Multigranular Manipulations for OLAP Querying

Gilles Hubert and Olivier Teste

Abstract. Decisional systems are based on multidimensional databases improving OLAP analyses. This chapter describes a new OLAP operator named "BLEND" that performs multigranular analyses. This operation transforms multidimensional structures when querying in order to analyze measures according to several granularity levels like one parameter. We study valid uses of this operation in the context of strict hierarchies. Experiments within a R-OLAP implementation show the light cost of the operator.

Keywords: Decision Support Systems, Multidimensional Databases, OLAP Querying, Multigranular Analysis.

1 Introduction

Decision support systems are experiencing a great boost in development because of their capacity to effectively support analyses on available data in the organizations. These decision systems are elaborated starting from the operational system of an organization: the data identified as relevant for decision makers are extracted, transformed, then loaded (Vassiliadis *et al.*, 2002) in a centralized storage space called data warehouse. In order to improve querying and analysis of these stored data, specific techniques of data organization were developed (Kimball, 1996) based on multidimensional databases (MDB). This type of modeling considers the data to be analyzed as points in a space with several dimensions, thus forming a datacube of data (Gray *et al.*, 1996). Decision makers who use these systems visualize an excerpt of the datacubes, generally a "slice" with only two dimensions of a

Gilles Hubert · Olivier Teste
Université de Toulouse, IRIT (UMR 5505), équipe SIG, 118 Route de Narbonne,
31062 Toulouse cedex 9, France
e-mail: {Gilles.Hubert,Olivier.Teste}@irit.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 97–112.
springerlink.com

cube. From this structure, called multidimensional table (Gyssens and Lakshmanan, 1997), the decision maker can interact with operations. The best known operations are drilling operations which consist in modifying the graduation of an analysis axis (levels of granularity) and rotations which consist in changing the cube slice. One speaks about online analysis or about OLAP ("On-line Analytical Processing") (Ravat *et al.*, 2008).

This environment offers an adapted framework to decision makers' analyses; however the imposed structure can prove to be imperfect or become obsolete. Let us consider sale amounts analyzed according to French customers and American customers. Within this framework, a decision maker may want to use the graduation according to the country for the French customers while wishing to use a different graduation simultaneously, for example the states for the customers of the USA. Indeed, for some analyses, it is necessary to compare a country like France with different geographic entities like states to compare information equivalent in size, population size, etc. The objective of this paper is to propose a solution allowing these manipulations described as *multigranular*.

2 Related Work

There exist mainly two approaches for MDB modeling: an approach based on the datacube metaphor whereby an MDB is represented by cubes; and an approach known as multidimensional modeling whereby a MDB is represented by a star or constellation schema (Kimball, 1996). Several field surveys (Chaudhuri and Dayal, 1997; Vassiliadis and Sellis, 1999) and comparative studies (Abelló *et al.*, 2006; Ravat *et al.*, 2008) are available.

One of the first works extends the aggregation operation in the OLAP context (Gray *et al.*, 1996). Since then, a great number of operations were defined; however due to lack of consensus on a reference model, the proposals for OLAP operations still were neither clearly identified nor defined within an algebra following the example of the relational approach. A comparative study of the many existing proposals is available in Romero and Abelló (2007).

To our knowledge no proposal can answer our problems. The closest solutions propose mechanisms aiming at personalizing an MDB by transforming its values and its structures. In Espil and Vaisman (2001) the rule-based language IRAH is introduced to allow decision makers to change value groupings between two graduations. However, this approach does not make it possible to transform the hierarchical structures of the graduations initially defined in the MDB. This approach does not allow multigranular analyzes by combining existing graduations; e.g., it allows a cisgenic organism denoted (C1, CIS) to become a transgenic organism denoted (C1, TRA). Our approach aims at generating a new graduation both composed of organisms (C1) and categories such as transgenic (TRA). More recently, Favre *et al.* (2007) introduced a mechanism based on "If-Then" rules in order to integrate users' knowledge to change the MDB schema. This mechanism allows users to add new graduations individually. Although these solutions allow a certain adaptation of a MDB it raises two problems: firstly, the transformation process is tricky and tedious because based on the definitions of rules expressed by the decision maker, and secondly, coherence and confidence with the stored decisional data are not guaranteed any more. Introducing direct means to access values in update mode renders inoperative the usual processes of data cleaning and consolidation.

Other works in MDB evolution context proposed operations to transform the hierarchies modeled initially (Blaschka *et al.*, 1999; Hurtado *et al.*, 1999; Eder *et al.*, 2003). In Blaschka *et al.* (1999), an operation to insert a new parameter is presented. The operation "Reports Levels", defined in Hurtado *et al.* (1999), makes it possible to transform the hierarchical organization of parameters. Other transformation operations such as "Split" related to parameter values are described in Eder *et al.* (2003). This work offers a framework allowing the evolution of hierarchies, but does not really correspond to multigranular transformations. These operations can be diverted to transform an MDB. However, our goal is different as it aims to help reorganize the values between two graduations, and this, during the analysis process, without impacting the data physically stored in the MDB.

