MULTIDIMENSIONAL SCHEMA EVOLUTION
Integrating New OLAP Requirements

Mohamed Neji, Ahlem Nabli, Jamel Feki, Faiez Gargouri
Laboratory MIRACL, ISIM Institute, BP 1030-3018, Sfax. Tunisie

Keywords: Data warehouse, Data mart, Schema evolution, OLAP requirement.

Abstract: Multidimensional databases are an effective support for OLAP processes. They improve the enterprise decision-making. These databases evolve with the decision maker requirements evolution and, are sensitive to data source changes. In this paper, we are interested in the evolution of the data mart schema due to the raise of new OLAP needs. Our approach determines first, what functional data marts will be able to cover a new requirement, if any, and secondly, decides on a strategy of integration. This leads either to the alteration of an existing data mart schema or, to the creation of a new schema suitable for the new requirement.

1 INTRODUCTION

Decisional systems based on data warehouses (DWs). In a previous work, we proposed a top down DW design approach where requirements are expressed as two-dimensional sheets. This approach generates data mart (DM) schemes (Feki, 2004), (Nabli and al., 2005) and (Soussi and al., 2004).
However, these requirements evolve and may require additional data.

Recently, literature has brought forward the problem of evolutions in the multidimensional structures and, new models have been proposed. The updating models (Blascka and al., 1999), focus on mapping data into the most recent version of the structure, whereas tracking history models (Bliujute and al., 1998), (Chamoni and al., 1999), (Eder and al., 2001), (Mendelzon and al., 2000) and (Pedersen, 2001) keep trace of the evolution of the system. The approach in (Chamoni and al., 1999) develops a multidimensional temporal model.
The model of (Eder and al, 2001) proposes mapping functions that allow conversions between structure versions. It provides a partial solution, which neither takes schema evolution and time consistent presentation into account, nor considers complex dimension structures.

In (Pedersen and al., 2001), the authors propose a conceptual model focusing on imprecision and complex dimension structures. However, their model does not provide the means to reporting data in any other versions than the latest.

In this context of study, (Body and al., 2003) present a model with validity periods and a multiversion concept. They distinguished between schema evolution and dimension instance evolution. This work presents a list of operations for schema changes and a set of operations for the instance dimension changes. Similarly, we consider two levels of evolution; the intention level (schema) and the extension level (data).

In particular, we are interested with DM schema evolution due to the emergence of new OLAP requirements. To do so, we develop two main steps: one comparison step, it is to identify which DM schema may be altered, and one adaptation step to make the necessary alterations on the DM schema.

In the remainder, section 2 will present the multidimensional concepts, our notation and describes the structure of OLAP requirements. Section 3 describes our approach of MS evolution. Section 4 outlines the proposed method and sets future works.

2 MULTIDIMENSIONAL CONCEPTS

Fact
Each fact reflects the information of the subject that has to be analyzed.

Definition. A **fact** *F* is defined as (*fname*, *Mf*) where:
- *fname* is the name of a fact,

Neji M., Nabli A., Feki J. and Gargouri F. (2006).
MULTIDIMENSIONAL SCHEMA EVOLUTION - Integrating New OLAP Requirements.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - DISI*, pages 331-334
DOI: 10.5220/0002461303310334

- $Mf = \{m^F_1, m^F_2 \ldots, m^F_n\}$ is a finite set of measures, each measure m^F_i is defined as $m^F_i = (NameM^F_i, FuncM^F_i)$ where:
- $NameM^Fi$ is the name of a measure,
-$FuncM^Fi$ is an aggregate function.

Dimension
A dimension is the axis according to which a fact will be analyzed. It is made up of a finite set of attributes; some of them take part to define various levels of detail (hierarchies), whereas others are less significant but used, for instance, to label results. The latter are said weak attributes.
*Definition***.** A ***dimension*** *d* is defined as $(d^N, Att, HIER)$ where:
- d^N is the name of a dimension,
- *Att* is a set of all attributes of *d* (including weak attributes),
- $HIER = \{h^1_d, h^2_d .., h^t_d\}$ is a set of hierarchies of *d.*
The attributes of a dimension *d* are organized in hierarchies.

