_2 , one Display for each state is generated and then translated into a partial query. Each partial query is executed separately.
- The first multidimensional table MT_0 is built based on the union of each partial query's results. Its restriction predicate is an empty, while the validation period of MT_0 is from 1995 to 1998 (cf. figure 6).

Figure 6. MT_0 after the execution of $\text{Display}^{\text{multi-states}}$ operator.

LINEORDER SUM(REVENUE)		PART HPART_BR	
		ALL_P	ALL
CUSTOMER HCUS_GEO	ALL_C		
	ALL		47457301783232
R = \emptyset , I = [1995, 1998[

As all chosen dimensions and hierarchies are available from 1995 to 1998 (states E_1 and E_2), the $\text{Display}^{\text{multi-states}}$ is executed without adaptation of input parameters. The returned results are the sum of revenues of all the line-orders in the SSB MDW. The decision-maker needs more multi-states analysis operators in order to, for instance, navigate at different granularity levels.

Changing analysis granularity

We extend two operators which allow changing the granularity levels of analysis, namely $\text{Drilldown}^{\text{multi-states}}$ and $\text{Rollup}^{\text{multi-states}}$. They both take several variables as inputs: a set of optional *temporal intervals*, a currently displayed *multidimensional table*, an analysis *dimension*, a *parameter* and an *adaptation method*. During execution, both operators exploit the navigation paths defined by the currently displayed hierarchy. As output, a new *multidimensional table* is produced containing information at a different granularity level.

$\text{Drilldown}^{\text{multi-states}}$ operator

The $\text{DrillDown}^{\text{multi-states}}$ operator allows displaying information at a finer granularity level on currently displayed dimension in several states.

Let $ax^K \in Ax^K \mid ax^K = (D^{ax^K}, H^{ax^K}, P^{ax^K})$ be the analysis axis affected by the $\text{Drilldown}^{\text{multi-states}}$ in the currently displayed multidimensional table MT_k

Table 2. Algebraic multi-states drilldown operator

DrillDown ^{multi-states} ($[I]$; MT_k , D_i ; P_{inf} ; [<i>Adaptation</i>]) = MT_{K+1} ;	
Input	<ul style="list-style-type: none"> • $I \subseteq \mathcal{T}_{TIMES}$: a set of optional temporal intervals • MT_k: the multidimensional table currently displayed • $D_i = D^{ax^K}$: one of the analysis axes currently displayed <p>$P_{inf} \in \mathcal{A}^{D_i}$: the chosen parameter or weak attribute on dimension D_i, $ID_{D_i} \prec^{H^{ax^K}} P_{inf}$ and $\forall P_m \in P^{ax^K}, P_{inf} \prec^{H^{ax^K}} P_m$</p> <ul style="list-style-type: none"> • <i>Adaptation</i> = {<i>BREADTH</i>, <i>DEPTH</i>}: an optional parameter to specify the adaptation method. If no method is specified, the analysis operator takes <i>breadth-first</i> adaptation method by default.
Output	<p>$MT_{K+1} = (S^{K+1}; Ax^{K+1}; R^{K+1}; I^{k+1})$ analysis results MT such as</p> <ul style="list-style-type: none"> • Analysis subject $S^{K+1} = S^K$ • Analysis axes $Ax^{K+1}, \forall ax_j^{K+1} \in Ax^{K+1} ax_j^{K+1} = (D_j^{K+1}, H_j^{K+1}, P_j^{K+1})$ such as <ul style="list-style-type: none"> ○ Dimension $D_j^{K+1} = D_j^K$, ○ Hierarchy $H_j^{K+1} = H_j^K$, ○ Parameter $P_j^{K+1} \in \mathcal{P}^{H_j^{K+1}} (D_j^{K+1} = D_i \rightarrow P_j^{K+1} = \langle P_{inf} \rangle \vee \langle P_{\lambda}^{**} \rangle) \wedge (D_j^{K+1} \neq D_i \rightarrow P_j^{K+1} = \langle P_1^{D_j^K}, P_2^{D_j^K}, \dots \rangle)$ • Selection predicates on dimensions and/or fact $R^{K+1} = R^K$ • Validation period $I^{k+1} = I \vee I^{k+1} = I_k \vee I^{k+1} = I_{\lambda}^{**}$

* Adjusted parameter $P_{\lambda} \in \mathcal{P}^{H_j^{K+1}} | P_{inf} \prec^{H_j^K} P_{\lambda}$

** Adjusted temporal interval $I_{\lambda} \subseteq I \wedge I_{\lambda} = T_m \cup \dots \cup T_k (1 \leq m < k < n)$

After executing the DrillDown^{multi-states}, the decision-maker obtains a new MT with one analysis axis unchanged (i.e. $D_j^{K+1} \neq D_i \rightarrow P_j^{K+1} = \langle P_1^{D_j^K}, P_2^{D_j^K}, \dots \rangle$) and the other analysis axis displaying information at a finer granularity level (i.e. $D_j^{K+1} = D_i \rightarrow P_j^{K+1} = \langle P_{inf} \rangle$). However, as certain parameters at low granularity levels exist only in most recent states but not in former ones, it is very likely that some input parameters become *missing attributes* in some involved states. The *missing attributes* should be adjusted before being displayed in the new MT (i.e. $D_j^{K+1} = D_i \rightarrow P_j^{K+1} = \langle P_{\lambda} \rangle$). The Drilldown^{multi-states} operator follows the algorithm below while being executed.

Algorithm 2: Drilldown^{multi-states} ($[I]; MT_k; D; P; [Adaptation]$) = MT_{k+1}

Input: Set of temporal intervals I , currently displayed multidimensional table MT_k , currently displayed dimension D , chosen parameter P , optional adaptation method $Adaptation$.

Output: new multidimensional table MT_{k+1}

Begin

1. Let H_{actual} be the actually displayed hierarchy on dimension D in the current MT_k ,
 $H_{actual} \in \mathcal{H}^D$
2. If $I \neq \emptyset$ then,
3. Find the maximal subset of states $E' \subseteq \mathcal{E}$, $\forall E_j \in E' \mid T_j \cap I \neq \emptyset$
4. Else
5. $E' =$ the set of states involved in MT_k , $E' \subseteq \mathcal{E}$
6. End if
7. If $P \in MA^{E'}$ then
8. If $Adaptation = DEPTH$ then
9. Find the maximal subset of states $E'' \subseteq E' \mid P \notin MA^{E''}$
10. $E' = E''$
11. Else
12. Let $P_{Drilldown} = P$
13. Let $P = \emptyset$
14. While $P_{Drilldown} <^{H_{actual}} ALL_D \wedge P = \emptyset$
15. If $\forall E_j \in E' \mid P_{Drilldown} \in \mathcal{A}^{D_j}$ then
16. $P = P_{Drilldown}$
17. Else
18. $P_{Drilldown}$ increases one granularity level
19. End if
20. End While
21. End if
22. End if
23. If $E' = \emptyset \vee P = \emptyset$
24. Impossible operation, display a warning message.
25. Else
26. For $E_j \in E'$
27. Let $MT_K^{E_j}$ be the part of MT in the state E_j
28. Translate Drilldown ($MT_K^{E_j}; D; P$) into query Q_j
29. Global query $Q = Q \cup Q_j$
30. $I_{TM} = I_{TM} \cup (T_k \cap I)$
31. End for
32. New $MT_{k+1} =$ former MT_k updated with the results of query Q , $I_{k+1} = I_{TM}$, $R_{k+1} = R_k$
33. End if

End

When a *missing attribute* is involved in an analysis, the DrillDown^{multi-states} operator should firstly take adaptive actions based on the chosen adaptation method as follows:

- According to *Depth-first* method, the DrillDown^{multi-states} operator keeps the specified parameter P unchanged while reducing the temporal intervals. The objective is to find a subset of states that each of them contains the parameter P.
- According to *Breadth-first* method, the DrillDown^{multi-states} operator augments the chosen parameter's granularity level while keeping the same temporal intervals. This approach aims to find the first common parameter among all involved states.

Example 5. After executing the previous Display^{multi-states} operator, the decision-maker continues her/his analysis by consulting orders' revenues by *CUSTKEY* and *CATEGORY* for the same period as before. Moreover, she/he points out in this analysis the temporal intervals is more important than the depth of the involved granularity levels. This analysis needs implies the use of two Drilldown^{multi-states} operators.

