Graphical Querying of Multidimensional Databases

Franck Ravat, Olivier Teste, Ronan Tournier, and Gilles Zurfluh

IRIT, Institut de Recherche en Informatique de Toulouse
Université Toulouse 3 (Paul Sabatier), 118 route de Narbonne
F-31062 Toulouse CEDEX9, France
{ravat,teste,tournier,zurfluh}@irit.fr

Abstract. This paper provides an answer-oriented multidimensional analysis environment. The approach is based on a conceptual point of view. We define a conceptual model that represents data through a constellation of facts and dimensions and we present a query algebra handling multidimensional data as well as multidimensional tables. Based on these two propositions, in order to ease the specification of multidimensional analysis queries, we define a formal graphical language implemented within a prototype: GraphicOLAPSQL.

1 Introduction

OLAP (On-Line Analytical Processing) [6] systems allow analysts to improve the decision-making process by consulting and analysing aggregated historical business or scientific data. One of the reasons for the absence of standard in Multidimensional DataBases (MDB) modelling is the lack of conceptual level in commercial tools [20]. Indeed, analysts and decision-makers manipulate logical and physical concepts instead of manipulating conceptual elements.

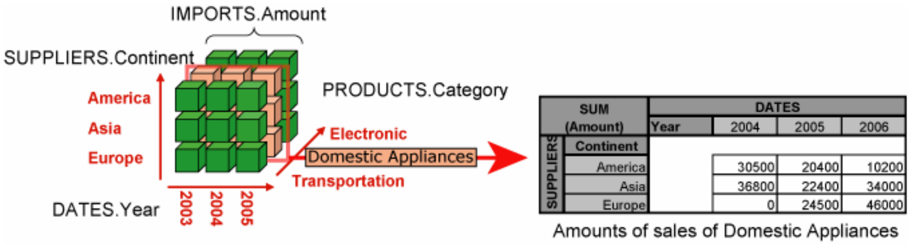
1.1 Context and Motivations

OLAP or multidimensional modelling [13] represents data as points in a multidimensional space with the use of the cube (or hypercube) metaphor. The following figure presents the data of a commercial activity (*imports amounts*) of a company analysed according to three analysis axes (*suppliers*, *dates*, *products*).

MDB are modelled through subjects of analysis, named *facts* and analysis axes, named *dimensions* [13]. These structures are grouped into a *star schema* [13]. Facts are groupings of analysis indicators, named *measures*. Dimensions are composed of hierarchically ordered attributes which model the different detail levels of the axis–data granularity. Notice that, in spite of a decade of research in OLAP systems, concepts and systems exist without uniform theoretical basis [17, 20].

OLAP analysis is performed through interactive exploration and analysis of indicators according to several analysis axes. Analysts need a decisional data representation semantically rich, which clearly distinguishes subjects and axes of analysis as well as structure and content. Moreover, this representation must be independent of any implementation choice. In order to facilitate decision-making, analysts need a multidimensional data visualisation interface adapted to the display of the analysis data. A

Y. Ioannidis, B. Novikov, and B. Rachev (Eds.): ADBIS 2007, LNCS 4690, pp. 298–313, 2007.



SUM (Amount)		DATES			
		Year	2004	2005	2006
SUPPLIERS	Continent				
	America		30500	20400	10200
	Asia		36800	22400	34000
	Europe		0	24500	46000

Amounts of sales of Domestic Appliances

Fig. 1. Cube representation of a MDB, with a slice corresponding to domestic appliances

graphic manipulation language is required, which must be homogeneously positioned between reporting restitution (query) commands and manipulation (analysis) commands.

1.2 Related Works

Without a model based on consensus for multidimensional data, several propositions have been made. Several surveys may be found [24, 1]. According to [24], these models may be classified into two categories. First, works based on the “cube model” [2, 15, 7, 12], have the following issues: 1) weakness in modelling the fact (subject of analysis) or its measures (analysis indicators); 2) little or no conceptual modelling of dimensions (analysis axes) with no explicit capture of their hierarchical structure; and 3) no separation between structure and content. The second category called “multidimensional model” overcomes these drawbacks and is semantically richer. It allows a precise specification of each multidimensional component and notably the different aggregation levels for measures [14, 18, 26, 1]. These models are based on the concepts of fact and dimension possibly with multiple hierarchies. A hierarchy defines a point of view of an analysis axis.

The first works on OLAP manipulation algebras extended relational algebra operators for the cube model [11, 2, 15, 12, 19]. To counter the inadaptability of relational algebra for manipulating multidimensional structures in an OLAP context, numerous works provided operations for specifying and manipulating cubes [4, 5, 18, 1, 8]. These works are not user-oriented for the following reasons: 1) they do not focus on an adapted data structure for displaying decisional data to the user; 2) they are based on partial sets of OLAP operations. Hardly any multidimensional model provides multi-fact and multi-hierarchies as well as a set of associated operations.

Despite of more than a decade of research in the field of multidimensional analysis, very little attention has been drawn on graphical languages. In [5], the authors present a graphical multidimensional manipulation language associated to a conceptual representation of the multidimensional structures. Although the authors define a rather complete manipulation algebra and calculus, the high level graphical language offers very little manipulations in comparison. In [3], the authors offer an intermediate solution, with more manipulations but the system uses complex forms for the multidimensional query specifications. Neither solution provides a restitution interface. [23] and [22] are advanced visualisation tools. The first one offers an impressive pivot table that adapts its display according to the analysis data type, whereas the second offers an arborescent view with multiple scales and very specific manipulations. Here, neither proposition provides a formal description of the manipulation language.

Microsoft Excel Pivot tables, although very expressive restitution interfaces, do not provide many dynamic manipulations. On the other hand, other commercial tools offer extensive manipulations (Business Objects[1], Cognos BI[2], Tableau[3], Targit[4]…). But all these tools display the multidimensional structures of the MDB within an arborescent view, rendering impossible comparative analyses between different subjects sharing analysis axes. Moreover, the representation used mixes completely MDB structures and content. The user completely lacks an adapted conceptual view of the MBD concepts [20].

Nowadays, decision-makers whish to perform their own analyses, but they lack the knowledge to manipulate multidimensional structures with the use of multidimensional query algebras or with adapted procedural query languages. On the other hand, commercial tools provide adapted manipulation languages but lack: 1) rigorous reference to multidimensional operations, 2) a uniform theoretical basis as well as 3) an adapted conceptual view of the multidimensional elements of the underlying MDB. Moreover, these tools sacrifice analysis coherence for analysis flexibility.

1.3 Aims and Contributions

In this context, in order to ensure access to company data, we intend to define a multidimensional OLAP language to ease multidimensional data analysis specification and manipulation. This language must: 1) disregard all logical and implementation constraints; 2) manipulate concepts close to analysts' point of view; 3) be based on a stable theoretical basis to provide consistency and ensure analysis coherence; and 4) provide interaction with the analyst through an incremental and graphical interface.

The paper contributions may be summarized in four points.

1) A conceptual representation of multidimensional data structures. This graphic conceptual view eases users to understand the available analyses supported by the MDB schema. Analysts express queries from the graphic conceptual view, using graphic elements of the MDB schema.

2) A display interface adapted to multidimensional analyses for representing the query results. Contrarily to pivot tables or commercial software output, this interface takes into account the hierarchical nature of multidimensional data in order to ensure analysis consistency. The tabular structure may be directly manipulated by users and analyst may specify complex analyses by incremental definition of complex queries.

3) A user-oriented multidimensional query algebra that uses the model elements as input and the multidimensional table (mTable) as output structure. This algebra is based on a closed minimal core of operators that may be combined together.

4) A graphic language that allows users to express operations using a reduced set of formal primitives. Each primitive handles multidimensional concepts independently of their implantation. This language is complete with regard to the algebraic core and provides incremental manipulations of the mTable, allowing fine analysis tuning.

