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 : 15244

The contribution was presented at EDA 2015:
https://eda2015.ulb.ac.be/

To cite this version : Ravat, Franck and Song, Jiefu and Teste, Olivier *Analysis Framework for Reduced Data Warehouse.* (2015) In: 11e Journées Francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2015), 2 April 2015 - 3 April 2015 (Bruxelles, Belgium).

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

Analysis Framework for Reduced Data Warehouse

Franck Ravat*, Jiefu Song*, Olivier Teste**

* IRIT - Université Toulouse I Capitole, 2 Rue du Doyen Gabriel Marty
F-31042 Toulouse Cedex 09
** IRIT - Université. Toulouse II Jean Jaurès, 1 Place Georges Brassens
F-31703 Blagnac Cedex
***{ravat, song, teste}@irit.fr

Abstract. Our aim is to define a framework supporting analysis in MDW with reductions. Firstly, we describe a modeling solution for reduced MDW. A schema of reduced MDW is composed of states. Each state is defined as a star schema composed of one fact and its related dimensions valid for a certain period of time. Secondly, we present a multi-state analysis framework. Extensions of classical drilldown and rollup operators are defined to support multi-states analyses. Finally we present a prototype of our framework aiming to prove the feasibility of concept. By implementing our extended operators, the prototype automatically generates appropriate SQL queries over metadata and reduced data.

1 Introduction

Nowadays, Multidimensional Data Warehouse (MDW) is a widely used component in decision support systems. A MDW schema is based on facts (analysis subjects) and dimensions (analysis axes). The facts contain analysis indicators while the dimensions organize analysis parameters according to their level on hierarchies: from the minimal (most detailed) granularity to the maximal (most general) granularity.

In a MDW, data are stored permanently and new data is steadily added. As result, a MDW stores a huge volume of data in which the analyst may be lost during her/his analyses. On the other hand, the relevance of MDW data decreases with age: detailed information is generally considered essential for recent data Skyt et al. (2008), while more aggregated information can usually satisfy the need of analysis over older data. For instance, an analyst may have interest in analyzing published news by subthemes for the last four years. However, as most of today's subthemes did not exist before, the subtheme granularity level may be proved useless for an older period. As a result the analyst may have no more interest in analyzing published news by subtheme over the last ten years but by a higher and more stable granularity level, such as news' theme.

Facing large volumes of data among which a great amount of inadequate data are found, our aim is to both increase the efficiency of analysis and facilitate the analysts' task. To this end, we

provide a conceptual MDW model which provides only pertinent data over time. Meanwhile we also develop a framework compatible with our conceptual MDW model in order to provide powerful analysis tools to decision-makers.

– The MDW model with reduction increases analysis efficiency by allowing analysts to remove useless temporal granularity levels according to their needs. As detailed information loses its value over time, we intend to implement selective deletion at low granularity levels.
– The compatible framework permits to manage reduced data and model analysis process by including analysis operators and a graphical interface.

This paper is composed of the following sections. Section 2 describes a state of the art of data reduction. Section 3 presents preliminary concepts of reduced MDW with the help of a case study. Section 4 describes our analysis framework compatible with reduced MDW by emphasizing modeling principles of metamodel. Section 5 focuses on extended analysis operators. Section 6 presents a prototype showing the feasibility of concepts.

2 Related work

Reducing data allows us to both decrease the quantity of irrelevant data in decision making and increase future analysis quality Udo and Afolabi (2011) . In the context of decision support, data reduction is a technique originally used in the field of data mining Okun and Priisalu (2007)Udo and Afolabi (2011).

In the data warehouse context, Garcia-Molina et al. (1998) were the first to propose solutions for data deletion. They study data expiration in materialized views so that they are not affected but maintained after updates with the help of a set of standard predefined views. No discussion about carrying out analyses in data warehouse after data expiration can be found in this work.

In the multidimensional approaches, Chen et al. (2002) propose an architecture allowing the integration of data streams into a MDW by reduce their size. The size reducing process is predefined and automatically executed by partially aggregating the data cube; it makes sure the detailed information is only available for a certain period of time. But this work only focuses on the fact table, and no analysis support component is included in proposed architecture. Skyt et al. (2008) present a technique for progressive data aggregation of a fact. This study intends to specify data aggregation criteria of a fact due to higher levels of dimensions. Although the authors propose techniques to query reduced multidimensional data, they fail to provide metamodel permitting to manage data reduction in MDW. Kimball and Ross (2011) define the concept of Slowly Changing Dimensions (SCDs) and indicate that data may change within a dimension even though it occurs less frequently than in a fact. They propose three basic modeling solutions for managing dimensional data changes, namely overwrite old data, create new record for each change and keep data changes as alternative values in MDW. Golfarelli and Rizzi (2009) point out that not only data but also MDW schema can change over time according to user's requirements. SCD, however, does not provide solution for handling schema changes and it does not take user's need into account.

In Iftikhar and Pedersen (2011), a gradual data aggregation solution based on conception, implementation and evaluation is proposed. This solution is based on a table containing different temporal granularities: second, minute, hour, month and year. Unfortunately this work does not discuss possibilities of carrying out analysis over gradual aggregated data.