- The first Drilldown^{multi-states} operator is carried out on the dimension *CUSTOMER*. As the parameter *CUSTKEY* only exists in the state E_1 from 1997 to 1998 (cf. figure 4), the first Drilldown^{multi-states} operator has to adapt itself in accordance to the *Breadth-first* adaptation method: the temporal intervals from 1995 to 1998 remains unchanged, while the parameter *CUSTKEY* is replaced by *CITY* which is the first common parameter in the states E_1 (1995-1997) and E_2 (1997-1998).
- The second Drilldown^{multi-states} operator is performed on the dimension *PART*. Its execution does not need any adaptive action, because the parameter *CATEGORY* is always available on the dimension *PART* from 1995 to 1998 (cf. figures 1 and 2).

The *Breadth-first* approach results in a combination of analysis operators as follows: Drilldown^{multi-states} ((Drilldown^{multi-states} (MT_0 , *CUSTOMER*, *CITY*, *BREADTH*)), *PART*, *CATEGORY*, *BREADTH*). After executing these two operators, the decision-maker receives the analysis results presented in figure 7.

Figure 7. MT_1 obtained according to *Breadth-first* approach.

LINEORDER SUM(REVENUE)		PART HPART_BR		
		CATEGORY	MFGR#11	MFGR#12
CUSTOMER HCUS_GEO	CITY			
	ALGERIA 0		6708292181	6793630778
	BRAZIL 1		8153363902	8243026709

R = ∅, I = [1995, 1998[

Example 6. Suppose another decision-maker carries out the same analysis described in the example 5, but she/he puts more emphasis on the depth of involved granularity level. In this case the temporal intervals of analysis have to be reduced to $[t_{1997}, t_{1998}[$ in order to keep the analysis granularity to *CUSTKEY* level. The *Depth-first* approach brings about a different combination of analysis operators: Drilldown^{multi-states} ((Drilldown^{multi-states} (MT_0 , *CUSTOMER*, *CUSTKEY*, *DEPTH*)), *PART*, *CATEGORY*). The results of analysis are shown in figure 8.

Figure 8. Another version of analysis results obtained according to Depth-first approach.

LINEORDER SUM(REVENUE)		PART HPART_BR		
		CATEGORY	MFGR#11	MFGR#12
CUSTOMER HCUS_GEO	CUSTKEY			
	1		8341249	13136766
	2		665266	5851875

R = ∅, I = [1997, 1998[

Rollup^{multi-states} operator

The Rollup^{multi-states} operator consists in moving from finer granularity data to coarser granularity data on a currently displayed dimension in several states (cf. table 3).

Let $ax^K \in Ax^K \mid ax^K = (D^{ax^K}, H^{ax^K}, P^{ax^K})$ be the analysis axis affected by the Rollup^{multi-states} in the currently displayed multidimensional table MT_k .

Table 3. Algebraic multi-states rollup operator

Rollup ^{multi-states} (I ; MT_k ; D_i ; P_{sup}) = MT_{K+1} ;	
Input	<ul style="list-style-type: none"> • $I \subseteq \mathcal{T}_{TIMES}$: a set of optional temporal intervals • MT_k: multidimensional table currently displayed • $D_i = D^{ax^K}$: one of the analysis axes currently displayed • $P_{sup} \in \mathcal{A}^{D_i}$: chosen parameter or weak attribute on dimension D_i, $P_{sup} \prec^{H^{ax^K}} ALL_{D_i}$ and $\forall P_m \in \mathcal{P}^{ax^K} \mid P_m \prec^{H^{ax^K}} P_{sup}$
Output	$MT_{K+1} = (S^{K+1}; Ax^{K+1}; R^{K+1}; I^{k+1})$ analysis results MT such as <ul style="list-style-type: none"> • Analysis subject $S^{K+1} = S^K$ • Analysis axes Ax^{K+1}, $\forall ax_j^{K+1} \in Ax^{K+1} \mid ax_j^{K+1} = (D_j^{K+1}, H_j^{K+1}, P_j^{K+1})$ such as <ul style="list-style-type: none"> ◦ Dimension $D_j^{K+1} = D_j^K$, ◦ Hierarchy $H_j^{K+1} = H_j^K$, ◦ Parameter $P_j^{K+1} \subseteq \mathcal{P}^{H_j^{K+1}} \mid (D_j^{K+1} = D_i \rightarrow P_j^{K+1} = \langle P_{sup} \rangle) \wedge (D_j^{K+1} \neq D_i \rightarrow P_j^{K+1} = \langle P_1^{D_j^K}, P_2^{D_j^K}, \dots \rangle)$ • Selection predicates on dimensions and/or fact $R^{K+1} = R^K$ • Validation period $I^{k+1} = I \vee I^{k+1} = I_k$

Rollup^{multi-states} operator reveals data at a higher granularity level. As currently displayed granularity levels are already the common ones in all involved states, chosen parameter denoted P_{sup} presents no doubt in all concerned states. No adaptation method needs to be specified for Rollup^{multi-states} operator (cf. algorithm 3).

Algorithm 3: Rollup^{multi-states} ($[I]; MT_k; D; P_{sup}$) = MT_{k+1}

Input: Set of temporal intervals I , currently displayed multidimensional table MT_k , currently displayed dimension D , parameter P_{sup} .

Output: new multidimensional table MT_{k+1}

Begin

1. Let H_{actual} be the actually displayed hierarchy on dimension D in the current MT_k
 $H_{actual} \in \mathcal{H}^D$
2. Let P_{actual} be the actually displayed parameter in hierarchy H_{actual} in MT_k , $P_{actual} \in \mathcal{A}^D$
3. If $I \neq \emptyset$ then,
4. Find the maximal subset of states $E' \subseteq \mathcal{E}$, $\forall E_j \in E' \mid T_j \cap I \neq \emptyset$
5. Else
6. $E' =$ the set of states involved in MT_k , $E' \subseteq \mathcal{E}$
7. End if
8. If $P \prec^{H_{actual}} P_{actual}$ then
9. Impossible operation, display a warning message.
10. Else
11. For $E_j \in E'$
12. Let $MT_K^{E_j}$ be the part of MT in the state E_j
13. Translate Rollup ($MT_K^{E_j}; D; P$) into query Q_j
14. Global query $Q = Q \cup Q_j$
15. $I_{TM} = I_{TM} \cup (T_k \cap I)$
16. End for
17. New $MT_{K+1} =$ former MT_k updated with the results of query Q , $I_{k+1} = I_{TM}$, $R_{k+1} = R_k$
18. End if

End

Example 7. The decision-maker continues her/his analysis after the execution of Drilldown^{multi-states} according to the *Breadth-first* adaptation method (cf. figure 7). Haven't found a great difference of revenues according to *CATEGORY* and *CITY*, the decision-maker wants to analyze the revenue at *MFGR* level of the dimension *PART*. The only analysis operator involved is presented as follows: Rollup^{multi-states} ($MT_{k+1}, PART, MFGR$). After the execution of this analysis operator, the decision-maker obtains a new MT presented in figure 9. She/he concludes that the *MFGR* of *PART* has not great influence on the revenue realized in a city.

Figure 9. MT_2 after the execution of Rollup^{multi-states}.

LINEORDER SUM(REVENUE)		PART HPART_BR		
		MFGR	MFGR#1	MFGR#2
CUSTOMER HCUS_GEO	CITY			
	ALGERIA 0		34097428973	34962799012
	BRAZIL 1		41346033704	41252993688

R = \emptyset , I = [1995, 1998[

Changing the content of an analysis axis

The rotation operator allows decision-makers to (a) replace a currently displayed dimension by a new one and (b) switch to a new analysis hierarchy on the same or a new dimension. Its algebraic representation is shown in the table 4.

Let $ax^K \in Ax^K \mid ax^K = (D^{ax^K}, H^{ax^K}, P^{ax^K})$ be the analysis axis affected by $Rotate^{multi-states}$ in the currently displayed multidimensional table MT_k .

