

Algebraic and graphic languages for OLAP manipulations

Franck Ravat (1), Olivier Teste (2), Ronan Tournier (3), Gilles Zurfluh (4)

(1)

Université Toulouse I, France
IRIT – Université Toulouse I
2 rue du doyen G. Marty
F-31042 Toulouse Cedex 9, FRANCE
e-mail : ravat@irit.fr
phone number : (+33)5.61.55.74.35
fax number : (+33)5.61.55.62.58

(2)

Université Toulouse III, France
IRIT – Université Paul Sabatier
118, route de Narbonne
F-31062 Toulouse cedex 9, FRANCE
e-mail : teste@irit.fr
phone number : (+33)5.61.55.74.35
fax number : (+33)5.61.55.62.58

(3)

Université Toulouse III, France
IRIT – Université Paul Sabatier
118, route de Narbonne
F-31062 Toulouse cedex 9, FRANCE
e-mail : tournier@irit.fr
phone number : (+33)5.61.55.72.65
fax number : (+33)5.61.55.62.58

(4)

Université Toulouse I
IRIT – Université Toulouse I
2 rue du doyen G. Marty
F-31042 Toulouse cedex 9, FRANCE
e-mail : zurfluh@irit.fr
phone number : (+33)5.61.55.63.23
fax number : (+33)5.61.55.62.58

Algebraic and graphic languages for OLAP manipulations

ABSTRACT

This paper deals with multidimensional analyses. Analysed data are designed according to a conceptual model as a constellation of facts and dimensions, which are composed of multi-hierarchies. This model supports a query algebra defining a minimal core of operators, which produce multidimensional tables for displaying analysed data. This user-oriented algebra supports complex analyses through advanced operators and binary operators. A graphical language, based on this algebra, is also provided to ease the specification of multidimensional queries. These graphical manipulations are expressed from a constellation schema and they produce multidimensional tables.

KEYWORDS

Multidimensional manipulations - OLAP algebra - Graphical query language - Constellation - Decision support systems - OLAP - Data warehouse

INTRODUCTION

As competitiveness increases in the business world, as faster reactivity is more than ever required, the decision making process has become a major focus of research and is increasingly assisted with information technologies. OLAP (On-Line Analytical Processing) systems, aim to ease the decision making process with a multidimensional data presentation. The use of Multidimensional DataBases (MDB) provides a global view of company data, and enables decision-makers to gain insight into an enterprise performance through fast and interactive access to data (Colliat, 1996). Unfortunately, in spite of a decade of research in OLAP systems, concepts and systems exist without uniform theoretical basis (Niemi et al., 2003), (Rizzi et al., 2006).

Context and Related Works

Without a model based on a consensus for multidimensional data, many propositions have been made. Multidimensional models rest upon cube or hyper-cube metaphor. Several surveys may be found in (Chaudhuri and Dayal, 1997), (Blaschka et al., 1998), (Vassiliadis and Sellis, 1999), (Pedersen et al., 2001), (Torlone, 2003) and (Abelló et al., 2006).

The first works, based on a “cube model” that present data in the form of n-dimensional cubes (Agrawal et al., 1997), (Li and Wang, 1996), (Datta and Thomas, 1999), (Gyssens and Lakshmanan, 1997), have the following drawbacks:

- 1) weakness in modelling the fact (subject of analysis) and its Key Performance Indicators (KPI or measures);
- 2) little or no conceptual modelling of dimensions (analysis axes) with no explicit capture of their hierarchical structure;
- 3) no separation between structure and content.

The second category called “multidimensional model” overcomes these drawbacks and it is semantically richer. It allows a precise specification of each multidimensional component

(Lehner, 1998), (Pedersen et al., 2001), (Trujillo et al., 2003), (Abelló et al., 2003) and (Abelló et al., 2006). Models of this category are based on the concepts of fact and dimension. Dimension attributes are organised in hierarchies. A hierarchy defines a point of view (or analysis perspective) of an analysis axis and is composed of the different aggregation levels of the measures. To our knowledge, hardly any multidimensional model provides a combined multi-fact and multi-hierarchy representation.

From a manipulation point of view, the first works on OLAP manipulation algebras extended relational algebra operators for the cube model (Gray et al., 1996), (Agrawal et al., 1997), (Li and Wang, 1996), (Gyssens and Lakshmanan, 1997), (Rafanelli, 2003). To counter the inadaptability of relational algebra for manipulating multidimensional structures in an OLAP context, numerous works provided operations for specifying and manipulating a cube (Cabibbo and Torlone, 1997, 1998), (Pedersen et al., 2001), (Abelló et al., 2003) and (Franconi and Kamble, 2004). These works are not user-oriented (Abelló et al., 2003) for the following reasons: 1) they do not define an adapted structure for displaying decisional data to the user; 2) they are based on partial sets of OLAP operations; and 3) the defined operations do not easily represent OLAP manipulations of decision-makers (Ravat et al., 2006a).

Multidimensional OLAP analyses consist in exploring interactively multidimensional databases by drilling, rotating, selecting and displaying data. Although there is no consensus on a common core of a minimal set of operations for a multidimensional algebra, most papers offer a support of these operation categories:

- **Drilling:** these operations allow navigating through the hierarchical structure of the analysis axes, in order to analyse a measure with more or less precision. Drilling upwards (*roll-up*) consists in displaying the data with a coarser level of detail; e.g. rollup allows changing corporate sales initially displayed by months into sales displayed by years. The

opposite, drilling downwards (*drill-down*) consists in displaying the data with a finer level of detail.

- **Selections:** these operations allow the user to work on a subset of the available data. *Slice* specifies a restriction predicate on dimension data while *Dice* specifies a restriction predicate on fact data.
- **Rotations:** these operations allow changing analysis axes (rotation of dimensions), changing the subject of analysis (rotation of facts or drill-across), or changing an analysis perspective within the same dimension (rotation of hierarchies).

Some authors have also presented additional operations:

- **Fact modification:** these operations allow decision-makers to add and to remove a measure (analysis indicator or KPI) in the current analysis.
- **Dimension modification:** these operations enable the insertion of dimensional attributes into a fact (push operation) or measures into a dimension (pull operation).
- **Ordering:** these operations allow decision-makers to change the order of the values of dimension parameters or to insert a parameter in another place in a hierarchy (nest operation).
- **Set operations:** some authors offer to use union, difference and intersection operations.

The following table summarizes the available operations in various propositions. It also gives the authors' specific names for these operations.

Research Works		(Grouping Algebra) Li and Wang 1996	Agrawal et al., 1997	Gyssens and Lakshmanan, 1997	(MD) Cabbibo and Torlone 1997, 1998	Lehner, 1998	Pedersen et al., 2001	(YAMF) Abelló et al., 2003	(GMD) Franconi et al., 2004	Ravat et al., 2006
Drilling	Finer level	Roll ,Cube	Join			DrillDown ⁽²⁾ , Split		DrillDown		Drill Down
	Coarser level	Roll, Aggregation	Merge	Summerization	RollUp, Aggregation	RollUp, Merge, Aggregation	Aggregation	RollUp		RollUp
Selection	Factual values			Slice (Selection)	Selection			Dice, Projection		
	Dimensional values		Restriction	Dice (Selection)	Selection		Selection		Slice, Multi-Slice ⁽³⁾	
Rotation	Fact							DrillAcross		FRotate
	Dimension							ChangeBase		DRotate
	Hierarchy									HRotate
Fact Modification	Adding a measure		Projection	Projection					Derived measures	
	Suppress a measure		Projection	Projection						
Dimension Modification	Reducing dimensions	Cube Aggregation	Projection, Destroy-Dimension		Simple Projection		Projection		Projection	
	Push		Push	Fold ⁽⁴⁾						
	Pull		Pull	Unfold						
Ordering	Ordering			Classification						
	Nesting	Transfer								Nest
Set operators	Union	Union ⁽⁶⁾	Union ⁽⁶⁾	Union ⁽⁶⁾			Union ⁽⁵⁾	Union ⁽⁶⁾	Union ⁽⁶⁾	
	Intersect		Intersection	Intersection					Intersection	
	Difference		Difference	Difference			difference		Difference	
	Join	RC-Join (Relation to dimension)	join cubes	join cubes	Join ⁽¹⁾		Identity-based Join, Group		join cubes	
Model Structure		Cube	Cube	2D-Table	MD (f-table)	MD	MD	Cube	Cube	2D-Table (MT)
Other Operations		Add dimension		cartesian product	cartesian product					
Comments				provides calculus language	provides graphic language, query calculus			provides SQL translation		provides assertional language

MD=Multidimensional; ⁽¹⁾=no restriction; ⁽²⁾=no hierarchy conservation; ⁽³⁾=specified on a range; ⁽⁴⁾=generalized push; ⁽⁵⁾=on dimensions; ⁽⁶⁾=identical cubes only;

Table 1: A comparison of different multidimensional languages.

Without a complete column, table 1 shows that current research works are incomplete with regard to the different operation categories.

Expressing queries with these algebraic operators is a difficult task for decision-makers. As a consequence, more adapted languages have to be defined. Graphical languages used to specify multidimensional analyses are very present in commercial tools, but despite this, very little attention has been drawn on graphical languages within research on decision support systems. In (Cabbibo and Torlone, 1998), the authors present a graphical multidimensional manipulation language associated to a conceptual representation of the multidimensional structures. Although the authors define a manipulation algebra and calculus, the high level graphical language offers very limited manipulations in comparison. In (Böhnlein et al., 2002), the authors offer an intermediate solution, with more manipulations but the system uses complex forms for query specifications. Neither of these two solutions provide a

restitution interface. (Stolte et al., 2002) and (Sifer, 2003) are advanced restitution tools. The first one offers an impressive pivot table that adapts its display according to the analysed data type, whereas the second offers an arborescent view with multiple scales and very specific manipulations. Neither proposition provides a formal description of the manipulation language.

Microsoft Excel Pivot tables (or DynamiCube from Data Dynamics), although very expressive restitution interfaces, do not provide many dynamic manipulations (especially rotations). On the other hand, other commercial tools offer extensive manipulations (Business Objects¹, Cognos BI², Tableau³, Targit⁴...). 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 for the specification analyses (Rizzi et al., 2006). Moreover, commercial tools lack formal manipulation languages and these languages are not complete with regard to the different operation categories previously described.

Nowadays, decision-makers wish 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 (Niemi et al., 2003) as well as 3) an adapted conceptual view of the multidimensional elements of the underlying MDB (Rizzi et al., 2006). Moreover, these tools sacrifice analysis coherence for analysis flexibility.

Motivation

In this context, in order to ensure access to multidimensional OLAP analyses of company data we intend to define a user-oriented query language composed of a formalised algebra and a

graphic language. The query language has to be based upon: 1) data structures of the MDB close to the user's point of view and an adapted display structure for multidimensional data; 2) algebraic operators allowing the expression of multidimensional OLAP analyses and 3) an incremental graphic query specification. These requirements are detailed in the following:

- 1) Being user-oriented (Abelló et al., 2003), this language has to be based on a model disregarding implementation issues. We intend to define a multidimensional conceptual model, semantically richer than cube models. In order to facilitate correlations between analysis subjects and to analyse measures through different aggregation levels, the model will support multi-fact and multi-hierarchy representations. Contrary to previous works, our objective is also to offer an adapted structure to return analysis data to decision-makers. A n -dimensional cube ($n > 2$) is hardly workable by decision-makers (Gyssens and Lakshmanan, 1997) (Maniatis et al., 2005) and disregards the dimension's hierarchical structure.
- 2) The algebra should provide a set of operators expressing all operations that an analyst may perform. In order to ensure complex OLAP analyses, the algebra must also support operator combinations.
- 3) The use of graphic query languages ease query specifications for the end-user compared with algebraic expressions. Contrary to commercial software that provide an arborescent view of the multidimensional elements, the graphic language should operate on an explicit graphic view of the multidimensional conceptual schema. In the same way, OLAP analysis queries should be expressed directly on the graphic representation in an incremental way.

Paper Contributions and Outline

In order to fulfil our goals, we define in the next section, a conceptual multidimensional model used as a basis for our query language. Section 3 introduces a formal algebraic

language allowing the restitution of analyses in a multidimensional table (MT). In the manner of the relational algebra, we define a minimal core of multidimensional OLAP operators that may be combined together, expressing complex queries. This core is extended by the adjunction of second level operators and binary operators presented in sections 4 and 5. Finally, section 6 specifies an incremental graphic language, which is complete with regard to the algebra core and it operates directly on the conceptual elements of a multidimensional schema.

CONCEPTUAL MODELLING

In this section, the model we define is close to the user's point of view and it is independent of implementation choices. This conceptual Multidimensional DataBase (MDB) model is based on facts, dimensions and hierarchies. This model facilitates correlations between several subjects of analysis through a constellation of facts and dimensions and it supports several data granularities according to which subjects may be analysed.

Concepts

A constellation regroups several subjects of analysis (facts), which are studied according to several analysis axes (dimensions) possibly shared between facts. A constellation extends star schemas (Kimball, 1996), which are commonly used in the multidimensional context.

Definition. A *constellation* C_s is defined as $(N^{C_s}, F^{C_s}, D^{C_s}, Star^{C_s})$ where

- N^{C_s} is a constellation name,
- $F^{C_s} = \{F_1, \dots, F_m\}$ is a set of facts,
- $D^{C_s} = \{D_1, \dots, D_n\}$ is a set of dimensions,
- $Star^{C_s} : F^{C_s} \rightarrow 2^{D^{C_s}}$ associates each fact to its linked dimensions.

The notation 2^X represents the powerset of the set X .

A dimension models an analysis axis; *e.g.* it reflects information according to which subjects of analysis will be analysed. A dimension is composed of attributes (dimension properties).