3 Contribution and Organization

The main contribution of this article is the proposal of a new manipulation in MDB facilitating multigranular analyses. A multigranular analysis combines the same analyzed measurements according to data resulting from several parameters: for example, we make possible the analysis of agricultural surfaces according to geographical values of different levels such as USA and European surfaces.

We extend the OLAP algebra, defined in our laboratory (Ravat *et al.*, 2008), by the multigranular analysis operator "BLEND". We carry out a study of the various possible uses of the operator in the context of strict hierarchies (Malinowski and Zimányi, 2006). We propose an operation that transforms the current hierarchy and the contours limits of the operator. Lastly, we experiment the operation in the context of a R-OLAP implementation.

An advantage of the suggested solution, is to make possible this type of analysis during analysis runtime whereas it would require complete data reorganization as well as associated ETL processes in a traditional context. The construction of an MDB is a tedious task and difficult to reproduce according to each analytical need. Applying these transformations during analysis runtime without impacting the real data organization facilitates sharing the MDB.

Section 4 presents the MDB model, i.e. the conceptual representation we adopt. We define a new operator called "BLEND" in Sect. 5. We show the various possible cases of multigranular manipulation authorized in the context of strict hierarchies (Malinowski and Zimányi, 2006). Section 6 describes the implementation of the operator in a R-OLAP context.

4 Multidimensional Modeling and OLAP Manipulations

This section describes our multidimensional framework based on a conceptual view displaying MDB structures as a graphical conceptual view. Our model allows users to disregard technical and storing constraints and sticks closer to decision makers' view (Golfarelli *et al.*, 2002). It allows a clear distinction between structural elements and values and offers a workable visualization for decision makers (Gyssens and Lakshmanan, 1997).

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

Definition 1. A *constellation* C is defined as $(NC, FC, DC, StarC)$ where:

- NC is a constellation name,
- $FC = \{F_1, \ldots, F_m\}$ is a set of facts,
- $DC = \{D_1, \ldots, D_n\}$ is a set of dimensions,
- $StarC : FC \rightarrow 2^{DC}$ associates each fact to its linked dimensions.

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

Definition 2. A *dimension*, noted $D \in DC$, is defined as (N^D, A^D, H^D) where:

- N^D is a dimension name,
- $A^D = \{a_1^D, \ldots, a_u^D\} \cup \{id^D, All\}$ is a set of attributes,
- $H^D = \{H_1^D, \ldots, H_v^D\}$ is a set of hierarchies.

Dimension attributes (also called parameters or levels) are organized 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 can be analyzed; 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 3. A *hierarchy* of a dimension D, noted $H \in HD$, is defined as $(N^H, Param^H, Weak^H)$ where:

- N^H is a hierarchy name,
- $Param^H = \langle id^D, p_1^H, \ldots, p_v^H, All \rangle$ is an ordered set of attributes, called *parameters*, which represent useful graduations along the dimension, $\forall k, p_k^H \in A^D$,
- $Weak^H : Param^H \rightarrow 2^{A^D - Param^H}$ is a function possibly associating each parameter to one or several *weak attributes*.

All hierarchies of a dimension start with a same parameter, noted id^D called *root parameter* and end with a same parameter, noted All called *extremity parameter*.

A fact reflects information that has to be analyzed according to dimensions. This analyzed information is modeled through one or several indicators, called measures; for example, a fact data may be sale amounts occurring in shops every day. The notation $D \in StarC(F)$ represents that the dimension D is linked to the fact F.

Definition 4. A *fact*, noted $F \in FC$, is defined as (N^F, M^F) where:

- N^F is a name of fact,
- $M^F = \{f_1(m_1^F), \ldots, f_w(m_w^F)\}$ is a set of *measures* associated with an aggregate function.

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

Definition 5. A *multidimensional table T* is defined as (S, L, C, R) where:

- $S = (F^S, M^S)$ represents the analyzed subject through a fact $F^S \in FC$ and a set of projected measures $M^S = \{f_1(m_1), \ldots, f_x(m_x)\}$ where $\forall i \in [1..x], m_i \in M^F$,
- $L = (DL, HL, PL)$ represents the horizontal analysis axis where $PL = \langle All, p_{max}^{HL}, \ldots, p_{min}^{HL} \rangle$, $HL \in H^{DL}$ and $DL \in StarC(F^S)$, HL is the current hierarchy of DL,
- $C = (DC, HC, PC)$ represents the vertical analysis axis where $PC = \langle All, p_{max}^{HC}, \ldots, p_{min}^{HC} \rangle, HC \in H^{DC}$ and $DC \in StarC(F^S)$, HC is the current hierarchy of DC,
- $R = pred_1 \wedge \ldots \wedge pred_t$ is a normalized conjunction of predicates (restrictions of dimension data and fact data).