Table 4. Algebraic multi-states rotate operator

Rotate ^{multi-states} ($[I]$; $MT_k, D_{old}; D_{new}; H_{new}; [Adaptation]$) = MT_{K+1} ;	
Input	<ul style="list-style-type: none"> • $I \subseteq \mathcal{T}_{TIMES}$: a set of optional temporal intervals • MT_k: the multidimensional table currently displayed • $D_{old} = D^{ax^K}$: one of the analysis axes currently displayed to be replaced • $D_{new} \in \mathcal{D}$: a new analysis axis to replacing D_{old} • $H_{new} \in H^{D_{new}} \wedge H_{new} \neq H^{ax^K}$: a hierarchy on the dimension D_{new} to be displayed in the new MT_{k+1} • $Adaptation = \{BREADTH, DEPTH\}$: an optional parameter to specify the adaptation method. If no method is specified, the analysis operator takes <i>breadth-first</i> adaptation method by default.
Output	$MT_{K+1} = (S^{K+1}; Ax^{K+1}; R^{K+1}; I^{k+1})$ analysis results MT such as <ul style="list-style-type: none"> • Analysis subject $S^{K+1} = S^K$ • Analysis axes $Ax^{K+1}, \forall ax_j^{K+1} \in Ax^{K+1} \mid ax_j^{K+1} = (D_j^{K+1}, H_j^{K+1}, P_j^{K+1})$ such as <ul style="list-style-type: none"> ○ Dimension $D_j^{K+1} = D_j^K \vee D_j^{K+1} = D_{new} \vee D_j^{K+1} = \emptyset$, ○ Hierarchy $H_j^{K+1} = H_j^K \vee H_j^{K+1} = H_{new} \vee H_j^{K+1} = \emptyset$, ○ Parameter $P_j^{K+1} = P_j^K \vee P_j^{K+1} = ALL^{D_c} \vee P_j^{K+1} = \emptyset$ • Selection predicates on dimensions and/or fact $R^{K+1} = R^K$ • Validation period $I^{k+1} = I \vee I^{k+1} = I_k \vee I^{k+1} = I_\lambda^*$

* Adjusted temporal interval $I_\lambda \subseteq I \wedge I_\lambda = T_m \cup \dots \cup T_k$ ($1 \leq m < k < n$)

There may be several cases where the $Rotate^{multi-states}$ can be used:

- When decision-makers need to change one currently displayed analysis axis, they should indicate the dimension to be replaced (D_{old}), the new dimension to be displayed (D_{new}) and one valid hierarchy (H_{new}) on the new dimension.
- When decision-makers want to switch to a new hierarchy on the same dimension, they should keep the currently displayed dimension ($D_{new} = D_{old}$) while choosing a valid hierarchy (H_{new}) different from the current displayed hierarchy H^{ax^K} .

- If decision-makers decide to remove one analysis axis, they may simply assign a *NULL* value to the new dimension (D_{new}) as well as the new hierarchy (H_{new}).

After execution of $\text{Rotate}^{\text{multi-states}}$ operator, the unaffected analysis axis remains unchanged (i.e. $D_j^{K+1} = D_j^K$, $H_j^{K+1} = H_j^K$, $P_j^{K+1} = P_j^K$). The replaced dimension and hierarchy may be the ones chosen by decision-makers (i.e. $D_j^{K+1} = D_{\text{new}}$, $H_j^{K+1} = H_{\text{new}}$). In this case, the displayed parameter is systematically positioned to the maximal granularity level (i.e. $P_j^{K+1} = \text{ALL}^{D_c}$). In the case where a missing element (dimension and/or hierarchy) is involved, the $\text{Rotate}^{\text{multi-states}}$ operator adapts the input variables according to the adaptation method chosen by decision-makers.

We propose the following algorithm to take into account the specificities of reduced MDW while executing the $\text{Rotate}^{\text{multi-states}}$ operator.

Algorithm 4: $\text{Rotate}^{\text{multi-states}}([I]; MT_k; D_{\text{old}}; D_{\text{new}}; H_{\text{new}}; [Adaptation]) = MT_{k+1}$

Input: Set of temporal intervals I , currently displayed multidimensional table MT_k , currently displayed dimension D_{old} , chosen dimension to be displayed D_{new} , chosen hierarchy to be displayed H_{new} , optional adaptation method $Adaptation$.

Output: new multidimensional table MT_{k+1}

Begin

1. If $I \neq \emptyset$ then,
2. Find the maximal subset of states $E' \subseteq \mathcal{E}$, $\forall E_j \in E' \mid T_j \cap I \neq \emptyset$
3. Else
4. $E' =$ the set of states involved in MT_k , $E' \subseteq \mathcal{E}$
5. End if
6. If $D_{\text{new}} \in \text{MD}^{E'}$ then
7. If $Adaptation = \text{DEPTH}$ then
8. Find the maximal subset of states $E'' \subseteq E' \mid D_{\text{new}} \notin \text{MD}^{E''}$
9. $E' = E''$
10. Else
11. $D_{\text{new}} = \emptyset$, $H_{\text{new}} = \emptyset$
12. End if
13. Else
14. If $D_{\text{new}} \neq \emptyset \wedge H_{\text{new}} \in \text{MH}^{E'}$
15. If $Adaptation = \text{DEPTH}$ then
16. Find the maximal subset of states $E'' \subseteq E' \mid H_{\text{new}} \notin \text{MH}^{E''}$
17. $E' = E''$
18. Else
19. Find a hierarchy $H_k^{D_{\text{new}}} \in \mathcal{H}^{D_{\text{new}}} \mid H_k^{D_{\text{new}}} \notin \text{MH}^{E'}$
20. $H_{\text{new}} = H_k^{D_{\text{new}}}$
21. End if
22. End if
23. End if
24. For $E_j \in E'$
25. Let $MT_K^{E_j}$ be the part of MT in the state E_j

26. Translate Rotate ($MT_K^{E_j}$; D_{old} ; D_{new} ; H_{new} ;) into query Q_j
 27. Global query $Q = Q \cup Q_j$
 28. $I_{TM} = I_{TM} \cup (T_k \cap I)$
 29. End for
 30. New MT_{K+1} = former MT_k updated with the results of query Q , $I_{k+1} = I_{TM}$, $R_{k+1} = R_k$
- End**

Example 8. Based on the former analysis results presented in MT_2 (cf. figure 9), the decision-maker wants to analyze revenue according to *YEARMONTH* in the hierarchy *HDATE* of the dimension *DATE*. This analysis is translated to a combination of two analysis operators: Drilldown^{multi-states} (Rotate^{multi-states} (MT_2 , *CUSTOMER*, *DATE*, *HDATE*), *DATE*, *YEARMONTH*).

- A Rotate^{multi-states} should firstly be executed to change the horizontal analysis axis to the hierarchy *HDATE* on the dimension *DATE*. By definition, the displayed parameter after the execution of rotation operator is the extreme parameter. In this case, the parameter at the highest granularity level on dimension *DATE* is displayed, namely *ALL_D*.
- In order to reach to the parameter *YEARMONTH* on the dimension *DATE*, we apply a Drilldown^{multi-states} to the horizontal analysis axis.

In this analysis, neither of the operators contains missing element. The adaptation part in the algorithm 4 (i.e. lines 6-23) and that in the algorithm 2 (i.e. lines 7-25) are not processed during the execution. At the end of the analysis, the decision-maker obtains the results in figure 10.

Figure 10. MT_3 obtained after the rotation of analysis axis.

LINEORDER SUM(REVENUE)		PART HPART_BR		
		MFGR	MFGR#1	MFGR#2
DATE HDATE	YEARMONTH			
	Jan1995		227038697547	226683548464
	Feb1995		203211459188	202809213967

R = \emptyset , I = [1995, 1998[

Adding restriction predicates

During decision-making process, decision-makers may have interests to focus on a subset of data. Therefore, we propose the Select^{multi-states} operator which allows removing fact and dimension instances that do not satisfy the specified selection criteria. The Select^{multi-states} operator regroups the *Dice* and *Slice* operators proposed in (Agrawal, Gupta, & Sarawagi, 1997). Its algebraic representation is shown in the table 5.

Table 5. Algebraic multi-states select operator

Select ^{multi-states} ([I]; MT_k , <i>pred</i> ; [<i>Adaptation</i>]) = MT_{K+1} ;	
Input	<ul style="list-style-type: none"> • $I \subseteq \mathcal{T}_{TIMES}$: a set of optional temporal intervals • MT_k: the multidimensional table currently displayed • $pred \neq \emptyset$: a set of restriction predicates to be applied to non-temporal

	<p>dimension and/or fact *</p> <ul style="list-style-type: none"> • <i>Adaptation</i> = {<i>BREADTH</i>, <i>DEPTH</i>}: an optional parameter to specify the adaptation method. If no method is specified, the analysis operator takes <i>breadth-first</i> adaptation method by default.
Output	<p>$MT_{K+1} = (S^{K+1}; Ax^{K+1}; R^{K+1}; I^{K+1})$ analysis results MT such as</p> <ul style="list-style-type: none"> • Analysis subject $S^{K+1} = S^K$ • Analysis axes $Ax^{K+1}, \forall ax_j^{K+1} \in Ax^{K+1} ax_j^{K+1} = (D_j^{K+1}, H_j^{K+1}, P_j^{K+1})$ such as <ul style="list-style-type: none"> ○ Dimension $D_j^{K+1} = D_j^K$ ○ Hierarchy $H_j^{K+1} = H_j^K$ ○ Parameter $P_j^{K+1} = P_j^K$ • Selection predicates on dimensions and/or fact $R^{K+1} = R^K \cap pred$ • Validation period $I^{K+1} = I \vee I^{k+1} = I_k \vee I^{k+1} = I_\lambda^{**}$

* the set of restriction predicates *pred* cannot include temporal predicates. All temporal predicates should be expressed through the temporal intervals *I* in order to update the validation period of MT_{k+1} (Cf. definition 9).