*Definition***.** A ***hierarchy*** h^d_i of a dimension *d* is an acyclic path defined as $(N^{HMid}, ParamF, Att\text{-}F)$ where:
- N^{hid} is the name of a hierarchy,
- $ParamF=<p1, p2\ldots,pn>$ is an ordered list of attributes used in h^d_i ,
- *Att-F* is a function which associates an attribute *pi* to the set of its weak-attributes with $\forall$ i$\in$ [1..n], $Att\text{-}F(pi)=\{ at_e, ..,at_r \}$ and $\forall$ j$\in$ [e..r], $at_j \in Att$ and $at_j \notin ParamF$.

Multidimensional Schema
A DM is characterized by its MS which can be either a ***star schema*** analyzing a single fact examined according to dimensions or, a ***constellation schema*** gathering several facts with shared dimensions. In our approach, each schema belongs to one specific application domain.
Definition A *multidimensional schema* is defined as a tuple $(N^{sch}, N^{D\text{-}sch}, F^{sch}, DIM, Funct)$ where:
- N^{sch} is the name of a multidimensional schema,
- $N^{D\text{-}sch}$ is the name of the schema domain,
- $F^{sch} = \{F_1, F_2,..,F_s\}$ is a finite set of facts,
- $DIM = \{d_1, d_2\ldots, d_v\}$ is a finite set of dimensions,
- *Funct* is a function which associates a fact F_i to the list of its dimensions with $\forall$i$\in$ [1..s], $Funct(f_i) = \{d_i\ldots, d_p\}$ with $\forall$ j $\in$ [i..p], $d_j \in DIM$.

OLAP requirement structure
In our approach (Feki, 2004), which aims at developing a computer aided design tool, we propose to collect user requirements in a format familiar to the decision makers, i.e., as structured sheets (Figure 1). A sheet defines the fact to be analyzed and its domain, its measures and dimensions.

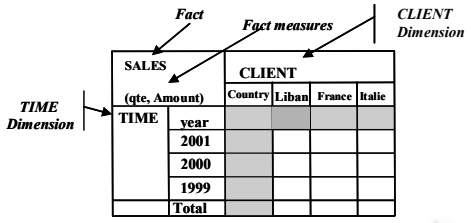

Figure1: Example of two-dimensional sheet.

Note that the structure of a sheet T can be seen as a special star schema since it has a single fact.

3 DM SCHEMA EVOLUTION

DM schema evolve due to several causes, among them : (i) changes in the source structure or (ii) changes in the decisional user needs. In this work we address the problem of DM schema evolution due to changes in OLAP needs. These changes can affect:
- subjects of analysis (fact and/or measures) or,
- axes of analysis (dimensions and hierarchies).

To know whether a new requirement may be covered by existing DMs or not, and how to realize it, we propose a two-phase approach: *a comparison phase*, it is to compare the OLAP requirement with the functional DM schemes, and an *adaptation phase* that is to adapt a DM schema according to the new requirement. Only the first one is presented in this paper.

The comparison phase compares the OLAP requirement (sheet) with the existing DM schemes to identify one of the following cases:
a)There is a schema that covers the requirement,
b)The requirement is partially satisfied, an alteration of a schema is necessary,
c)The requirement is completely not satisfied, the creation of a new MS is required.

The following algorithm *Search_sch*, identifies one of the above situations.
Inputs:
-T is a sheet representing an OLAP requirement,
-S= $\{S_1, S_2\ldots, S_n\}$: a set of *n* stored MS belonging to *m* domains of analysis (m$\leq$ n).
Output:
- A case is identified from a), b) or c) of above.