Example 1. We consider an MDB to analyze the surface of parcels of land with genetically modified (GM) organisms around the world. The constellation is composed of one fact and three dimensions (see Fig. 1). The graphical notations we adopt are inspired from the notations of (Golfarelli *et al.*, 1998). According to formal definitions the graphical constellation is defined as follows:

('C1', $\{F^{DISTRIBUTION}\}, \{D^{ORGANISM}, D^{DATE}, D^{GEOGRAPHY}\}, \{(F^{DISTRIBUTION}, \{D^{ORGANISM}, D^{DATE}, D^{GEOGRAPHY}\})\}$)

The fact denoted $F^{DISTRIBUTION}$ is defined as follows:

('DISTRIBUTION', {SURFACE})

The dimension denoted $D^{GEOGRAPHY}$ is defined as follows:

('GEOGRAPHY', {PARCEL, STATE, REGION, COUNTRY, CONTINENT, DENSITY}, {HGEO, HST}) where:

- HGEO = ('GEO', ⟨PARCEL, REGION, COUNTRY, CONTINENT⟩, {(COUNTRY, {DENSITY})})
- HST = ('ST', ⟨PARCEL, STATE, COUNTRY, CONTINENT⟩, {(COUNTRY, {DENSITY})})

A decision maker displays data into multidimensional tables; T_1 displays surfaces according to continents and organism types. This MT is transformed into T_2 using a combination of drill-down and roll-up operations for displaying surfaces according to country and organism varieties. T_2 allows the decision maker to compare parcels with GM organisms (GTS-Soya, Corn BT176 and Mon 810) as well as parcels without GM organisms.

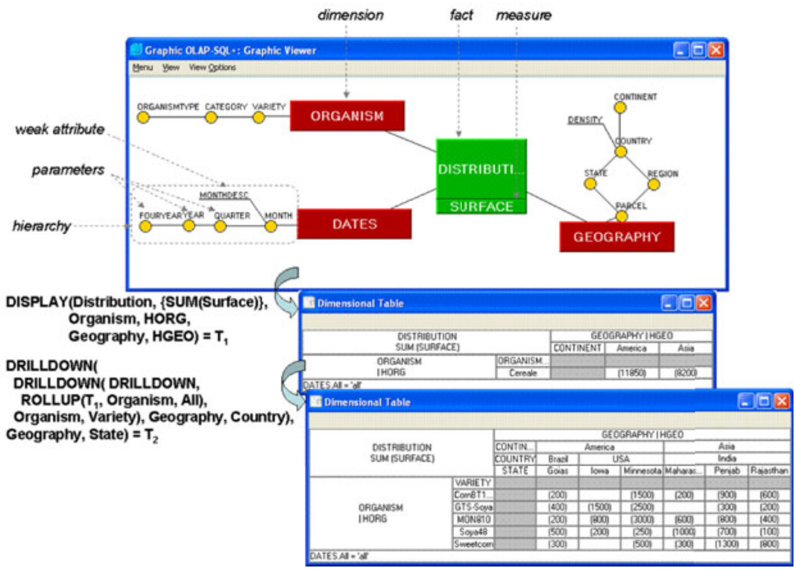


Fig. 1 Star schema example (constellation composed of only one fact) and two multidimensional tables resulting from OLAP operations

5 Operator "BLEND"

In order to answer our problems of multigranular analyses, we define an operation to transform dimension parameters. This operation named "BLEND" is applied to an MT in order to modify the headings of the lines or the columns.

5.1 Algebraic Operator

Definition 6. The operation of multigranular transformation of a MT is defined by: $BLEND(T_{SRC}, D, P_{sup}(s_{sup}), P_{inf}(s_{inf}), pred) = T_{RES}$

- $T_{SRC} = (S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC})$ is the source MT to transform
- $D \in \{DL_{SRC}, DC_{SRC}\}$ is one of the dimensions of the MT T_{SRC}
- P_{sup} and P_{inf} are consecutively displayed parameters of the dimension D such that P_{sup} is the parameter hierarchically higher than P_{inf},

- $s_{sup} \in \{+,-\}$ and $s_{inf} \in \{+,-\}$ are tags indicating the conservation $(+)$ or not $(-)$ of the parameter associated in T_{RES}; the use of the tags and their various combinations are studied in an exhaustive way in the following section 5.2,
- *pred* is a selection predicate that determines the values resulting from the parameters P_{sup} and P_{inf} to build the definition field of the new parameter noted $P_{sup}\_P_{inf} \in T_{RES}$,
- T_{RES} is the resulting MT.

The predicate *pred* is used to compute the sets E_{sup} and E_{inf}, which gather the values resulting from the parameters P_{sup} and P_{inf} taking part in the construction of the new parameter field:

- E_{sup} contains the values of P_{sup} selected by *pred*,
- E_{inf} contains the values of P_{inf} selected by $\neg$ *pred*.

Constraint 1. *The predicate noted* pred *in the definition of operator "BLEND" is valid if and only if* $E_{sup} \cap ancestor(E_{inf}) = \varnothing$ *with:*

- $ancestor(E_{inf})$ *indicates the values of* $dom(P_{sup})$ *related to* E_{inf},
- $dom(P_{sup})$ *indicates the field definition of* P_{sup}.