The previous works only focuses on reduction of fact table or data changes within dimensions. Iftikhar and Pedersen (2010) and Iftikhar and Pedersen (2011) use a temporal table for gradual data reduction. Analysts' needs are ignored in data reduction process. None of the previous work fully support carrying out analysis in reduced MDW. Facing to these issues, our goal is more ambitious as it aims to meet several challenges mentioned in Golfarelli and Rizzi (2009):

– We propose a solution for handling the complete MDW schema changes by generalizing the mechanism of reduction. In consequence all of the dimensions as well as the fact are susceptible to sustain reductions to different granularity levels so as to fully satisfy analyst's needs. The information judged useless is aggregated and then deleted from MDW in order to provide only necessary data for analysis.
– We face issues of carrying out analysis in MDW whose schema changes over time. We provide decision-makers with an analysis-support framework applicable to MDW with reduction. The framework contains allows managing both reduced MDW and analysis process.

3 Preliminary concepts of reduced MDW

We will firstly describe a case study of data reduction in MDW that fulfills decision-makers' needs. This case study intends to give a first glance at selective deletion of data in MDW as well as data model notations. Then we define a formal presentation for reduced MDW. At the end of the section we describe a set of reduction operators permitting to define reduced MDW.

3.1 Case study

This case study shows a complete multidimensional schema progression that fulfills the analyst's needs. A MDW populated by the RSS streams allows decision-makers to analyze the number of published news from her/his favorite websites. Containing over ten million tuples in the fact table, this MDW face the first important problem of performance while a decision-maker carries out analyses. Moreover, most of the old detailed data become obsolete nowadays, they are no more used in analysis process and should be deleted according to user' needs. More precisely, a decision-maker expresses her/his needs as followed: (a) during the last four years, news analysis is carried out with reference to lowest levels of granularity (subtheme, city and publication date); (b) in the previous period from 2000 to 2010, analyses are summarized according to news' theme, country mentioned in the news and month of publication because no daily analysis referring to subtheme and city is required; (c) before 2000, only aggregated information about published news by quarter and by continent makes sense.

The following three figures represent the conceptual multidimensional schemas fulfilling user's needs. Each schema is based on star schemas introduced in Golfarelli et al. (1998). A star

schema is based on a subject of analysis (fact) related to different dimensions. Each fact is composed of one or more indicators. For instance, in figure 1 the fact named "*FNews*" contains one indicator: number of published news (*NBN*). A dimension models an analysis axis; it represents information according to which subjects of analysis are to be dealt with. For instance, the "*FNews*" fact is associated to 3 dimensions: *DTheme*, *DGeography* and *DTimes*. Dimension attributes (also called parameters or levels) are organized according to one or more hierarchies such as *HTHM* on dimension *DTheme*.

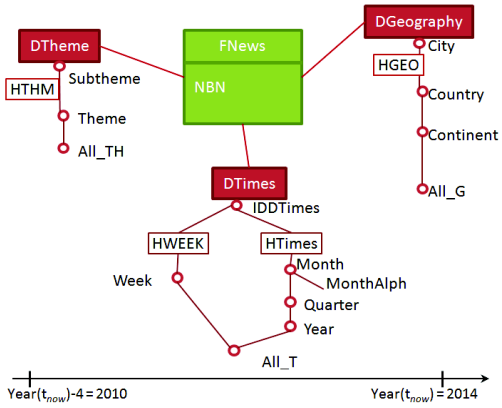

FIG. 1: MDW schema valid from 2010 to 2014

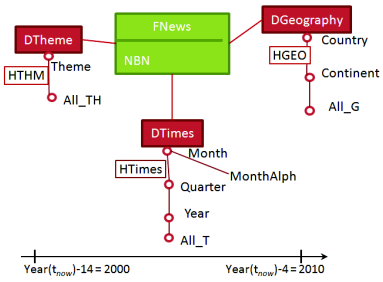

FIG. 2: MDW schema valid from 2000 to 2010

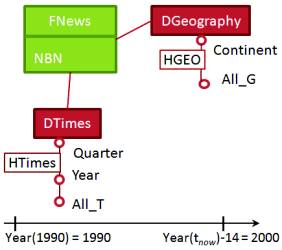

FIG. 3: MDW schema valid from 1990 to 2000

3.2 Concepts

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.