[1] Business Objects XI from http://www.businessobjects.com/

[2] Cognos Business Intelligence 8 from http://www.cognos.com/

[3] Tableau 2 from http://www.tableausoftware.com/

[4] Targit Business Intelligence Suite from http://www.targit.com/

The paper layout is as follows: section 2 defines the concepts and formalisms of the multidimensional model, section 3 presents the algebraic operators and section 4 defines the graphic language.

2 Multidimensional Model: Constellation

In this section we present our multidimensional framework based on a conceptual view displaying MDB structures as a graphical conceptual view as well as a data interface displaying decisional data through a multidimensional table (mTable). Our model allows users to disregard technical and storing constraints, sticks closer to decision-makers' view [10], eases correlation between various subjects of analysis [21, 26] with multiple perspective analysis axes [16]. It also allows a clear distinction structural elements and values and offers a workable visualisation for decision-makers [12]. We invite the reader to consult [25] for a more complete discussion on multidimensional models and definition of the constellation model.

2.1 Concepts

A constellation regroups several subjects of analysis (facts), which are studied according to several analysis axes (dimensions) possibly shared between facts. It extends star schemas [13] commonly used in the multidimensional context.

A *constellation Cs* is composed of a set $F^{Cs}=\{F_1,\ldots,F_m\}$ of facts F_i, a set $D^{Cs}=\{D_1,\ldots,D_n\}$ of dimensions D_i and a function $Star^{Cs}$ linking dimensions and facts together. A fact represents a subject of analysis. It is a conceptual grouping of analysis indicators called measures. It reflects information that has to be analysed according to several dimensions. A *fact* F_i is composed of a set $M^{Fi}=\{M_1,\ldots,M_w\}$ of *measures* M_i. Analysis axes are dimensions seen through a particular perspective, namely a hierarchy. This hierarchy of parameters represents the different graduations of the analysis axis. A *dimension* D_i is composed of a set H^{Di} of hierarchies H_i and a set A^{Di} of attributes (namely parameters and weak attributes). A dimension D_i linked to the fact F_i is noted $D_i \in Star^{Cs}(F_i)$. A *hierarchy* H_i is an ordered list $Param^{Hi}=<p_1,\ldots,p_{np}>$ of *parameters* p_i and $\forall p_i \in Param^{Hi}$, $\forall H_i \in H^{Di} \Rightarrow p_i \in A^{Di}$. Note that p_1 is named *root parameter* while p_{np} is the *extremity parameter*. *Weak attributes* may be associated to a parameter in order to complete its semantic. Note that, within the rest of the paper no distinction will be made between parameters and weak attributes.

Notations. An aggregated measure M_i is noted $f_{AGGi}(M_i)$. $level^H(p_i)$ is the position of p_i in $Param^H$. The notation $p_i \in H$ is a simplified notation for $p_i \in Param^H$. The set of values of a parameter p_i is $dom(p_i)=<v_{min},\ldots,v_{max}>$.

Graphic notations, based on [9], offer a clear global conceptual view. These notations highlight subjects of analysis (facts) and their associated axes of analyses (dimensions and hierarchies). See Figure 2 for an example.

2.2 Multidimensional Table (MTable)

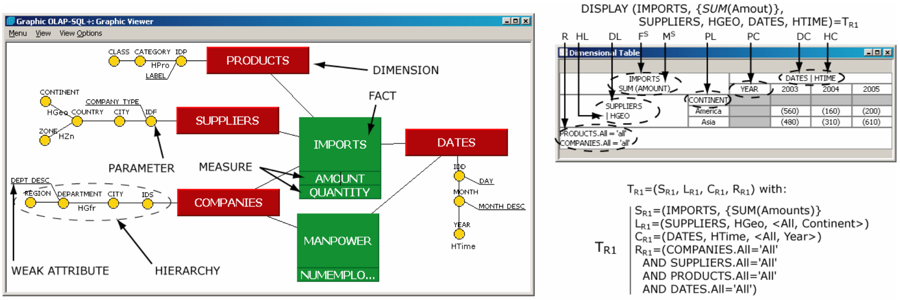
OLAP analysis consists in analysing key performance indicators (measures) according to several analysis axes. As in [12], we offer a tabular visualisation called multidimensional table (mTable), displaying a fact and detailed information of two dimensions.

A *multidimensional table* is defined by $T=(S, C, L, R)$, where $S=(F^S, M^S)$ represents the analysed *subject* through a fact F^S and a set of aggregated measures $M^S=\{f_{\text{AGG1}}(M_1),\ldots, f_{\text{AGGv}}(M_v)\}$; $C=(DC,HC,PC)$ represents the column analysis axis where $PC=<p^C_{max},\ldots p^C_{min}>$, HC is the *current hierarchy* of column dimension DC; $L=(DL, HL, PL)$ represents the line analysis axis where $PL=<p^L_{max},\ldots p^L_{min}>$, HL is the *current hierarchy* of the line dimension DL; and $R=pred_1 \wedge\ldots\wedge pred_t$ is a normalised conjunction of predicates (restrictions of dimension and fact data). An mTable has the following constraints: $DC \in Star^{Cs}(F^S)$, $DL \in Star^{Cs}(F^S)$, $HC \in H^{DC}$, $HL \in H^{DL}$, $\exists\, p_i \in PC \mid p_i \in HC$, $\exists\, p_i \in PL \mid p_i \in HL$, $\exists\, M_i \in M^S \mid M_i \in M^{FS}$, $F^S \in F^{Cs}$.

The $DISPLAY^{Cs}$ operator constructs an mTable (T_{RES}) from a constellation Cs. The expression is $DISPLAY^{Cs}(F^S, M^S, DL, HL, DC, HC) = T_{RES}$ where: $M^S=\{f_1(M_1),\ldots,f_x(M_x)\}$, $\forall i \in [1..x]$, $M_i \in M^F$ and $T_{RES}=(S_{RES}, L_{RES}, C_{RES}, R_{RES})$ is the output mTable, where: $S_{RES} = (F^S, M^S)$; $L_{RES}=(DL, HL, <p^L_{max}, p^L_{min}>)$, where $p^L_{max}=All$ and $p^L_{min}=p^{DL}_{np}$; $C_{RES}=(DC, HC, <p^C_{max}, p^C_{min}>)$, where $p^C_{max}=All$ and $p^C_{min}=p^{DC}_{np}$; and

$$R_{RES} = \bigwedge_{\forall i, D_i \in Star^C(F^S)} D_i.ALL = 'all' .$$

In the following figure is displayed a constellation allowing the analysis of importing companies activity as well as their manpower. An mTable shows an analysis example: total import amounts by years and by continent of origin of the suppliers.



DISPLAY (IMPORTS, {*SUM*(Amout)},
SUPPLIERS, HGEO, DATES, HTIME)=T_{R1}

		DATES | HTIME			
IMPORTS SUM (AMOUNT)		YEAR	2003	2004	2005
SUPPLIERS | HGEO	CONTINENT				
	America		(560)	(160)	(200)
	Asia		(480)	(310)	(610)
PRODUCTS.All = 'all' COMPANIES.All = 'all'					

$T_{R1}=(S_{R1}, L_{R1}, C_{R1}, R_{R1})$ with:

T_{R1}
S_{R1}=(IMPORTS, {SUM(Amounts)}
L_{R1}=(SUPPLIERS, HGeo, <All, Continent>)
C_{R1}=(DATES, HTime, <All, Year>)
R_{R1}=(COMPANIES.All='All'
AND SUPPLIERS.All='All'
AND PRODUCTS.All='All'
AND DATES.All='All')

Fig. 2. Example of a constellation and an associated analysis (mTable T_{R1})

3 OLAP Algebra

Multidimensional OLAP analyses consist in exploring interactively constellation data. The following algebra allows manipulation and retrieval of data from a constellation through nested expressions of algebraic operators. The OLAP algebra provides a minimal core of operators, all operators may not be expressed by any combination of other core operators; this ensures the *minimality* of the core. Although there is no consensus on a set of operations for a multidimensional algebra, most papers offer a partial support of seven operation categories see [25] for more details. Based on these categories, our algebraic language allows the presentation of analysis data in an mTable.