Definition. A *dimension*, noted $D \in D^{Cs}$, is defined as (N^D, A^D, H^D, I^D) where

- N^D is a dimension name,
- $A^D = \{a_{1, \dots, u}^D\} \cup \{id^D, All\}$ is a set of attributes,
- $H^D = \{H_{1, \dots, v}^D\}$ is a set of hierarchies,
- $I^D = \{i_{1, \dots, p}^D\}$ is a set of *dimension instances*.

Dimension attributes (also called parameters or levels) are organised according to one or more hierarchies. Hierarchies represent a particular vision (perspective) of a dimension. Each attribute represents one data granularity according to which measures could be analysed; for example, along the *store* dimension, a hierarchy could group *individual stores* into *cities* and *cities* into *countries*. Weak attributes (attributive properties) complete the parameter semantics, e.g. the *name* of an *individual store*.

Definition. A *hierarchy* of a dimension D , noted $H_i \in H^D$, is defined as $(N^{Hi}, Param^{Hi}, Weak^{Hi})$ where

- N^{Hi} is a hierarchy name,
- $Param^{Hi} = \langle id^D, p_{1, \dots, vi}^{Hi}, All \rangle$ is an ordered set of attributes, called *parameters*, which represent useful graduations along the dimension, $\forall k \in [1..vi], p_k^{Hi} \in A^D$,
- $Weak^{Hi} : Param^{Hi} \rightarrow 2^{A^D - Param^{Hi}}$ is a function possibly associating each parameter to one or several *weak attributes*.

For a hierarchy H_i we introduce $A^{Hi} = Param^{Hi} \cup (\bigcup_{j=1}^{j=vi} Weak^{Hi}(p^{Hj}))$ the set of the hierarchy

attributes, $A^{Hi} \subseteq A^D$. We also define the function *level*: $A^D \rightarrow \mathbb{N}^+$, noted $level^{Hi}(a_i)$, which returns the order of a_i in the list $Param^{Hi}$ (note that the level of a weak attribute is the level of its associated parameter).

All hierarchies in one dimension start with a same parameter, noted id^D called *root parameter*.

All these hierarchies end with a same parameter, noted All called *extremity parameter*. We

represent dimensions using a graphical formalism, which extends notations introduced in

(Golfarelli et al., 1998). Each path starting from Id and ending by All represents a hierarchy.

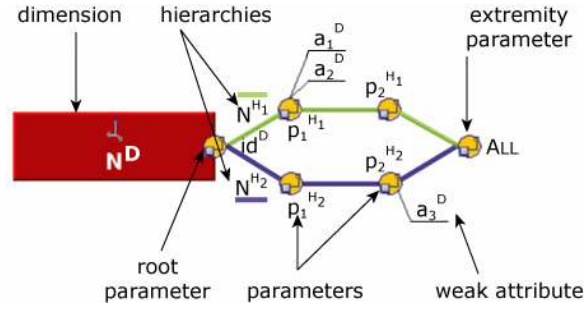


Figure 1: Graphical formalism of dimensions and hierarchies.

A fact reflects information that has to be analysed according to dimensions. This analysed information is modelled through one or several indicators, called measures; for example, a fact data may be *sale amounts* occurring in shops every day. The notation $D_i \in \text{Star}^{\text{Cs}}(F)$ represents that the dimension D_i is linked to the fact F_j .

Definition. A *fact*, noted $F \in F^{\text{Cs}}$, is defined as $(N^F, M^F, I^F, \text{IStar}^F)$ where

- N^F is a name of fact,
- $M^F = \{f_1(m_1^F), \dots, f_w(m_w^F)\}$ is a set of *measures* associated with an aggregate function,
- $I^F = \{i_1^F, \dots, i_q^F\}$ is a set of *fact instances*,
- $\text{IStar}^F : I^F \rightarrow I^{D_1} \times \dots \times I^{D_n}$ is a function ($\forall k \in [1..n], D_k \in \text{Star}^{\text{Cs}}(F)$), which respectively associates fact instances to their linked dimension instances.

Figure 2 illustrates graphical notations used to represent facts.

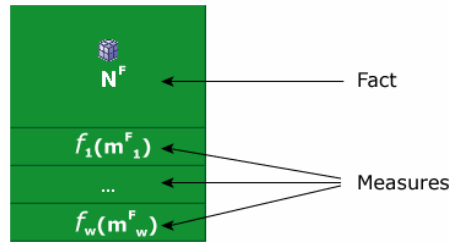


Figure 2: Graphical formalism of facts.

Case study

The case we study is taken from the meteorology domain. We define a multidimensional database, which allows users to analyse weather forecasts (atmospheric measures and

forecasted measures) according to dates, geographic locations, sensor devices and forecast models.

Figure 3 shows a constellation, which supports analyses about weather forecasts. It is composed of two facts, named *Forecast* and *Measures*, and four dimensions, named *Model*, *Dates*, *Geography*, and *Sensor*. Note that the extremity parameter (All) is not displayed in the graphical representation as this parameter tends to confuse users (Malinowsky and Zimányi, 2006).

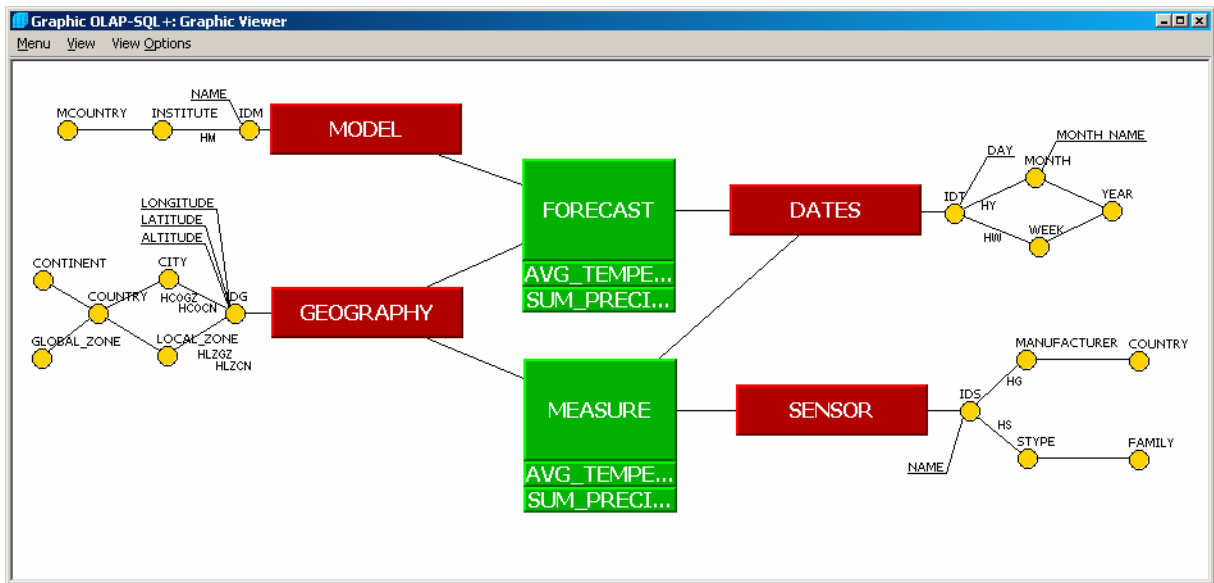


Figure 3: Example of constellation schema.

The dimension named Geography, which represents geographic information, is defined by

$(N^{GEO}, A^{GEO}, H^{GEO}, I^{GEO})$ where

- $N^{GEO} = \text{'Geography'}$,
- $A^{GEO} = \{\text{Latitude, Longitude, Altitude, City, Local_Zone, Country, Continent, Global_Zone}\} \cup \{\text{IdG, All}\}$,
- $H^{GEO} = \{\text{H}_{COGZ}, \text{H}_{COCN}, \text{H}_{LZGZ}, \text{H}_{LZCN}\}$,
- $I^{GEO} = \{i^{GEO}_1, i^{GEO}_2, \dots, i^{GEO}_p\}$.

Examples of dimension instances are presented in the next table.

IdG	Latitude	Longitude	Altitude	City	Local_Zone	Country	Continent	Global_Zone	All
i_1^{GEO}	N43.36	E1.26	145M	Toulouse	MP	France	Europe	North	All
i_2^{GEO}	S34.0	E151.0	0M	Sydney	NSW	Australia	Oceania	South	All
...

Table 2: Example of dimension instances.

This dimension is composed of four hierarchies, noted $H_{\text{COGZ}} = (N^{\text{HCOGZ}}, \text{Param}^{\text{HCOGZ}}, \text{Weak}^{\text{HCOGZ}})$; $H_{\text{LZCN}} = (N^{\text{HLZCN}}, \text{Param}^{\text{HLZCN}}, \text{Weak}^{\text{HLZCN}})$; $H_{\text{LZGZ}} = (N^{\text{HLZGZ}}, \text{Param}^{\text{HLZGZ}}, \text{Weak}^{\text{HLZGZ}})$; and $H_{\text{COCN}} = (N^{\text{HCOCN}}, \text{Param}^{\text{HCOCN}}, \text{Weak}^{\text{HCOCN}})$.

The hierarchy named H_{COGZ} is specified by:

- $N^{\text{HCOGZ}} = \text{'HCOGZ'}$,
- $\text{Param}^{\text{HCOGZ}} = \langle \text{IdG}, \text{City}, \text{Country}, \text{Global_Zone}, \text{All} \rangle$,
- $\text{Weak}^{\text{HCOGZ}} = \{ \text{IdG} \rightarrow \{ \text{Latitude}, \text{Longitude}, \text{Altitude} \} \}$.

We shall only present Param^{Hx} for the three other hierarchies: $\text{Param}^{\text{HCOCN}} = \langle \text{IdG}, \text{City}, \text{Country}, \text{Continent}, \text{All} \rangle$; $\text{Param}^{\text{HLZCN}} = \langle \text{IdG}, \text{Local_Zone}, \text{Country}, \text{Continent}, \text{All} \rangle$; and $\text{Param}^{\text{HLZGZ}} = \langle \text{IdG}, \text{Local_Zone}, \text{Country}, \text{Global_Zone}, \text{All} \rangle$.

Weather forecasts may be analysed through the fact noted $\text{FORC} = (N^{\text{FORC}}, M^{\text{FORC}}, I^{\text{FORC}}, \text{IStar}^{\text{FORC}})$ where

- $N^{\text{FORC}} = \text{'Forecast'}$,
- $M^{\text{FORC}} = \{ \text{SUM}(\text{Precipitation}), \text{AVG}(\text{Temperature}) \}$,
- $I^{\text{FORC}} = \{ i_1^{\text{FORC}}, i_2^{\text{FORC}} \dots i_q^{\text{FORC}} \}$,
- $\text{IStar}^{\text{FORC}} = \{ i_k^{\text{FORC}} \rightarrow (i_{rk}^{\text{MOD}}, i_{sk}^{\text{DAT}}, i_{tk}^{\text{GEO}}) \mid \forall k \in [1..q], i_k^{\text{FORC}} \in I^{\text{FORC}} \wedge \exists i_{rk}^{\text{MOD}} \in I^{\text{MOD}} \wedge \exists i_{sk}^{\text{DAT}} \in I^{\text{DAT}} \wedge \exists i_{tk}^{\text{GEO}} \in I^{\text{GEO}} \}$.

Examples of fact instances are presented in the next table.

I ^{FORC}			IStar ^{FORC}		
I ^{FORC}	SUM(Precipitation)	AVG(Temperature)	I ^{MOD}	I ^{DAT}	I ^{GEO}
I ^{FORC} ₁	0	20	I ^{MOD} ₃	I ^{DAT} ₁	I ^{GEO} ₁
I ^{FORC} ₂	30	25	I ^{MOD} ₁	I ^{DAT} ₁	I ^{GEO} ₂
...

Table 3: Example of fact instances.

Multidimensional Table Structure

Constellation schemas depict MDB structures whereas user analyses are based on tabular representations (Gyssens and Lakshmanan, 1997) where structures and data are displayed. The visualisation structure that we define is a multidimensional table (MT), which displays data from one fact and two of its linked dimensions.

Definition. A multidimensional table T is defined as (S, L, C, R) where

- $S=(F^S, M^S)$ represents the analysed subject through a fact $F^S \in F^{Cs}$ and a set of projected measures $M^S = \{f_1(m_1), \dots, f_x(m_x)\}$ where $\forall i \in [1..x], m_i \in M^{Fs}$,
- $L=(DL, HL, PL)$ represents the horizontal analysis axis where $PL = \langle \text{All}, p_{\max}^{HL}, \dots, p_{\min}^{HL} \rangle$, $HL \in H^{DL}$ and $DL \in \text{Star}^{Cs}(F^S)$, HL is the *current hierarchy* of DL ,
- $C=(DC, HC, PC)$ represents the vertical analysis axis where $PC = \langle \text{All}, p_{\max}^{HC}, \dots, p_{\min}^{HC} \rangle$, $HC \in H^{DC}$ and $DC \in \text{Star}^{Cs}(F^S)$, HC is the *current hierarchy* of DC ,
- $R = \text{pred}_1 \wedge \dots \wedge \text{pred}_i$ is a normalised conjunction of predicates (restrictions of dimension data and fact data).

Figure 4 depicts an example of MT, which displays precipitation forecasts according to the temporal axis and the geographic axis. Note that a MT represents an excerpt of data recorded in a constellation.

$T_1 = (S_1, L_1, C_1, R_1)$ with:

- $S_1 = (\text{FORC}, \{\text{SUM}(\text{Precipitation}), \text{AVG}(\text{Temperature})\});$
- $L_1 = (\text{GEO}, \text{HCOCN}, \langle \text{All}, \text{Continent} \rangle);$
- $C_1 = (\text{DAT}, \text{HY}, \langle \text{All}, \text{Year} \rangle);$
- $R_1 = \text{GEOGRAPHY.All} = \text{'all'} \wedge \text{DATES.All} = \text{'all'} \wedge \text{MODEL.All} = \text{'all'}.$

FORECAST		DATES HY	
SUM (SUM_PRECIPITATION), AVG (AVG_TEMPERATURE)		YEAR	2006
GEOGRAPHY HCOCN		CONTINENT	
		America	(0, 15)
		Europ	(20, 17.5)
		Oceania	(80, 25)

MODEL.All = 'all' AND DATES.All = 'all' AND GEOGRAPHY.All = 'all'

Figure 4: Example of a MT (textual definition and graphic representation).