– A MDW is defined by $S = (n^S; \mathcal{E}; \mathcal{M}ap)$, where $n^S \in N$ is the name of the MDW; $\mathcal{E} = E_1; ...; E_n$ is a set of states composing the MDW; Reduction function $\mathcal{M}ap : \mathcal{E} \rightarrow \mathcal{E} | \mathcal{M}ap(E_k) = E_{k+1}$ defines the state named E_{k+1} obtained by the reduction of E_k.
– Each state $Ei = (F_i; \mathcal{D}_i; T_i)$ is a star schema defined for a temporal period, where $F_i \in F$ is a *fact* representing a subject of analysis; $\mathcal{D}_i = \{D_{times}; D_1; ...; D_m\} \subseteq D$ is a set of *dimensions* associated to the fact with necessarily a temporal dimension denoted D_{times}. To define the T_i temporal interval, we adopt a linear and discrete time model approaching time in granular way through time observation units Wang et al. (1997). A temporal interval T_k of state E_k is defined by a couple of instants. These instants can be fixed (temporal grains such as the year of *1990*) or dynamic (defined with the instant t_{now}).
– A *fact* denoted F_i is defined by $F_i = (n^{Fi}, M^{Fi})$, where n^{Fi} is the fact name; M^{Fi} is a set of *measures* or indicators, $\forall i, j | i < j \rightarrow M_j \subseteq M_i$. A *dimension* denoted D_i is defined by (n^{Di}, A^{Di}, H^{Di}), where n^{Di} is the dimension name; A^{Di} is the set of the *attributes* of the dimension; H^{Di} is a set of *hierarchies*, $\forall i, j | i < j \rightarrow D_j \subseteq D_i \wedge H_j \subseteq H_i \wedge A_j \subseteq A_i$.
– A *hierarchy*, denoted H_j (abusive notation of $H_j^{D_i}$) is defined by $(n^{Hj}, P^{Hj}, \prec^{Hj}, Weak^{Hj})$, where n^{Hj} is the hierarchy name; P^{Hj} is a set of attributes called *parameters*; $\prec^{Hj}$ is an antisymmetric and transitive binary relation between parameters; $Weak^{Hj}$ is an application that associates to each parameter a set of dimension attributes, called *weak attributes*. Hierarchies organize the attributes of a dimension, from the finest graduation (root parameter denoted ID_{Di}) to the most general graduation (extremity parameter denoted ALL_{Di}). Thus, a hierarchy defines the valid navigation paths on an analysis axis.
– Analysis results are presented in forms of *multidimensional table*, denoted MT_i, $\forall i \in [1..n]$, which is defined by $(S^i; Ax^i; R^i; I^i)$, where $S^i \in F$ is the analysis subject; $Ax^i = \{ax_1^i, ax_2^i, [, ax_3^i [, ax_4^i, \ldots]]\}$ is the set of analysis axes currently presented, among which ax_1^i and ax_2^i are the horizontal and vertical displayed axes respectively; $R^i =< pred_1, pred_2, \ldots >$ is a set of selection predicates filtering displayed analysis results; $I^i \subseteq D_{TIMES}$ is a set of temporal interval representing the validation period of MT_i.

4 Multi-states analysis framework

Data reduction brings changes to not only MDW's schema but also its instances. In regard to schema changes, a MDW is no more modeled as single static schema but a set of schemas (states) over time. As for instance changes, a reduced MDW no longer contains all detailed information over time. Judged useless by analysts, certain detailed information in recent states would be aggregated and then deleted. As consequence, dimensions and facts in recent and older states certainly have different instances, as they contain different attributes and measures respectively.

Due to the schema and instances changes, existing metamodel for MDW as well as analysis operators is no more applicable to reduced MDW. In order to manage different states in MDW and provide decision-makers with efficient analysis tools, we propose an analysis framework compatible with reduced MDW. In this section, we first present the architecture of our framework and the roles of its mains components. Then we detail *Data Management* parts along

with a deeper discussion about how reduced MDW can be managed through our framework. Details about *Analysis Engine* can be found in section 5.

4.1 Architecture of multi-states analysis framework

The multi-states analysis framework is composed of three components, namely *Data Management*, *Analysis Engine* and *Interactive Restitution* (cf. figure 4). Each part has specific roles and interacts with others.

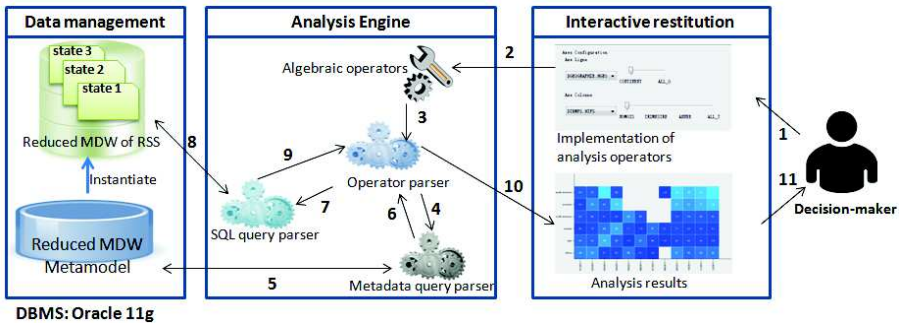

FIG. 4: Main components of multi-states analysis frameworks

– The *Data Management* part accommodates a metamodel and a set of reduced MDW. The metamodel allows managing reduced MDW, new MDW can be defined by instantiating the metamodel.
– The *Analysis Engine* part 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 to tools and implementation languages. More sophisticated analysis operations can be realized via composition of algebraic operators.
  – The *operator parser* (a) translates operators in algebraic form into queries over metadata, (b) analyzes metadata queries' results and generates corresponding SQL queries which interrogate concerned MDW and its states, (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* part contains (a) a graphical implementation of analysis operators in order to facilitate decision-makers' tasks and (b) a graphical interface.