- **Drilling:** these operations allow navigating through the hierarchical structure along analysis axes, in order to analyse measures with more or less precision. Drilling upwards (operator: ROLLUP) consists in displaying the data with a coarser level of detail; *e.g.* import amounts analysed by months then analysed by year. The opposite, drilling downwards (DRILLDOWN) consists in displaying the data with a finer level of detail.

- **Selections:** these operations (SELECT) allow the specification of restriction predicates on fact or dimension data. This operation is also known as “slice/dice” [2].

- **Rotations:** these operations allow: changing an analysis axis by a new dimension (ROTATE); changing the subject of analysis by a new fact (FROTATE); and changing an analysis perspective by a new hierarchy (HROTATE).

Some authors have also presented complementary operations:

- **Fact modification:** These operations allow the modification of the set of selected measures. They allow adding (ADDM) or removing (DELM) a measure to the current analysis.

- **Dimension modification:** These operations allow more flexibility in analysis. They allow converting a dimensional element into a subject (PUSH), thus “pushing” a parameter into the subject; or converting a subject into a dimensional element (PULL), thus “pulling” a measure out of the subject. They also allow nesting (NEST). It is a structural reordering operation. It allows changing the order of parameters in a hierarchy but it also allows adding in a hierarchy a parameter from another dimension. The consequence is to be able to display in the bi-dimensional mTable more than two dimensions. This compensates for the 2D limitation of the bi-dimensional table.

- **Ordering:** these operations allow changing the order of the values of dimension parameters or inserting a parameter in another place in a hierarchy. Switching (SWITCH) is an operation that eases analysis allowing regrouping columns or lines together independently of the order of the parameter values in lines or columns. It is a value reordering operation. Notice that switching as well as nesting may “break” hierarchies’ visual representation in the mTable. Contrarily to commercial software which allows this operation without warning the user, the use of these specific operators allows a query system to warn the user analysis incoherence risks.

- **Aggregation:** This operation allows the process of totals and subtotals of the displayed data (AGGREGATE). If applied to all displayed parameters, it is equivalent to the Cube operator [11]. The reverse operation is UNAGGREGATE.

The following table describes the symbolic representation (algebraic expression) and the necessary conditions of input and output semantics for each operator. We invite the reader to consult [25] for more detailed specifications.

3.1 Minimal Closed Algebraic Core of OLAP Operators

To ensure *closure* of the core, each operator takes as input a source mTable (T_{SRC}) and produces as output a result mTable (T_{RES}). T_{SRC}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC}) where:

$S_{SRC} = (F^S, M^S)$ with $M^S=\{f_1(M_1),\ldots,f_x(M_x)\}$; $L_{SRC} = (DL, HL, PL)$ with $PL=<All, p_{Lmax},\ldots, p_{Lmin}>$; $C_{SRC} = (DC, HC, PC)$ with $PC=<All, p_{Cmax},\ldots, p_{Cmin}>$; and $R_{SRC} = (pred_1 \wedge\ldots\wedge pred_t)$. $T_{RES}=(S_{RES}, L_{RES}, C_{RES}, R_{RES})$.

We have also defined advanced operators as well as set operators but due to lack of space, we invite the reader to consult [25] for complete specification.

Table 1. OLAP Algebraic operators

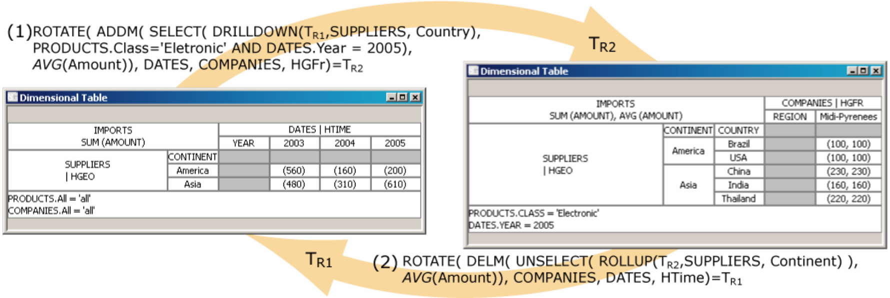
Operation	Semantics
DRILLDOWN(T_{SRC}, D, p_i) = T_{RES}	
Conditions	$D\in\{DC, DL\}$; $(D=DC, \exists p_i\in HC \mid level(pi)<level(p_{Cmin})) \vee (D=DL, \exists p_i\in HL \mid level(p_i)<level(p_{Lmin}))$
Output	$T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ where: if D=DL then $L_{RES}=(DL, HL, <All, p^{L}_{lmax},\ldots, p^{L}_{lmin}, p_i>)$ and $C_{RES}=C_{SRC}$; if D=DC then $L_{RES}=L_{SRC}$ and $C_{RES}=(DC, HC, <All, p^{C}_{cmax},\ldots\ p^{C}_{cmin}, p_i>)$.
ROLLUP(T_{SRC}, D, p_i) = T_{RES}	
Conditions	$D\in\{DC, DL\}$; $(D=DC, \exists p_i\in HC \mid level(p_i)>level(p_{Cmin})) \vee (D=DL, \exists p_i\in HL \mid level(p_i)>level(p_{Lmin}))$.
Output	$T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ where: if D=DL then $L_{RES}=(DL, HL, <All, p^{L}_{lmax},\ldots, p_i>)$ and $C_{RES}=C_{SRC}$; if D=DC then $L_{RES}=L_{SRC}$ and $C_{RES}=(DC, HC, <All, p^{C}_{cmax},\ldots, p_i>)$.
SELECT(T_{SRC}, pred) = T_{RES} / UNSELECT(T_{SRC})=T_{RES}	
Conditions	$pred= pred_1 \wedge\ldots\wedge pred_t$, $pred_i$ is a predicate on $dom(M_i)$ or $dom(p_i)$, $M_i\in M^{FS} \wedge p_i\in A^D \wedge D\in Star^{Cs}(F^S)$
Output	$T_{RES}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{RES})$ where R_{RES} = pred. For UNSELECT, $R_{RES}=\varnothing$
ROTATE(T_{SRC}, D_{old}, D_{new} ,H_{new}) = T_{RES}	
Conditions	$D_{old}\in\{DC,DL\}$; $D_{new}\in Star^{Cs}(F^s) \wedge H_{new}\in H^{Dnew}$
Output	$T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ where: if D_{old}=DL then $L_{RES}=(D_{new}, H_{new}, <All, p^{Dnew}_{np}>)$ and $C_{RES}=C_{SRC}$; if D_{old}=DC then $L_{RES}=L_{SRC}$ and $C_{RES}=(D_{new}, H_{new}, <All, p^{Dnew}_{np}>)$. Where p^{Dnew}_{np} is the coarser-granularity parameter
ADDM(T_{SRC},$f_i(M_i)$) = T_{RES}	
Conditions	$M_i\in M^{Fs} \wedge f_i(M_i)\notin M^S$
Output	$T_{RES}=(S_{RES}, L_{SRC}, C_{SRC}, R_{SRC})$ where: $S_{RES}=(F^S, \{f_1(M_1),\ldots,f_x(M_x),f_i(M_i)\})$.
DELM(T_{SRC},$f_i(M_i)$) = T_{RES}	
Conditions	$f_i(M_i)\in M^S \wedge \|M^S\|>1$.
Output	$T_{RES}=(S_{RES}, L_{SRC}, C_{SRC}, R_{SRC})$ where: $S_{RES}=(F^S, \{f_1(M_1),\ldots,f_{i-1}(M_{i-1}),f_{i+1}(M_{i+1}),\ldots,f_x(M_x)\})$. This operator may not remove the last displayed measure of the subject
PUSH(T_{SRC}, D, p_i) = T_{RES}	
Conditions	$p_i\in H \wedge H\in H^D \wedge D\in Star^{Cs}(F^s)$.
Output	$T_{RES}=(S_{RES}, L_{SRC}, C_{SRC}, R_{SRC})$ where: $S_{RES}=(F^S, \{f_1(M_1),\ldots,f_x(M_x), p_i\})$ and $p_i\notin H$.
PULL(T_{SRC},$f_i(M_i)$, D) = T_{RES}	
Conditions	$M_i\in M^{Fs} \wedge f_i(M_i)\notin M^S \wedge D\in\{DC,DL\}$.
Output	$T_{RES}=(S_{RES}, L_{RES}, C_{RES}, R_{SRC})$ where: $S_{RES}=(F^S, \{f_1(M_1),\ldots,f\text{i-1}(\text{Mi-1},\ldots,f_x(M_x), p_i\})$ and if D=DL then $L_{RES}=(DL, HL, <All, p_{Lmax},\ldots,p_{Lmin},f_i(M_i)>)$ and $C_{RES}=C_{SRC}$, if D=DC then $L_{RES}=L_{SRC}$ and $C_{RES}=(DC, HC, <All, p_{Cmax},\ldots, p_{Cmin},f_i(M_i)>)$
NEST(T_{SRC}, D, p_i, D_{nested}, p_{nested}) = T_{RES}	
Conditions	$D\in\{DC,DL\} \wedge (D=DC, p_i\in PC \vee D=DL, p_i\in PL) \wedge D_{nested}\in Star^{Cs}(F^S) \wedge p_{nested}\in A^{Dnested}$
Output	$T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ where: if D=DL then $L_{RES}=(DL, HL, <All, p_{Lmax},\ldots, p_i, p_{nested},\ldots, p_{Lmin}>)$ and $C_{RES}=C_{SRC}$; if D=DC then $L_{RES}=L_{SRC}$ and $C_{RES}=(DC, HC, <All, p_{Cmax},\ldots, p_i, p_{nested},\ldots, p_{Cmin}>)$
SWITCH(T_{SRC}, D, p_i, v_x, v_y) = T_{RES}	
Conditions	$D\in\{DC,DL\} \wedge p_i\in PC \wedge v_x\in dom(p_i) \wedge v_y\in dom(p_i) \wedge dom(p_i)=<v_{min},\ldots v_x,\ldots v_y,\ldots, v_{max}>$.
Output	$T_{RES}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC})$ where: $dom(p_i)=<v_{min},\ldots v_y,\ldots v_x,\ldots, v_{max}>$