A MT is built from a constellation using the operator named DISPLAY.

Definition. $DISPLAY(N^{Cs}, F^S, M^S, DL, HL, DC, HC) = T_{RES}$ where

- N^{Cs} is the constellation name,
- $F^S \in F^{Cs}$ is the displayed fact,
- $M^S = \{f_1(M_1), \dots, f_x(M_x)\}$, where $\forall i \in [1..x]$, $M_i \in M^{FS}$ is a measure of F^S to be displayed in the MT,
- $DL \in Star^{Cs}(F^S)$ and $DC \in Star^{Cs}(F^S)$ are respectively the horizontal and vertical dimensions,
- $HL \in H^{DL}$ and $HC \in H^{DC}$ are selected hierarchies, which are used to display parameters,
- $T_{RES} = (S_{RES}, L_{RES}, C_{RES}, R_{RES})$ is the resulting multidimensional table, where: $S_{RES} = (F, \{f_1(M_1), \dots, f_x(M_x)\})$; $L_{RES} = (DL, HL, \langle All, p^{HL}_{vl} \rangle)$; $C_{RES} = (DC, HC, \langle All, p^{HC}_{vc} \rangle)$; and $R_{RES} = \bigwedge_{\forall i, D_i \in Star^C(F^S)} D_i..ALL = 'all'$.

Example 1. Let us consider users who wish to display precipitation forecasts according to months grouped into years and according to cities grouped into countries and continents. This OLAP analysis is calculated according to the following algebraic expression:

$DISPLAY('Weather Constellation', FORC, \{SUM(Precipitation), AVG(Temperature)\},$

$GEO, H_{COCN}, DAT, H_Y) = T_1$. The resulting MT (T_1) is displayed in Figure 4;

MINIMAL CORE OF THE OLAP ALGEBRA

In relational databases, the relational algebra is a procedural query language composed of operators. Queries are specified by sequences of relational algebraic operators that manipulate relations (database components). This language is a closed set of operators. Each one operates on one or more relations and yields a relation. More elaborated languages (e.g. SQL) are built on this algebra.

In the same way, our objective is to provide an algebra and an associated more elaborated language. The OLAP algebra we define is associated to the multidimensional conceptual model described above. This algebra is a “procedural” query language that allows manipulation and retrieval of data from a MDB through nested expressions of multidimensional algebraic operators. It provides a stable basis for the specification of more elaborated languages, notably languages adapted to decision-makers such a graphical query language. Moreover, the algebra represents also specific algorithms corresponding to each elementary operator.

Formally the algebraic operations take as input a source MT, noted $T_{SRC}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC})$, and producing as output a new MT, noted $T_{RES}=(S_{RES}, L_{RES}, C_{RES}, R_{RES})$. Each output MT can further be manipulated using operations of the same algebra. This property is called closure. As relational algebra we specify a minimal core of operators and a set of advanced operators composed of combined core operators.

The minimal core is a small set of *operators* that allows the analyst to manipulate MT in useful ways. We define three categories of multidimensional OLAP manipulation operators:

- **modifying analysis precision:** it consists in moving the analysis details along a hierarchy (DRILLDOWN or ROLLUP) or selecting data of a multidimensional schema (SELECT).
- **changing analysis criteria:** it consists in: 1) replacing an analysis axis by another one (ROTATE); 2) transforming the subject of the analysis (ADDM/DELM); 3) transforming an analysis axis by adding or removing a dimension attribute (PUSH/PULL); and 4) adding attributes from external dimensions in a displayed analysis axis (NEST).
- **changing MT presentation:** it consists in: 1) switching parameter values of a displayed dimension; 2) add totals and subtotals in a MT.

Each operation has the following input: $T_{SRC}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC})$ where

- $S_{SRC}=(F^S, \{f_1(m_1), \dots, f_x(m_x)\}), \forall i, 1 \leq i \leq x, m_i \in M^S,$

- $L_{SRC}=(DL, HL, \langle All, p_{max}^{HL}, \dots, p_{min}^{HL} \rangle), DL \in Star(F^S),$
- $C_{SRC}=(DC, HC, \langle All, p_{max}^{HC}, \dots, p_{min}^{HC} \rangle), DC \in Star(F^S),$
- $R_{SRC}=(pred_1 \wedge \dots \wedge pred_i).$

Modifying the analysis precision

The operation called DRILLDOWN consists in moving from coarser-granularity data to finer-granularity data. The opposite operator, which modifies the analysis from finer-granularity data to a coarser granularity, is called ROLLUP.

Definition. $DRILLDOWN(T_{SRC}, D, Lvl_{inf}^{(*)}) = T_{RES}.$

Input:

- $D \in \{DC, DL\}$ is the dimension, on which the drilling downward operation is applied,
- Lvl_{inf} is a lower attribute in the current hierarchy of D . The intermediate graduation levels between the finer graduation of T_{SRC} and the new graduation are not displayed.

Output: $T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ is the resulting multidimensional table such as

- if $D=DL$ then $L_{RES}=(DL, HL, \langle All, p_{max}^{HL}, \dots, p_{min}^{HL}, Lvl_{inf} \rangle)$ and $C_{RES}=C_{SRC},$
- if $D=DC$ then $L_{RES}=L_{SRC}$ and $C_{RES}=(DC, HC, \langle All, p_{max}^{HC}, \dots, p_{min}^{HC}, Lvl_{inf} \rangle).$

(*) A level, noted Lvl , may represent a parameter p^D , a parameter with a list of weak attributes $p^D(a^D_1, a^D_2, \dots),$ or a list of weak attributes (a^D_1, a^D_2, \dots) of $p^D,$ which is not displayed.

Definition. $ROLLUP(T_{SRC}, D, Lvl_{sup}) = T_{RES}$ where

Input:

- $D \in \{DC, DL\}$ is the dimension, on which the drilling upward operation is applied,
- Lvl_{sup} is a coarser-graduation level used in $T_{RES},$ the finer graduations are deleted.

Output: $T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ is the resulting multidimensional table such as

- if $D=DL$ then $L_{RES}=(DL, HL, \langle All, p_{max}^{HL}, \dots, Lvl_{sup} \rangle)$ and $C_{RES}=C_{SRC},$
- if $D=DC$ then $L_{RES}=L_{SRC}$ and $C_{RES}=(DC, HC, \langle All, p_{max}^{HC}, \dots, Lvl_{sup} \rangle).$

The selection, noted SELECT, operates on a MT and removes the data that do not satisfy the condition. This condition may be expressed on dimension attribute values as well as on fact measure values. Note that this operator realises “slicing/dicing” manipulations in a MDB terminology (Agrawal et al., 1997).

Definition. $SELECT(T_{SRC}, pred) = T_{RES}$ where

Input: $pred = pred_1 \wedge \dots \wedge pred_i$ is a normalized selection predicate (conjunction of disjunctions) on the fact F^S and/or its linked dimensions ($D_i \mid D_i \in Star^{Cs}(F^S)$).

Output: $T_{RES} = (S_{SRC}, L_{SRC}, C_{SRC}, R_{RES})$ is the resulting MT where $R_{RES} = pred$.

Example 2. From the previous example in Figure 4, decision-makers changes the precision of forecast analysis. The focus is on European continent (European countries and cities). Three operators compose the algebraic expression (output MT in Figure 5):

$DRILLDOWN(DRILLDOWN(SELECT(T_1, Geography.Continent = 'Europ'), Geography, Country), Geography, City) = T_2$.

FORECAST				DATES HY	
SUM (SUM_PRECIPITATION), AVG (AVG_TEMPERATURE)				YEAR	2006
GEOGRAPHY HCOCN		CONTINENT	COUNTRY	CITY	
		Europ	France	Toulouse	(20, 17.5)

MODEL.All = 'all' AND DATES.All = 'all' AND GEOGRAPHY.CONTINENT = 'Europ'

Figure 5: Multidimensional table (T_2) resulting from three algebra operators.

$T_2 = ((FORC, \{SUM(Precipitation), AVG(Temperature)\}),$

$(GEO, HCO, <All, Continent, Country, City>),$

$(DAT, HY, <All, Year>),$

$GEOGRAPHY.Continent = 'Europ' \wedge DATES.All = 'all' \wedge MODEL.All = 'all')$.

Changing analysis criteria

The rotation, noted ROTATE, allows changing one analysis axis by another one in a MT. It also may be used to change the current hierarchy by another one belonging to the same dimension.

Definition. $ROTATE(T_{SRC}, D_{old}, D_{new}, H_k^{D_{new}}) = T_{RES}$ where

Input:

- $D_{old} \in \{DC, DL\}$ is a dimension of T_{SRC} to be replaced,
- D_{new} is the dimension replacing $D_{old} \in \{DC, DL\}$ in the resulting multidimensional table,

- $H_k^{D_{new}}$ is the current hierarchy of D_{new} (positioned on the coarser-granularity parameter).
- Output:** $T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ is the resulting multidimensional table such as
- if $D_{old}=DL$ then $L_{RES}=(D_{new}, H_k^{D_{new}}, \langle All, p^{HD_{new}}_{vl} \rangle)$ and $C_{RES}=C_{SRC}$,
 - if $D_{old}=DC$ then $L_{RES}=L_{SRC}$ and $C_{RES}=(D_{new}, H_k^{D_{new}}, \langle All, p^{HD_{new}}_{vc} \rangle)$.

The operation that adds a measure, noted ADDM and the operation that deletes a measure, noted DELM, allow the modification of the analysed measure set.

Definition. $ADDM(T_{SRC}, f_i(m_i)) = T_{RES}$ where

Input: $f_i(m_i) \notin M^S = \{f_1(m_1), \dots, f_x(m_x)\}$ is a measure of F^S ($f_i(m_i) \in M^{FS}$) to be added to T_{SRC} ,

Output: $T_{RES}=(S_{RES}, L_{SRC}, C_{SRC}, R_{SRC})$ is the resulting multidimensional table where

$S_{RES} = (F^S, \{f_1(m_1), \dots, f_x(m_x), f_i(m_i)\})$.

Definition. $DELM(T_{SRC}, f_i(m_i)) = T_{RES}$ where

Input: $f_i(m_i) \in M^S$ is a measure to be suppressed from T_{SRC} ,

Output: $T_{RES}=(S_{RES}, L_{SRC}, C_{SRC}, R_{SRC})$ is the resulting multidimensional table where

$S_{RES}=(F^S, \{f_1(m_1), \dots, f_{i-1}(m_{i-1}), f_{i+1}(m_{i+1}), \dots, f_x(m_x)\})$. Note that this operator may not remove the last measure of T_{SRC} .

Example 3. Now, decision-makers complete the previous analysis by changing the analysis criteria; e.g. they change temporal granularity by the dimension named MODEL and they focus their analysis by deleting unnecessary measure. The algebraic expression is $ROTATE(DELM(T_2, SUM(Precipitation)), DATES, MODEL, HM) = T_3$ and Figure 6 displays the output MT.

FORECAST				MODEL HM		
AVG (AVG_TEMPERATURE)				MCOUNTRY	France	UK
GEOGRAPHY		CONTINENT	COUNTRY	CITY		
HCOCN		Europ	France	Toulouse	(17.5)	(15)
DATES.All = 'all' AND MODEL.All = 'all' AND GEOGRAPHY.CONTINENT = 'Europ'						

Figure 6: Multidimensional table resulting from OLAP algebra manipulations.

$T_3 = ((FORC, \{AVG(Temperature)\}),$
 $(GEO, HCO, \langle All, Continent, Country, City \rangle),$

(MOD, HM, <All, MCountry>),

GEOGRAPHY.Continent = 'Europ' \wedge DATES.All = 'all' \wedge MODEL.All = 'all').

The restructuring operations, noted PUSH and PULL, consist in combining the dimension attributes with the measures. The PUSH operator converts dimension attributes into measures.

The PULL operator is the converse of the PUSH operator; it converts measures into parameters.

Definition. PUSH(T_{SRC} , D, p) = T_{RES} where

Input:

- $D \in \text{Star}^{CS}(F^S)$ is a dimension,
- $p \in H_k^D$ is the parameter of D to be converted into a measure,

Output: $T_{RES}=(S_{RES}, L_{SRC}, C_{SRC}, R_{SRC})$ is the resulting multidimensional table where $S_{RES}=(F^S, \{f_1(m_1), \dots, f_x(m_x), p\})$.

Definition. PULL($T_{SRC}, f_i(m_i), D$) = T_{RES} where

Input:

- $f_i(m_i) \in M^S$ is a measure of the current fact to be converted into a parameter of D,
- $D \in \{DC, DL\}$ is the dimension, which is extended with the new converted parameter.

Output: $T_{RES}=(S_{RES}, L_{RES}, C_{RES}, R_{SRC})$ is the resulting multidimensional table such as

$S_{RES}=S_{SRC}-\{f_i(m_i)\}$, and

- if $D=DL$ then $L_{RES}=(DL, HL, \langle \text{All}, p_{\max}^{HL}, \dots, p_{\min}^{HL}, f_i(m_i) \rangle) \wedge C_{RES}=C_{SRC}$,
- if $D=DC$ then $L_{RES}=L_{SRC} \wedge C_{RES}=(DC, HC, \langle \text{All}, p_{\max}^{HC}, \dots, p_{\min}^{HC}, f_i(m_i) \rangle)$.

The nesting operator, noted NEST, allows the user to include dimension attributes of dimensions that are not displayed in the displayed dimensions of a multidimensional table.