This multi-states analysis framework guaranties the transparency of data reduction in MDW. Decision-makers carry out analysis via graphical implementation of algebraic operators (arrow tagged "1" in figure 4) and then receive a global analysis result (arrow tagged "11" in figure

4). No knowledge about schema and instance evolution of MDW is required for effectuating analysis via our framework. This is thanks to the *Data Management* and *Analysis Engine* parts of our framework which adapt themselves to users' needs.

4.2 Data management

The *Data Management* part of framework permits to (a) manage MDW composed of one or several states and (b) define new reduced MDW. As we can see from the figure 5, the metamodel embodies all proposed concepts. The graphical notation of our conceptual metamodel is based on UML class diagram.

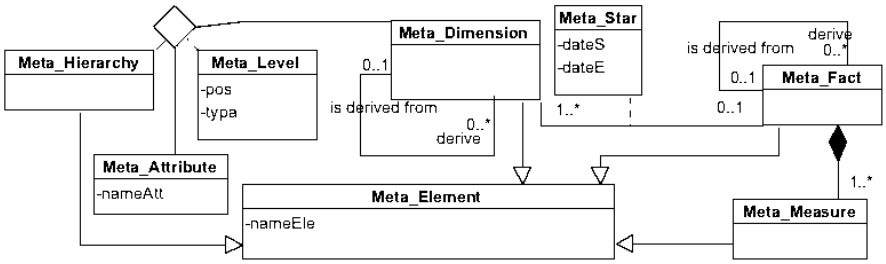

FIG. 5: UML class diagram metamodel of the reduced MDW

Firstly, as all facts, dimensions and hierarchies have a name, the name of elements is centralized and managed by the class notated *Meta_Element* which is the base of our metamodel. All the rest is considered as specialized classes of *Meta_Element*. Secondly the notion of state is represented by the association class *Meta_star*. This association class possesses a temporal interval between a start date and an end date. Thirdly, the fact and the dimension are embodied respectively by the classes *Meta_Fact* and *Meta_Dimension*. Each of these classes possesses a recursive association denoted *Derive* pointing to itself. Fourthly, by definition a fact contains a set of measures while a measure belongs to one and only one fact. This rule is expressed by the relationship notated *Contain* between the class *Meta_mesure* and the class *Meta_fact*. The ternary association in our metamodel permits to organize attributes according to their level on a hierarchy of a dimension, which corresponds to concept hierarchy $H_j = (n^{Hj}, P^{Hj}, \prec^{Hj}, Weak^{Hj})$.

5 Analysis

One of the core components of our framework is *Analysis Engine*. In this section we present more details about multi-states analysis processing steps. Detailed studies about emblematic and the most used operators (drilldown and rollup) are also effectuated. To explain how elementary analysis operations are carried out, we present execution algorithms for algebraic operators.

5.1 Analysis processing

To facilitate the decision-maker's tasks, she/he only interactively selects the *MT* components: the displayed measures of a fact, the displayed attributes of dimensions as well as the temporal interval of the analysis. The system converts the selection into corresponding analysis operators. All operators supporting multi-states analysis are processed according to the five steps showed in figure 6. More precisely, through a set of temporal intervals chosen by decision-makers, the analysis operator firstly determines the states in which analysis is carried out before adjusting input parameters if necessary. Then it splits up into several classical operators, each one of them is applied to a single concerned state along with input parameters eventually adjusted. Next each partial operator is translated and executed independently to get partial results. At last all partial results are combined in order to return a unique global result to decision-makers.

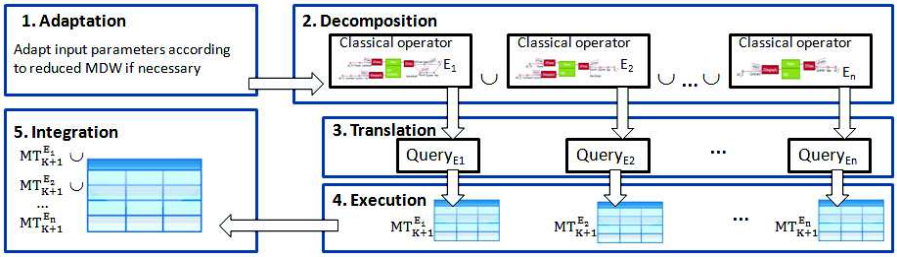

FIG. 6: Analysis processing steps of multi-states analysis operators

Our proposition of analysis processing differs from the one in Morzy and Wrembel (2004). Firstly, Morzy and Wrembel (2004) involves only issues at querying level, no discussion about analysis operators can be found. Secondly, decision-makers should provide the exact set of schemas in which analysis is carried out, which requires her/him to have a profound knowledge about schema evolution in MDW. At last, analysis results are possible to be presented separately, which makes the interaction with data more complex.

Conversely, our proposition of analysis processing has multiple advantages. First of all, it depends on formally defined analysis operators which are extensions of classical operators. Existing analysis tools and languages are fully reusable to the extended operator. Secondly, data reduction in MDW is entirely transparent to decision-makers. While carrying out analysis, decision-makers do not necessarily need to know schema evolutions in MDW. The multi-states operators systematically search concerned states and adapt themselves if necessary. Moreover, multi-states analysis operators provide a single analysis result even if information may come from several states. No information about states in MDW is presented to decision-makers, which reinforces the transparency of data reduction.

