



# **Towards Conceptual Multidimensional Design in Decision Support Systems**

Olivier Teste

## **► To cite this version:**

Olivier Teste. Towards Conceptual Multidimensional Design in Decision Support Systems. 5th East-European Conference on Advances in Databases and Information Systems - ADBIS'01, Sep 2001, Vilnius, Lithuania. pp.77-88. <hal-00479851>

**HAL Id: hal-00479851**

**<https://hal.science/hal-00479851v1>**

Submitted on 3 May 2010

**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.



HAL Authorization

# Towards Conceptual Multidimensional Design in Decision Support Systems

Olivier Teste

Université Paul Sabatier - IRIT/SIG  
118 Route de Narbonne - 31062 Toulouse cedex 04 (France)  
e-mail: teste@irit.fr

**Abstract.** Multidimensional databases support efficiently on-line analytical processing (OLAP). In this paper, we depict a model dedicated to multidimensional databases. The approach we present designs decisional information through a constellation of facts and dimensions. Each dimension is possibly shared between several facts and it is organised according to multiple hierarchies. In addition, we define a comprehensive query algebra regrouping the more popular multidimensional operations in current commercial systems and research approaches. We introduce new operators dedicated to a constellation. Finally, we describe a prototype that allows managers to query constellations of facts, dimensions and multiple hierarchies.

## 1 Introduction

In order to improve decision-making process in companies, decision support systems are built from sources (operational databases). These dedicated systems are based on the data warehousing approach [4, 11, 24]. A data warehouse [11] stores large volumes of data, which are extracted from multiple, distributed, autonomous and heterogeneous data sources [4, 11, 24] and they are available for querying.

### 1.1 The Problem

In previous works, we specified a functional architecture of the decision support systems [18, 19], based on a dichotomy of two repositories; a data warehouse collects source data, which is relevant for the decision-makers, and it keeps data changes over the time whereas data marts are deduced from the data warehouse and they are dedicated to specific analyse (each data mart is subject-oriented). This architecture distinguishes several issues laying the foundation of our study (*cf.* figure 1).

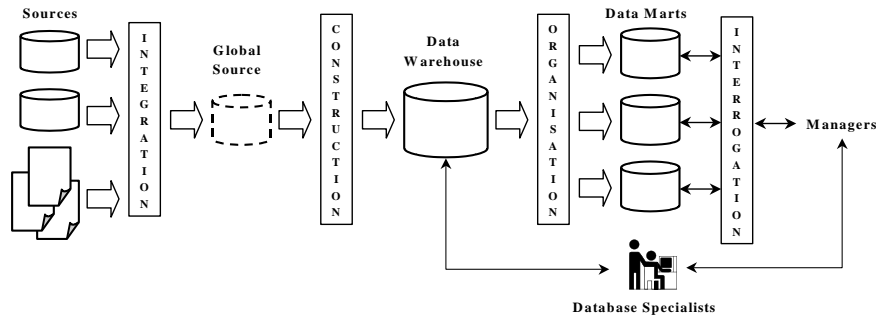
- The **integration** generates a global source from data sources; it is virtual and it is described according to the ODMG data model. The motivating for using the object paradigm at the integration is that it has proven to be successful in complex data modelling [2].
- The **construction** generates a data warehouse as a materialised view [8] over the global source. It is not organised according to a multidimensional model [12]. We justify this choice by the fact that this modelling generates a lot of redundant data [4, 11, 12] limiting efficient warehouse management. We defined a flexible temporal object-oriented data warehouse model in [18, 19].

Due to manager requirements, we provide two approaches for improving the decision making process.

In the first approach the managers exploit the warehouse data to make global analyses. They are helped by database specialists who can directly query warehouse data using powerful and expressive languages. This approach has the advantage of allowing global analyses of the decisional information.

In the second approach the managers make themselves their analyses. They require advanced tools that facilitate analyses and multidimensional operations. We provide a solution based on two steps:

- The **organisation** models data for supporting efficiently OLAP (“On-Line Analytical Processing”) applications [5] in several subject-oriented data marts. The data marts may be designed according to a multidimensional model [12, 17].
- The **interrogation** exploits decisional information. The managers improve their decisions through advanced tools facilitating OLAP applications.