This operation enables the use of parameters from several dimensions in the 2D space of the MT.

Definition. NEST($T_{SRC}, D, Lvl, D_{\text{nested}}, Lvl_{\text{nested}}$) = T_{RES} where

Input:

- $D \in \{DC, DL\}$ is a dimension and Lvl is its level,
- $D_{\text{nested}} \in \text{Star}^{CS}(F^S)$ is the dimension from which the nested level is taken,
- Lvl_{nested} is the nested level of D_{nested} .

Output: $T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC})$ is the resulting multidimensional table such as

- if $D=DL$ then $L_{RES}=(DL, HL, \langle All, p^{HL}_{max}, \dots, Lvl, Lvl_{nested}, \dots, p^{HL}_{min} \rangle) \wedge C_{RES}=C_{SRC}$,
- if $D=DC$ then $L_{RES}=L_{SRC} \wedge C_{RES}=(DL, HD, \langle All, p^{HC}_{max}, \dots, Lvl, Lvl_{nested}, \dots, p^{HC}_{min} \rangle)$.

Example 4. Decision-makers complete the previous analysis; they change the MT presentation by nesting year into Geography dimension. They also modify analysis precision by rolling up to continents and countries. The algebraic expression is $NEST(ROLLUP(T_3, GEOGRAPHY, Country), GEOGRAPHY, Country, DATES, Year) = T_4$ and Figure 7 displays the output MT.

FORECAST				MODEL HM		
AVG (AVG_TEMPERATURE)				MCOUNTRY	France	UK
GEOGRAPHY HCOCN	CONTINENT	COUNTRY	YEAR			
	Europ	France	2006		(17.5)	(15)

DATES.All = 'all' AND MODEL.All = 'all' AND GEOGRAPHY.CONTINENT = 'Europ'

Figure 7: Multidimensional table resulting from nest operator.

$T_4 = ($ (FORC, $\{AVG(Temperature)\}$),
 (GEO, HCO, $\langle All, Continent, Country, Year \rangle$),
 (MOD, HM, $\langle All, MCountry \rangle$),
 GEOGRAPHY.Continent = 'Europ' \wedge DATES.All = 'all' \wedge MODEL.All = 'all').

Changing the multidimensional table presentation

The switching operation, noted SWITCH, permutes two values of a parameter from a dimension allowing a specific order in the displayed values.

Definition. $SWITCH(T_{SRC}, D, att, v_1, v_2) = T_{RES}$ where

Input:

- $D \in \{DC, DL\}$ is a displayed dimension,
- $att \in A^D$ is an attribute of the dimension D , on which the switching of the values v_1 and v_2 is applied. Note that $dom(att) = \langle \dots v_1, \dots v_2, \dots \rangle$ in T_{SRC} .

Output: $T_{RES}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC})$ is the resulting multidimensional table where

$\text{dom}(\text{att}) = \langle \dots v_2, \dots v_1, \dots \rangle$.

The operation to calculate aggregates, noted AGGREGATE, allows aggregating values in line or column of a multidimensional table. This operation realises the Cube operator defined by (Gray et al., 1996).

Definition. $\text{AGGREGATE}(T_{\text{SRC}}, D, f(\text{att})) = T_{\text{RES}}$ where

Input:

- $D \in \{\text{DC}, \text{DL}\}$ is a dimension,
- att is the attribute of D on which applies the aggregation function $f(\text{sum}, \text{avg}, \dots)$ with $\text{dom}(\text{att}) = \langle v_1, \dots, v_x \rangle$ in T_{SRC} .

Output: $T_{\text{RES}} = (S_{\text{SRC}}, L_{\text{SRC}}, C_{\text{SRC}}, R_{\text{SRC}})$ is the resulting multidimensional table where $\forall i \in [1..x], \text{dom}(\text{att}) = \langle v_1, f(v_1), \dots, v_x, f(v_x) \rangle$. Each initial value is completed by the aggregation value.

Summary

In this section we have defined a core of eleven operators, dispatched in three categories. The following table summarises this proposal. These operators may be combined together.

Categories	Operators
modifying analysis precision	DRILLDOWN / ROLLUP SELECT
changing analysis criteria	ROTATE PUSH / PULL ADDM / DELM NEST
changing MT presentation	SWITCH AGGREGATE

Table 4: OLAP Algebra core.

ADVANCED OPERATORS OF THE OLAP ALGEBRA

The minimal core of the algebra allows the expression of more or less complex analyses on constellation data. However some analyses demand numerous combinations of different basic operators. In order to improve the processing of complex queries, we define a set of advanced operations (created by combinations of basic operators). The interest is twofold: the analysis query specification is simplified and the system processing of advanced operations may be optimized in relation to the equivalent combination of basic operators.

Operator	Description	Syntax and its translation
FROTATE	This operation consists in using a new fact in the MT while preserving the characteristics of the current analysis axes (DC and DL). The new fact must share at least the two selected dimensions with the initial fact.	$FROTATE(T_{SRC}, F_{new}, \{f_1(m_1), \dots, f_x(m_x)\}) = History^{(*)}(T_{SRC}, DL, History(T_{SRC}, DC, DISPLAY(N^{CS}, F_{new}, \{f_1(m_1), f_2(m_2), \dots\}, DL, HL, DC, HC)))$
HROTATE	The rotation of hierarchies in a MT consists in changing the current hierarchy in line or column.	$HROTATE(T_{SRC}, D, H_k^D) = ROTATE(T_{SRC}, D, D, H_k^D)$
ORDER	This operation consists in ordering values of parameters in an ascending or descending order ($ord \in \{'asc', 'dsc'\}$).	$ORDER(T_{SRC}, D, p, ord) = SWITCH(\dots(SWITCH(T_{SRC}, D, p, v_1, v_2), \dots), D, p, v_a, v_b)$
PLOT	This operation consists in displaying data according to a unique parameter of a dimension.	$PLOT(T_{SRC}, D, Lvl) = DRILLDOWN(ROLLUP(T_{SRC}, D, All), D, Lvl)$
UNSELECT	The unselecting operation consists in cancelling all selections on dimensions and facts. This operation builds a MT starting from all the characteristics of an initial MT but withdrawing all restrictions on the domain values.	$UNSELECT(T_{SRC}) = SELECT(T_{SRC}, F^S.All='all' \wedge (\bigwedge_{D_i \in Star^{CS}(F^S)} D_i.All='all'))$

Table 5: Advanced operators.

(*) $\text{History}(T_{\text{old}}, \text{obj}, T_{\text{new}}) = T_R$ represents the history of operations that were applied in T_{old} on obj (dimension or fact) and that must be applied on T_{new} . Note that the fact rotation operation, noted FROTATE, is equivalent to the Drill-Across operation (Abelló et al., 2003).

BINARY OPERATORS OF THE OLAP ALGEBRA

All previously defined operations are unary operators. In order to manipulate two MT, we provide a set of binary operators. From two MT, a binary operator builds a third MT by applying a union, an intersection or a difference operation. In order to be applied, these operations need compatible or semi-compatible MT as inputs. The following sections present the compatibility of input tables and the set of binary operators.

Compatibility of input tables

Input MT, which are noted $T_{\text{SRC1}} = (S_{\text{SRC1}}, L_{\text{SRC1}}, C_{\text{SRC1}}, R_{\text{SRC1}})$ and $T_{\text{SRC2}} = (S_{\text{SRC2}}, L_{\text{SRC2}}, C_{\text{SRC2}}, R_{\text{SRC2}})$, must be compatibles to applied binary operators. Note that $\forall i \in [1..2]$,

- $S_{\text{SRC}i} = (F, \{m^{\text{SRC}i}_1, \dots, m^{\text{SRC}i}_s\})$,
- $L_{\text{SRC}i} = (DL^{\text{SRC}i}, HL^{\text{SRC}i}, \langle \text{All}, p^{\text{DL/SRC}i}_1, \dots, p^{\text{DL/SRC}i}_{cl} \rangle)$,
- $C_{\text{SRC}i} = (DC^{\text{SRC}i}, HC^{\text{SRC}i}, \langle \text{All}, p^{\text{DC/SRC}i}_1, \dots, p^{\text{DC/SRC}i}_{cc} \rangle)$,
- $R^{\text{SRC}i} = \text{pred}^{\text{SRC}i}_1 \wedge \dots \wedge \text{pred}^{\text{SRC}i}_t$.

Definition. Two tables T_{SRC1} and T_{SRC2} are *compatible tables* if and only if

- S_{SRC1} and S_{SRC2} are compatibles; *e.g.* they have the same number of measures, noted $\{m^{\text{SRC1}}_1, \dots, m^{\text{SRC1}}_s\}$ and $\{m^{\text{SRC2}}_1, \dots, m^{\text{SRC2}}_s\}$, and the type^(*) of corresponding measures is the same in both S_{SRC1} and S_{SRC2} , $\forall i \in [1..s]$, $\text{type}(m^{\text{SRC1}}_i) = \text{type}(m^{\text{SRC2}}_i)$,
- L_{SRC1} and L_{SRC2} are compatibles; *e.g.* they have same structure (same dimension $DL^{\text{SRC1}} = DL^{\text{SRC2}}$, same hierarchy $HL^{\text{SRC1}} = HL^{\text{SRC2}}$, same ordered set of displayed dimension attributes $\langle \text{All}, p^{\text{DL SRC1}}_1, \dots, p^{\text{DL SRC1}}_{cl} \rangle = \langle \text{All}, p^{\text{DL SRC2}}_1, \dots, p^{\text{DL SRC2}}_{cl} \rangle$). Notice that domains of dimension attributes are not necessarily equals.
- C_{SRC1} and C_{SRC2} are compatibles; *e.g.* they have same structure.

(*) Note that $\text{type}(\text{att})$ gives the set of all possible values of the attribute whereas $\text{dom}(\text{att})$ gives the set of attribute values ($\text{dom}(\text{Att}) \subseteq \text{type}(\text{Att})$).

We introduce the semi-compatibility property allowing binary operations between two MT, which are not strictly compatibles.

Definition. Two tables T_{SRC1} and T_{SRC2} are *semi-compatible tables* if and only if

- L_{SRC1} and L_{SRC2} are compatibles,
- C_{SRC1} and C_{SRC2} are compatibles.

Example 5. In the following table, we display three MT noted T_a , T_b , and T_c .

- T_a is not compatible with T_b and T_c because $C_a \neq C_b$ and $C_a \neq C_c$ while
- T_b and T_c are compatibles because $S_b = S_c$, $L_b = L_c$ and $C_b = C_c$. Note that $\text{dom}(\text{Continent})$ in T_b is not equal to $\text{dom}(\text{Continent})$ in T_c .

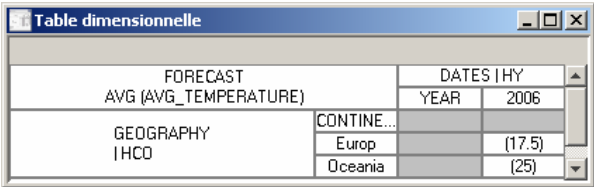
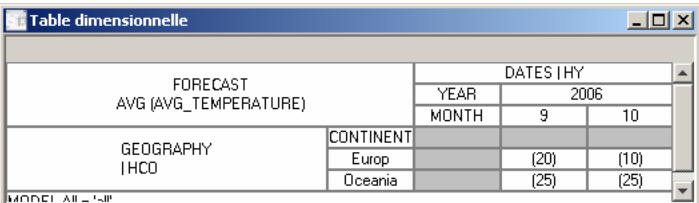
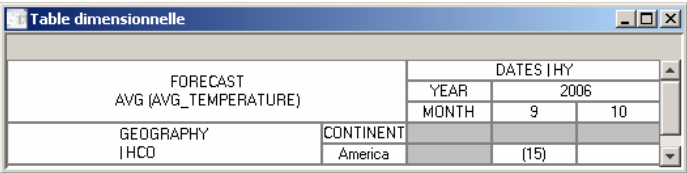
<p>$S_a = (\text{Forecast}, \{\text{AVG}(\text{Temperature})\})$ $L_a = (\text{Geography}, \text{HCO}, \langle \text{All}, \text{Continent} \rangle)$ $C_a = (\text{Dates}, \text{HY}, \langle \text{All}, \text{Year} \rangle)$</p>	 <p style="text-align: center;">Figure 8: MT noted T_a.</p>
<p>$S_b = (\text{Forecast}, \{\text{AVG}(\text{Temperature})\})$ $L_b = (\text{Geography}, \text{HCO}, \langle \text{All}, \text{Continent} \rangle)$ $C_b = (\text{Dates}, \text{HY}, \langle \text{All}, \text{Year}, \text{Month} \rangle)$</p>	 <p style="text-align: center;">Figure 9: MT noted T_b.</p>
<p>$S_c = (\text{Forecast}, \{\text{AVG}(\text{Temperature})\})$ $L_c = (\text{Geography}, \text{HCO}, \langle \text{All}, \text{Continent} \rangle)$ $C_c = (\text{Dates}, \text{HY}, \langle \text{All}, \text{Year}, \text{Month} \rangle)$</p>	 <p style="text-align: center;">Figure 10: MT noted T_c.</p>

Table 6: Compatibilities between MT.

Set operators

Set operators can be applied between two tables, which may be compatibles or semi-compatibles. These operations consist in combining two MT using union, intersection or minus operations (Gyssens and Lakshmanan, 1997), (Agrawal et al. 1997).

Definition. $SET(T_{SRC1}, T_{SRC2} [, fn]) = T_{RES}$ where $SET \in \{UNION, INTERSECT, MINUS\}$.

Input T_{SRC1} and T_{SRC2} are two compatible or semi-compatible MT, and

fn is an optional calculus function, which is applied on corresponding measures from both T_{SRC1} and T_{SRC2} .