For simplicity we will say that pred *must define two sets of values "disjoined" in comparison with the hierarchical organization.*

Constraint 2. *The composition of "BLEND" operators is not commutative. The user must build his manipulations taking into account the order of the parameters* P_{sup} *and* P_{inf}*, but also the order of the combinations of the multigranular transformations.*

5.2 Transformation Cases

The operator "BLEND" modifies the existing hierarchy by substituting a new parameter to one of the existing parameters (or both) or by integrating a new parameter in addition to the existing parameters. The interest of the operation is to allow the user to transform the existing hierarchy by the user replacing the initial hierarchy considered obsolete directly in the MT without reconstructing the MDB.

The integration of the new parameter can be carried out according to four scenarios:

- either the parameter replaces both existing P_{sup} and P_{inf} (Tab. 1-a);
- or the parameter replaces the P_{inf} parameter (Tab. 1-b);
- or the parameter replaces the P_{sup} parameter (Tab. 1-c);
- or the parameter is inserted between the parameters P_{sup} and P_{inf} (Tab. 1-d).

The tags added to the two parameters P_{sup} and P_{inf} indicate the selected scenario. The tag $(-)$ indicates that the parameter must not appear in the result while the tag $(+)$ indicates the opposite. In this way it is possible to transform two parameters by creating a new multigranular parameter, while maintaining whole or part of the possibilities of initial navigations (with drilling operations). For example, in Tab. 1, the scenario (a) removes the possibilities of drilling on the countries and the states (only the multigranular parameter is available) whereas (d) maintains the two initial parameters.

It is important to note that we present here only the possibilities which maintain strict hierarchies (Malinowski and Zimányi, 2006) in which any value of the lower parameter can be dependent on only one value of the higher parameter.

5.3 Operator Closure Property

The definition of the "BLEND" operator respects the closure property: it is applied to a MT and produces a new MT. This property allows chaining successive operations in order to operate complex transformations.

Example 2. Let us consider a complex analysis in which a decision maker wishes to compare the cereal surfaces between American states, a country such as Brazil and the Asian continent. This analysis is multigranular on three levels since it uses a continent, a country, and American states (subdivisions of a country). Starting from the MT T_2, we chain the two following multigranular transformations:

BLEND(BLEND(T_2, Geography, Country $(-)$**, State** $(-)$**, Country<>'USA'), Geography, Continent** $(-)$**, Country-State** $(-)$**, Continent='Asia') =** T_3

The following figures illustrate the sequence of the two operations with the corresponding multigranular transformations.

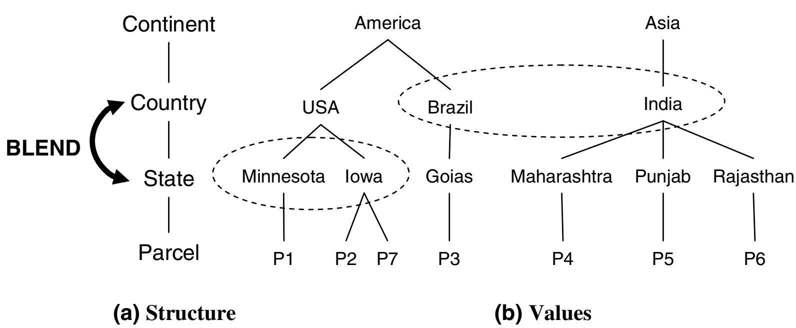


Fig. 2 Initial structure of Geography in T_2

The sequence of the two "BLEND" operations induces a multigranular transformation of the data of T_2. The resulting table T_3 is presented in Fig. 5.

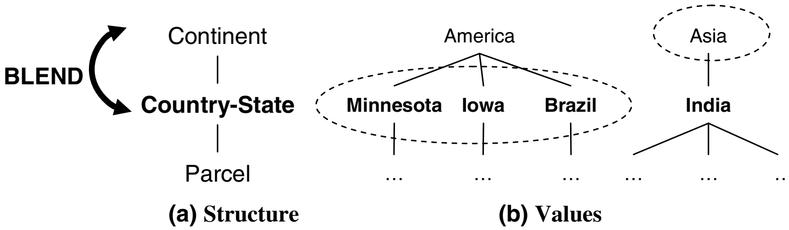


Fig. 3 Intermediate structure of Geography after the first “BLEND” Fig. 2

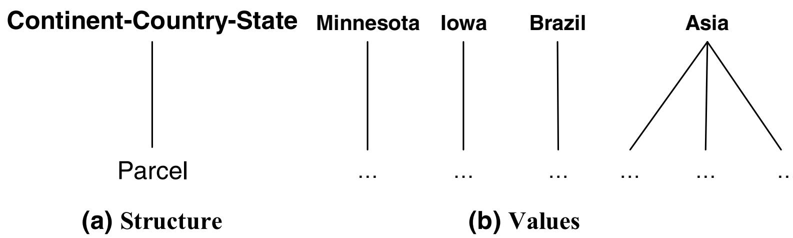
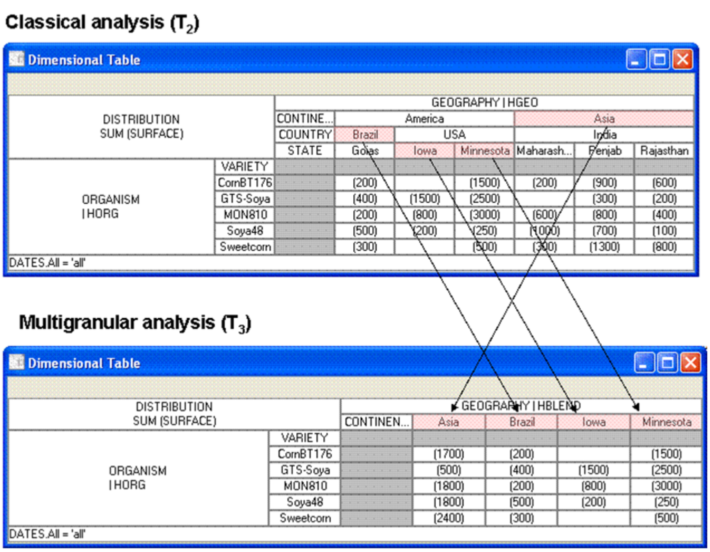