**Fig. 1.** Architecture of Decision Support Systems.

In this paper, we focus on this approach based on data mart generations where relevant data is stored “*multidimensionally*”. We depict a multidimensional model and we define a multidimensional query algebra.

## 1.2 Related Work

In academic research, multidimensional modelling has enjoyed spectacular growth [6]. One of the significant development is the proposal of the data cube operator [7]. Several approaches treat data as n-dimensional cubes where the data is divided in measures (facts) and dimensions [7, 9, 12], but the hierarchy between the parameters is not captured explicitly by the schema. Therefore, several proposals provide structured cube models, which capture dimension hierarchies [1, 3, 13, 14, 17]. Some models provide statistical objects where a structured hierarchy is related to an explicit aggregation function on a single measure supporting a set of queries [20]. To model dimensions of complex structures, several models were made in an object oriented framework [3, 16, 21]. Also, some proposals exploits the temporal nature of the multidimensional modelling [10, 15, 16].

Most of these proposals introduce constraints and specific modelling choices as ROLAP, MOLAP and OOLAP. Nevertheless, in [22] the authors provide a full conceptual approach through the *starER* model, which combines the star structure with the semantically rich constructs of the ER model. In the same way, the model we present is independent of the ROLAP, OOLAP or MOLAP context.

Moreover, existing approaches design a multidimensional database as a star schema [12]. This approach integrates only one fact. We argue that an extended multidimensional model in which a multidimensional database is designed as a constellation of facts and dimensions is a more efficient way for improving a powerful multidimensional modelling [17]. This extended model needs a query language integrating the more popular operations in current commercial systems and research approaches as well as some operations related to the constellation organisation. The main contribution of this paper is the comprehensive multidimensional query algebra that we define. We provide formal definitions of the most important multidimensional operations and we define two new operations related to the constellation organisation.

### 1.3 Paper Outline

Section 2 defines a multidimensional model supporting facts, shared dimensions and multiple hierarchies, independently of ROLAP, OOLAP or MOLAP contexts. Section 3 presents the query algebra related to the multidimensional model. Section 4 describes extensions of our prototype GEDOOH.

## 2 A Multidimensional Model

In the architecture that we depict in figure 1, a data mart is subject-oriented; it is dedicated to a specific class of users and it regroups all relevant information for supporting their decisional requirements. The data mart must be modelled “*multidimensionally*” for improving analyses and decision making processes [12].

The multidimensional model we define is based on the idea of the “*constellation*” [17], in which data marts are composed of several facts and dimensions; each dimension is shared between facts and it can be associated to one or several hierarchies. Therefore, the managers can handle several facts according to shared dimensions, facilitating comparisons between several measures.

### 2.1 Facts

A fact reflects information that have to be analysed; for example, a factual data is the amount of sales occurring in shops.

**Definition 1.** A **fact**  $F$  is defined by a tuple  $(fname, M^{fname})$  where

- $fname$  is a name,
- $M^{fname} = \{m_1, m_2, \dots, m_m\}$  is a set of attributes where each  $m_k$  represents one measure.

### 2.2 Dimensions and Hierarchies

A dimension reflects information according to which data of facts will be analysed. A dimension is organised through parameters, which conform to one or several hierarchies; the dimensions of interest may be the shop location, the time,...

**Definition 2.** A **dimension**  $D$  is defined by a tuple  $(dname, A^{dname}, H^{dname})$  where

- $dname$  is a name,
- $A^{dname}$  is a set of attributes,
- $H^{dname} = \langle H^{dname}_1, H^{dname}_2, \dots, H^{dname}_h \rangle$  is an ordered set of hierarchies ( $H^{dname}_1$  is called the current hierarchy).

The parameters are organised according to hierarchies. Within a dimension, values of different parameters are related through a family of roll up functions, denoted  $\rho_{roll}$ , according to each hierarchy defined on them. A roll up function  $\rho_{roll}^{H(p_j \rightarrow p_{j'})}$  associates a value  $v$  of a parameter  $p_j$  with a value  $v'$  of an upper parameter  $p_{j'}$  in the hierarchy  $H$ .