Output: T_{RES} is the resulting multidimensional table such as

- if fn is specified, $S_{RES} = (F^{SRC1}, \{m^{SRC1}_{1, \dots, s}\})$, else if fn is not specified, $S_{RES} = (F^{SRC1}, \{m^{SRC1}_{1, \dots, s1}, m^{SRC2}_{1, \dots, s2}\})$; e.g. measures from T_{SRC1} and T_{SRC2} are not regrouped into T_{RES} .
- $L_{RES} = (DL^{SRC1}, HL^{SRC1}, \langle All, p^{DL RES}_{1, \dots, cl} \rangle)$ where $\forall i \in [1..cl]$,
 - if $SET = UNION$, $dom(p^{DL RES}_i) = dom(p^{DL SRC1}_i) \cup dom(p^{DL SRC2}_i)$,
 - if $SET = INTERSECT$, $dom(p^{DL RES}_i) = dom(p^{DL SRC1}_i) \cap dom(p^{DL SRC2}_i)$,
 - if $SET = MINUS$, $dom(p^{DL RES}_i) = dom(p^{DL SRC1}_i) \setminus dom(p^{DL SRC2}_i)$,
- $C_{RES} = (DC^{SRC1}, HC^{SRC1}, \langle All, p^{DC RES}_{1, \dots, cc} \rangle)$ where $\forall i \in [1..cc]$,
 - if $SET = UNION$, $dom(p^{DL RES}_i) = dom(p^{DL SRC1}_i) \cup dom(p^{DL SRC2}_i)$,
 - if $SET = INTERSECT$, $dom(p^{DL RES}_i) = dom(p^{DL SRC1}_i) \cap dom(p^{DL SRC2}_i)$,
 - if $SET = MINUS$, $dom(p^{DL RES}_i) = dom(p^{DL SRC1}_i) \setminus dom(p^{DL SRC2}_i)$,
- if $SET = UNION$, $R_{RES} = R_{SRC1} \vee R_{SRC2}$, if $SET = INTERSECT$, $R_{RES} = R_{SRC1} \wedge R_{SRC2}$, if $SET = MINUS$, $R_{RES} = R_{SRC1} \wedge \neg R_{SRC2}$.

Note that both union and intersection are commutative and associative operations whereas the minus operation is neither commutative nor associative.

Example 6. Decision-makers want to analyse data from T_b with data from T_c . The expression $UNION(T_b, T_c)$ produces $T_{R1} = ((Forecast, \{AVG(Temperature)\}), (Geography, HCO, \langle All, Continent \rangle), (Dates, HY, \langle All, Year, Month \rangle), Model.All='all')$ with $dom(Continent) = \{Europ, Oceania, America\}$.

FORECAST AVG (AVG_TEMPERATURE)		DATES HY	
		YEAR	2006
		MONTH	9 10
GEOGRAPHY HCOCN		CONTINENT	
		America	(15)
		Europ	(20) (10)
		Oceania	(25) (25)

MODEL.All = 'all' AND DATES.All = 'all' AND GEOGRAPHY.All = 'all'

Figure 11: Multidimensional table resulting from union.

GRAPHICAL QUERY LANGUAGE

Although the algebra is a powerful tool, it may not be directly used by a decision-maker. We provide a user-oriented graphic language avoiding the specification of long tedious textual queries. The graphic language is based on the algebraic operations. With this language the user expresses the different OLAP manipulations disregarding the algebraic syntax.

This section starts by defining the principles of the language. The second part provides formal specifications and the third focuses on the completeness with regard to the algebraic core.

Principles

The graphic language is based on an environment providing a display of the conceptual multidimensional elements and an analysis display interface presenting the results of the multidimensional OLAP analysis. Query specification uses both interfaces.

Constellation display interface

This interface presents to the user the schema of a MDB, with a graph (see Figure 3). In this graph, notations are inspired by (Golfarelli et al., 1998), each node is a fact (green) or a dimension (red) whereas links between facts and dimensions are represented by edges.

Measures are directly associated to fact nodes. Each dimension is represented by a sub-tree of parameters (yellow circles) and weak attributes (underlined texts). Additional visualisations are provided for special cases (see appendix-A).

Analysis display interface

This interface is a bi-dimensional table representing a multidimensional table defined in section 2.3. This interface, called MT, is composed of eight “drop zones” that may be used for query specification. These zones are depicted in the following table. It evolves incrementally, according to decision-maker manipulations.

Factual Header (Fact + Measure(s))		Dimensional Header (Column)	
		Parameter Header (Column)	Parameter Values / Dimensional Values (Column)
Dimensional Header (Line)	Parameter Header (Line)		
	Parameter Values / Dimensional Values (Line)		Factual Values
Restrictive Predicate Zone			

Figure 12: MT zones.

Query specification

In order to specify multidimensional OLAP analyses, the decision-maker expresses his queries with the use of the two previously defined interfaces:

- Users may drag elements from the graph displayed in the constellation display interface onto a MT drop zone. He may also drag elements from a MT zone to another one.
- Alternatively users may call a contextual menu on elements of the graph or on the elements in the MT. They then select the item in the menu corresponding to the desired operation.

In order to ensure consistency during the specification of the analyses, the user is guided along the query specification process: incompatible operations with the ongoing analysis are deactivated. As a consequence, the user cannot create erroneous analyses. Each operation takes as input a multidimensional element and is applied to the current MT. If complementary information is needed by the system, the user specifies it through dialog boxes.

Formal specification of the graphic operations

The system provides users with a set of operations for specifying graphically multidimensional OLAP analyses. Users start by defining a MT and employ the manipulation language to modify the MT.

Formal specifications of graphical operations are as follows: OPERATION(E), where E is a multidimensional element. This element may be: a fact (F), a measure ($f_i(m_i^F)$ or $f_i(m_i)$), a dimension (D), a hierarchy (H_i^D or H) or a parameter (p_i^D or p_i). This element may be either in the constellation display interface or in a MT, e.g. the column dimension DC is in the graph representation, but also in the column dimensional header. Notation: $p_i \in H \Leftrightarrow (p_i \in \text{Param}^H \vee (\exists p_j \in \text{Param}^H \mid p_i \in \text{Weak}^H(p_j)))$

MT definition

A MT may be in two different modes: **definition mode** where only a few graphic operations are available; and **alteration mode** where all other operations are available.

In definition mode, graphic manipulations are limited to those that may specify the DISPLAY operation in order to define the initial display of the MT: DIS_SUBJ to specify the displayed subject, DIS_COL and DIS_LN for the specification of the line and column axis. The MT stays in definition mode as long as the displayed fact and the column and line dimensions are not all specified, i.e. for a MT noted $T=(S, L, C, R)$ as long as $(S=\emptyset \vee L=\emptyset \vee C=\emptyset)$.

Example. The following algebraic expression permits to display the multidimensional table T_1 (see example.1):

Ex1: DISPLAY('Weather Constellation', FORC, {SUM(Precipitation), AVG(Temperature)}, GEO, HCOCN, DAT, HY) = T_1
--

Within the graphical context, T_1 (displayed in Figure 4) is defined by a sequence of three graphic actions which may be executed in any order. The MT is built incrementally and the

display is updated with each new instruction. The following steps (1, 2 and 3) are displayed in Figure 13:

- 1) Subject selection (DIS_SUBJ): the user selects the fact FORECAST, drags its graphic icon and drops it in the factual header of the MT. This action defines F^S , and automatically selects all its measures in M^S , i.e. the two measures $SUM(Precipitation)$ and $AVG(Temperature)$. Alternatively, the user may drag each measure one after the other and drop them in the factual header. The user may also use contextual menus to select a fact and/or its measures.
- 2) Column dimension selection: the user selects the dimension DATES and drags its graphic icon and drops it in the column dimensional header, defining DC. If a dimension is composed of a single hierarchy, this latest is automatically selected and the parameter of highest level which is just below the extremity parameter "All": p_{vc}^{HC} is displayed. If the system may not isolate this current hierarchy, automatic selection fails, which is the case here. The user is then prompted to select the correct hierarchy: HY. Alternatively, the user may drag the hierarchy edge or the desired parameter p_i . As above, the user may rely on contextual menus.
- 3) Line dimension selection: the user operates as previously described. In the example, the user drags the hierarchy edge HCOCN. In the end, the user obtains T_1 , displayed in Figure 4.

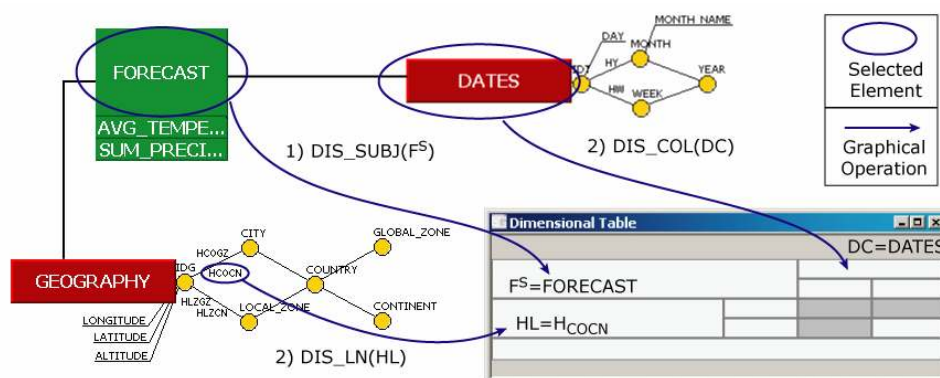


Figure 13: Example of a graphical definition of an MT.

The next table presents the different possibilities to define a MT with the graphic instructions. In the table (as in all other tables of this section) detailed instructions are presented only for the column definition (DC, HC,...), as they are identical to lines (DL, HL,...). “?” stands for an element that the user may have to select through a user prompt. In the DISPLAY case, this happens if: 1) DC is selected and DC has more than one hierarchy, i.e. $\|H^{DC}\|>1$; 2) p_i is selected and it belongs to more than one hierarchy, i.e. $\exists H \in H^{DC}, H \neq HC \mid p_i \in H \wedge p_i \in HC$; or 3) the edge between p_i and p_{i+1} represents more than one hierarchy, i.e. $\exists H \in H^{DC}, H \neq HC \mid p_i \in H \wedge p_i \in HC \wedge p_{i+1} \in H \wedge p_{i+1} \in HC$.

Notice that the user may also directly select a parameter p_i which is not the highest parameter in the hierarchy, i.e. $level^{HC}(p_i) < level^{HC}(p_{vc})$.

Source element of the action (and conditions)		Algebraic equivalent
Subject specification : DIS SUBJ(E)		
1)	$E=F \mid F \in F^{Cs}$	DISPLAY($N^{Cs}, F^S, M^S, DL, HL, DC, HC$) with: $F^S=F, M^S=M^F$
2)	$E=f_i(m_i) \mid f_i(m_i) \in M^F \wedge F \in F^{Cs}$	DISPLAY($N^{Cs}, F^S, M^S, DL, HL, DC, HC$) with: $F^S=F, M^S=\{f_i(m_i)\}$
Column specification: DIS COL(E)		
3)	$E=D \mid \exists H \in H^D \wedge D \in D^{Cs}$	DISPLAY($N^{Cs}, F^S, M^S, DL, HL, DC, HC$) with: DC=D, HC=H?
4)	$E=H \mid H \in H^D \wedge D \in D^{Cs}$	DISPLAY($N^{Cs}, F^S, M^S, DL, HL, DC, HC$) with: DC=D, HC=H
5)	$E=p_i \mid p_i \in H \wedge H \in H^D \wedge D \in D^{Cs}$	DISPLAY($N^{Cs}, F^S, M^S, DL, HL, DC, HC$) with: DC=D, HC=H?
6)	$E=p_i \mid p_i \in H \wedge H \in H^D \wedge D \in D^{Cs} \wedge level^{HC}(p_i) < level^{HC}(p_{vc})$	PLOT(DISPLAY($N^{Cs}, F^S, M^S, DL, HL, DC, HC$), DC, p_i) with: DC=D, HC=H?
Line specification: DIS LN(E), identical to columns for selecting DL and HL.		
If $(DC \neq \emptyset \wedge F^S \neq \emptyset)$ then $DC \in Star^{Cs}(F^S)$ and If $(DL \neq \emptyset \wedge F^S \neq \emptyset)$ then $DL \in Star^{Cs}(F^S)$		

Table 7: Graphical specifications of the algebraic DISPLAY operator.

Figure 14 presents the graphical “drag and drop” actions that may be used to define a MT. In order to specify the analysis subject, the user drags the graphic icon representing the subject from the constellation graph, onto the MT and drops it in the factual header zone (executing a DIS-SUBJ operation). To specify an analysis axis, the user has to drop the element in a

dimensional header zone (executing a DIS_COL on the column header and a DIS_LN on the line header). In the following figure, operations numbered from 1) to 6) correspond to the six lines of the Table 7.

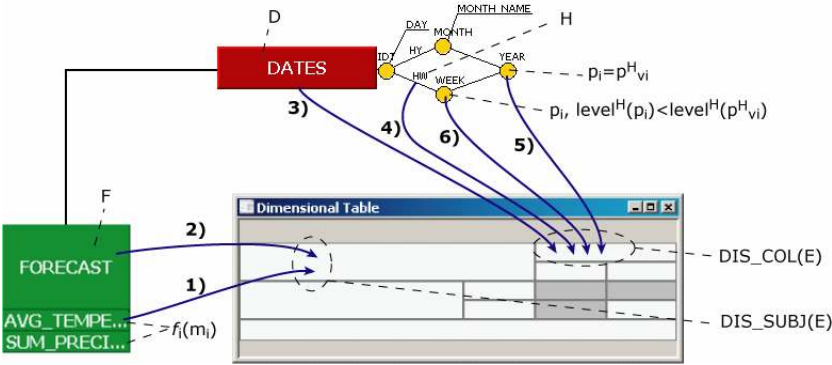


Figure 14: Actions that may be used to define a MT.