5.2 Extended analysis operators

Ravat et al. (2008) define a set of generic algebraic analysis operators based on MDW with one

stable schema over time. Even though these analysis operators sustain flexible and adaptable to different classical modeling solutions (star and constellation models), they become incompatible with reduced MDW. Firstly, classical analysis operators are conceived to manipulate one unique schema rather than a set of schemas evolving over time. Secondly, classical analysis operators do not provide solutions for handling heterogeneities of instances in different states. Conceived to modify analysis precision, the classical drilldown and rollup operators are highly affected by data reduction in MDW. An analysis involving an inexistent parameter, for instance, is theoretically impossible according to classical analysis operators.

Facing to these issues, we propose two extended analysis operators: $Drilldown^{multi-states}$ and $Rollup^{multi-states}$, in order to support multi-states analyses in reduced MDW. They both take a set of *temporal intervals*, a *multidimensional table* currently displayed, an analysis *dimension* and a *parameter* as input. A new *multidimensional table* is produced as output containing information at lower or higher granularity level. As data reduction is completely transparent, decision-makers have no information about which dimension D_i in state E_i to choose to carry out analyses. For the sake of simplicity, we allow decision-maker to simply specify the name of dimension n^{Di} in which analyses are carried out. The analysis operators adapt themselves by finding the corresponding dimensions D_i in state E_i while executing. In order to distinguish dimensions D_i in state E_i (cf. section 3.2) from dimension chosen by decision-makers, in the following paper we add "$_E_i$" as suffix to dimension in state E_i, while the name of dimension chosen by decision-makers is without suffix.

5.2.1 $Drilldown^{multi-states}$ operator

$Drilldown^{multi-states}$ ([I] ; MT_k, D_i ; P_{inf})= MT_{K+1} ;	
Input	- $I \subseteq D_{TIMES}$: a set of optional temporal intervals on dimension D_{TIMES} - MT_k: multidimensional table currently displayed - D_i : analysis axis currently displayed - P_{inf} : chosen parameter on dimension D_i
Output	- $MT_{K+1} = (S^{K+1}; Ax^{K+1}; R^{K+1}; I^{k+1})$ analysis result MT such as - Analysis subject $S^{K+1} = S^K$ - Analysis axes$Ax^{K+1} \subseteq D, \forall ax_j^{K+1} \in Ax^{K+1} \vert ax_j^{K+1} = (D_j^{K+1}, H_j^{K+1}, P_j^{K+1})$ such as - Dimension $D_j^{K+1} = D_j^K$, - Hierarchy $H_j^{K+1} = H_j^K$, - Parameter $P_j^{K+1} \subseteq P^(H_j^{K+1}) \vert (D_j^{K+1} = D_i \rightarrow P_j^{K+1} =< P_{inf} > \vee < P_\lambda{}^1 >) \wedge (D_j^{K+1} \neq D_i \rightarrow P_j^{K+1} =< P_1^{D_j^K}, P_2^{D_j^K}, \dots >)$ - Selection predicates on dimensions and/or fact $R^{K+1} = R^K$ - Validation period $I^{k+1} = I \vee I^{k+1} = I^k$

1. Adjusted parameter $P_\lambda \in P^{H_j^{K+1}} \vert P_{inf} \prec^{H_j^K} P_\lambda$

TAB. 1: Algebraic multi-states drilldown operator

The$Drilldown^{multi-states}$ operator permits to display information at a finer granularity level on currently displayed dimension in several states. As certain parameters of low granularity exist only in most recent states but not in former ones, it is very likely that some input parameters are not present in all states involved in analysis, especially for $Drilldown^{multi-states}$ operator. For example, from 2000 to 2014, hierarchy *HGeo* on dimensions *DGeography_E1* and *DGeography_E2* can sustain a drilldown operation to *Continent* and *Country* levels. But a classical drilldown until to *City* level is impossible because the parameter *City* exists no longer in "E2" state after reduction. To handle this issue, we propose to augment parameter's granularity level until finding the first common parameter among all involved states(cf. table 1). The execution algorithm of Drilldownmulti-states is as follows.

Algorithm 1: $Drilldown^{multi-states}$ ([*I*] ; MT_k, D_i ; P_{inf})
Input: Set of temporal intervals *I*, displayed multidimensional table MT_k, displayed dimension *D*, parameter *P*. Output: new multidimensional table MT_{k+1}