**Definition 3.** A hierarchy  $H^{dname}_i$  is defined by a tuple  $(hname, P^{hname})$  where

- $hname$  is a name,
- $P^{hname} = \langle p_{i1}, p_{i2}, \dots, p_{hi} \rangle$  is an ordered set of parameters where  $\forall j \in [i_1..h_i]$ ,  $p_j \in A^{dname}$ .

Note that  $A^{dname}$  contains a distinguished parameter  $all$ , such that  $dom(all) = \{All\}$ . This attribute defines the upper granularity of hierarchies; for every hierarchy  $H^{dname}_j \in H^{dname}$ ,  $H^{dname}_j = \langle p_{i1}, p_{i2}, \dots, all \rangle$ .

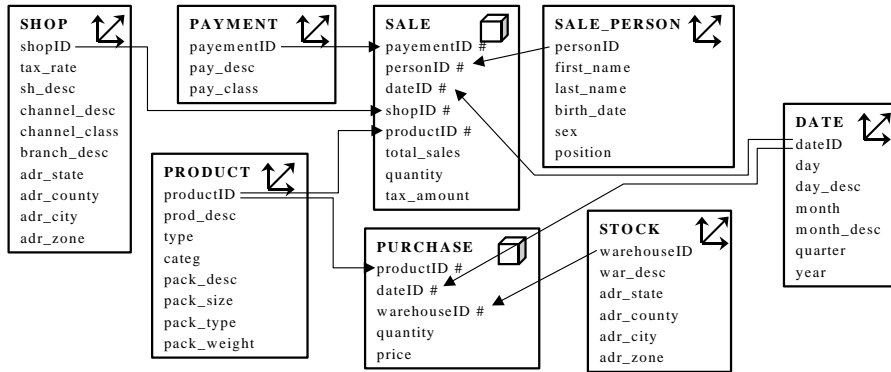
### 2.3 Constellation Schema

A data mart is modelled according to a constellation schema; it is composed of several facts and several dimensions, which are possibly shared between facts.

**Definition 4.** A constellation schema  $S^{DM}$  is defined by a tuple  $(sname, FACT, DIM, Param^{sname})$  where

- $sname$  is a name,
- $FACT = \langle F_1, F_2, \dots, F_u \rangle$  is an ordered set of facts ( $F_i$  is called the current fact),
- $DIM = \{D_1, D_2, \dots, D_v\}$  is a set of dimensions,
- $Param^{sname}: FACT \rightarrow 2^{DIM}$  is a function such that  $Param^{sname}(F_i) = \langle D_i^1, D_i^2, \dots, D_i^{w_i} \rangle$ . It returns an ordered set of dimensions which are associated to the fact  $F_i$  (for the current fact  $F_i$ ,  $D_i^1$  and  $D_i^2$  are called the current dimensions).

**Example.** The case we study is taken from commercial domain and it concerns shop channels. The data mart supports analyses about sales and purchases related to various commercial shop channels and commercial warehouses. Figure 2 represents the example of a data mart; we use extended relational notations: ‘ $\square$ ’ represents a fact and ‘ $\boxplus$ ’ represents a dimension. Note that the constellation is named “channalyse”.



**Fig. 2.** Example of a Constellation Schema.

Along each dimension, the administrator defines one, or possibly several, hierarchy(ies). These hierarchies offer various views for analysed data; e.g. users can analyse sales according to dates and shops, and they can display analysed data with an administrative organisation of the country ( $adr\_state$ ,  $adr\_county$ ,  $adr\_city$ ) or with a specific organisation ( $adr\_city$ ,  $adr\_zone$ ). The hierarchies are defined as follows.