** Adjusted temporal interval $I_\lambda \subseteq I \wedge I_\lambda = T_m \cup \dots \cup T_k$ ($1 \leq m < k < n$).

The $Select^{multi-states}$ operator consists in adding restriction predicates (*pred*) to restrict both dimension and fact instances without changing the structure of MT (i.e. $Ax^{K+1} = Ax^K$). After the execution of $Select^{multi-states}$ operator, the new MT_{k+1} updates the restriction predicates (i.e. $R^{K+1} = R^K \cap pred$). Note that a selection predicate denoted $pred_k$ can be applied to any fact or non-temporal dimension instances, even if they are not displayed in the currently displayed MT_k .

Like other multi-states analysis operators, the $Select^{multi-states}$ operator cannot avoid involving *missing dimension* and/or *missing attribute*. If a restriction predicate $pred_k$ cannot be satisfied due to the involved missing element, it will be removed from the set of restriction predicates *pred*. As an extreme example, if none of the restriction predicates can be satisfied, the set of restriction predicates in MT_{k+1} remains unchanged (i.e. $pred = \emptyset \rightarrow R^{K+1} = R^K$).

Algorithm 5: $Select^{multi-states}([I]; MT_k, pred; [Adaptation]) = MT_{k+1}$

Input: Set of temporal intervals *I*, currently displayed multidimensional table MT_k , chosen selection predicates *pred*, optional adaptation method *Adaptation*.

Output: new multidimensional table MT_{k+1}

Begin

1. If $I \neq \emptyset$ then,
2. Find the maximal subset of states $E' \subseteq \mathcal{E}, \forall E_j \in E' | T_j \cap I \neq \emptyset$
3. Else
4. $E' =$ the set of states involved in $MT_k, E' \subseteq \mathcal{E}$
5. End if
6. For $pred_k \in pred$
7. If $pred_k$ contains a predication on non-temporal dimension instances


```

8.   Let  $D_{pred_k}$  be the involved dimension,  $D_{pred_k} \in \mathcal{D}$ 
9.   Let  $P_{pred_k}$  be the involved parameter,  $P_{pred_k} \in \mathcal{A}^{D_{pred_k}}$ 
10.  If  $D_{pred_k} \in MD^{E'}$  then
11.    If Adaptation = DEPTH then
12.      Find the maximal subset of states  $E'' \subseteq E' \mid D_{pred_k} \notin MD^{E''}$ 
13.       $E' = E''$ 
14.    Else
15.       $pred = pred \setminus pred_k$ 
16.      Display a warning message about the deletion of  $pred_k$ 
17.    End if
18.  Else
19.    If  $P_{pred_k} \in MA^{E'}$  then
20.      If Adaptation = DEPTH then
21.        Find the maximal subset of states  $E'' \subseteq E' \mid P_{pred_k} \notin MA^{E''}$ 
22.         $E' = E''$ 
23.      Else
24.         $pred = pred \setminus pred_k$ 
25.        display a warning message about the deletion of  $pred_k$ 
26.      End if
27.    End if
28.  End if
29. End if
30 End for
31 For  $E_j \in E'$ 
32   Let  $MT_K^{E_j}$  be the part of MT in the state  $E_j$ 
33   Translate Select ( $MT_K^{E_j}; pred$ ) into query  $Q_j$ 
34   Global query  $Q = Q \cup Q_j$ 
35    $I_{TM} = I_{TM} \cup (T_k \cap I)$ 
36 End for
37 New  $MT_{K+1}$  = former  $MT_k$  updated with the results of query  $Q$ ,  $I_{k+1} = I_{TM}$ ,  $R_{k+1} = R_k \cap pred$ 
End

```

Example 9. The formerly obtained MT_3 shows the sum of revenue according to *MONTH* and *MFGR* (cf. figure 10). Now the decision-maker wants to analyze the orders placed by *FRENCH* customers concerning *EUROPEEN* suppliers, and the sum of orders' revenue must be over 20 million. This analysis can be translated into one *Select*^{multi-states} operator including three restriction predicates: *Select*^{multi-states} ($MT_3; SUPPLIER.S_REGION = 'EUROPE' \wedge CUSTOMER.C_NATION = 'FRANCE' \wedge SUM(LO_REVENUE) > 200000000$).

During the execution of the *Select*^{multi-states} operator, the lines 10 to 28 are skipped since none of the restriction predicate includes missing element. After the execution, the decision-maker obtain the results shown in figure 11.

Figure 11. MT_4 obtained after applying three restriction predicates.

LINEORDER SUM(REVENUE)		PART HPART_BR		
		MFGR	MFGR#1	MFGR#2
DATE HDATE	YEARMONTH			
	Aug1995		2017184117	
	Jan1998			2106853624

R = SUPPLIER.S_REGION = 'EUROPE' ^ CUSTOMER.C_NATION = 'FRANCE' ^ SUM(LO_REVENUE) > 2000000000, l = [1995, 1998]				

Remark. In the literature, we can find more OLAP operators other than basic ones. They are usually used to support more sophisticated analysis operations. Since most of the complex analyses can be realized through combinations of commonly-used operators, we do not discuss advanced OLAP operators in this paper.

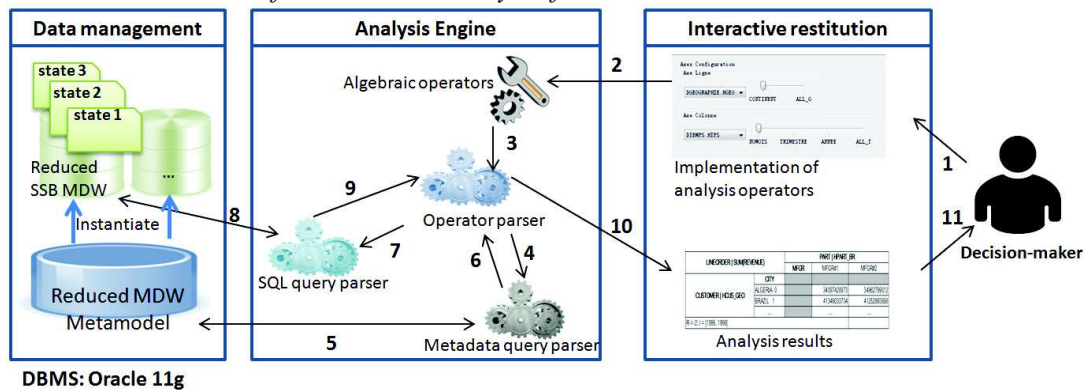
IMPLEMENTATION

Based on the proposed modeling solutions for reduced MDW and analysis operators, we develop a multi-states analysis framework using Java JDK 7 and Oracle 11g. We firstly introduce the global architecture of the framework. Secondly, we present two modules in the framework, namely *Data Management* and *Analysis Engine*.

Architecture of multi-states analysis framework

The multi-states analysis framework is composed of three modules, namely *Data Management*, *Analysis Engine* and *Interactive Restitution* (cf. figure 12). Each part has specific roles and interacts with others.

Figure 12. Main modules of multi-states analysis frameworks



The *Data Management* module accommodates a metamodel and reduced MDWs. The metamodel allows managing reduced MDW, new MDW can be defined by instantiating the metamodel.