Once a fact has been designated as a subject and two dimensions have been specified as line and column axes, the MT switches to alteration mode authorizing all the other operations.

MT manipulations

In alteration mode, a MT may be modified by the use of operations of the following sets:

- **A set of six display-oriented operations.** The three operations previously defined display an element as subject, in lines or in columns (DIS_SUBJ, DIS_LN, and DIS_COL). In addition to these three operations, the set is enriched by two “insertion” operations. These operations allow decision-makers to “insert” an element in the current line or column (INS_LN and INS_COL) of the analysis display interface. Moreover, a deletion operation is provided (DEL).
- **A set of specific operations:** these operations allow the execution of specific algebraic operators. In order to ease understanding, they have the same name as their algebraic equivalent. These operations allow the restriction of the displayed data (SELECT), the inversion of two displayed values (SWITCH), the addition of aggregated data

(AGGREGATE), the removal of previously aggregated data (UNAGGREGATE) and the inversion of all displayed values (ORDER).

- **A set of two drilling operations: DRILLDOWN, ROLLUP.** As all drilling operations may not be expressed by the use of the graphic-oriented operations, these two operations are defined.

Each of these graphic operations executes a sequence of core and/or advanced algebraic operations.

The display operations are the equivalent of dropping elements into the factual header or the two dimensional headers; the insert operations are the equivalent of dropping elements into the parameter headers; and the delete operation is the equivalent of dragging an element which is in the MT and drop it out of the MT. The following figure summarises the different correspondences between graphic operations and “drag and drop” actions on a particular drop zone.

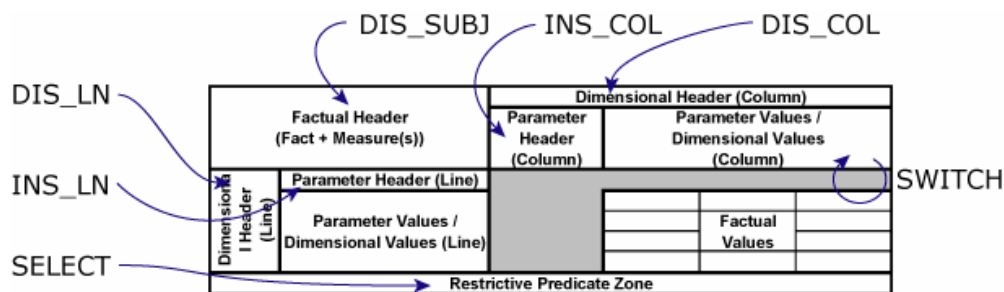


Figure 15: Correspondance between graphic operations and drag and drop actions.

Example. The following algebraic expression permits to display the multidimensional table T_1 (see example.2):

Ex2: DRILLDOWN(DRILLDOWN(SELECT(T_1 , Geography.Continent = 'Europ'),
 Geography, Country), Geography, City) = T_2

Within the graphical context, T_2 (displayed in Figure 5) is defined by a sequence of three graphic actions. A MT is built incrementally and the display is updated with each new graphic action. These actions are described below (see Figure 16):

- 1) **Restriction of the line values:** the user selects the parameter on which he wishes to add a predicate. He drags the parameter graphic icon from the graph and drops it in the restriction zone of the MT, thus executing a `SELECT(Continent)`. He will be prompted to specify the restriction predicate which will be applied: “=Europ”.
- 2) **Adding to the display parameters:** the user selects the graphic icon representing the parameter `Country`, drags it onto the MT and drops it in the line parameter header, thus executing `INS_LN(Country)`.
- 3) **Adding to the display parameters:** the user repeats the operation with the parameter `City`: `INS_LN(City)`. Alternatively, the decision-maker could have used the contextual menus in order to execute two `DRILLDOWN` operations on `GEOGRAPHY` dimension. In the end, decision-makers obtain T_2 displayed in Figure 5.

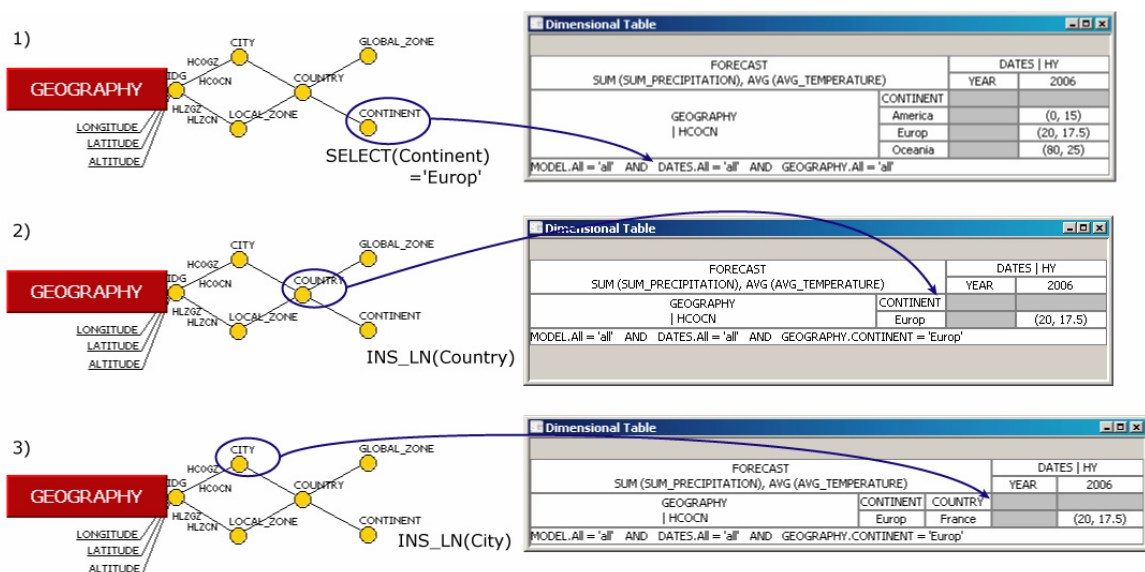


Figure 16: Example of a graphical definition of an MT.

The next paragraphs present complete formal specification of graphic instructions of display-oriented operations and their translation in algebraic language (see tables). These formal specifications are followed by their equivalent graphic “drag and drop” manipulations. Other specifications (specific operations and drilling operations) are in appendix B.

Display-oriented operations: DIS_SUBJ, DIS_LN and DIS_COL

The graphic instruction DIS_SUBJ is used to designate an element that will be displayed as a subject of a MT. Formal specifications may be found in Table 8. Its corresponding “drag and drop” actions are displayed in Figure 17. In this figure, the numbers from 1 to 5 correspond to each line of the Table 8.

DIS_SUBJ(E)		
Source element of the action (and conditions)		Algebraic equivalent (T_{RES})
1)	$E=F^S$	History(T_{SRC} , DL, History(T_{SRC} , DC, DISPLAY(N^{CS} , F^S , $\{f_i(m_1), \dots, f_w(m_w)\}$), DL, HL, DC, HC))), $\forall i \in [1..w], f_i(m_i) \in M^{FS}$
2)	$E=F_{new} \mid F_{new} \neq F^S \wedge DC \in Star^{Cs}(F_{new}) \wedge DL \in Star^{Cs}(F_{new})$	FROTATE(T_{SRC} , F_{new} , $\{f_i(M_1), \dots, f_w(M_w)\}$), $\forall i \in [1..w], M_i \in M^{F_{new}}$
3)	$E=f_i(m_i) \mid f_i(m_i) \notin M^S \wedge f_i(m_i) \in M^{FS}$	ADDM(T_{SRC} , $f_i(m_i)$)
4)	$E=f_i(m_i) \mid f_i(m_i) \notin M^{FS} \wedge f_i(m_i) \in M^{F_{new}} \wedge DC \in Star^{Cs}(F_{new}) \wedge DL \in Star^{Cs}(F_{new})$	FROTATE(T_{SRC} , F_{new} , $\{f_i(m_i)\}$)
5)	$E=p_i \mid \forall p_i \in A^{D_{new}} \wedge D_{new} \in Star^{Cs}(F^S)$	PUSH(T_{SRC} , D_{new} , p_i)

Table 8: Formal specification of DIS_SUBJ(E).

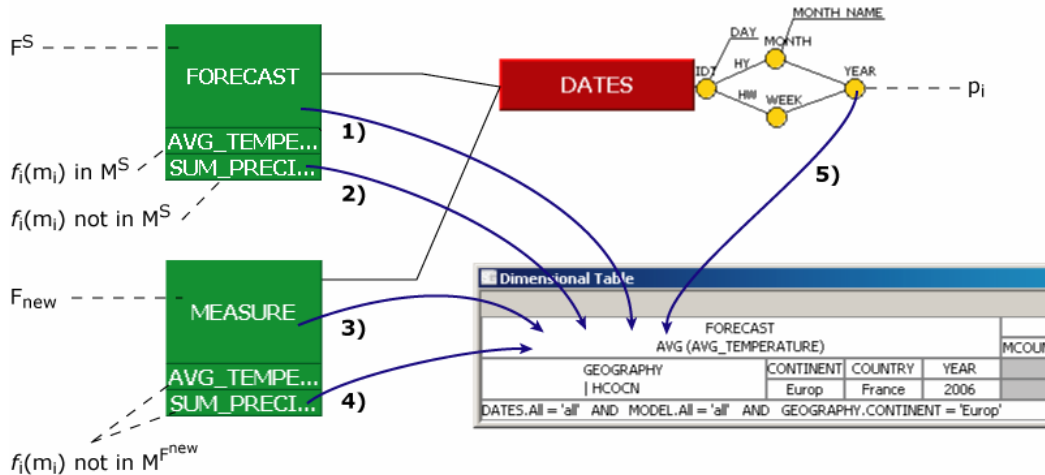


Figure 17: Possible drag and drop manipulations for the instruction DIS_SUBJ(E).

The graphic instruction DIS_COL (respectively DIS_LN) is used to replace the elements currently displayed in columns (resp. lines). Formal specifications of DIS_COL may be found in Table 9 and corresponding “drag and drop” actions are displayed in Figure 18. As previously, the specification of DIS_LN is not presented.

DIS_COL(E)		
Source element of the action (and conditions)		Algebraic equivalent (T_{RES})
1)	$E=f_i(m_i) \mid f_i(m_i) \in M^{FS}$	PULL(T_{SRC} , $f_i(m_i)$, DC)
2)	$E=DC$	ROLLUP(T_{SRC} , DC, p_{vc}^{DC}) the display of HC is reset on the most general parameter: $p_{Cmin} = p_{Cmax} = p_{vc}^{DC}$

3)	$E=D_{new} \mid D_{new} \neq DC, D_{new} \in Star^{Cs}(F^S)$	ROTATE($T_{SRC}, DC, D_{new}, H_{new}?$), with $H_{new} \in H^{D_{new}}$
4)	$E=HC$	ROLLUP(T_{SRC}, DC, p^{DC}_{vc}) the display of HC is reset on the most general parameter: $p_{Cmin}=p_{Cmax}=p^{DC}_{vc}$
5)	$E=H_{new} \mid H_{new} \neq HC \wedge H_{new} \in H^{DC}$	HROTATE(T_{SRC}, DC, H_{new})
6)	$E=H_{new} \mid H_{new} \notin H^{DC} \wedge H_{new} \in H^{D_{new}} \wedge D_{new} \in Star^{Cs}(F^S)$	ROTATE($T_{SRC}, DC, D_{new}, H_{new}$)
7)	$E=p_i \mid p_i \in PC \vee (p_i \notin PC \wedge p_i \in HC)$	PLOT(T_{SRC}, DC, p_i)
8)	$E=p_i \mid p_i \notin HC \wedge p_i \in H_{new} \wedge H \in H^{DC}$	PLOT(HROTATE(T_{SRC}, DC, H_{new}), DC, p_i)
9)	$E=p_i \mid p_i \notin A^{DC} \wedge p_i \in A^{D_{new}} \wedge D_{new} \in Star^{Cs}(F^S)$	PLOT(ROTATE($T_{SRC}, DC, D_{new}, H_{new}?$), D_{new}, p_i), with $H_{new} \in H^{D_{new}}, p_i \in H_{new}$

Table 9: Formal specification of DIS_COL(E), DIS_LN for lines.

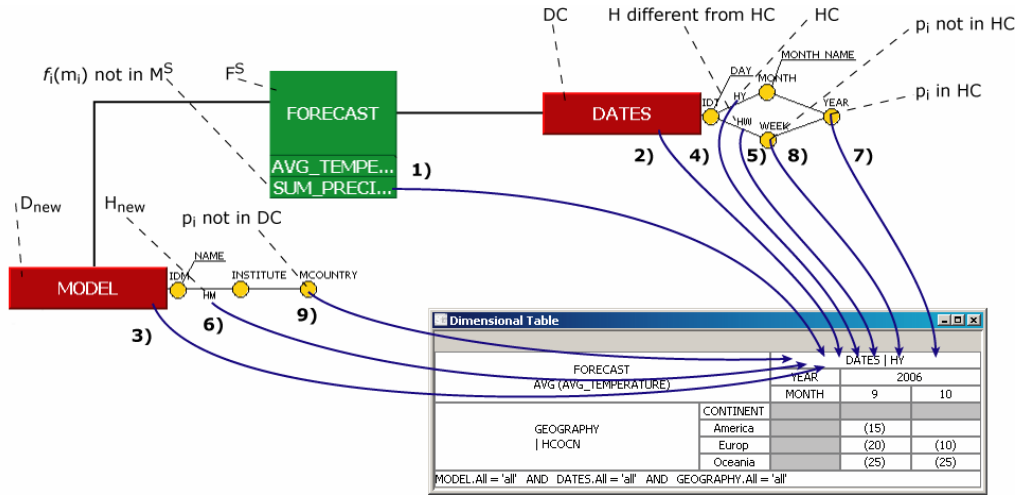


Figure 18: Possible drag and drop manipulations for the instruction DIS_COL(E).