Table 1. (*continued*)

AGGREGATE(T_{SRC}, D, $f_i(p_i)$) = T_{RES}	
Conditions	$D \in \{DC, DL\} \wedge (D=DC, p_i \in PC \vee D=DL, p_i \in PL) \wedge dom(p_i)=<v_{min},\ldots,v_{max}>$, $f_i \in \{SUM, AVG, MIN, MAX, COUNT\ldots\}$.
Output	$T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ where: If D=DL then in L_{RES}, $dom(p_i)$ changes in PL: $dom(p_i)=<v_{min},f_i(v_{min}),\ldots,v_{max},f_i(v_{max})>$. Each initial value v_j of p_i is completed by the aggregation value $f_i(v_j)$ and $C_{RES}=C_{SRC}$; If D=DC then $L_{RES}=L_{SRC}$ and in C_{RES}, $dom(p_i)$ changes in PC: $dom(p_i)=<v_{min},f_i(v_{min}),\ldots,v_{max},f_i(v_{max})>$. Each initial value v_j of p_i is completed by the aggregation value $f_i(v_j)$
UNAGGREGATE(T_{SRC}) = T_{RES}	
Conditions	$\exists p_i \in PL \vee \exists p_i \in PC \mid dom(p_i)=<v_{min},f_i(v_{min}),\ldots,v_{max},f_i(v_{max})>$.
Output	$T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ where: $\forall p_i \in PL$ and $\forall p_i \in PC$, $dom(p_i)=<v_{min},\ldots, v_{max}>$, i.e. the previously aggregated values are removed from $dom(p_i)$

3.2 Example

The decision-maker proceeds with the previous analysis (see T_{R1} in figure 2) by focussing his observations on the average *import amounts* in *2005* for *electronic products*. He also wishes to refine the analysis by visualizing amounts more precisely by *country* of origin of the *suppliers* while modifying the column axis to observe the measures by importing *company*. In order to do so, this complex query is specified by a combination of several basic operators from the algebra core: 1) DRILLDOWN on the SUPPLIER axis; 2) SELECT of 'Electronic' PRODUCTS during year 2005; 3) ADDM for the indicator: AVG(Amount); and 4) ROTATE dimensions: DATES and COMPANY. The complete algebraic expression (1) and the corresponding mTable (T_{R2}) are displayed in the following figure. In the same way, the reverse operation combination is presented (2).



(1)ROTATE(ADDM(SELECT(DRILLDOWN(T_{R1},SUPPLIERS, Country), PRODUCTS.Class='Eletronic' AND DATES.Year = 2005), *AVG*(Amount)), DATES, COMPANIES, HGFr)=T_{R2}

IMPORTS SUM (AMOUNT)		DATES | HTIME			
		YEAR	2003	2004	2005
SUPPLIERS | HGEO	CONTINENT				
	America		(560)	(160)	(200)
	Asia		(480)	(310)	(610)

PRODUCTS.All = 'all'
COMPANIES.All = 'all'

IMPORTS SUM (AMOUNT), AVG (AMOUNT)			COMPANIES | HGFR	
			REGION	Midi-Pyrenees
SUPPLIERS | HGEO	CONTINENT	COUNTRY		
	America	Brazil		(100, 100)
		USA		(100, 100)
	Asia	China		(230, 230)
		India		(160, 160)
		Thailand		(220, 220)

PRODUCTS.CLASS = 'Electronic'
DATES.YEAR = 2005

(2) ROTATE(DELM(UNSELECT(ROLLUP(T_{R2},SUPPLIERS, Continent)), *AVG*(Amount)), COMPANIES, DATES, HTime)=T_{R1}

Fig. 3. Example of algebraic manipulations: how to obtain T_{R2} from T_{R1} (1) and vice-versa (2)

4 OLAP Graphic Language

This section presents an incremental graphic language, which is complete with regard to the algebra core and operating directly on the conceptual elements of the constellation model to stay closer to decision-makers view. As core operators may be combined together, algebraic queries may end up being very complex. The graphic language eases such expressions by providing some simple manipulations hiding this complexity to the user.

4.1 Principles

We have developed a tool composed of two interfaces. The first allows the display of a conceptual graph representing the constellation. The second is the visual representation of the mTable. Both are displayed in figure 1. The user specifies the operations using drag and drop actions or contextual menus. Moreover the mTable supports incremental On-Line Analytical Processing; *e.g.*, the mTable components may be removed, replaced and new components may be added. The mTable display adapts itself after each manipulation.