Algorithm **SEARCH_SCH**

```
BEGIN
For each Sch_i in S do
Begin
1. If F ∈ Sch_i then // integrate T in
    Sch_i
  1.1. If D ⊆ DIM   then
       If  M ⊆ Sch_i.F.mf
          Then    T is satisfied
          else   Add M to Sch_i.F.mf
  1.2. else
      Begin
       1.2.1. determine the set D_add of
            dimensions to be added
       1.2.2. for each d ∈ D_add   do
            Add dimension d to DIM
       1.2.3. determine the set D_alt of
            dimensions to be altered
       1.2.4. for each d ∈ D_alt   do
            1.2.4.1. to determine the
            set H_add of hierarchies to be
            added
            1.2.4.2 for each h ∈  H_add do
              add the hierarchie to d
            1.2.4.3.  determine the set
                 H_alt of hierarchies to
                 be altered
            1.2.4.4. For each h ∈  H_alt
                    do
              begin
              - to determine the strong
              and weak attributes of h
              to add to d
              - to add the strong and
              weak attributes to h
              End
       1.2.5  If M ⊆ set of mesures F
          then measures are satisfied
       1.2.6 Else
        add measures to sch_i for the
fact F
      End
End
2. else
   begin // call IDENTIFY_SCH algorithm
       Result := IDENTIFY_SCH (T,S)
          If Result = {} then
              Create new star schema
           Else Add T to Result
    End
END.
```

If the algorithm identifies case b) and the fact $F \notin S_i$ $\forall$ i $\in$ [1..n] then it raises the problem of choosing which DM schema has to be modified. This requires the identification of the candidate MS and how to choose one; i.e., the MS that has the maximum of common elements with the new requirement. To carry out this choice, we use the similarity factor (Soussi and al., 2005) which is a metric measuring the relevance of the integration.We define two similarity metrics, one for measures and one for dimensions.

Dimension Similarity Metric ***SimD (T, S_i)***

It measures the relevance of the integration of the requirement in a MS, it is based on dimensions.

$$SimD(T,S_i)=\begin{cases}0.75 \text{ if } n=p \text{ or } n<m\\ p/q \text{ otherwise}\end{cases}$$

- n: number of dimensions of T
- m: number of dimensions of S_i
- p : number of common dimensions between T and S_i
- q : number of different dimensions = n+m-p

Measures Similarity Metric ***SimM (T, S_i)***

It measures the relevance of the integration of the requirement in a MS; it is based on measures.

$$SimD(T,S_i)=\begin{cases}0.75 \text{ if } n=p \text{ or } n<m\\ p/q \text{ otherwise}\end{cases}$$

n, m, p and q are as above replacing dimensions by measures.

To decide whether the fact F in T leads to the construction of a new MS or it will be integrated into an existing one, we define two vectors:

1- A *Dimension Similarity Vector* DSV containing all values of SimD (T,S_i) $\forall$ i $\in$ [1..n].

2- A *Measure Similarity Vector* MSV containing all values of SimM (T,S_i) $\forall$ i $\in$ [1..n].

The integration of requirement T into an existing DM can occur if and only if the maximum of at least one vector is greater than 1/3.

The following algorithm *IDENTIFY_SCH* identifies a schema, among several candidate MS, where the requirement should be integreted. It uses DSV and MSV vectors.

Algorithm ***IDENTIFY_SCH***

- P_{simil} : a threshold parameter indicating the minimal value beyond which the integration of the requirement can be carried out.

Max_d := Max (DSV)

Max_m := Max (MSV)

- VmaxD : a subset of S such as $\forall$ i $\in$[1..p] p$\leq$n VmaxD(i) = Max_d.
- VmaxM : a subset of S such as $\forall$ i $\in$[1..q] q$\leq$n VmaxM(i) = Max_m.
- Int_sch : a set of schemes common to VmaxD and VmaxM
- nb : number of rows in HSM

Inputs:

- T: a sheet representing an OLAP requirement analyse a fact according to *n* dimensions where $F^T \notin S_i$ $\forall$ i $\in$ [1..n]
- S ={S_1,S_2,……,S_n} : *n* stored MS belonging to *m* domains (m$\leq$ n).