- $H_{shop_1}^{shop} = ("h\_shop\_channel", <shopID, channel\_class, branch\_desc, all>)$
- $H_{shop_2}^{shop} = ("h\_shop\_administrative", <shopID, city, county, state, all>)$
- $H_{shop_3}^{shop} = ("h\_shop\_zone", <shopID, city, zone, all>)$
- $H_{payment_1}^{payment} = ("h\_payment", <paymentID, pay\_class, all>)$
- $H_{person_1}^{person} = ("h\_person\_position", <personID, position, all>)$
- $H_{product_1}^{product} = ("h\_product\_category", <prodID, type, categ, all>)$
- $H_{date_1}^{date} = ("h\_date\_gregorian", <dateID, day, month, quarter, year, all>)$
- $H_{stock_1}^{stock} = ("h\_stock\_administrative", <warehouseID, city, county, state, all>)$
- $H_{stock_2}^{stock} = ("h\_stock\_zone", <warehouseID, city, zone, all>)$

### 3 A Comprehensive Multidimensional Query Algebra

Here, we express in a query algebra the most popular OLAP operators introduced in the scientific literature and we provide new operators related to the constellation organisation.

#### 3.1 Data Displaying: “n-table”

A constellation is displayed within an n-table according to columns, rows and planes. The current fact  $F_I$  is used to define the displayed plane. The current dimensions  $D_I^1$  and  $D_I^2$  of the current fact define displayed lines and rows. For each current dimension, the upper level is displayed according to the current hierarchy. Note that because of the constellation feature, we do not display the complete information stored in data marts; more precisely, only the measures of the current fact are displayed according to the current dimensions and their current hierarchies.

**Example.** We deal with the previous example. The current fact is “sale” and the current dimensions are “shop” and “payment” displayed according to the hierarchies “h\_shop\_channel” and “h\_payment”.

**Table 1.** Example of a Constellation Displaying.

Sale		Shop / $H_{shop_1}^{shop}$				
		branch_desc	BR1	BR2	BR3	BR4
Payment / $H_{payment_1}^{payment}$	pay_class	total_sales, tax_amount, quantity				
	PC1		(58,6, 2)	(67,7, 3)	(58,6, 1)	(68,7, 2)
	PC2		(60,6, 3)	(55,6, 3)	(50,5, 1)	(65,7, 3)
	PC3		(45,5, 1)	(50,5, 1)	(52,5, 1)	(64,6, 2)
Sale.person.position="manager"						
Product.categ="C1"						
Date.year=2000						

#### 3.2 Multidimensional Operations

We first define relational operators in the multidimensional algebra; we adopt the most popular operators (Join, Aggregate, Union, Intersect, and Difference). The operation Slice and Dice is used on a dimension and it removes values of the dimension that do not satisfy a restricted condition. Note that this operator realises selecting (or restricting) in relational terminology.

Because the complete information stored in data marts is not displayed, we define rotate operators for displaying measures according to various parameters. We adopt rotate operators introduced in [1], and we define a new rotation between facts.

**Definition 5.** The **DRotate** operation permutes two dimensions  $D_i$  and  $D_j$  of a fact  $F$ .  $DRotate(Sh, F, D_i, D_j) = Sh'$  where

- $Sh = (sname, FACT, DIM, Param^{sname})$  is a constellation schema,
- $F \in FACT$  is a fact,
- $D_i \in DIM$  and  $D_j \in DIM$  are two dimensions |  $Param(F) = \langle \dots, D_i, \dots, D_j, \dots \rangle$ .

$Sh' = (sname, FACT, DIM, Param^{sname'})$  where  $Param^{sname'}(F) = \langle \dots, D_j, \dots, D_i, \dots \rangle$  and  $\forall F \in FAI, F_k \neq F, Param^{sname'}(F_k) = Param^{sname}(F_k)$ .

**Definition 6.** The **HRotate** operation permutes two hierarchies  $H^{dname_i}$  and  $H^{dname_j}$  of a dimension  $D$ .  $HRotate(Sh, D, H^{dname_i}, H^{dname_j}) = Sh'$  where

- $Sh = (sname, FACT, DIM, Param^{sname})$  is a constellation schema,
- $D \in DIM$  is a dimension,
- $H^{dname_i} \in H^{dname}$  and  $H^{dname_j} \in H^{dname}$  are two hierarchies |  $H^{dname} = \langle \dots, H^{dname_i}, \dots, H^{dname_j}, \dots \rangle$ .