Let H_{actual} be the actually displayed hierarchy
Let P_{actual} be the actually displayed parameter
If $P_{actual} \prec^{H_{actual}} P \vee ALL_D \prec^{H_{actual}} P$then
  Impossible operation
Else
  Find the subset of states $E_i \rightarrow \forall E_j \in E_i | I_{E_j} \in I \vee I_{MT_k}$ –Adaptation
  Let $P_{Drilldown} = P$
  Let $r = FALSE$
  While $ALL_D \prec^{H_{actual}} P_{Drilldown} \wedge r = FALSE$
    If $\forall E_j \in E_i | P_{Drilldown} \in A^D_{E_j}$ then
      $r = TRUE$
    Else
      $P_{Drilldown}$ increases one granularity level
    End if
  End While
  If $r = FALSE$ then
    Impossible operation
  Else
    For $E_j \in E_i$ – Decomposition
      Let $MT_K^{E_j}$ be the part of MT in states E_j
      Translate $Drilldown(MT_K^{E_j}, D, P_{Drilldown})$into query Q – Translation
      $MT_{K+1}^{E_j}$= Result of query Q – Execution
      $MT_{K+1} = MT_{K+1} \bigcup MT_{K+1}^{E_j}$ – Integration
    End for
  End if
End if

For instance, if a decision-maker carries out an analysis to *City* level in "E1" and "E2" states, the initial $Drilldown^{multi-states}$ ([2000,2014], MT_k, *DGeography*, *City*) operator will be handled as follows: (a) the first step *Adaptation* replaces inexistent parameter by the nearest superior parameter. In this case, *City* is replaced by *Country* which is the nearest superior

common parameter on hierarchy *HGEO* of dimensions *DGeography_E1* and *DGeography_E2*. We obtain by consequence a new operator with an adjusted parameter $Drilldown^{multi-states}$ ([2000, 2014], MT_k, *DGeography*, *Country*); (b) the second step *Decomposition* decomposes the adjusted operator into several classical drilldown operators. Each one of them is then applied to a cube containing one single state deducted from the set of intervals; (c) next we find the classical *Translation* phase which transforms each drilldown operator into an independent SQL query; (d) the following phase *Execution* executes independently each SQL query to obtain a set of partial analysis results in forms of cube; (e) at last Integration phase gathers all individual results obtained from each state in order to form one single analysis result.

5.2.2 $Rollup^{multi-states}$ operator

$Rollup^{multi-states}$ ([I] ; MT_k, D_i ; P_{inf})= MT_{K+1} ;	
Input	- $I \subseteq D_{TIMES}$: a set of optional temporal intervals on dimension D_{TIMES} - MT_k: multidimensional table currently displayed - D_i : analysis axis currently displayed - P_{sup} : chosen parameter on dimension D_i
Output	- $MT_{K+1} = (S^{K+1}; Ax^{K+1}; R^{K+1}; I^{k+1})$ analysis result MT such as - Analysis subject $S^{K+1} = S^K$ - Analysis axes $Ax^{K+1} \subseteq D, \forall ax_j^{K+1} \in Ax^{K+1} \mid ax_j^{K+1} = (D_j^{K+1}, H_j^{K+1}, P_j^{K+1})$ such as - Dimension $D_j^{K+1} = D_j^K$, - Hierarchy $H_j^{K+1} = H_j^K$, - Parameter $P_j^{K+1} \subseteq P(H_j^{K+1}) \mid (D_j^{K+1} = D_i \rightarrow P_j^{K+1} =< P_{sup} >) \wedge (D_j^{K+1} \neq D_i \rightarrow P_j^{K+1} =< P_1^{D_j^K}, P_2^{D_j^K}, \dots >)$ - Selection predicates on dimensions and/or fact $R^{K+1} = R^K$ - Validation period $I^{k+1} = I \vee I^{k+1} = I^k$

TAB. 2: Algebraic multi-states rollup 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 2).

$Rollup^{multi-states}$ operator reveals data at a higher granularity level. As currently displayed granularity levels exist in all concerned states, chosen parameter denoted P_{sup} presents no doubt in all concerned states. For example, if the decision-makers wants to roll up analysis level from *Country* to *Continent* after the previous analysis, she/he can simply carry out the following operator: $Rollup^{multi-states}$ ([2000, 2014], MT_{k+1}, *DGeography*, *Continent*). No special treatment is needed in Adaptation step (cf. algorithm 2).

Algorithm 2: $Rollup^{multi-states}$ ($[I]$; MT_k, D_i ; P_{inf})
Input: Set of temporal intervals *I*, displayed multidimensional table MT_k, displayed dimension *D*, parameter *P*. Output: new multidimensional table MT_{k+1}

1 Let H_{actual} be the actually displayed hierarchy
2 Let P_{actual} be the actually displayed parameter
3 If $P \prec^{H_{actual}} P_{actual}$then
4 Impossible operation
5 Else
6 Find the subset of states $E_i \rightarrow \forall E_j \in E_i | I_{E_j} \in I \vee I_{MT_k}$
7 For $E_j \in E_i$ – Decomposition
8 Let $MT_K^{E_j}$ be the part of MT in states E_j
9 Translate $Rollup(MT_K^{E_j}, D, P)$into query Q – Translation
10 $MT_{K+1}^{E_j}$= Result of query Q – Execution
11 $MT_{K+1} = MT_{K+1} \bigcup MT_{K+1}^{E_j}$ – Integration
12 End for
13 End if

6 Implementation

Based on our proposed framework, we develop a prototype in Java. The prototype implements the reduced MDW of our case study. As our previous work Atigui et al. (2014) shows that queries are more efficiently computed within a reduced *Star Schema* in DBMS Oracle 11g, we chose to implement our case study with the R-OLAP schema presented in figure 7. As a first step, the aim of our prototype is to demonstrate the feasibility of carrying out multi-states analysis in reduced MDW. Issues about execution efficiency of multi-states analysis operators will be included in our future work. Moreover, we choose to implement in the first place the reduced MDW in a R-OLAP environment, because R-OLAP suffers from poorer query execution performance than M-OLAP and H-OLAP Vassiliadis and Sellis (1999), especially in an era of massively abundant data. Thus, one of the main focuses of our future work is to evaluate analysis efficiency in other OLAP environments, such as M-OLAP and H-OLAP.