Using drag and drop, the user may select a graphic icon representing a multidimensional element displayed in the graph, drag it onto the mTable and then drop it in one of the mTable zones (see Figure 4-left), thus directly specifying the resulting mTable. The mTable builds itself incrementally as the elements are dropped into position. In order to ensure consistency during multidimensional OLAP analyses, the user is guided throughout the process: incompatible operations with the ongoing analysis are deactivated, i.e. operations that do not meet conditions. This forbids the user to create erroneous analyses. In the same way, the user may also "move" elements by dragging them from the mTable and dropping them into another zone of the same mTable.

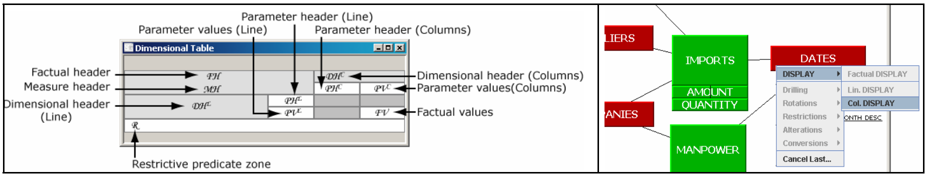


Fig. 4. mTable drop zones (left) and a contextual menu called on the dimension *DATES* (right)

Alternatively, the user may use contextual menus called upon a multidimensional element either displayed in the constellation graph or in the mTable. For example, in Figure 4-right the user designates the DATES dimension to be displayed in columns.

In some cases, in order to ensure the proper execution of the operation, the system might ask the user with dialog boxes complementary information that may not be provided by the graphic operation's context (designated by '**?**' in formal specifications). Formally, each operation takes as input a component of the multidimensional graph: a fact F, a measure M_i, a dimension D, a hierarchy H or a parameter p_i.

4.2 Graphic Definition of a mTable

An mTable is defined once the three major elements have been specified, i.e. in T_{RES} ($S_{RES} \neq \varnothing \wedge L_{RES} \neq \varnothing \wedge C_{RES} \neq \varnothing$). Three of the display-oriented operations allow the specification of the elements of the DISPLAY instruction: DIS_SUBJ to select the subject: F^S and M^S and DIS_COL (respectively DIS_LN) for the definition of the

Table 2. Graphic specification of the DISPLAY operator

Source element of the action (and conditions)	Algebraic equivalent
Subject specification : DIS_SUBJ(E)	
$E=F \mid F \in F^{Cs}$	DISPLAY(N^{Cs}, F^S, M^S, DL, HL, DC, HC) with $F^S=F$ and $M^S=\{f_1\mathbf{?}(M_1),\ldots f_w\mathbf{?}(M_w)\})$
$E=M_i \mid M_i \in M^F \wedge F \in F^{Cs}$	DISPLAY(N^{Cs}, F^S, M^S, DL, HL, DC, HC) with $F^S=F$, $M^S=\{f_i\mathbf{?}(M_i)\}$
Column specification: DIS_COL(E)	
$E=D \mid \exists\ H \in H^D \wedge D \in D^{Cs}$	DISPLAY(N^{Cs}, F^S, M^S, DL, HL, DC, HC) with DC=D, HC=H**?**
$E=H \mid H \in H^D \wedge D \in D^{Cs}$	DISPLAY(N^{Cs}, F^S, M^S, DL, HL, DC, HC) with DC=D, HC=H
$E=p_i \mid p_i \in H \wedge H \in H^D \wedge D \in D^{Cs}$	DRILLDOWN(ROLLUP(DISPLAY(N^{Cs}, F^S, M^S, DL, HL, DC, HC), DC, All), DC, p_i) with DC=D, HC=H**?**
Line specification: DIS_LN(E), same as lines but specifies DL and HL	

The following constraints must be verified: $DC \in Star^{Cs}(F^S) \wedge DL \in Star^{Cs}(F^S)$.
? = complementary information that might be necessary.

column axis: DC and HC (resp. the line axis: DL and HL). The mTable is built incrementally as the user specifies the different elements. Formal specifications are presented in the following table.

4.3 Graphic Manipulation of a mTable

Once the mTable is defined it switches to alteration mode and all other operations are available. A total of twelve graphic operations split into three operation categories are available: display-oriented operations, mTable modification oprations and drilling operations.

The three previous operations (DIS_SUBJ, DIS_COL and DIS_LN) no longer specify a DISPLAY instruction. Instead they now allow the designation of elements that are to replace previously displayed ones in the mTable. DIS_SUBJ, now allows to replace the previous fact; to add measures (ADDM) and to display a parameter as a measure (PUSH). DIS_COL (respectively DIS_LN) replaces the current elements displayed in columns (resp. lines) with the newly specified one. With these two instructions, the user may also ask to display a measure as a parameter in columns or in lines (PULL). In addition to these instructions, the system also provides three others (INS_LN, INS_COL and DEL). INS_COL (resp. INS_LN) allow to insert an element in columns (resp. in lines) enabling nesting (NEST) and to drilling downwards (DRILLDOWN). Finally the deletion operation (DEL) is a very handy operation allowing to remove a displayed component from the mTable. This operation replaces complex algebraic expressions. These six operations are the contextual menu equivalent of most of the drag and drop actions.

The operations that modify the mTable display structure consist in four operations, identical to their algebraic equivalent (SELECT, SWITCH, AGGREGATE and UNAGGREGATE). Apart from AGGREGATE, the other operations may also be expressed using drag and drop. Table 3 presents the equivalence between graphical operation names and drag and drop actions.

Table 3. Correspondance between graphic operations and drag and drop actions

Graphic operation	Drag and drop action
DIS_SUBJ	drop an element into factual header
DIS_COL	drop an element into column dimensional header
DIS_LN	drop an element into line dimensional header
INS_COL	drop an element into column parameter header$^{(1)}$
INS_LN	drop an element into line parameter header$^{(1)}$
DEL	drag an element outside the mTable
SELECT	drag an element into the restriction zone
SWITCH	select a parameter value and drop it on another value of the same parameter
UNAGGREGATE	drag an aggregated parameter value outside the mTable
$^{(1)}$= if the instruction requires it, the position where the element is dropped within the header is used as complementary information.	

Finally as drilling operations are the heart of multidimensional OLAP analyses, the graphic language has two specific operations (ROLLUP and DRILLDOWN), identical to their algebraic equivalent although part of these operations may be done by the use of the display-oriented operations.

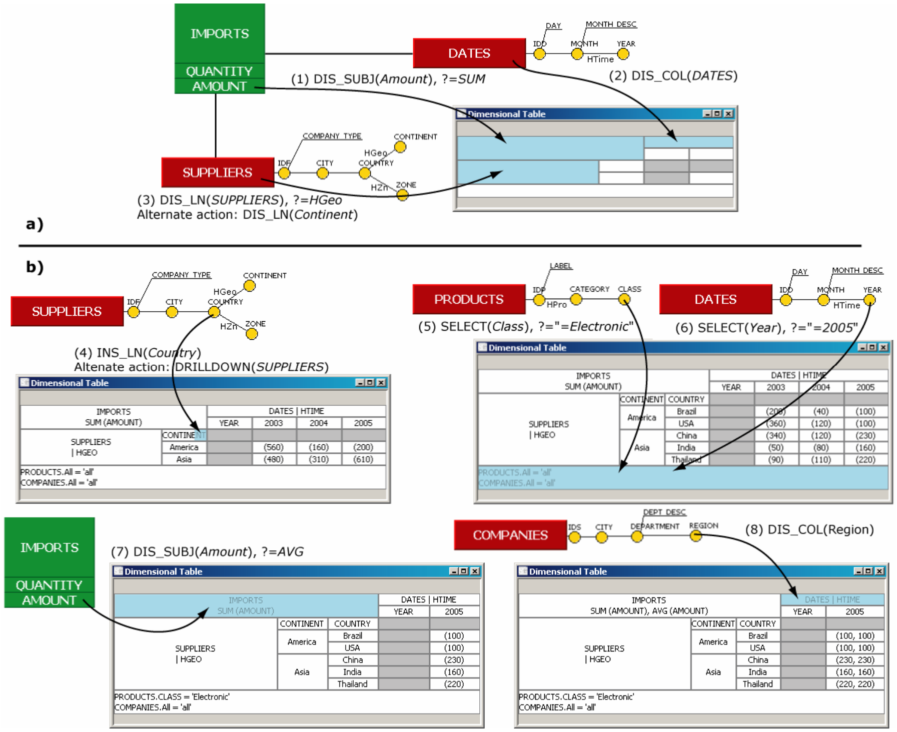
Formal specifications are available in appendix. Unfortunately, due to lack of space, we invite the reader to consult [25] for the complete specification of the graphic language.