$Sh' = (sname, FACT, DIM', Param^{sname'})$  where

- $DIM' = DIM - \{D_i\} + \{D_j\}$  |  $D_i' = (dname_i, P^{dname_i}, H^{dname_i} = \langle \dots, H^{dname_j}, \dots, H^{dname_i}, \dots \rangle)$
- $\forall D \in FACT$ , if  $D_i \in Param^{sname}(F)$  then  $Param^{sname'}(F) = Param^{sname}(F) - \{D_i\} + \{D_j\}$  else  $Param^{sname'}(F) = Param^{sname}(F)$ .

**Definition 7.** The **FRotate** operation permutes two facts  $F_i$  and  $F_j$  of a constellation schema  $Sh$ .  $FRotate(Sh, F_i, F_j) = Sh'$  where

- $Sh = (sname, FACT, DIM, Param^{sname})$  is a constellation schema,
- $F_i \in FACT$  and  $F_j \in FACT$  are two facts |  $FACT = \langle \dots, F_i, \dots, F_j, \dots \rangle$ .

$Sh' = (sname, FACT', DIM, Param^{sname'})$  where  $FACT' = \langle \dots, F_j, \dots, F_i, \dots \rangle$ .

**Example.** We complete the previous example. Managers change the dimensions in order to analyse measures according to other parameters. They permute “Shop” and “Date” as well as “Payment” and “Product”.  $DRotate(DRotate("channalyse", Payment, Product), Shop, Date)$

**Table 2.** N-table Representing the Constellation after Rotations.

Sale		Date / H <sup>date</sup> <sub>1</sub>			
		year	1998	1999	2000
Product / H <sup>product</sup> <sub>1</sub>	categ	total_sales, tax_amount, quantity			
	C1		(58,6, 2)	(67,7, 3)	(58,6, 1)
	C2		(60,6, 3)	(55,6, 3)	(50,5, 1)
	C3		(45,5, 1)	(50,5, 1)	(52,5, 1)
Sale_person.position="manager"					
Payment.pay_class="PC1"					
Shop.branch_class="BRI"					

The positions (values) of each parameter are ordered. We introduce one operator for changing these positions.

**Definition 8.** The operation **Switch** permutes two positions (values)  $pos_{j1}$  and  $pos_{j2}$  of a parameter  $p$ .  $Switch(Sh, d, p, pos_{j1}, pos_{j2}) = Sh'$  where

- $Sh = (sname, FACT, DIM, Param^{sname})$  is a constellation schema,
- $D \in DIM$  is a dimension,
- $p \in P^{dname}$  is a parameter of the current hierarchy |  $p \in H^{dname}_1$ ,
- $pos_{j1} \in dom(p)$  and  $pos_{j2} \in dom(p)$  are two positions (values) of the parameter  $p$ .

$Sh'$  is the result where  $pos_{j1}$  and  $pos_{j2}$  are permuted in the hierarchy  $H^{dname}_1$ .

The RollUp and DrillDown operations are probably the most important operations for OLAP; they allow users to change data granularities.

**Definition 9.** The **DrillDown** operation inserts into the current hierarchy of a dimension  $D_i$ , a parameter  $p_j$  at a lower granularity.  $DrillDown(Sh, D_i, p_j)=Sh'$  where

- $Sh=(sname, FACT, DIM, Param^{sname})$  is a constellation schema,
  - $D_i \in DIM$  is a dimension such that  $D_i=(dname_i, P^{dname_i}, H^{dname_i})$ ,
  - $p_j \in P^{dname_i}$  is a parameter (it will be integrated in the current hierarchy of  $D_i$ ).
- $Sh'=(sname, FACT, DIM', Param^{sname'})$  where
- $DIM'=DIM-\{D_i\}+\{D_i'\} \mid D_i'=(dname_i, P^{dname_i}+\{p_j\}, H^{dname_i'}=<<p_j>+H^{dname_i}, H^{dname_2}, \dots, H^{dname_h}>)$  and
  - $\forall F \in FACT$ , if  $D_i \in Param^{sname}(F)$  then  $Param^{sname'}(F)=Param^{sname}(F)-\{D_i\}+\{D_i'\}$  else  $Param^{sname'}(F)=Param^{sname}(F)$ .