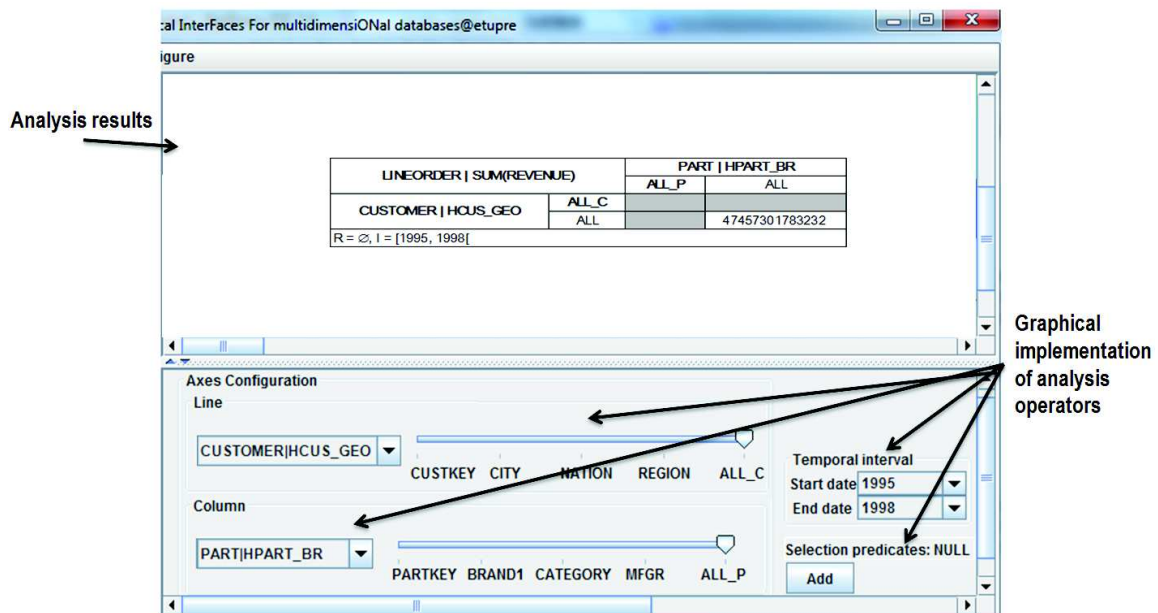
The *Analysis Engine* module is designed for R-OLAP environment. It is composed of a set of algebraic operators and three parsers :

- The set of *algebraic operators* defines elementary operations that decision-makers can carry out while analyzing. The definition of algebraic operators is independent of tools and implementation languages.
- The *operator parser* (a) translates algebraic operators into queries over metadata, (b) analyzes metadata queries' results and generates corresponding SQL queries over different states of MDW, (c) receives partial SQL query results and combines them together before sending one global result to graphical interface;
- The *metadata query parser* (a) receives and executes queries over metadata generated by *operator parser*, (b) returns metadata queries' results to *operator parser*;
- The *SQL query parser* (a) receives SQL queries generated by *operator parser* and executes them in corresponding MDW and states, (b) returns partial SQL queries' results to *operator parser*.

The *Interactive Restitution* module contains (a) a graphical implementation of analysis operators in order to facilitate decision-makers' tasks and (b) a graphical interface showing analysis results. The interface of the prototype (cf. figure 13) is divided into two parts:

- Analysis results are shown on the top of the graphical interface
- The multi-states operators are implemented with the help of graphical components on the bottom of the interface.

Figure 13. Graphical interface of interactive restitution module

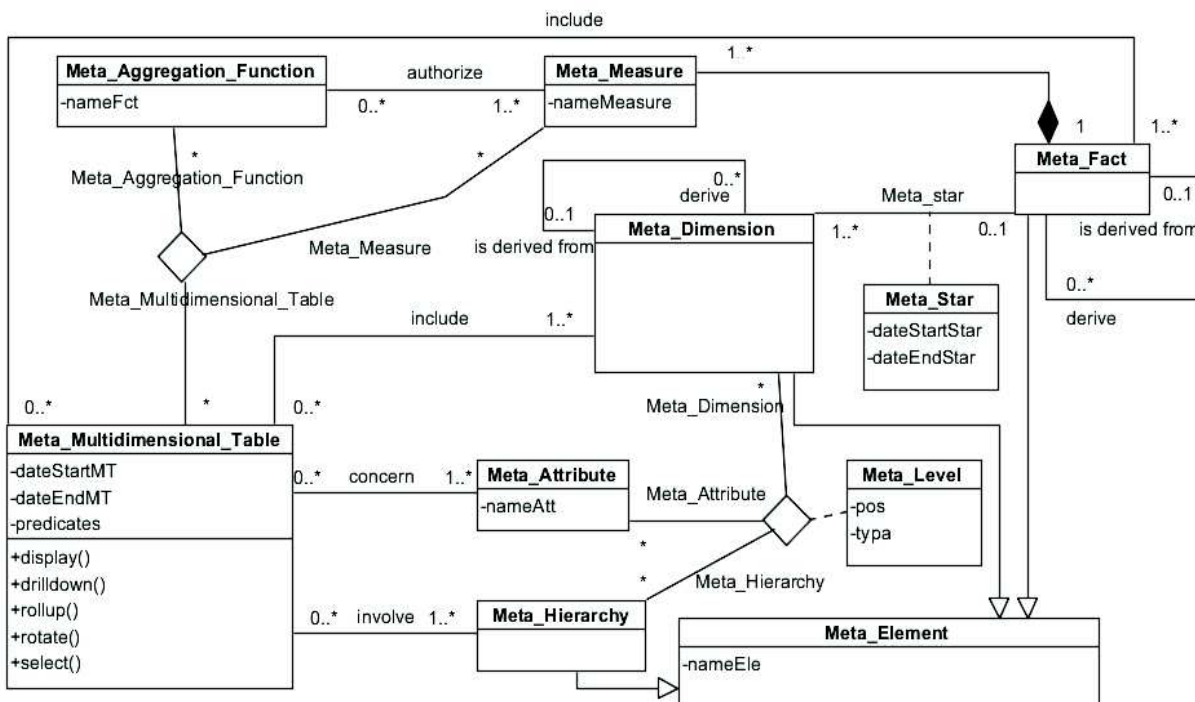


This multi-states analysis framework guarantees the transparency of data reduction in MDW. Decision-makers carry out analysis via graphical implementation of algebraic operators (arrow tagged "1" in figure 12) and then receive a global analysis result (arrow tagged "11" in figure 12).

Data management

The *Data Management* module allows (a) managing MDW composed of one or several states, (b) defining new reduced MDW and (c) automating the translation process from algebraic operators to SQL queries. Due to the limited space, we present a simplified metamodel which contains only essential components concerning multi-states OLAP analysis. Other components related to the constraints (cf. section *Preliminary Concepts Of Reduced MDW*) are not shown below. The graphical notation of the conceptual metamodel is based on UML class diagram (cf. figure 14).

Figure 14. UML class diagram metamodel of the reduced MDW



As we can see from the figure 14, the metamodel embodies all proposed concepts. To illustrate the particularities of reduced MDW and how it can be managed by the metamodel, we implement and instantiate the metamodel with the MDW of the case study. To distinguish the name of different types of elements, we add a suffix to “nameEle” attributes for each type of elements while implementing the specialized classes of the class *Meta_Element*. For instance, the “nameEle” for class *Meta_Dimension* is denoted as “nameEle_Dim” afterwards, while the “nameEle” for class *Meta_Fact* is denoted as “nameEle_Fact” in the following examples. We also add “_E_i” as suffix to the dimensions in the state E_i during implementation.

Firstly, the concept of state is implemented with the help of association *Meta_star*, it connects a dimension to a fact with a start date (dateStartStar) as well as an end date (dateEndStar). As we can see from the figure below, the fact *LINEORDER_E1* is connected with four dimensions, namely *CUSTOMER_E1*, *SUPPLIER_E1*, *DATE_E1* and *PART_E1*. The validation period of state E₁ is presented by the validation periods of all these *Fact-Dimension* associations, which is from 1997 to 1998 (cf. figure 15).

Figure 15. Content of *Meta_star*

NAMEELE_FACT	NAMEELE_DIM	DATESTARTSTAR	DATEENDSTAR
LINEORDER_E1	CUSTOMER_E1	1997	1998
LINEORDER_E1	SUPPLIER_E1	1997	1998
LINEORDER_E1	DATE_E1	1997	1998
LINEORDER_E1	PART_E1	1997	1998

Secondly, the derivation between states is managed by two recursive association both denoted *Derive* in metamodel. After implementation, the recursive associations are translated to foreign keys named *NAMEELE_DIM_DERIVED_FROM* and *NAMEELE_FACT_DERIVED_FROM*: they allow associating a derivate dimension and fact to a parent dimension and fact respectively. For instance, the dimension named *CUSTOMER_E1* of state *E₁* is associated to its parent dimension named *CUSTOMER* of original MDW (cf. figure 16), while the fact of state *E₁* named *LINEORDER_E1* is bonded with the parent fact named *LINEORDER* (cf. figure 17).