4.4 Example

In this section we show the same example as in section 3.2. The analyst wishes to create the mTable T_{R1} then manipulate it in order to obtain T_{R2} (see figure 2). First the analyst executes three operations to create T_{R1}: 1) drag the icon representing the measure *Amount* and dropping it in the factual header and then selecting *SUM* as desired aggregation function; 2) drag the dimension *Dates* and dropping it in the column dimensional header; and 3) drag the dimension *Suppliers* and dropping it in the line dimensional header and select *HGEO* as the hierarchy to be used (see figure 3a). Notice that instead of this last action, the user may drag the parameter *Continent*, thus avoiding specifying the hierarchy. In the figure, one may find the formal specifications and alternative actions. These actions may be expressed through contextual menus or drag and drop actions.

The analyst then modifies T_{R1} with five operations to obtain T_{R2}: 4) drag the parameter *Country* into the right part of the line parameter header (dragging it in the center would replace the previously displayed parameter); 5) (resp. 6) drag the parameter Class (resp. Year) onto the restriction zone and specify *='Electronic'* (resp. *=2005*); 7) drag the measure *Amount* into the factual header and select the aggregation function *AVG* (Average); and 8) drag the parameter *Region* into the column dimensional header (see figure 3b). After each graphic operation, the mTable updates itself, thus the decision-maker may stop or decide to change his analysis after each new operation.

In the following figure blue zones (light grey) underneath each arrow are displayed by the mTable according to the context of the action. This indicates to the user where



IMPORTS SUM (AMOUNT)		DATES | HTIME			
		YEAR	2003	2004	2005
SUPPLIERS | HGEO	CONTINENT				
	America		(560)	(160)	(200)
	Asia		(480)	(310)	(610)

PRODUCTS.All = 'all'
COMPANIES.All = 'all'

IMPORTS SUM (AMOUNT)			DATES | HTIME			
			YEAR	2003	2004	2005
SUPPLIERS | HGEO	CONTINENT	COUNTRY				
	America	Brazil		(200)	(40)	(100)
		USA		(360)	(120)	(100)
	Asia	China		(340)	(120)	(230)
		India		(50)	(80)	(160)
		Thailand		(90)	(110)	(220)

PRODUCTS.All = 'all'
COMPANIES.All = 'all'

IMPORTS SUM (AMOUNT)			DATES | HTIME	
			YEAR	2005
SUPPLIERS | HGEO	CONTINENT	COUNTRY		
	America	Brazil		(100)
		USA		(100)
	Asia	China		(230)
		India		(160)
		Thailand		(220)

PRODUCTS.CLASS = 'Electronic'
COMPANIES.All = 'all'

IMPORTS SUM (AMOUNT), AVG (AMOUNT)			DATES | HTIME	
			YEAR	2005
SUPPLIERS | HGEO	CONTINENT	COUNTRY		
	America	Brazil		(100, 100)
		USA		(100, 100)
	Asia	China		(230, 230)
		India		(160, 160)
		Thailand		(220, 220)

PRODUCTS.CLASS = 'Electronic'
COMPANIES.All = 'all'

Fig. 5. (a) Drag and drop actions to build T_{R1}; (b) successive actions to generate T_{R2} from T_{R1}

the dropped element will go, thus, for example, if it will be inserted within already displayed elements (action 4 in figure 3b) or if it will replace one or all a set of displayed elements (action 8 in figure 3b).

Manipulations with drag and drop actions and contextual menus are complete with regard to the algebraic core. Due to lack of space, we invite the reader to consult [25] for more details.

5 Conclusion

The aim of this paper is to define a user-oriented multidimensional OLAP analysis environment. In order to do so we have defined a conceptual model modelling multidimensional elements with the concepts of fact, dimension and hierarchy and representing them through a constellation. Based on this model, we defined an OLAP query algebra, formally specifying operations executed by decision-makers. The algebra is composed of a core of operators that may be combined to express complex analysis queries. As such a language is inappropriate for users, we defined a graphical query language based on this algebra. This language allows decision makers to express multidimensional OLAP analysis queries and it is complete with regard to the algebraic core.

Compared to actual solutions, our proposition has the advantages of providing: 1) a global view of the analysis data (with the constellation graph); 2) manipulation of high-level concepts (direct manipulation of the conceptual elements); 3) analysis correlations (analysing different facts with common dimensions); 4) analysis coherence during manipulations (usage of hierarchies within the mTable); and if necessary 5) as much flexibility as actual solutions, i.e. disregarding analysis coherence (usage of the NEST operator), but warning the decision-maker of incoherence risks. As in [20], we argue that the use of conceptual representations eases query specifications for decision-makers. Moreover, the graphical language must be as expressive and as robust as complex multidimensional algebraic query languages.

The algebraic and graphic languages have been validated by their implementation in a prototype (figures in this paper are screen captures). The prototype is based on an implementation of the multidimensional concepts in a ROLAP environment with the DBMS Oracle *10g*. The client is a java application composed of over a hundred classes and other components such as JGraph and Javacc.

As the majority of industrial implementations are relational, for a future work, we intend to complete this proposition by the specification of a transformation process, mapping multidimensional operations into an optimal combination of relational operators.

References

1. Abelló, A., Samos, J., Saltor, F.: YAM2: a multidimensional conceptual model extending UML. Information Systems (IS) 31(6), 541–567 (2006)
2. Agrawal, R., Gupta, A., Sarawagi, S.: Modeling Multidimensional Databases. In: 13th Int. Conf. Data Engineering (ICDE), pp. 232–243. IEEE Computer Society, Los Alamitos (1997)
3. Böhnlein, M., Plaha, M., Ulbrich-vom Ende, A.: Visual Specification of Multidimensional Queries based on a Semantic Data Model. In: Vom Data Warehouse zum Corporate Knowledge Center (DW), pp. 379–397. Physica-Verlag, Heidelberg (2002)
4. Cabibbo, L., Torlone, R.: Querying Multidimensional Databases. In: Cluet, S., Hull, R. (eds.) Database Programming Languages. LNCS, vol. 1369, pp. 319–335. Springer, Heidelberg (1998)
5. Cabibbo, L., Torlone, R.: From a Procedural to a Visual Query Language for OLAP. In: 10th Int. Conf. on Scientific and Statistical Database Management (SSDBM), pp. 74–83. IEEE Computer Society, Los Alamitos (1998)
6. Codd, E.F.: Providing OLAP (On Line Analytical Processing) to user analysts: an IT mandate, Technical report, E.F. Codd and Associates (1993)
7. Datta, A., Thomas, H.: The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Decision Support Systems (DSS) 27(3), 289–301 (1999)
8. Franconni, E., Kamble, A.: The GMD Data Model and Algebra for Multidimensional Information. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 446–462. Springer, Heidelberg (2004)
9. Golfarelli, M., Maio, D., Rizzi, S.: The Dimensional Fact Model: A Conceptual Model for Data Warehouses. International Journal of Cooperative Information Systems (IJCIS) 7(2-3), 215–247 (1998)