**Definition 10.** The **RollUp** operation inserts into the current hierarchy of a dimension  $D_i$ , a parameter  $p_j$  corresponding to an upper granularity.  $RollUp(Sh, D_i, p_j)=Sh'$  where

- $Sh=(sname, FACT, DIM, Param^{sname})$  is a constellation schema,
  - $D_i \in DIM$  is a dimension  $\mid D_i=(dname_i, P^{dname_i}, H^{dname_i})$ ,
  - $p_j \in parameters^{dname_i}$  is a parameter, which will be integrated in the current hierarchy of the dimension  $D_i$ .
- $Sh'=(sname, FACT, DIM', Param^{sname'})$  where
- $DIM'=DIM-\{D_i\}+\{D_i'\} \mid D_i'=(dname_i, P^{dname_i}+\{p_j\}, H^{dname_i'}=<H^{dname_i}+<p_j>, H^{dname_2}, \dots, H^{dname_h}>)$  and
  - $\forall F \in FACT$ , if  $D_i \in Param^{sname}(F)$  then  $Param^{sname'}(F)=Param^{sname}(F)-\{D_i\}+\{D_i'\}$  else  $Param^{sname'}(F)=Param^{sname}(F)$ .

In order to ease analyses, in [1] the authors introduce operations allowing an uniform treatment of parameters and measures; one operator converts parameters into measures and another one creates parameters from specified measures. We adopt these operations in the constellation framework.

**Definition 11.** The **Push** operation converts parameters into measures.  $Push(Sh, d, p, f)=Sh'$  where

- $Sh=(sname, FACT, DIM, Param^{sname})$  is a constellation schema,
  - $D \in DIM$  is a dimension,
  - $p \in P^{dname}$  is a parameter of the dimension  $D$ ,
  - $F \in FACT$  is a fact  $\mid D \in Param(F)$ .
- $Sh'=(sname, FACT', DIM', Param^{sname'})$  where
- $FACT'=FACT-\{F\}+\{F'\} \mid F'=(fname, M^{fname}+\{p\})$ ,
  - $DIM'=DIM-\{D_i\}+\{D_i'\} \mid P^{dname_i'}=P^{dname_i}-\{p\}$  and
  - $Param^{sname'}(F') = Param^{sname}(F)-\{D_i\}+\{D_i'\}, \quad \forall F'' \in FACT, \quad F'' \neq F',$   
 $Param^{sname'}(F'') = Param^{sname}(F'')$ .

**Definition 12.** The **Pull** operation converts measures into parameters.  $Pull(Sh, F, m, D)=Sh'$  where

- $Sh=(sname, FACT, DIM, Param^{sname})$  is a constellation schema,
- $F \in FACT$  is a fact,
- $m \in M^{fname}$  is a measure of the fact  $F$ ,



- $D \in DIM$  is a dimension  $| D \in Param(F)$ .
- $Sh' = (sname, FACT', DIM', Param^{sname'})$  where
- $FACT' = FACT - \{F\} + \{F'\} | F' = (fname, M^{fname} - \{m\})$ ,
- $DIM' = DIM - \{D_i\} + \{D_i'\} | P^{dname_i} = P^{dname} + \{m\}$  and
- $Param^{sname'}(F') = Param^{sname}(F) - \{D_i\} + \{D_i'\}, \quad \forall F'' \in FACT, \quad F'' \neq F',$   
 $Param^{sname'}(F'') = Param^{sname}(F).$

To convert a constellation into several star schemas (constellations composed of one fact), we introduce two operators. They allow users to reduce schemas.

**Definition 13.** The **TSplit** operation generates several sub schemas from a constellation schema according to its facts. Each generated schema is composed of one fact.  $TSplit(Sh) = \{Sh_1, \dots, Sh_u\}$  where

- $Sh = (sname, FACT, DIM, Param^{sname})$  is a constellation schema.
- $\forall i \in [1..u], Sh_i = (sname, FACT', DIM', Param^{sname'})$  is a resulting sub schema. Its a constellation schema composed of one fact such that  $FACT' = \{F_i\}$ ,  $DIM' = \{D | D \in DIM \wedge D \in Param(F_i)\}$  and  $Param'(F_i) = Param(F_i)$ .