Display-oriented operations: INS_LN, INS_COL and DEL

The graphic instruction INS_COL (respectively INS_LN) is used to insert an element into a column (respectively a line). Formal specification for INS_COL may be found in Table 10.

As before, INS_LN is not presented. The corresponding “drag and drop” actions are presented in Figure 19. Notice that actions represented by dashed arrows take into account the exact position where the drop action took place. E.g. if the element is dropped between Year and Month, the element is inserted between both. More details are available in appendix B.

INS COL(E)		
Source element of the action (and conditions)	Algebraic equivalent (T_{RES})	
1)	$E=f_i(m_i) \mid f_i(m_i) \in M^{FS}$	PULL($T_{SRC}, f_i(m_i), DC$)
2)	$E=DC$	DRILLDOWN(...(DRILLDOWN(ROLLUP(T_{SRC}, DC, p^{DC}_{vc}), DC, p^{DC}_{vc-1})...), DC, p^{DC}_i) all parameters of HC are displayed: $PC = Param^{HC}$
3)	$E=HC$	DRILLDOWN(...(DRILLDOWN(ROLLUP(T_{SRC}, DC, p^{DC}_{vc}), DC, p^{DC}_{vc-1})...), DC, p^{DC}_i) all parameters of HC are displayed: $PC = Param^{HC}$

4)	$E=p_i \mid p_i \in PC$	NEST(DELETE(T_{SRC} , DC, p_i), DC, $p_i?$, DC, p_i), with $p_i \in PC$
5)	$E=p_i \mid p_i \notin PC \wedge p_i \in HC$	DRILLDOWN(T_{SRC} , DC, p_i), if $level^{HC}(p_i) < level^{HC}(p_{Cmin})$ DRILLDOWN(...(DRILLDOWN(ROLLUP(T_{SRC} , DC, p_i), DC, p_{i+1})...), DC, p_{Cmin}), if $level^{HC}(p_i) > level^{HC}(p_{Cmin})$ ⁽¹⁾
6)	$E=p_i \mid p_i \notin HC \wedge p_i \in H \wedge H \in H^{DC}$	NEST(T_{SRC} , DC, $p_i?$, DC, p_i), with $p_i \in PC$
7)	$E=p_i \mid p_i \notin A^{DC} \wedge p_i \in A^{D_{new}} \wedge D_{new} \in Star^{Cs}(F^S)$	NEST(T_{SRC} , DC, $p_i?$, D_{new} , p_i), with $p_i \in PC$
⁽¹⁾ = p_i not displayed ($p_i \notin PC \wedge p_i \in HC$), $p_{C_{i+1}}$ (respectively $p_{C_{i-1}}$) is the attribute immediately inferior (resp. superior) to p_i in PC: $level^{HC}(p_{C_{i-1}}) = level^{HC}(p_i) - 1$ (resp. $level^{HC}(p_{C_{i+1}}) = level^{HC}(p_i) + 1$)		

Table 10: Formal specification of INS_COL(E), INS_LN for lines.

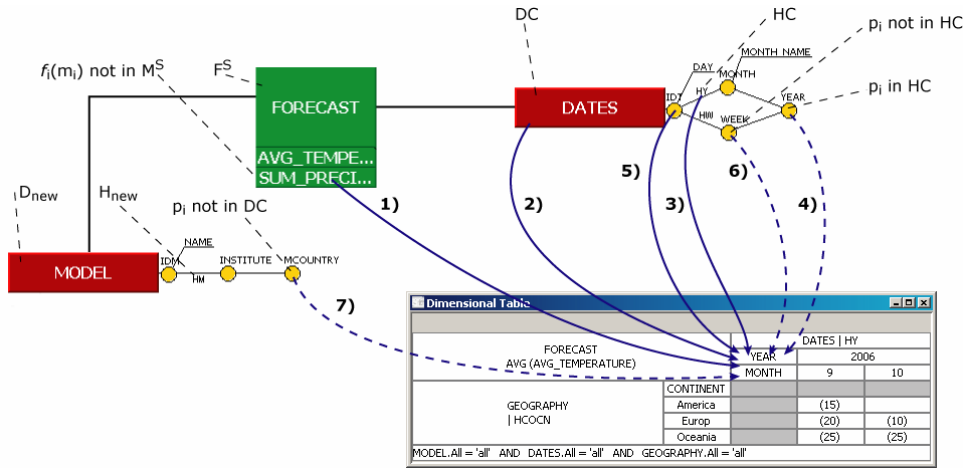


Figure 19: Possible drag and drop manipulations for the instruction INS_COL(E).

The graphic instruction DEL is used to remove a displayed element in a MT. When removing all the elements of a MT zone such as the fact, the user is prompted to designate a new one. Formal specifications are provided in Table 11 and “drag and drop” actions are described in Figure 20.

DEL(E)		
Source element of the action (and conditions)		Algebraic equivalent (T_{RES})
1)	$E=F^S$	FRotate(T_{SRC} , $F_{new}?$, $\{f_i(m_1), \dots, f_w(m_w)\}$), $\forall i \in [1..w] m_i \in M^{F_{new}}$)
2)	$E=f_i(m_i) \mid f_i(m_i) \in M^S$	DELM(T_{SRC} , $f_i(m_i)$)
3)	$E=DC$	ROTATE(T_{SRC} , DC, $D_{new}?$, $H?$), $D_{new} \in Star^{Cs}(F^S)$, $H \in H^{D_{new}}$
4)	$E=HC$	ditto above: DEL(DC)
5)	$E=p_i \mid p_i \in PC$	DELETE(DC, p_i)
6)	$E=R$ (zone de restriction) ⁽¹⁾	UNSELECT(T_{SRC})
⁽¹⁾ = This operation is done on the restriction zone (R) of the MT.		

Table 11: Formal specification of DEL(E).

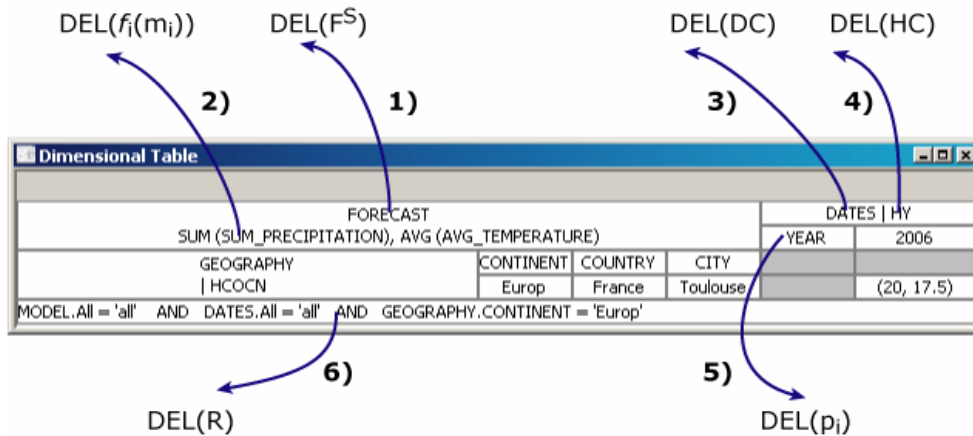


Figure 20: Possible drag and drop manipulations for the instruction `INS_COL(E)`.

Completeness of the language

The graphic OLAP query language rests upon two possible user interactions: drag and drop (DnD) or contextual menus. The DnD actions may be executed from the conceptual graph towards the MT, or from the MT to the same MT, or from the MT to outside the MT. The contextual menu may be called on the conceptual graph elements or on the elements in the MT. Manipulations with the contextual menus are assisted with dialog boxes. It is easy to understand that the contextual menus have a greater expressive power in the number of different operations that they may allow. As specified in the following table, graphical manipulations are complete with regard to the algebraic operators.

		Core Operators											
		Rotate	DrillDown	RollUp	Nest	Switch	Aggregate	Unaggregate	Push	Pull	AddM	DelM	Select
Drag and Drop	Graph-mTable	x	x	x	x				x	x	x		x
	mTable-mTable				o	x			o	o			o
	mTable-outside	x		o				x				x	
Contextual Menu	Graph	x	x	x	x		x	x	x	x	x	x	x
	mTable	x	x	x		x	x	x	x	x	x	x	x
Graphic support of the operations		x	x	x	x	x	x	x	x	x	x	x	x

o = partial support of the operator; p=complete support of the operator

Table 12: Graphic actions for each operator / operation.

Implementation

Our prototype is built using Java JDK 1.5 on top of the Oracle 10g DBMS. Composed of a hundred of classes, it allows the definition and manipulation of a R-OLAP constellation as well as visualizing and querying the multidimensional data with the use of a graph and an MT (the figures illustrating the present article are screenshots from the prototype).

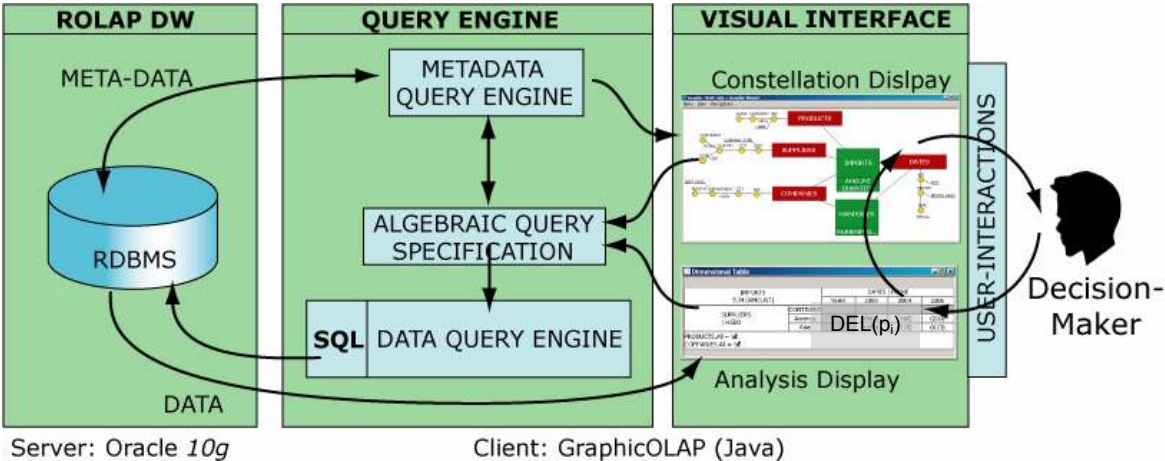


Figure 21: Prototype architecture.

The visual interface allows decision-makers to specify analysis queries with the constellation display interface as well as with the analysis display interface. The constellation display is done with the use of meta-data describing the R-OLAP data warehouse architecture. Graphic queries are then translated into algebraic expressions, which are in turn, translated into SQL queries. Correctness of query expressions is validated through meta-data. These SQL queries are sent to the R-OLAP data warehouse and results are sent back to the MT which updates its display. Algebraic translation between graphic and algebraic expressions is done according to tables throughout this section and in appendix.

CONCLUDING REMARKS

The goal of this paper is to provide a user-oriented OLAP environment. The provided solution is based on three points: a conceptual model that disregards implementation issues, an OLAP

query algebra that provides formal specifications of manipulation operators and a graphic language easing graphic multidimensional OLAP analyses.

The conceptual model we define allows a multidimensional representation of the available data. This model represents a constellation of facts, dimensions, and hierarchies. The constellation has the advantage to ease correlations between analysis subjects (multi-fact modelling). The model, based on multi-hierarchical analysis axes, provides multiple data analysis perspectives. This model acts as a basis for of OLAP manipulation languages.

To formalize multidimensional OLAP analyses, we first define a query algebra. It allows the selection of analysis data to be displayed from a multidimensional schema (operator Display).

This algebra uses a decisional analysis display structure adapted to decision-makers: a multidimensional table (MT), i.e. a hierarchical bi-dimensional table. This algebra allows decision-makers to express the different operations, which may be performed during an analysis. Query correctness is guaranteed through the closure property supported by the algebra. Each operator produces as output a multidimensional table that is a compatible input for the other operators.

We defined a minimal core and advanced operations. The core operators allow the following modifications:

- the data granularity or the analysis precision (Drilldown, Rollup and Select);
- the analysis criteria (Rotate, Push, Pull, Nest, AddM and DelM);
- the structure of the resulting table (Switch, Aggregate).

In order to simplify complex OLAP query expression and to improve query processing, we provide a set of advanced operators defined by combinations of the core algebra operators.

Finally to analyse two MT, we provide three set operators based on compatible or semi-compatible tables.

Although user-oriented our algebra is difficult for decision-makers. As a consequence, we have defined a graphic language. This language is based on two interfaces. The first one displays the graphical schema of a multidimensional database and the second displays the resulting multidimensional table. To perform a multidimensional OLAP analysis, the decision-maker selects directly an element of the multidimensional schema and drags it in the result MT. These drag and drop actions may be completed by a set of menus that specify the analysis elements. The graphic language allows the definition of an analysis in an incremental way and at each step, the decision-maker visualises the resulting MT. The graphic language is complete with regard to the algebraic core. For each graphic instruction, we have specified the equivalent combination of algebraic operators. Finally, the language has been validated by its implementation in a prototype (figures throughout this paper are screen captures). The prototype is based on an implementation of the multidimensional concepts in a ROLAP environment with the DBMS Oracle. The graphic language is composed of a hundred java classes and other components.

The next step of these works is to take into account multidimensional schema data evolutions. Versions could be used for the data evolution tracking (Ravat et al., 2006b), and within this framework, OLAP manipulation operators should be revised. Another step in OLAP systems is the integration of XML data. This will require adapted multidimensional models and methodologies (Rusu et al., 2005), (Nassis et al., 2005) (Messaoud et al., 2006) as well as adapted analysis operators (Ravat, et al., 2007).