Fig. 4 Final structure of Geography in T_3 after the second “BLEND” Fig. 3



DISTRIBUTION SUM (SURFACE); ORGANISM | HORG; GEOGRAPHY | HGEO

CONTINENT	America	America	America	Asia	Asia	Asia
COUNTRY	Brazil	USA	USA	India	India	India
STATE / VARIETY	Goias	Iowa	Minnesota	Maharash...	Penjab	Rajasthan
CornBT176	(200)		(1500)	(200)	(900)	(600)
GTS-Soya	(400)	(1500)	(2500)		(300)	(200)
MON810	(200)	(800)	(3000)	(600)	(800)	(400)
Soya48	(500)	(200)	(250)	(1000)	(700)	(100)
Sweetcorn	(300)		(500)	(300)	(1300)	(800)

DATES.All = 'all'

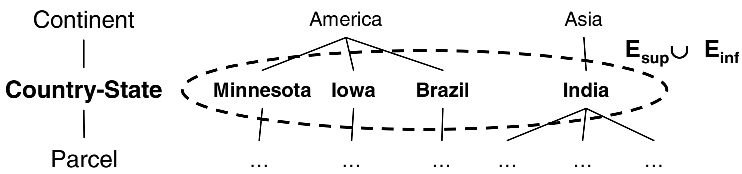
DISTRIBUTION SUM (SURFACE); ORGANISM | HORG; GEOGRAPHY | HBLEND

CONTINEN... / VARIETY	Asia	Brazil	Iowa	Minnesota
CornBT176	(1700)	(200)		(1500)
GTS-Soya	(500)	(400)	(1500)	(2500)
MON810	(1800)	(200)	(800)	(3000)
Soya48	(1800)	(500)	(200)	(250)
Sweetcorn	(2400)	(300)		(500)

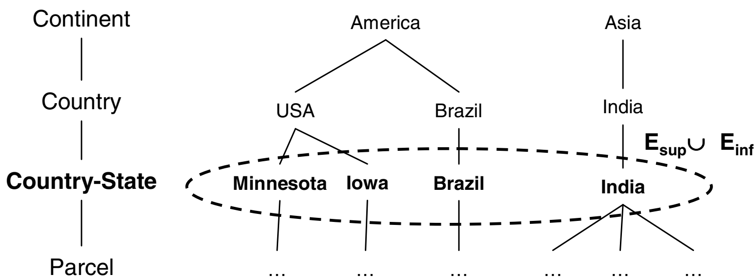
DATES.All = 'all'

Fig. 5 Principle of multigranular transformations

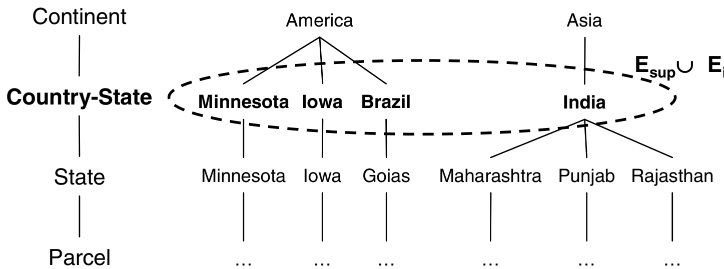
Table 1 Four possibilities of modification of the hierarchy



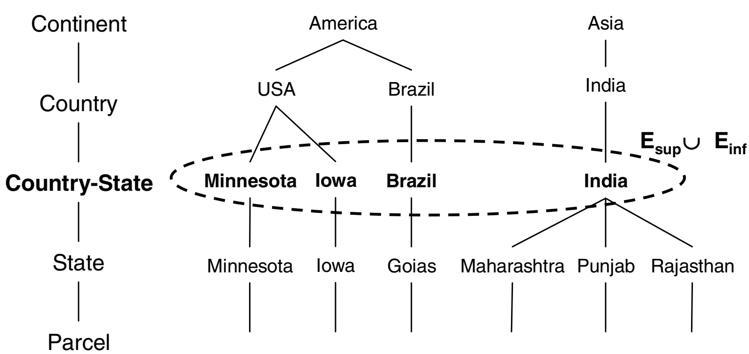
(a) BLEND (T_{SRC}, Geography, Country (−), State (−), Country <>'USA')