10. Golfarelli, M., Rizzi, S., Saltarelli, E.: WAND: A Case Tool for Workload-Based Design of a Data Mart. In: 10th National Convention on Systems Evolution for Data Bases, pp. 422–426 (2002)
11. Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. In: 12th Int. Conf. on Data Engineering (ICDE), pp. 152–159. IEEE Computer Society, Los Alamitos (1996)
12. Gyssen, M., Lakshmanan, L.V.S.: A Foundation for Multi-Dimensional Databases. In: 23rd Int. Conf. on Very Large Data Bases (VLDB), pp. 106–115. Morgan Kaufmann, San Francisco (1997)
13. Kimball, R.: The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses, 2nd edn. John Wiley & Sons, Inc., Chichester (2003)
14. Lehner, W.: Modeling Large Scale OLAP Scenarios. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 153–167. Springer, Heidelberg (1998)
15. Li, C., Wang, X.S.: A Data Model for Supporting On-Line Analytical Processing. In: 5th Int. Conf. on Information and Knowledge Management (CIKM), pp. 81–88. ACM, New York (1996)
16. Malinowski, E., Zimányi, E.: Hierarchies in a multidimensional model: From conceptual modeling to logical representation. J. of Data & Knowledge Engineering (DataK) 59(2), 348–377 (2006)
17. Niemi, T., Hirvonen, L., Jarvelin, K.: Multidimensional Data Model and Query Language for Informetrics. Wiley Periodicals 54(10), 939–951 (2003)
18. Pedersen, T., Jensen, C., Dyreson, C.: A foundation for capturing and querying complex multidimensional data. Information Systems (IS) 26(5), 383–423 (2001)
19. Rafanelli, M.: Operators for Multidimensional Aggregate Data. In: Multidimensional Databases: Problems and Solutions. ch. 5, pp. 116–165. Idea Group Inc. (2003)
20. Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J.: Research in data warehouse modeling and design: dead or alive? In: 9th Int. Workshop on Data Warehousing and OLAP (DOLAP), pp. 3–10. ACM, New York (2006)
21. Sapia, C., Blaschka, M., Höfling, G.: Dinter: Extending the E/R Model for the Multidimensional Paradigm. In: Kambayashi, Y., Lee, D.-L., Lim, E.-p., Mohania, M.K., Masunaga, Y. (eds.) Advances in Database Technologies. LNCS, vol. 1552, pp. 105–116. Springer, Heidelberg (1999)
22. Sifer, M.: A Visual Interface Technique for Exploring OLAP Data with Coordinated Dimension Hierarchies. In: Int. Conf. on Information and Knowledge Management (CIKM), pp. 532–535. ACM, New York (2003)
23. Stolte, C., Tang, D., Hanrahan, P.: Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. IEEE Trans. Vis. Comput. Graphics (TVCG) 8(1), 52–65 (2002)
24. Torlone, R.: Conceptual Multidimensional Models. In: Rafanelli, M. (ed.) Multidimensional Databases: Problems and Solutions. ch. 3, pp. 69–90. Idea Group Inc. (2003)
25. Tournier, R.: OLAP model, algebra and graphic language for multidimensional databases. Scientific Report n° IRIT/RR—2007-6–FR, IRIT, Université Paul Sabatier (Toulouse 3), France
26. Trujillo, J.C., Luján-Mora, S., Song, I.: Applying UML for designing multidimensional databases and OLAP applications. In: Siau, K. (ed.) Advanced Topics in Database Research, vol. 2, pp. 13–36. Idea Group Publishing (2003)

A Appendix

This section provides tables with the formal specifications of the graphic operations. Each operation takes as input a component of the multidimensional graph: E (F, M_i, D, H, p_i) or a specific element of the mTable: X. They may also require complementary information notified by '**?**'. Only column operation specifications are specified (line specification are identical.

Table 4. Formal specification of the display-oriented graphic operations

Source element of the action (and conditions)	Algebraic equivalent
Operation for displaying as subject: DIS_SUBJ(E)	
$E=F^S$	History$^{(1)}$ $(T_{SRC}, DL, \text{History}(T_{SRC}, DC, \text{DISPLAY}(N^{CS}, F^S, \{f_1(M_1),\ldots,f_w(M_w)\}, DL, HL, DC, HC))), \forall i\in[1..w], M_i\in M^{FS}$
$E=F_{new}\mid F_{new}\neq F^S\wedge DC\in Star^{Cs}(F^{new})\wedge DL\in Star^{Cs}(F^{new})$	History$^{(1)}$$(T_{SRC}, DL, \text{History}(T_{SRC}, DC, \text{DISPLAY}(N^{CS}, F_{new}, \{f_1(M_1),\ldots,f_w(M_w)\}, DL, HL, DC, HC))), \forall i\in[1..w], M_i\in M^{Fnew}$
$E=M_i \mid M_i\in M^S$	ADDM$(T_{SRC},f_i\mathbf{?}(M_i))$ with $f_i(M_i)\notin M^S$
$E=M_i \mid M_i\notin M^S \wedge M_i\in M^{FS}$	ADDM$(T_{SRC},f_i\mathbf{?}(M_i))$
$E=M_i \mid M_i\notin M^{FS} \wedge M_i\in M^{Fnew} \wedge DC\in Star^{Cs}(F^{new})\wedge DL\in Star^{Cs}(F^{new})$	History$^{(1)}$$(T_{SRC}, DL, \text{History}(T_{SRC}, DC, \text{DISPLAY}(N^{CS}, F_{new}, \{f_i\mathbf{?}(M_i)\}, DL, HL, DC, HC)))$
$E=p_i \mid \forall p_i\in A^{Dnew}\wedge D_{new}\in Star^{Cs}(F^S)$	PUSH(T_{SRC}, D_{new}, p_i)
Operation for displaying in columns: DIS_COL(E) (DIS_LN for lines)	
$E=M_i \mid M_i\in M^{FS}$	PULL$(T_{SRC},f_i\mathbf{?}(M_i), DC)$
$E=DC$	ROLLUP$(T_{SRC}, DC, p^{DC}{}_{np})$, the display of HC is reset on the most general parameter: $p_{Cmin}= p_{Cmax}=p^{DC}{}_{np}$
$E=D_{new}\mid D_{new}\neq DC\wedge D_{new}\in Star^{Cs}(F^S)$	ROTATE$(T_{SRC}, DC, D_{new}, H\mathbf{?})$, with $H\in H^{Dnew}$
$E=HC$	ROLLUP$(T_{SRC}, DC, p^{DC}{}_{np})$, the display of HC is reset on the most general parameter: $p_{Cmin}= p_{Cmax}=p^{DC}{}_{np}$
$E=H_{new} \mid H_{new}\neq HC \wedge H_{new}\in H^{DC}$	ROTATE$(T_{SRC}, DC, DC, H_{new})$ $\|H^{DC}\|>1$, i.e. $\exists H_{new}\in H^{DC} \mid H_{new}\neq HC$
$E=H_{new} \mid H_{new}\notin H^{DC}\wedge H_{new}\in H^{Dnew} \wedge D_{new}\in Star^{Cs}(F^S)$	ROTATE$(T_{SRC}, DC, D_{new}, H_{new})$
$E=p_i \mid p_i\in PC \vee (p_i\notin PC\wedge p_i\in HC)$	DRILLDOWN(ROLLUP$(T_{SRC}$,DC,All),DC, p_i)
$E=p_i \mid p_i\notin HC \wedge p_i\in H_{new}\wedge H\in H^{DC}$	DRILLDOWN(ROLLUP(ROTATE$(T_{SRC}, DC, DC, H_{new})$, DC,All),DC, p_i) $\|H^{DC}\|>1$, i.e. $\exists H_{new}\in H^{DC} \mid H_{new}\neq HC$
$E=p_i \mid p_i\notin A^{DC} \wedge p_i\in A^{Dnew} \wedge D_{new}\in Star^{Cs}(F^S)$	DRILLDOWN(ROLLUP(ROTATE$(T_{SRC}, DC, D_{new} ,H_{new})$, D_{new},All), D_{new}, p_i), with $H_{new}\in H^{Dnew}$
Operation for inserting in columns: INS_COL(E) (INS_LN for lines)	
$E=M_i \mid M_i\in M^{FS}$	PULL$(T_{SRC},f_i\mathbf{?}(M_i), DC)$
$E=DC$	DRILLDOWN(…(DRILLDOWN(ROLLUP$(T_{SRC}, DC, p^{DC}{}_{np})$, DC, $p^{DC}{}_{np-1}$)…), DC, $p^{DC}{}_1$) all parameters of HC are displayed: $PC = Param^{HC}$
$E=D_{new} \mid D_{new}\neq DC\wedge D_{new}\in Star^{Cs}(F^S)$	ROTATE$(T_{SRC}, DC, D_{new}, H_{new}\mathbf{?})$, with $H_{new}\in H^{Dnew}$
$E=HC$	DRILLDOWN(…(DRILLDOWN(ROLLUP$(T_{SRC}, DC, p^{DC}{}_{np})$, DC, $p^{DC}{}_{np-1}$)…), DC, $p^{DC}{}_1$) all parameters of HC are displayed: $PC = Param^{HC}$
$E=p_i \mid p_i\in PC \wedge PC\neq\{All, p_i\}$	NEST(DRILLDOWN(…(DRILLDOWN(ROLLUP$(T_{SRC}$, DC, p_{i+1}), DC, p_{i-1})…), DC, p_{Cmin}), DC, $p_j\mathbf{?}$, DC, p_i), with $\{p_{i+1}, p_i, p_{i-1},\ldots,p_{Cmin}\}\in PC$