**Definition 14.** The **Split** operation generates several sub schemas from a constellation schema, which is composed of one fact. Each generated sub schema results from a selection.  $Split(Sh, D, p) = \{Sh_1, Sh_2, \dots, Sh_s\}$  where

- $Sh = (sname, FACT, DIM, Param^{sname})$  is a constellation schema.
- $D \in DIM$  is a dimension,
- $p \in P^{dname}$  is a parameter of the dimension  $D | dom(p) = \{pos_1, pos_2, \dots, pos_s\}$ .
- $\forall i \in [1..s], Sh_i = (sname, FACT, DIM, Param^{sname})$  is a resulting sub schema according to the slice operation  $Slice(Sh, D, pred(pos_i))$ .

## 4 Implementation

In previous works, we have implemented a prototype allowing administrators both to define and to generate data warehouses and data marts. This prototype is called **GEDOOH**. It is based on three components: a graphical interface, an automatic data warehouse generator, and an automatic data mart generator.

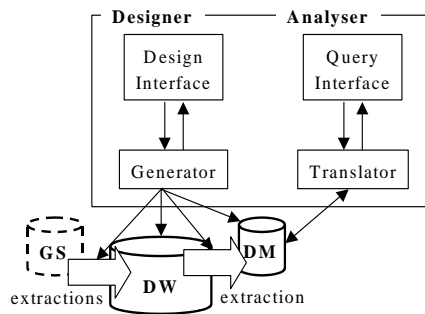


Fig. 3. GEDOOH Architecture.

GEDOOH helps administrators in designing data warehouses and data marts. It is based on extended UML notations for displaying schemas.

- Firstly, the administrator defines a data warehouse from a graph of the global source (or a data mart from a graph of the data warehouse).
- Secondly, the generators create automatically the data warehouse (or the data mart) according to the graphical definitions. The schema, the first extraction (which populates the data warehouse or the data mart) and the refresh process are generated.

This tool is implemented in Java (jdk 1.3) on top of a relational database management system (Oracle) and it is operational; its source code represents approximately 8000 lines of Java code.

Now, we are implementing extensions in order to validate the model we present in this paper and its associated query algebra. We add a user component allowing the managers both to display and to query constellation schemas of the generated data marts. The extension (the user component) is composed of two parts: an interface and a query translator.

- The **query interface** displays an n-table representing a constellation. This component uses internal structures of the displayed information. Each multidimensional operation is treated by the query translator component.
- The **query translator** translates each multidimensional operation in a relational query. This component sends the relational query to the RDBMS and it translates the result in internal structures.

Sale		Shop / Hierarchy				
Payment / Hierarchy	pay_class	branch_desc	BR1	BR2	BR3	BR4
		total_sales, tax_amount, quantity				
	PC1		(58,6,2)	(67,7,3)	(58,6,1)	(68,7,2)
	PC2		(60,6,3)	(55,6,3)	(50,5,1)	(65,7,3)
	PC3		(45,5,1)	(50,5,1)	(62,5,1)	(64,6,2)

Sale person.position="manager"  
 Product.category="C1"  
 Date.year=2000

Fig. 4. Example of a Constellation Displaying through the GEDOOH Query Interface.

## 5 Conclusion

We first introduce an architecture of decision support systems distinguishing several issues and laying the foundation for our study. Based on the architecture, this paper deals with the data mart designing and querying.

The multidimensional model we define is based on the idea of the “constellation”, in which data marts are composed of several facts and dimensions; each dimension is shared between facts and it can be associated to one or several hierarchies. Shared dimensions facilitates comparisons between several measures according to the same dimensional data organisation (same hierarchies, same parameters...). This approach provides a unified framework for the multidimensional modelling independently of the ROLAP, OOLAP or MOLAP context. We develop a query algebra for the data marts. We express in a comprehensive algebra the most popular OLAP operators and we provide new operators related to the constellation organisation (FRotate, TSplit).

We are currently working on extending the tool GEDOOH. It allows administrators to generate a data warehouse from sources through a graphical interface. We have extended algorithms for generating data marts from data warehouses. Now, we develop solutions for querying data marts based on the query algebra.