Figure 16. Content of *Meta_Dimension*

NAMEELE_DIM	NAMEELE_DIM_DERIVED_FROM
CUSTOMER	
SUPPLIER	
DATE	
PART_E1	
CUSTOMER_E1	CUSTOMER
SUPPLIER_E1	SUPPLIER
DATE_E1	DATE
DATE_E1	DATE

Figure 17. Content of *Meta_FACT*

NAMEELE_FACT	NAMEELE_FACT_DERIVED_FROM
LINEORDER	
LINEORDER_E1	LINEORDER

Thirdly, the relationships between levels are managed by association class *Meta_Level* containing two variables: (a) the position *Pos* shows the granularity level of an attribute in comparison to the others; (b) the type of attribute *Typa* indicates if it belongs to parameter or weak attribute in a *hierarchy*. As we can see from the figure 18 which shows the content of association *Meta_Level*, the attributes on the dimension *CUSTOMER_E1* are organized on two hierarchies: *HCUS_MKT* and *HCUS_GEO*. Each hierarchy organizes the attributes according to their granularity level. For example, hierarchy *HCUS_GEO* starts from attribute *CUSTKEY* at position “1” and ends up with extremity attribute *ALL_C* at position “5”. By definition, a parameter may be associated with one or several descriptive attributes (weak attributes) on the same level. That explains why four attributes of different types *CUSTKEY* (parameter) and *NAME*, *PHONE* as well as *ADDRESS* (weak attribute) share the same position in the hierarchy *HCUS_GEO*.

Figure 18. Content of Meta_Level

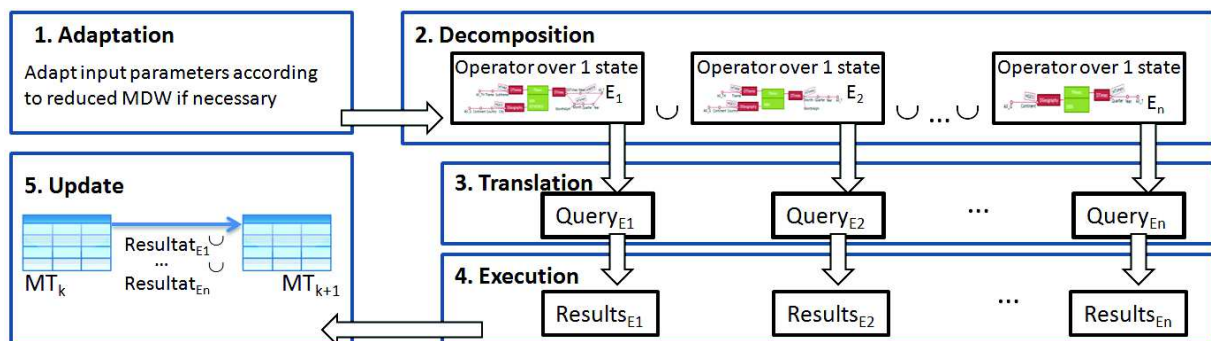
NAMEELE_DIM	NAMEELE_HIERARCHY	NAMEATT	POS	TYPA
CUSTOMER_E1	HCUS_MKT	CUSTKEY	1	PARAMETER
CUSTOMER_E1	HCUS_MKT	MKTSEGMENT	2	PARAMETER
CUSTOMER_E1	HCUS_MKT	ALL_C	3	PARAMETER
CUSTOMER_E1	HCUS_GEO	CUSTKEY	1	PARAMETER
CUSTOMER_E1	HCUS_GEO	NAME	1	WEAK-ATTRIBUTE
CUSTOMER_E1	HCUS_GEO	PHONE	1	WEAK-ATTRIBUTE
CUSTOMER_E1	HCUS_GEO	ADDRESS	1	WEAK-ATTRIBUTE
CUSTOMER_E1	HCUS_GEO	CITY	2	PARAMETER
CUSTOMER_E1	HCUS_GEO	NATION	3	PARAMETER
CUSTOMER_E1	HCUS_GEO	REGION	4	PARAMETER
CUSTOMER_E1	HCUS_GEO	ALL_C	5	PARAMETER

Analysis Engine

One of the core modules of our framework is *Analysis Engine*. In this section we present more details about multi-states analysis processing steps.

To facilitate decision-makers' tasks, they only interactively selects the MT components. The system converts the selection into the corresponding OLAP analysis operator. The execution of analysis operator follows its associated algorithm which is implemented according to five steps in *Analysis Engine* (cf. figure 19). More precisely, through a set of temporal intervals chosen by decision-makers, the analysis operator firstly determines the states in which the analysis is carried out before adjusting input parameters if necessary. Then it splits up into several operators, each of them is applied to a single state along with eventually adjusted parameters. Next each partial operator is translated and executed independently to get partial results. At last, the currently displayed multidimensional table denoted MT_k is updated with all partial results obtained. Only the global results presented in the new multidimensional table MT_{k+1} are returned to decision-makers.

Figure 19. Analysis processing steps of multi-states analysis operators



EXPERIMENTAL ASSESSMENTS

After implementation, we carry out some experimental assessments through the graphical interface of the *Interactive Restitution* module. The experimental assessments aim at (a) illustrating how analysis over reduced data is carried out through multi-states analysis operators and (b) comparing the efficiency of analysis in reduced and unreduced MDW.

Efficacy of multi-states OLAP analysis

In this section we carry out OLAP analyses over reduced data through the graphical interface of the framework.

Starting from the MT shown in the figure13, a decision-maker is analyzing the sum of revenues of all *PART*s bought by all *CUSTOMER*s from 1995 to 1998 (i.e. states E_1 and E_2 of the reduced MDW). Wishing to consult the revenues by *MFGR* and by *CUSTKEY* to get more detailed information, she/he slides the cursors concerning the horizontal and vertical analysis axes to corresponding levels. The framework detects it concerns two multi-states drilldown operations: (a) a Drilldown^{multi-states} from *ALL_P* to *MFGR* in the hierarchy *HPART_BR* of the dimension *PART* and (b) another Drilldown^{multi-states} from *ALL_C* to *CUSTKEY* in the hierarchy *HCUS_GEO* of the dimension *CUSTOMERS*. Details about the execution of these two operators are as follows:

Drilldown^{multi-states} on dimension *PART*.

As no analysis interval has been specified, the first Drilldown^{multi-states} on dimension *PART* take the same validation period of the MT_0 in the figure 13. The involved states in this analysis are E_1 and E_2 (cf. lines 2 to 6 in the algorithm 2). Then, by consulting the metamodel, the analysis framework finds the specified parameter *MFGR* is available in both E_1 and E_2 . During this processing step, the first query applicable to metamodel has been generated (cf. figure 20).

Figure 20. The first generated query over metadata

```
SELECT SL.FACT_NAME AS FACT, SL.DIMENSION_NAME AS DIML, SC.DIMENSION_NAME AS DIMC
FROM META_STAR SL, META_FACT F, META_DIMENSION DL, META_DIMENSION DC, META_STAR SC
WHERE SL.FACT_NAME = F.NAME
      AND SC.FACT_NAME = SL.FACT_NAME
      AND F.NAME_PARENT = 'LINEORDER'
      AND DL.NAME = SL.DIMENSION_NAME
      AND DC.NAME = SC.DIMENSION_NAME
      AND DL.NAME = 'PART'
      AND DC.NAME = 'DATESSB'
      AND SL.DATES <= TO_DATE ('30-12-1998', 'DD-MM-YYYY')
      AND SL.DATEE >= TO_DATE ('01-01-1995', 'DD-MM-YYYY')
```

After receiving the results of the first query, the multi-states analysis framework transforms Drilldown^{multi-states} operator into several operators. Each operator is translated into a query applied to one single state. After the execution of the global query (cf. figure 21), the MT_0 is updated with the obtained results (cf. algorithm lines 26 to 32).