Now we will illustrate how multi-states analysis operators are transformed into SQL queries over metadata and reduced data. A decision-maker is analyzing published news in the world by month from 2000 to 2014 (states "E1" and "E2" in reduced MDW). Wishing to consult the number of published news by month and by continents to get more detailed information, she/he slides the cursor on line to *Continent* level. The framework detects it concerns a multi-states drilldown operation from ALL_G to *Continent* on dimensions *DGeography_E1* and *DGeography_E2*. After executing a $Drilldown^{multi-states}$ operator, two queries are generated by *Analysis Engine* of framework.

– By translating algebraic $Drilldown^{multi-states}$ operator, the prototype generates firstly the query applicable to metamodel (cf. left part in figure 8). It interrogates the metamodel in order to find states in which the drilldown operation is carried out and verify if the chosen parameter is available in concerned states. In this case parameter *Continent* presents in both "E1" and "E2" states.

E1

Fnews(NBN, KeyWords, IDDTheme#, IDDTimes#, IDDGeography#)
DGeography(IDDGeography, City, Country, Continent)
DTheme(IDDTheme, Subtheme, Theme)
DTimes(IDDTime, Month, NameMonth, Quarter, Week, Year)

E2

Fnews(NBN, Theme#, IDDTimes#, Country#)
DGeography(Country, Continent)
DTheme(Theme)
DTimes(Month, NameMonth, Quarter, Year)

E3

FNews(NBN, Continent#, Quarter#)
DGeography(Continent)
DTimes(Quarter, Year)

FIG. 7: R-OLAP schema of case study

– After receiving the results of the previous query , the prototype transforms the multi-states drilldown operator into several classical drilldown operator. Each one of classical drilldown is then translated into a query based on a single state. At last the prototype combines partial queries with SQL clause *UNION* (cf. right part in figure 8). In this way the prototype returns one global result for each analysis. Moreover, as each partial query could be run individually without effect on other queries' results, our approach facilitates parallel computing of partial queries in parallel DBMS environment (for instance Oracle Parallel, Oracle Distributed, etc.).

```
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.NAME = F.NAME
    AND SC.NAME = SL.NAME
    AND F.NAME_P = 'FNEWS'
    AND DL.NAME = SL.DIMENSION_NAME
    AND DC.NAME = SC.DIMENSION_NAME
    AND DL.NAME_P = 'DGEOGRAPHIE'
    AND DC.NAME_P = 'DTEMPS'
    AND SL.DATES <= TO_DATE
        ('30-12-2014', 'DD-MM-YYYY')
    AND SL.DATEE >= TO_DATE
        ('01-01-2000', 'DD-MM-YYYY')
```

```
(SELECT SUM(NBN) AS FNEWS, DIML.COUNTRY, DIMC.NUMMONTH
FROM FNEWS_E1 FAIT, DGEOGRAPHY_E1 DIML, DTIMES_E1 DIMC, DTIMES_E1 DIMT
WHERE FAIT.ID_DGEOGRAPHY_E1 = DIML.ID_DGEOGRAPHY_E1
    AND FAIT.ID_DTIMES_E1 = DIMT.ID_DTIMES_E1
    AND TO_DATE(DIMT.NUMMONTH, 'MM-RRRR')
        BETWEEN TO_DATE('01-01-2000', 'DD/MM/RRRR')
        AND TO_DATE('30-12-2014', 'DD/MM/RRRR')
    AND FAIT.ID_DTIMES_E1 = DIMC.ID_DTIMES_E1
GROUP BY DIML.COUNTRY, DIMC.NUMMONTH  )
UNION
(SELECT SUM(NBN) AS FNEWS, DIML.COUNTRY, DIMC.NUMMONTH
FROM FNEWS_E2 FAIT, DGEOGRAPHY_E2 DIML, DTIMES_E2 DIMC, DTIMES_E2 DIMT
WHERE FAIT.ID_DGEOGRAPHY_E2 = DIML.ID_DGEOGRAPHY_E2
    AND FAIT.ID_DTIMES_E2 = DIMT.ID_DTIMES_E2
    AND TO_DATE(DIMT.NUMMONTH, 'MM-RRRR')
        BETWEEN TO_DATE('01-01-2000', 'DD/MM/RRRR')
        AND TO_DATE('30-12-2014', 'DD/MM/RRRR')
    AND FAIT.ID_DTIMES_E2 = DIMC.ID_DTIMES_E2
GROUP BY DIML.COUNTRY, DIMC.NUMMONTH )
```

FIG. 8: Queries over metadata generated by prototype

7 Conclusion

This paper resides within the field of MDW. Our first objective is to specify reduced multidimensional schema over time in order to store only the useful data for decision support according to the needs of analysts. The second objective is to provide a framework allowing analysts to carry out multi-states analysis in reduced MDW.