Output:
Sch_res: set of candidate schemes where T could be integrated.

```
BEGIN
calculate  DSV, MSV, VmaxD and VmaxM
1. If (Maxd ≥ Psimil) or (Maxm ≥
Psimil)  then
  1.1.  Int_sch = VmaxD ∩ VmaxM
  1.2.  If Int_sch = ∅ then
    Begin
       1.2.1 SmaxDM:=0
       1.2.2 For each schema  S in
       VmaxD do
         Begin
            If SmaxDM < DSV(S) + MSV(S)
             then
               Begin
               SmaxDM := DSV(S) + MSV(S)
               Sch_res := Sch_res ∪ {S}
              End
         End
       1.2.3 Return Sch_res
      End
2. Else
   2.1 If |Int_sch | = 1 then
         Return Int_sch
   2.2 Else
         Begin
      2.2.1 Calculate HSM for all
            schemes in Int_sch
       2.2.2  SH_max:=0
        2.2.3 For each  schema S in
            Int_sch do
      Begin
            If SH_max< Σ_{j=1}^{nb} HSM(i,Sj)
            Begin
              SH_max= Σ_{j=1}^{nb} HSM(i,Sj)
              Sch_res := Sj
            end
      End
        2.2.4 Return Sch_res
  End
END
```

4 CONCLUSION

In this paper, we present the evolution of the DM schemes based on the OLAP requirements evolution. These requirements expressed as dimensional fact sheets are compared with the existing DM schemes in order to be integrated. For that, we have proposed two phases approach. The first phase compares the new requirement with the MS to detect the new requirement elements. The second phase, not presented in this paper, adapts a MS and, is based on a set of algebraic operators. We have introduced the concept of similarity as a metric to identify the candidate MS. This work is a part of an ongoing project.

We are interested with the development of a software tool. Especially, it is to visualize the MS graphically and to highlight the evolution impacts on the DW schema. Currently we are studying the effect such alteration on the DM data.

REFERENCES

Blaschka M., Sapia C. and Höfling G., "On Schema Evolution in Multidimensional Databases", *Proceedings of DaWak'99 Conference*, Florence, Italy, 1999.

Bliujute R., Saltenis S., Slivinskas G., Jensen C.S., "Systematic Change Managment in Dimensional Data Warehousing", *Proceedings of the 3rd International Baltic Workshop on DB and IS*, 1998.

Body M., Miquel M., Bedard Y., Tchounikine A., "Handling Evolutions in Multidimensional Structures", 19th International Conference on Data Engineering Sponsored by the IEEE Computer Society March 5 - March 8, 2003 Bangalore, India.

Chamoni P. and Stock S., "Temporal Structures in Data warehousing", *In Proceedings of the DaWaK'99 Conference*, Florence, Italy, 1999.

Eder J. and Koncilia C., "Evolution of Dimension Data in Temporal Data Warehouses", *Proceedings of the DaWaK'01 Conference*, Munich, Germany, 2001.

Feki J., "Vers une conception automatisée des entrepôts de données : Modélisation des besoins OLAP et génération de schémas multidimensionnels", *8th MCSEAI*, Tunisia, 2004.

Mendelzon A.O. and Vaisman A., "Temporal Queries in OLAP" *Proceedings of the 26th VLDB'00 Conference*, Cairo, Egypt, 2000.

Nabli A., Feki J. and Gargouri F.:" Automatic construction of multidimensional schema from OLAP requirements", AICCSA'05, 3-6 January, Cairo, Egypt.

Pedersen T.B., Jensen C.S. and Dyreson C.E., "A foundation for capturing and querying complex multidimensional data", *Information Systems Special Issue: Data Warehousing*, Vol 26, No 5, 2001.

Soussi A., Feki J., et Gargouri F., "Approche semi-automatisée de conception de schémas multidimensionnels valides"., EDA'05, 10 juin Lyon, France, 2005.