Table 4. (*continued*)

$E=p_i \mid p_i \notin PC \wedge p_i \in HC$	DRILLDOWN(T_{SRC}, DC, p_i), if $level^{HC}(p_i)<level^{HC}(p_{Cmin})$ or DRILLDOWN(…(DRILLDOWN(ROLLUP(T_{SRC}, DC, p_i), DC, p_{i+1})…), DC, p_{Cmin}), if $level^{HC}(p_i)>level^{HC}(p_{Cmin})$ [(2)]
$E=p_i \mid p_i \notin HC \wedge p_i \in H \wedge H \in H^{DC}$	NEST(T_{SRC}, DC, p_j**?**, DC, p_i), with $p_j \in PC$
$E=p_i \mid p_i \notin A^{DC} \wedge p_i \in A^{Dnew} \wedge D_{new} \in Star^{Cs}(F^S)$	NEST(T_{SRC}, DC, p_j**?**, D_{new}, p_i), with $p_j \in PC$
Delete operation: DEL(E) ou DEL(X) [(3)]	
$E=F^S$	History[(1)](T_{SRC}, DL, History(T_{SRC}, DC, DISPLAY(N^{CS}, F_{new}**?**, $\{f_1(M_1),\ldots,f_w(M_w)\}$, DL, HL, DC, HC))), $\forall i \in [1..w]\ M_i \in M^{Fnew}$ and $DC \in Star^{Cs}(F_{new})$ and $DL \in Star^{Cs}(F_{new})$
$E=f_i(M_i) \mid f_i(M_i) \in M^S$	DELM(T_{SRC}, $f_i(M_i)$)
E=DC	ROTATE(T_{SRC}, DC, D_{new}**?**, H**?**), $D_{new} \in Star^{Cs}(F^S)$, $H \in H^{Dnew}$
E=HC	ditto above: DEL(DC)
$E=p_i \mid p_i \in PC \wedge PC \neq <All, p_i>$	DRILLDOWN(…(DRILLDOWN(ROLLUP(T_{SRC}, DC, p_{i+1}), DC, p_{i-1})…), DC, p_{Cmin}), $\{p_{i+1}, p_i, p_{i-1},\ldots,p_{Cmin}\} \in PC$
X=R (zone de restriction) [(3)]	SELECT(T_{SRC}, F^S.All='all' $\wedge$ ($\bigwedge_{D_i \in Star^{Cs}(F^S)} D_i.All='all'$))

[(1)] = History(T_{old}, obj, T_{new})=T_R is producing T_R by applying on T_{new} the history of the operations that were applied in T_{old} on obj (dimension or fact).

[(2)] = p_i not displayed ($p_i \notin PC \wedge p_i \in HC$), p_{Ci+1} (respectively p_{Ci-1}) is the attribute immediately inferior (resp. superior) to p_i in PC: $level^{HC}(p_{Ci-1})=level^{HC}(p_i)-1$ (resp. $level^{HC}(p_{Ci+1}) =level^{HC}(p_i)+1$);

[(3)] = This operation is done on the restriction zone (R) of the mTable;

Table 5. Formal specification of the algebra-oriented graphic operations

Source element of the action (and conditions)	Algebraic equivalent
Data restriction: SELECT(E)	
$E=M \mid M \in M^{FS}$	SELECT(pred?), pred is a predicate on dom(M_i)
$E=p_i \mid p_i \in A^D$, $D \in Star^{Cs}(F^S)$	SELECT(pred?), pred is a predicate on dom(p_i)
Line/Column value inversion: SWITCH(X) [(1)]	
$X=val_x \mid val_x \in dom(p_i)$, $p_i \in PC$	SWITCH(T_{SRC}, DC, p_i, val_x, val_y?), with $val_y \in dom(p_i)$
Data aggregation: AGGREGATE(E) and UNAGGREGATE(E)	
$E=p_i \mid p_i \in PC$, $dom(p_i)=<v_1,...,v_x>$	AGGREGATE(T_{SRC}, DC, f_i?(p_i))
$E=p_i \mid p_i \in PC$, $dom(p_i)=<v_1,f_i(v_1),\ldots,v_x,f_i(v_x)>$	UNAGGREGATE(T_{SRC})

Table 6. Formal specification of the graphic drilling operations

Source element of the action (and conditions)	Algebraic equivalent
Drilling downwards: DRILLDOWN(E)	
$E=DC \mid \exists p_i \in HC \mid level^{HC}(p_i)=level^{HC}(p_{Cmin})-1$	DRILLDOWN(T_{SRC}, DC, p_{Cmin-1})[(1)]
$E=HC \mid \exists p_i \in HC \mid level^{HC}(p_i)=level^{HC}(p_{Cmin})-1$	DRILLDOWN(T_{SRC}, DC, p_{Cmin-1})[(1)]
$E=p_i \mid level^{HC}(p_i)>level^{HC}(p_{Cmin})$	DRILLDOWN(T_{SRC}, DC, p_i)
Drilling upwards: ROLLUP(E)	
$E=DC \mid \exists p_i \in HC \mid level^{HC}(p_i)=level^{HC}(p_{Cmin})+1$	ROLLUP(T_{SRC}, DC, p_{Cmin+1})[(1)]
$E=HC \mid \exists p_i \in HC \mid level^{HC}(p_i)=level^{HC}(p_{Cmin})+1$	ROLLUP(T_{SRC}, DC, p_{Cmin+1})[(1)]
$E=p_i \mid level^{HC}(p_i)<level^{HC}(p_{Cmin})$	ROLLUP(T_{SRC}, DC, p_i)

[(1)] p_{Cmin-1} (resp. p_{Cmin+1}) is the parameter immediately inferior (resp. superior) to p_{Cmin} in the list $Param^{HC}$ (i.e. $level^{HC}(p_{Cmin-1})+1=level^{HC}(p_{Cmin})=level^{HC}(p_{Cmin+1})-1$).