(b) BLEND (T_{SRC}, Geography, Country (+), State (−), Country <>'USA')



(c) BLEND (T_{SRC}, Geography, Country (−), State (+), Country <>'USA')



(d) BLEND (T_{SRC}, Geography, Country (+), State (+), Country <>'USA')

5.4 *Special Cases of the Operator*

Empty Selection

The predicate noted *pred* in the definition of “BLEND” could be an empty selection for the parameters P_{sup} or P_{inf}. If E_{sup}, respectively E_{inf}, is empty, then the operation is also valid. This special case consists in deleting the parameter P_{sup}, respectively P_{inf}. Note that it is possible to obtain this result using another combination of operators from the OLAP algebra (Ravat *et al.*, 2008). Note also that due to the definition of the operator, E_{sup} and E_{inf} cannot be empty at the same time.

Example 3. Let us consider a new operation that combines countries having a strong population density (Density $>$ 20) with states having a week population density. This multigranular transformation is defining as follows:

BLEND (T_3**, Geography, Country (**s_{sup}**), State (**s_{inf}**), Density** $>$ **20)**
where $s_{sup} \in \{+,-\}$ and $s_{inf} \in \{+,-\}$.

If country densities of the USA, Brazil and India are respectively 31.15, 21.60, and 300.24 hab/km^2, then the predicate ‘Density $>$ 20’ provides the following sets: E_{sup} = {‘USA’, ‘Brazil’, ‘India’} and $E_{inf} = \varnothing$. This special case where $E_{inf} = \varnothing$ consists in keeping the countries and deleting their states from the current analysis.

Root Parameter

Using the root parameter in the “BLEND” operator implies a dimension multigranular transformation and the associated measure values have to be recalculated. More precisely, deleting the root parameter values requires the aggregation of the associated measure values; e.g., each aggregated value is linked to an upper parameter value.

Example 4. Let us consider a multigranular transformation using the root parameter named ‘Parcel’. The decision maker wants to compare state surfaces of the USA and parcels of others countries. This multigranular transformation is defined as follows:

BLEND (T_3**, Geography, State (**s_{sup}**), Parcel (**s_{inf}**), Country = ‘USA’)**

This operation calculates sets such as E_{sup}={‘Minnesota’, ‘Iowa’} and E_{inf}= {‘P3’, ‘P4’, ‘P5’, ‘P6’}. In the resulting multidimensional table, measure values that are linked to the USA (‘P1’, ‘P2’ and ‘P7’) are aggregated to be linked to the states of E_{sup}. In the same way of the roll-up operations, the multigranular transformation uses the aggregation function defined from the initial constructor operation noted DISPLAY (see Fig. 1); in this example the SUM function is used.

Aggregation Functions

In this paper, we study the operator using the aggregation SUM. This approach can be generalized with every additive function (Golfarelli *et al.*, 1998).

The operator would be applied using other aggregation functions such as average, maximum. However, note that the average is an algebraic function (Gray *et al.*, 1996; Lenz and Thalheim, 2001); i.e. the implementation of the operator is more difficult because only part of the results may be pre-calculated using views, the rest must be calculated from detailed data. For example, the surface of continent is calculated by summing surfaces of countries whereas the temperature of continent cannot be calculated by averaging temperatures of countries. The temperature of a continent is calculated by averaging temperatures of the most detailed data.

6 Experiments within R-OLAP Context

The "BLEND" operation is implemented within the Graphic-OLAP tool (Ravat *et al.*, 2008) we have developed in our laboratory using the Java language and the Oracle DBMS. This prototype is implemented according to a R-OLAP approach: the architecture is based on a relational storage of the data and metadata while presenting various interfaces to the user.

The constellation of facts and dimensions is implemented through tables: a set of meta-tables describes the multidimensional structure and a set of tables stores the decisional data available for the analysis. To simplify, our presentation is limited to the tables that store the detailed data; we do not approach the problems of optimization by materialized views (Zhuge *et al.*, 1998; Kotidis and Roussopoulos, 1999). Within this simplified framework, the queries specified by the user are translated into an extraction SQL query on the tables storing the decisional data. Note that the database's structure complexity increases the metadata size. We do not take into account the quantity of meta-data because it does not impact the query process compared to the detailed data.

Example 5. The star schema (see Fig. 1) is stored in R-OLAP as a set of relations:
DATES(**id_dates**, month, monthdesc, quarter, year, fouryear)
ORGANISMS(**id_organisms**, variety, category, organismtype)
GEOGRAPHY(**id_geography**, parcel, state, region, country, density, continent)
DISTRIBUTION(**id_repartition**, **id_dates#**, **id_organisms#**, **id_geography#**, surface)

Let us reconsider the "BLEND" operations illustrated in Figs. 1, 2 and 3. The MT T_3 of Fig. 5 is obtained from the result of extraction queries generated by Graphic-OLAP. Table 2 shows the SQL queries generated for each operation.

6.1 Experiments with Standard Relational SQL

The experiments we made aim at estimating the operator costs. We study the cost of "BLEND" by translating this algebraic operator into its equivalent SQL query over the star schema. Two queries are compared:

- The first query (R1) uses an attribute that stores the multigranular transformation. This query simulates MDB, which would be modeled according to the user multigranular transformation needs.