Firstly, we define a MDW model which allows us to specify MDW schemata composed of a set of states varying over time. Each state consists of a star schema valid for a certain period of time. Secondly, we present a generic framework permits to manage both reduced and unreduced MDW through a metamodel and support multi-states analysis with the help of extended analysis operators. By instantiating the metamodel, we explain how reduced MDW can be managed by our framework. As regard to multi-states analysis operators, we present an algebraic form followed by an execution algorithm in order to show how analysis results are produced. One of the main advantages of our analysis framework is the transparency of data reduction: decision-makers can freely carry out analysis without the need for knowing schema evolution in MDW. Finally, we implement our multi-states analysis framework in order to show the feasibility of proposed concepts. We show how multi-states analysis operators are executed by giving concrete examples of automatically generated SQL queries.

In the future, we intend to integrate more analysis operators in a short term, such as Dice, Slice, etc. Moreover, even though our previous work has shown that queries were more efficiently computed in reduced R-OLAP MDW Atigui et al. (2014), the efficiency of multi-states analysis in M-OLAP and H-OLAP environments still remains to be evaluated. Our long term goals are to study influence of data reduction over pre-aggregated data.

References

Atigui, F., F. Ravat, J. Song, and G. Zurfluh (2014). Reducing multidimensional data. In *Data Warehousing and Knowledge Discovery*, pp. 208–220. Springer.

Chen, Y., G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang (2002). Olaping stream data: Is it feasible? In *Proc. Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM SIGMOD*, pp. 53–58. Citeseer.

Garcia-Molina, H., W. Labio, and J. Yang (1998). Expiring Data in a Warehouse. In *Proceedings of 24th International Conference on Very Large Data Bases, VLDB*, New York City, New York, USA, pp. 500–511.

Golfarelli, M., D. Maio, and S. Rizzi (1998). Conceptual design of data warehouses from e/r schemes. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, Volume 7, pp. 334–343. IEEE.
en

Golfarelli, M. and S. Rizzi (2009). A survey on temporal data warehousing. *International Journal of Data Warehousing and Mining 5*(1), 1–17.

Iftikhar, N. and T. B. Pedersen (2010). Using a Time Granularity Table for Gradual Granular Data Aggregation. In *Proceedings of the 14th East European Conference Advances in Databases and Information Systems, ADBIS*, Novi Sad, Serbia, pp. 219–233.

Iftikhar, N. and T. B. Pedersen (2011). A rule-based tool for gradual granular data aggregation. In *Proceedings of the 14th International Workshop on Data Warehousing and OLAP, DOLAP*, Glasgow, United Kingdom, pp. 1–8.

Kimball, R. and M. Ross (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.

Morzy, T. and R. Wrembel (2004). On querying versions of multiversion data warehouse. In *Proceedings of the 7th ACM international workshop on Data warehousing and OLAP*, pp. 92–101. ACM.

Okun, O. and H. Priisalu (2007). Unsupervised data reduction. *Signal Processing 87*(9), 2260–2267.

Ravat, F., O. Teste, R. Tournier, and G. Zurfluh (2008). Algebraic and graphic languages for olap manipulations. *International Journal of Data Warehousing and Mining, IJDWM 4*(1), 17–46.

Skyt, J., C. S. Jensen, and T. B. Pedersen (2008). Specification-based data reduction in dimensional data warehouses. *Information Systems Journal 33*(1), 36–63.

Udo, Ifiok, J. and B. Afolabi (2011). Hybrid Data Reduction Technique for Classification of Transaction Data. *Journal of Computer Science and Engineering 6*(2), 12–16.

Vassiliadis, P. and T. Sellis (1999). A survey of logical models for olap databases. *ACM Sigmod Record 28*(4), 64–69.

Wang, X. S., C. Bettini, A. Brodsky, and S. Jajodia (1997). Logical design for temporal databases with multiple granularities. *ACM Transactions on Database Systems (TODS) 22*(2), 115–170.

Résumé

Notre objectif est de définir un environnement permettant d'effectuer des analyses décisionnelles dans un entrepôt de données multidimensionnel (EDM) réduit. Dans un premier temps, nous proposons une modélisation pour EDM réduit. Un schéma d'EDM réduit est composé de plusieurs états. Chaque état est défini comme un schéma en étoile composé d'un fait et de ses dimensions valables pour une période déterminée. Dans un deuxième temps, nous présentons un environnement d'analyse multi-états. Des extensions des opérateurs classiques drilldown et rollup sont définies pour les analyses multi-états. Enfin, nous présentons un prototype afin de prouver la faisabilité du concept. En implantant nos opérateurs étendus, le prototype génère automatiquement des requêtes SQL adéquates sur des méta-données et des données réduites.