We will investigate meta-modelling issues and we plan to develop a method for designing decision support systems. We must provide a design method; it must be composed of models (with concepts and constraints), a complete process and a tool.

## 6 References

- [1] Agrawal R., Gupta A., Sarawagi A., "Modeling Multidimensional Databases", IBM Research Report, Semptember 1995, (published in the proceedings of ICDE'97).
- [2] Bukhres O.A., Elmagarmid A.K., "Object-Oriented Multidatabase Systems - A solution for Advanced Applications", Prentice Hall, ISBN 0-13-103813-2, 1993.
- [3] Cabibbo L., Torlone R., "A logical approach to multidimensional databases", EDBT'98, Valencia (Spain), March 1998.
- [4] Chaudhuri S., Dayal U., "An Overview of Data Warehousing and OLAP Technology", ACM SIGMOD Record, Tucson (Arizona, USA), May 1997.
- [5] Codd E.F., "Providing OLAP (on-line analytical processing) to user-analysts : an IT mandate", Technical Report, E.F. Codd and Associates, 1993.
- [6] Gatzia S., Jeusfeld M.A., Staudt M., Vassiliou Y., "Design and Management of Data Warehouses", Report on the DMDW'99, SIGMOD Record, Dec. 1999.
- [7] Gray J., Bosworth A., Layman A., Pirahesh H., "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals", Microsoft Technical Report MSR-TR-95-22, 1995.
- [8] Gupta, I.S. Mumick, "Maintenance of Materialized Views: Problems, Techniques, and Applications", IEEE Data Engineering Bulletin, 1995.
- [9] Gyssen M., Lakshmanan L.V.S., "A Foundation for Multi-Dimensional Databases", VLDB'97, Athens (Greece), August 1997.
- [10] Hurtado C., Mendelzon A., Vaisman A., "Maintaining Data Cubes under Dimension Updates", ICDE '99, San Diego (California, USA), March 1999.
- [11] Inmon W.H., "Building the Data Warehouse", J. Wiley&Sons, ISBN 0471-14161-5, 1994.
- [12] Kimball R., "The data warehouse toolkit", J. Wiley&Sons, 1996.
- [13] Lehner W., "Modeling Large Scale OLAP Scenarios", EDBT'98, Valencia (Spain), March 1998.
- [14] Li C., Wang X. S., "A Data Model for Supporting On-Line Analytical Processing", CIKM'96, Rockville (Maryland, USA), November 1996.
- [15] Mendelzon A.O., Vaisman A., "Temporal Queries in OLAP", VLDB'00, Cairo (Egypt), September 2000.
- [16] Pedersen T.B., Jensen C.S., "Multidimensional Data Modeling for Complex Data", ICDE'99, San Diego (California, USA), March 1999.
- [17] Pokorný J., Sokolowsky P., "A conceptual modelling perspective for Data Warehouses", Electronic Business Engineering, 4 Internationale Wirtschaftsinformatik 1999, Hrsg. A.-W. Scheer, M. Nuttgens, Heidelberg Physica-Verlag, 1999, pp.666-684.
- [18] Ravat F., Teste O., "A Temporal Object-Oriented Data Warehouse Model", DEXA'00, LNCS 1873, September 2000, London (UK).
- [19] Ravat F., Teste O., Zurfluh G., "Towards Data Warehouse Design", ACM CIKM'99, November 1999, Kansas City (Missouri, USA).
- [20] Shoshani A., "Statistical databases and OLAP: similarities and differences", invited talk, CIKM'96, Rockville (Maryland, USA), November 1996.
- [21] Trujillo J., Palomar M., "An Object Oriented Approach to Multidimensional Database Conceptual Modeling (OOMD)", DOLAP'98, Bethesda (USA), November 1998.
- [22] Tryfona N., Busborg F., Borch Christiansen J.G., "starER: A Conceptual Model for Data Warehouse Design", DOLAP'99, Kansas city (Missouri, USA), November 1999.
- [23] Widom J., "Research problems in data warehousing", ACM CIKM'95, 1995.