REFERENCES

-
- Abelló, A., Samos, J., & Saltor, F., (2003). Implementing operations to navigate semantic star schemas. *6th ACM international workshop on Data Warehousing and OLAP (DOLAP2003)*, ACM, pp.56-62.

- Abelló, A., Samos, J., & Saltor, F., (2006). YAM²: a multidimensional conceptual model extending UML. *Journal of Information Systems (IS)*, vol.31(6), Elsevier, pp. 541-56.
- Agrawal, R., Gupta, A., & Sarawagi, S., (1997). Modeling Multidimensional Databases. *13th Int. Conf. Data Engineering (ICDE)*, IEEE Computer Society, pp. 232–243.
- Blaschka, M., Sapia, C., Höfling, G., & Dinter, B., (1998). Finding Your Way through Multidimensional Data Models. *9th Int. Workshop on Database and Expert Systems Applications: Data Warehouse Design and OLAP Technology (DWDOT'98)*, IEEE Computer Society, pp.198-203.
- Böhnlein, M., Plaha, M., & Ulbrich-vom Ende, A., (2002). Visual Specification of Multidimensional Queries based on a Semantic Data Model. *Vom Data Warehouse zum Corporate Knowledge Center (DW2002)* Physica-Verlag, pp.379-397, 2002.
- Cabibbo, L., & Torlone, R., (1998). From a Procedural to a Visual Query Language for OLAP. *10th Int. Conf. on Scientific and Statistical Database Management (SSDBM 1998)*, IEEE Computer Society, pp. 74-83.
- Cabibbo L., & Torlone R., (1997). Querying Multidimensional Databases. *Database Programming Languages, 6th International Workshop (DBPL-6)*, LNCS 1369, Springer, pp. 319-335.
- Chaudhuri, S., & Dayal, U., (1997). An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, vol.26(1), pp. 65-74.
- Colliat, G., (1996). OLAP, Relational, and Multidimensional Database Systems. *SIGMOD Record*, vol.25(3), pp. 64-69.
- Datta, A., & Thomas, H., (1999). The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. *Journal of Decision Support Systems (DSS)*, vol.27(3), Elsevier, pp. 289-301.

- Franconni, E., & Kamble, A., (2004). The GMD Data Model and Algebra for Multidimensional Information. *16th Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, LNCS 3084, Springer, pp.446-462.
- Golfarelli, M., Maio, D., & Rizzi, S., (1998). Conceptual design of data warehouses from E/R schemes. *31st Hawaii Int. Conf. on System Sciences*.
- Gray, J., Bosworth, A., Layman, A., & Pirahesh, H., (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. *12th Int. Conf. on Data Engineering (ICDE)*, IEEE Computer Society, pp. 152–159.
- Gyssen, M., & Lakshmanan, L.V.S., (1997). A Foundation for Multi-Dimensional Databases. *23rd Int. Conf. on Very Large Data Bases (VLDB'97)*, Morgan Kaufmann, pp. 106–115.
- Kimball, R., (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*". John Wiley & Sons Inc, 2nd ed. 2003.
- Lamping, J., & Rao R., (1994). Laying out and visualizing Large Trees Using a Hyperbolic Space. *7th ACM Symposium on User Interface Software and Technology (UIST'94)*, ACM, pp.13-14.
- Lehner, W., (1998). Modeling Large Scale OLAP Scenarios. *6th Int. Conf. on Extending Database Technology (EDBT'98)*, LNCS 1377, Springer, pp. 153-167.
- Li, C., & Wang, X.S., (1996). A Data Model for Supporting On-Line Analytical Processing. *5th Int. Conf. on Information and Knowledge Management (CIKM'96)*, ACM, pp. 81-88, 1996.
- Malinowski, E., & Zimányi, E., (2006). Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Journal of Data & Knowledge Engineering (DKE)*, vol.59(2), Elsevier, pp. 348-377.
- Maniatis, A., Vassiliadis, P., Skiadopoulos, S., Vassiliou, Y., Mavrogonatos, G., & Michalarias, I., (2005). A Presentation Model & Non-Traditional Visualization for

- OLAP. *International Journal of Data Warehousing & Mining (DWM)*, vol.1(1), Idea Group Publishing, pp. 1-36.
- Messaoud, R. B.; Boussaid, O., & Rabaséda, S. L., (2006). A Data Mining-Based OLAP Aggregation of Complex Data: Application on XML Documents. *International Journal of Data Warehousing & Mining (DWM)*, vol.2(4), Idea Group Publishing, pp. 1-26.
- Nassis, V., Rajagopalapillai, R., Dillon, T.S., & Rahayu, W., (2005). Conceptual and Systematic Design Approach for XML Document Warehouses. *International Journal of Data Warehousing & Mining (DWM)*, vol.1(3), Idea Group Publishing, pp. 63-87.
- Niemi, T., Hirvonen, L., & Jarvelin, K., (2003). Multidimensional Data Model and Query Language for Informetrics. *Journal of the American Society for Information Science and Technology (JASIST)*, vol.54(10), pp 939-951.
- Pedersen, T.B., Jensen, C.S., & Dyreson, C.E., (2001). A foundation for capturing and querying complex multidimensional data. *Journal of Information Systems (IS)*, vol.26(5), Elsevier, pp. 383-423.
- Rafanelli, M., (2003). Operators for Multidimensional Aggregate Data. Chapter 5 of *Multidimensional Databases: Problems and Solutions*, M. Rafanelli (ed.), Idea Group Publishing, ISBN 1-59140-053-8, pp. 116-165.
- Ravat, F., Teste, O., & Zurfluh, G., (2006). Constraint-Based Multi-Dimensional Databases. Chap. XI of *Database Modeling for Industrial Data Management*, Zongmin Ma (ed.), Idea Group Publishing, pp.323-368.
- Ravat, F., Teste, O., & Zurfluh, G., (2006). A multiversion-based multidimensional model. 8th *Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK 2006)*, LNCS 4081, Springer, pp. 65-74.

- Ravat, F., Teste, O., & Tournier, R., (2007). OLAP Aggregation Function for Textual Data Warehouse. *International Conference on Enterprise Information Systems (ICEIS 2007)*, INSTICC Press.
- Rizzi, S., Abelló, A., Lechtenböcker, J., & Trujillo, J., (2006). Research in data warehouse modeling and design: dead or alive? *9th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2006)*, ACM, pp.3-10.
- Rusu, L. I.; Rahayu, J. W., & Taniar, D., (2005). A Methodology for Building XML Data Warehouses. *International Journal of Data Warehousing & Mining (DWM)*, vol.1(2), Idea Group Publishing, pp. 23-48.
- Stolte, C., Tang, D., & Hanrahan, P., (2002). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, vol.8(1), pp.52-65.
- Torlone, R., (2003). Conceptual Multidimensional Models. Chapter 3 of *Multidimensional Databases: Problems and Solutions*, M. Rafanelli (ed.), Idea Group Publishing, pp. 69-90.
- Trujillo, J. C., Luján-Mora, S., & Song I., (2003). Applying UML for designing multidimensional databases and OLAP applications. Chapter 2 of *Advanced Topics in Database Research Volume 2*, K. Siau (Ed.), Idea Group Publishing, pp. 13-36.
- Vassiliadis, P., & Sellis, T.K., (1999). A Survey of Logical Models for OLAP Databases. *SIGMOD Record* 28(4), pp. 64-69.

APPENDIX-A: CONSTELLATION GRAPHICAL VISUALISATION

If a dimension is composed of a complex hierarchical structure, the structure may be exploded, in order to see easily the different parameters of each hierarchy. This is useful as drilling operations follow the hierarchical order of the dimensional parameters. For example

in Figure 22, the dimension named GEOGRAPHY is composed of 6 parameters spread over 4 different hierarchies.

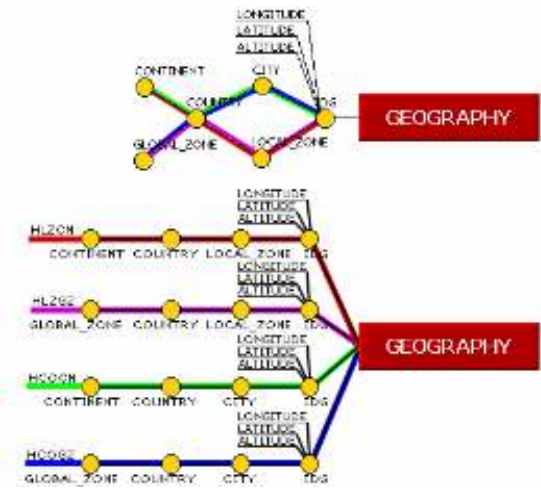


Figure 22: The GEOGRAPHY dimension in (top) compact version and (bottom) split version.

It is not exceptional that a corporate multidimensional database holds several facts and numerous dimensions. Users may be lost in large constellations. An alternative visualisation is provided in order to focus on relevant parts of the constellation graph. The constellation graph is projected on a hyperbolic space (Lamping and Rao, 1994) and decision-makers “move” quickly the graph in order to see relevant nodes.

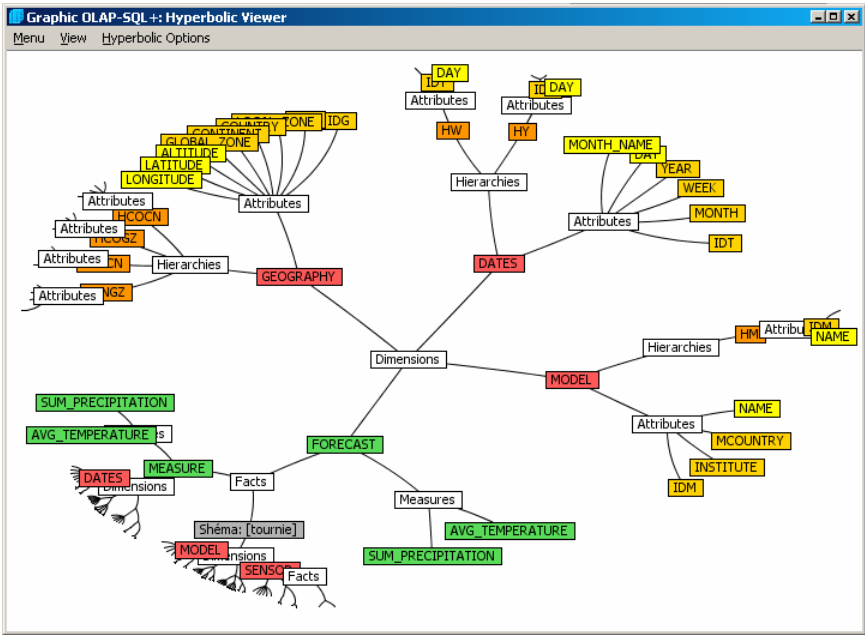


Figure 23: Hyperbolic view of a constellation.

APPENDIX-B: COMPLEMENTARY FORMAL SPECIFICATIONS

The following table presents specific operations that modify the MT display.

Source element of the action (and conditions)	Algebraic equivalent (T_{RES})
SELECT(E)	
$E = f_i(m_i) \mid f_i(m_i) \in M^{FS}$	SELECT(pred?), pred is a predicate on $\text{dom}(f_i(m_i))$
$E = p_i \mid p_i \in A^D, D \in \text{Star}^{Cs}(F^S)$	SELECT(pred?), pred is a predicate on $\text{dom}(p_i)$
SWITCH(X) ⁽¹⁾	
$X = \text{val}_x \mid \text{val}_x \in \text{dom}(p_i), p_i \in PC$	SWITCH($T_{SRC}, DC, p_i, \text{val}_x, \text{val}_y?$), with $\text{val}_y \in \text{dom}(p_i)$
AGGREGATE(E) and UNAGGREGATE(E)	
$E = p_i \mid p_i \in PC, \text{dom}(p_i) = \langle v_1, \dots, v_x \rangle$	AGGREGATE($T_{SRC}, DC, f_i?(p_i)$)
$E = p_i \mid p_i \in PC, \text{dom}(p_i) = \langle v_1, f_i(v_1), \dots, v_x, f_i(v_x) \rangle$	UNAGGREGATE(T_{SRC})
ORDER(E)	
$E = p_i \mid p_i \in PC$	ORDER($T_{SRC}, DC, p_i, \text{ord}?$)

Table 13: Formal specification of the algebra-oriented graphic operations.

Table 14 presents the formal specification of the two drilling operations.

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

Table 14: Formal specification of the graphic drilling operations.

Figure 24 illustrates some graphic drag and drop operations that may be used on a parameter header zone. These graphic manipulations consist in “inserting in line” (INS_LN). The position of drop actions determines the equivalent algebraic operations; e.g., in figure 23, the parameter noted *Global_Zone* is dropped between *Country* and *City*, thus this action is translated in a nest operator $\text{NEST}(T, \text{GEOGRAPHY}, \text{Country}, \text{GEOGRAPHY}, \text{Global_Zone})$.

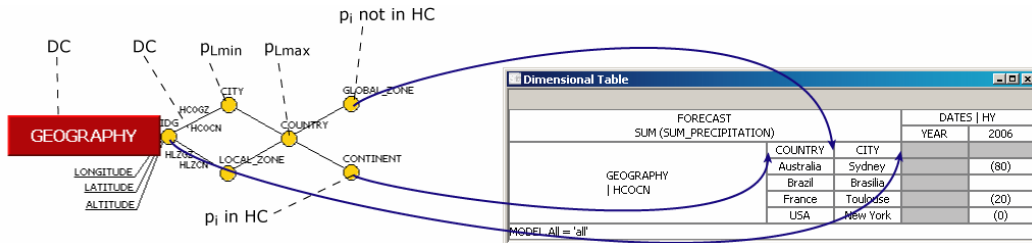


Figure 24: Some specific nesting and drilling interactions (INS_LN(E)).

-
- 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/>