Table 2 SQL translation of “BLEND”

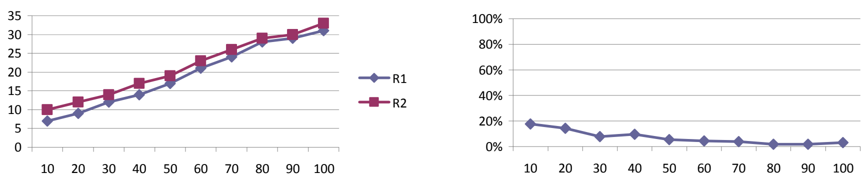
BLEND(T_2, Geography, Country(−), State(−), Country <> ‘USA’) = T_i <u>SQL translation</u>: **SELECT SUM**(surface) **AS** superficy, continent, country_state, variety **FROM** (**SELECT** surface, continent, pays **AS** country_state, variety **FROM** T_2 **WHERE** country<>’USA’ **UNION ALL** **SELECT** surface, continent, state **AS** country_state, variety **FROM** T_2 **WHERE NOT** (country<>’USA’)) **GROUP BY** continent, country_state, variety;	BLEND(T_i, Geography, Continent(−), Country-State(−), Continent = ‘Asia’) = T_3 <u>SQL translation</u>: **SELECT SUM**(surface) **AS** surface, continent_country_state, variety **FROM** (**SELECT** surface, continent **AS** continent_country_state, variety **FROM** T_i **WHERE** continent=’Asia’ **UNION ALL** **SELECT** surface, country_state **AS** continent_country_state, variety **FROM** T_i **WHERE NOT** (continent=’Asia’)) **GROUP BY** continent_country_state, variety;

- The second query (R2) calculates the multigranular transformation from the star schema.

Tuples were generated into the ROLAP database’s relations according to the following:

- |ORGANISM|= 250
- $10 \leq$ |GEOGRAPHY|≤ 100
- |DISTRIBUTION|= |ORGANISM|$\times$ |GEOGRAPHY|

Each relation was completed by multiple indexes on foreign keys. Values were generated using a random function but we make sure that the sizes of generated sets noted E_{sup} and E_{inf} are similar.



(a) (b)

Fig. 6 Experiment results of BLEND costs

The costs are calculated from the system cost (cost provided by the explain plan of Oracle 11g Application Server). The experiments aim at showing how much the operator “BLEND” costs. The size represents the number of tuples in GEOGRAPHY (from 10 to 100) and DISTRIBUTION (from 250×10 to 250×100); the size of ORGANISM is fixed to 250 tuples. Figure 6(a) compares the queries. Naturally (R2) is more expensive than (R1) due to computation of multigranular transformation. The cost is not very important (between 18% and 2%). As Fig. 6(b) shows, this result is interesting because the cost falls according to the relation sizes are increased.

We also investigated if results remain similar when sizes of E_{sup} and E_{inf} are different. We use |GEOGRAPHY|= 200 and |DISTRIBUTION|= 50000. Figure 7 shows costs of (R2) when sizes of E_{inf} and E_{sup} are modified: the axis x represents GEOGRAPHY size whereas |DISTRIBUTION|= |ORGANISM|$\times$ |GEOGRAPHY|. We can see that cost is constant, and the size difference between E_{inf} and E_{sup} seems not to influence the BLEND operator cost.

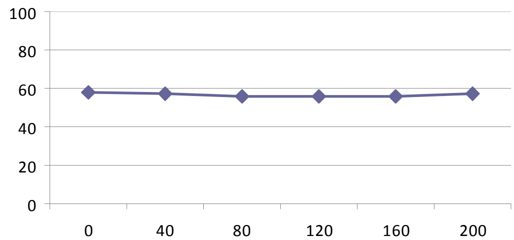


Fig. 7 R2 cost according to the data distribution between E_{sup} and E_{inf}

6.2 *Experiments with Oracle SQL3/OLAP*

We performed second experiment series in Oracle SQL3/OLAP using the GROUP BY CUBE operator. We compared (R2) with its equivalent query (R3) using the cube operator (Gray *et al.*, 1996). Figure 8 compares queries (R2) and (R3). We can note that the Oracle GROUP BY CUBE implementation is faster than the standard GROUP BY operation.

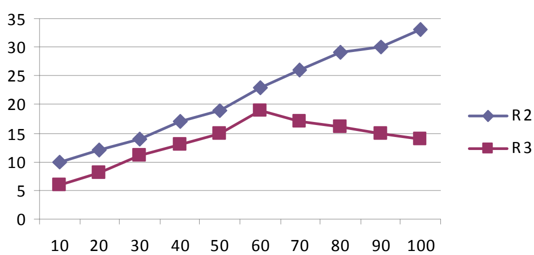


Fig. 8 Comparison with cube operator

7 Conclusion

This paper deals with complex analyses consisting in combining parameters of different granularities. Such analyses known as multigranular are not easily performed with traditional systems since they require organizing the data according to each analysis. We introduce a new algebraic operator for OLAP manipulations, called "BLEND". We study the limits of its use on strict hierarchies. The approach allows transforming a hierarchy by maintaining the initial possibilities of navigation. In order to establish the feasibility of this proposal, the operator has been implemented in a R-OLAP context within the Oracle DBMS.