Figure 21. The second generated query over reduced data

```

(SELECT SUM(NBN) AS LINEORDER, DIML.ALL_C, DIMC.MFGR
FROM LINEORDER_E1 FAIT, PART_E1 DIMC, CUSTOMER_E1 DIML, DATESSB_E1 DIMT
WHERE FAIT.ID_PART_E1 = DIMC.ID_PART_E1
AND FAIT.ID_DATE_E1 = DIMT.ID_DATE_E1
AND FAIT.ID_CUST_E1 = DIML.ID_CUST_E1
AND TO_DATE(DIMT.D_DATE, 'DD-MM-YYYY')
BETWEEN TO_DATE('01-01-1995', 'DD-MM-YYYY')
AND TO_DATE('30-12-1998', 'DD-MM-YYYY')
GROUP BY DIML.ALL_C, DIMC.MFGR )
UNION
(SELECT SUM(NBN) AS LINEORDER, DIML.ALL_C, DIMC.MFGR
FROM LINEORDER_E2 FAIT, PART_E2 DIMC, CUSTOMER_E1 DIML, DATESSB_E2 DIMT
WHERE FAIT.ID_PART_E2 = DIMC.ID_PART_E2
AND FAIT.ID_DATE_E2 = DIMT.ID_DATE_E2
AND FAIT.ID_CUST_E2 = DIML.ID_CUST_E2
AND TO_DATE(DIMT.D_DATE, 'DD-MM-YYYY')
BETWEEN TO_DATE('01-01-1995', 'DD-MM-YYYY')
AND TO_DATE('30-12-1998', 'DD-MM-YYYY')
GROUP BY DIML.ALL_C, DIMC.MFGR )

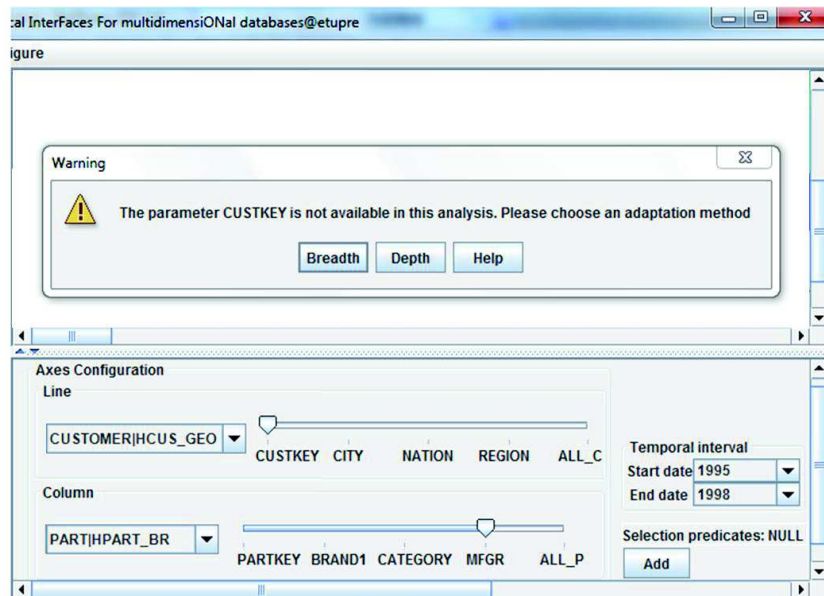
```

Drilldown^{multi-states} on dimension CUSTOMER .

The second Drilldown^{multi-states} operator is based on the intermediary MT₁ obtained after the first Drilldown^{multi-states}. The execution of this operator is more complex, since a *missing attribute CUSTKEY* is involved.

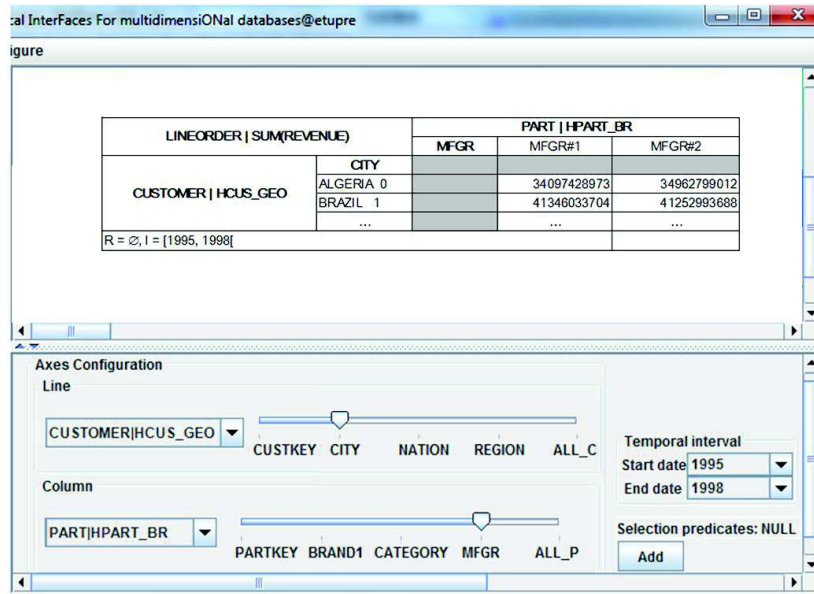
To deal with the *missing attribute*, the analysis framework firstly asks about the adaptation method to use (cf. figure 22). Since the decision-maker chooses the *breadth-first* option, the framework searches another parameter at a coarser granularity level in the hierarchy HCUS_GEO of dimension CUSTOMER (cf. algorithm 2 lines 12-20). In this case, the nearest parameter CITY happens to be the first common parameter in both E₁ and E₂.

Figure 22. Choice of adaptation method to deal with a missing attribute



Next, the Drilldown^{multi-states} is divided into two Drilldown operators. Each Drilldown operator is carried out over one state with the adapted parameter *CITY*. By using the union of all partial queries' results, the multi-states analysis framework updates the currently displayed MT₁. Further analyses can be carried out based on the newly obtained *multidimensional table* denoted MT₂ (cf. figure 23).

Figure 23. Analysis results obtained after two Drilldown^{multi-states} operations



Efficiency of multi-states OLAP analysis

In order to see how far the gain of execution time may reach within the reduced SSB MDW, we record the execution time of two Drilldown^{multi-states} operators in the previous section. In the control MDW without reduction (containing over 24 million tuples of synthetic data), the execution time reaches up to 6.83 and 4.678, while in the reduced MDW, their execution time is 0.15 and 0.098 second. The gain of execution time in the reduced MDW is significant: 97.8%.

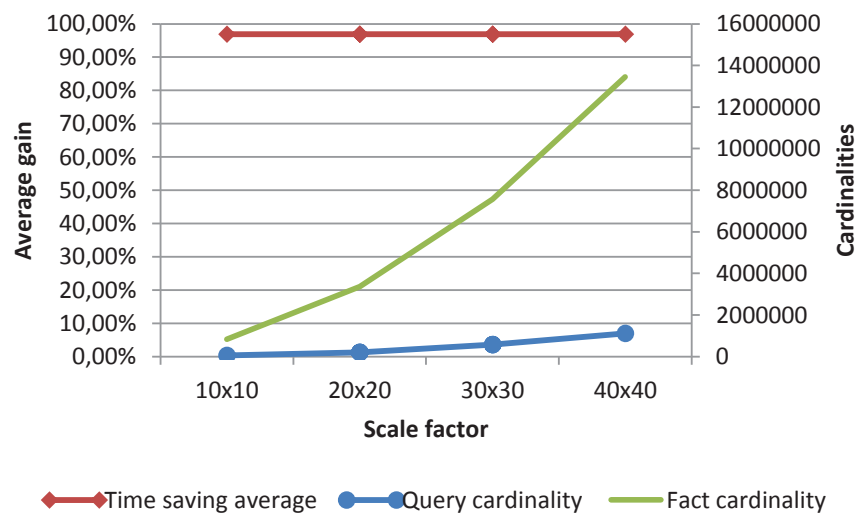
In R-OLAP environment, the multi-states OLAP operators are translated into SQL queries during execution. To evaluate the efficiency of multi-states OLAP analysis, we can study query execution efficiency in reduced and unreduced MDW. In our previous work (Atigui, Ravat, Song, Teste, & Zurfluh, 2015; Atigui, Ravat, Song, & Zurfluh, 2014), we have compared some key indicators about query execution efficiency by applying different types of queries to different types of R-OLAP MDW. Several experimental assessments are carried out in DBMS Oracle 11g with the following hardware configuration:

- OS: Red Hat Enterprise Linux Server release 5.9 (Tikanga)
- CPU: 2 x Intel(R) Xeon(R) E5410 @ 2.33GHz with 4 cores
- RAM: 5GB
- Disk: SAS 10K

Each time we take the experimental assessments, we execute two types of query in sixteen unreduced and reduced MDW. Four implementation strategies are proposed, such as a MDW based on one single table, a MDW based on a fact table and different dimension tables, a denormalized MDW implementation and a normalized MDW implementation. For each implementation, we propose four scales factors by varying the number of tuples within non-temporal analysis axis. By consequence, the volume of MDW ranges from 0.9 million to 14 million tuples.

After the experimental assessments, we conclude that whatever the query's type, the implementation strategy and scale factor, queries are always more efficiently calculated within reduced MDW. The reduced MDW allows saving, on average, up to 97% of query execution time comparing to unreduced MDW (cf. figure 24). This can show OLAP analysis is more efficiently processed in reduced MDW at execution level.

Figure 24. Average gain of execution time in reduced MDW.



CONCLUSION

Our objective is to both increase the efficiency of analysis and facilitate the analysts' task. To this end, we propose a solution for supporting analyses over data that are reduced over time.

Firstly, we provide a conceptual MDW model keeping only pertinent data over time. In order to aggregate and remove useless detailed data, data reduction is applied to the whole MDW schema according to decision-makers' needs. We propose a MDW composed of a set of states. Each state consists of a star schema valid for a certain period of time. Secondly, we propose a set of multi-states analysis operators compatible with reduced MDW. Without need for precise knowledge of different states in reduced MDW, a decision-maker can carry out common analysis operations with the proposed operators, such as displaying analysis results, drilling up and down in an analysis axis, changing to a new dimension or hierarchy and restricting analysis results by adding selection predicates. For each operator, we make an algebraic definition, an execution algorithm and an implementation strategy in R-OLAP environment. To the best of our knowledge, this is the first discussion about how OLAP analyses can be carried out in a reduced

MDW. At last, we develop a multi-states analysis framework. By illustrating how OLAP analyses are carried out through a graphical interface, we show the feasibility of proposed concepts. We also show that carrying analyses in reduced MDW is highly efficient, which allows saving up to 97.8% of execution time.

In this paper, we put emphasis on how to carry out OLAP analysis in reduced MDW composed of several states over time. The future extensions can take different forms. At the modeling level, the reduced MDW is currently modeled with star schema, which is sufficient for carrying out multi-states OLAP analyses. Therefore, a minor future extension consists in proposing a constellation model for reduced MDW. At analysis level, we intend to propose some advanced multi-states analysis operators by regrouping several elementary operators. At implementation level, in our previous work (Atigui et al., 2014; Ravat, Song, & Teste, 2015; Atigui et al., 2015), the reduced MDW have been implemented in R-OLAP environment. Thus, one extension consists in evaluating the efficacy and efficiency of data reduction in different environments, such as M-OLAP and H-OLAP.

REFERENCES

- Agrawal, R., Gupta, A., & Sarawagi, S. (1997). Modeling multidimensional databases (pp. 232–243). Presented at the 13th International Conference on Data Engineering, Birmingham: IEEE Comput. Soc. Press.
- Atigui, F., Ravat, F., Song, J., Teste, O., & Zurfluh, G. (2015). Facilitate Effective Decision-Making by Warehousing Reduced Data: Is It Feasible? *International Journal of Decision Support System Technology*, 7(3), 36–64.
- Atigui, F., Ravat, F., Song, J., & Zurfluh, G. (2014). Reducing Multidimensional Data. In L. Bellatreche & M. K. Mohania (Eds.), *Data Warehousing and Knowledge Discovery* (Vol. 8646, pp. 208–220). Cham: Springer International Publishing.
- Body, M., Miquel, M., Bédard, Y., & Tchounikine, A. (2002). A multidimensional and multiversion structure for OLAP applications (pp. 1–6). ACM Press.
- Boly, A., Hebrail, G., Boly, A., & Goutier, S. (2007). Forgetting data intelligently in data warehouses. In *IEEE International Conference on Research, Innovation and Vision for the Future* (pp. 220–227). New York: IEEE.
- Ciferri, C., Ciferri, R., Gómez, L., Schneider, M., Vaisman, A., & Zimányi, E. (2013). Cube Algebra: A Generic User-Centric Model and Query Language for OLAP Cubes. *International Journal of Data Warehousing and Mining*, 9(2), 39–65.

- Demuth, B., & Hussmann, H. (1999). Using UML/OCL Constraints for Relational Database Design. In R. France & B. Rumpe (Eds.), «UML»'99 — *The Unified Modeling Language* (Vol. 1723, pp. 598–613). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Eder, J., Koncilia, C., & Morzy, T. (2002). The COMET Metamodel for Temporal Data Warehouses. In *Advanced Information Systems Engineering* (Vol. 2348, pp. 83–99). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Elmasri, R. (2011). *Fundamentals of database systems* (6th ed). Boston: Addison-Wesley.
- Ghozzi, F., Ravat, F., Teste, O., & Zurfluh, G. (2003). Constraints and multidimensional databases (pp. 104–111). Presented at the 5th International Conference on Enterprise Information Systems - ICEIS'03, Angers, France.
- Golfarelli, M., Maio, D., & Rizzi, S. (1998). Conceptual design of data warehouses from E/R schemes. In *Thirty-First Annual Hawaii International Conference on System Sciences* (Vol. 7, pp. 334–343). Kohala Coast, HI: IEEE Computer Society.
- Golfarelli, M., & Rizzi, S. (2009). A Survey on Temporal Data Warehousing: *International Journal of Data Warehousing and Mining*, 5(1), 1–17.
- Gyssens, M., & Lakshmanan, L. V. S. (1997). A Foundation for Multi-dimensional Databases. In *VLDB '97 Proceedings of the 23rd International Conference on Very Large Data Bases* (pp. 106–115). Athens, Greece.
- Iftikhar, N., & Pedersen, T. B. (2011). A rule-based tool for gradual granular data aggregation. In *International Workshop on Data Warehousing and OLAP* (pp. 1–8). Glasgow, United Kingdom: ACM Press.
- Kimball, R. (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses*. New York: John Wiley & Sons.
- Last, M., & Maimon, O. (2000). Automated dimensionality reduction of data warehouses. In *International Workshop on Design and Management of Data Warehouses* (pp. 7.1–7.10). Stockholm, Sweden: CEUR Workshop Proceedings.

- Maniatis, A., Vassiliadis, P., Skiadopoulou, S., Vassiliou, Y., Mavrogonatos, G., & Michalarias, I. (2005). A Presentation Model & Non-Traditional Visualization for OLAP. *International Journal of Data Warehousing and Mining*, 1(1), 1–36.
- Mitrpanont, J. L., & Fugkeaw, S. (2006). Design and development of a multiversion OLAP application (pp. 493–497). Presented at the Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France: ACM Press.
- Morzy, T., & Wrembel, R. (2004). On querying versions of multiversion data warehouse. In *7th ACM international workshop on Data warehousing and OLAP* (pp. 92–101). Washington, DC, USA: ACM Press.
- O’Neil, P., O’Neil, E., Chen, X., & Revilak, S. (2009). The Star Schema Benchmark and Augmented Fact Table Indexing. In *Performance Evaluation and Benchmarking* (Vol. 5895, pp. 237–252). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Ravat, F., Song, J., & Teste, O. (2015). Analysis Framework for Reduced Data Warehouse (pp. 81–96). Presented at the Journées Francophones sur les Entrepôts de Données et l’Analyse en ligne (EDA 2015), Bruxelles.
- Ravat, F., Teste, O., Tournier, R., & Zurfluh, G. (2008). Algebraic and Graphic Languages for OLAP Manipulations. *International Journal of Data Warehousing and Mining*, 4(1), 17–46.
- Ravat, F., Teste, O., & Zurfluh, G. (2006). A Multiversion-Based Multidimensional Model. In A. M. Tjoa & J. Trujillo (Eds.), *Data Warehousing and Knowledge Discovery* (Vol. 4081, pp. 65–74). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Skyt, J., Jensen, C. S., & Pedersen, T. B. (2008). Specification-based data reduction in dimensional data warehouses. *Information Systems*, 33(1), 36–63.
- Udo, I. J., & Afolabi, B. (2011). Hybrid Data Reduction Technique for Classification of Transaction Data. *Journal of Computer Science and Engineering*, 6(2), 12–16.

- Wang, X. S., Bettini, C., Brodsky, A., & Jajodia, S. (1997). Logical design for temporal databases with multiple granularities. *ACM Transactions on Database Systems*, 22(2), 115–170.
- Wrembel, R. (2009). A Survey of Managing the Evolution of Data Warehouses. *International Journal of Data Warehousing and Mining*, 5(2), 24–56.
- Wrembel, R., & Bębel, B. (2007). Metadata Management in a Multiversion Data Warehouse. In *Journal on Data Semantics VIII* (Vol. 4380, pp. 118–157). Berlin, Heidelberg: Springer Berlin Heidelberg.