In the short term, a first prospect is to carry out a study on the possible techniques of operator optimization, in particular by exploiting lattices of materialized views set up within the MDB. The expression of the operation in our graphic language (Ravat *et al.*, 2007) also constitutes a direct extension of this work. We also project to study other principles of multigranular transformations in more complex contexts such as non-strict hierarchies.

References

Abelló, A., Samos, J., Saltor, F.: YAM2: a multidimensional conceptual model extending UML. Inf. Syst. 31(6), 541–567 (2006), `http://dx.doi.org/10.1016/j.is.2004.12.002`

Blaschka, M., Sapia, C., Höfling, G.: On Schema Evolution in Multidimensional Databases. In: Mohania, M., Tjoa, A.M. (eds.) DaWaK 1999. LNCS, vol. 1676, pp. 153–164. Springer, Heidelberg (1999)

Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Rec. 26(1), 65–74 (1997), `http://doi.acm.org/10.1145/248603.248616`

Eder, J., Koncilia, C., Mitsche, D.: Automatic Detection of Structural Changes in Data Warehouses. In: Kambayashi, Y., Mohania, M.K., Wöß, W. (eds.) DaWaK 2003. LNCS, vol. 2737, pp. 119–128. Springer, Heidelberg (2003)

Espil, M.M., Vaisman, A.A.: Efficient intensional redefinition of aggregation hierarchies in multidimensional databases. In: DOLAP 2001: Proceedings of the 4th ACM international workshop on Data warehousing and OLAP, pp. 1–8. ACM, New York (2001), `http://doi.acm.org/10.1145/512236.512237`

Favre, C., Bentayeb, F., Boussad, O.: Dimension Hierarchy Updates in Data Warehouses: a User-driven Approach. In: 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira, Portugal, pp. 206–211 (2007)

Golfarelli, M., Maio, D., Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schema. In: HICSS 1998: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, vol. 7, pp. 334–343. IEEE Computer Society, Washington (1998), `http://dx.doi.org/10.1109/HICSS.1998.649228`

Golfarelli, M., Rizzi, S., Saltarelli, E.: WAND: A CASE Tool for Workload-Based Design of a Data Mart. In: SEBD, pp. 422–426 (2002)

Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. In: ICDE 1996: Proceedings of the Twelfth International Conference on Data Engineering, pp. 152–159. IEEE Computer Society, Washington (1996)

Gyssens, M., Lakshmanan, L.V.S.: A Foundation for Multi-dimensional Databases. In: VLDB 1997: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 106–115. Morgan Kaufmann Publishers Inc., San Francisco (1997)

Hurtado, C.A., Mendelzon, A.O., Vaisman, A.A.: Maintaining Data Cubes under Dimension Updates. In: International Conference on Data Engineering, vol. 0, pp. 346–355 (1999), `http://doi.ieeecomputersociety.org/10.1109/ICDE.1999.754950`

Kimball, R.: The data warehouse toolkit: practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc., New York (1996)

Kotidis, Y., Roussopoulos, N.: DynaMat: a dynamic view management system for data warehouses. In: SIGMOD 1999: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pp. 371–382. ACM, New York (1999), `http://doi.acm.org/10.1145/304182.304215`

Lenz, H.-J., Thalheim, B.: OLAP Databases and Aggregation Functions. In: SSDBM 2001: Proceedings of the 13th International Conference on Scientific and Statistical Database Management, pp. 91–100. IEEE Computer Society, Washington (2001)

Malinowski, E., Zimányi, E.: Hierarchies in a multidimensional model: from conceptual modeling to logical representation. Data Knowl. Eng. 59(2), 348–377 (2006), `http://dx.doi.org/10.1016/j.datak.2005.08.003`

Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Graphical Querying of Multidimensional Databases. In: Ioannidis, Y.E., Novikov, B., Rachev, B. (eds.) ADBIS 2007. LNCS, vol. 4690, pp. 298–313. Springer, Heidelberg (2007)

Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Algebraic and Graphic Languages for OLAP Manipulations. IJDWM 4(1), 17–46 (2008)

Romero, O., Abelló, A.: On the Need of a Reference Algebra for OLAP. In: Song, I.Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 99–110. Springer, Heidelberg (2007)

Vassiliadis, P., Sellis, T.: A survey of logical models for OLAP databases. SIGMOD Record 28(4), 64–69 (1999), `http://doi.acm.org/10.1145/344816.344869`

Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Modeling ETL activities as graphs. In: Lakshmanan, L.V.S. (ed.) DMDW. CEUR Workshop Proceedings, vol. 58, pp. 52–61 (2002) CEUR-WS.org

Zhuge, Y., Garcia-Molina, H., Wiener, J.L.: Consistency Algorithms for Multi-Source Warehouse View Maintenance. Distrib. Parallel Databases 6(1), 7–40 (1998), `http://dx.doi.org/10.1023/A:1008698814840`