Wolfgang Lehner

Modeling large scale OLAP scenarios

Erstveröffentlichung in / First published in:

Advances in Database Technology (EDBT '98). Valencia 23.03 - 27.03.1998. Springer Nature, S. 151–167. ISBN 978-3-540-69709-1

DOI: https://doi.org/10.1007/BFb0100983

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

Modelling in OLAP

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

Modeling Large Scale OLAP Scenarios

Wolfgang Lehner

University of Erlangen-Nuremberg, Dept. of Database Systems, Martensstr. 3
D-91058 Erlangen, Germany

Abstract. In the recent past, different multidimensional data models were introduced to model OLAP ('Online Analytical Processing') scenarios. Design problems arise, when the modeled OLAP scenarios become very large and the dimensionality increases, which greatly decreases the support for an efficient ad-hoc data analysis process. Therefore, we extend the classical multidimensional model by grouping functionally dependent attributes within single dimensions, yielding in real orthogonal dimensions, which are easy to create and to maintain on schema design level. During the multidimensional data analysis phase, this technique yields in nested data cubes reflecting an intuitive two-step navigation process: classification-oriented 'drill-down'/ 'roll-up' and description-oriented 'split'/ 'merge' operators on data cubes. Thus, the proposed NESTED MULTIDIMENSIONAL DATA MODEL provides great modeling flexibility during the schema design phase and application-oriented restrictiveness during the data analysis phase.

1 Introduction

In the last few years, "Online Analytical Processing" (OLAP, [5]) and the corresponding multidimensional data model has become a major research area in the database community ([1], [10], [7], [9]). The promising turnover estimates for OLAP applications results in the appearance of different commercial products as well ([2], [8], [13], [14], etc.). One consequence of the OLAP-fever is the rejuvenation of the multidimensional data model. Moreover, the multidimensional view of data seems natural and appropriate for a wide range of scientific applications ([20]) such as market analysis, population analysis, geographic information analysis or cost accounting but needs a conceptual extension to model complex application scenarios appropriately.

To motivate our modeling approach and to show that the proposed approach ("THE NESTED MULTIDIMENSIONAL DATA MODEL") is 'not yet another data model' but provides necessary extensions in different directions, we refer to an example which stems from a cooperation with an industrial partner, a large european market retail research company. In their business, facts like sales or stock values of single articles in single shops at a specific period of time are monitored and collected to form the raw database ("micro data"). In a second phase, the raw database is analyzed in two ways: On the one hand, the data is aggregated along a predefined classification hierarchy. As pointed out later in more detail, one may imagine a classification hierarchy as a tree of 'high-level' business terms, identifying classes of single basic items ("Video" identifies all video equipment articles). On the other hand, the data is split into characteristic features of the single articles or shops. For example, each shop holds descriptive information about its purchase class or its shop type ('Cash&Carry', 'Retail', 'Hypermarket'). In the product dimension, each article of the 250.000 monitored products belongs to one of the 400

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

product families. Furthermore, each article is characterized by five attributes valid for all products (brand, package type, ...) and about 15 attributes which are valid only in the product family or product group to which the article belongs to (video system only for the product group "Video", water usage only for the product family "Washers"). Thus, a typical query may look like:

"Give me the total sales values for all product families subsumed by the product group 'Video' sold in shops in 'Germany' divided into different regions and split these values in such a way that the market shares of different brands are compared to the shop type where the corresponding articles have been sold",

which could be specified in SQL like:

```
select sum(SALES)
from ...
where Product_Group = 'Video',
      Shop_Country = 'Germany'
group by Product_Family, Product_Brand,
         Shop_Region, Shop_Type
```

SALES			Product_Group = 'Video'			
			CAMC		VCR	
			Sony	JVC	JVC	Grundig
Shop_Country = 'Germany'	North	C&C	12	11	37	58
		Retail	31	35	32	66
	South	HyperM	22	18	32	67
		Retail	51	46	54	57

According to the number of group-by attributes, this sample query would intuitively produce a 4-dimensional data cube. As we will show in this paper, this query corresponds only to a two-dimensional, but nested data cube (look at the prefixes of the attribute identifiers), where the additional characterizations like 'Brand' and 'ShopType' are nested into a classification based low-dimensional data cube stretched by product families and different geographic regions[1].

Structure of the paper

The paper is structured as follows: In the next section we will show that either the classical as well as all proposed extensions to the multidimensional model will fail in modeling the application scenario. Section 3 details the notion of a complex dimensional structure, whereas section 4 introduces the notion of 'Multidimensional Objects' which arises, when dimensional structures are brought into a multidimensional analysis context. Section 5 defines formally as well as guided by the ongoing market research example, the different operators on multidimensional objects. The paper concludes with a summary and a conclusion.

2 Related Work

As illustrated in figure 1, the general idea of the multidimensional data model is that each dimension of a multidimensional data cube, e.g. Products, Shops, or Time, can be seen as part of the primary key, spanning the cartesian product with the elements of the dimensions. Consequently, any combination of the composite primary key identifies exactly a single cell within the cube. In the classical multidimensional model as implemented in different OLAP products ([2], [8], [13], [14], etc.), classification hierarchies can be

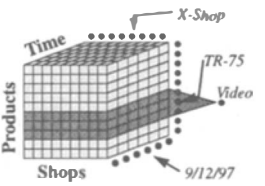


Fig. 1. General Idea of the multidim. Model

1. As pointed out in [6], a statistical table might be used to visualize a multi-dimensional data cube. Due to problems in drawing such cubes, we will follow this advice throughout the paper.

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

defined on the dimensional elements to identify business terms like 'Video' in figure 2. According to the classes within the hierarchies, operators are defined to slice/dice the data cube, i.e. selecting a subcube addressed by high-level terms.

To reflect dimensional attributes, i.e. features or properties describing single dimensional elements in the pure multidimensional model, each dimensional attribute must be modeled as an own dimension of the data cube. Referring again to the market research example, with 250.000 products classified in 400 product families and each family with about 15 features, the obviously three dimensional problem (Products, Shops, Time) explodes to a problem with 400*15=6000 dimensions to represent only the description of the articles (figure 2).

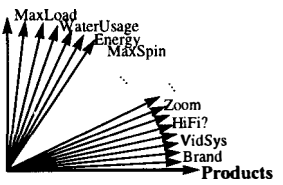


Fig. 2. Naive modeling approach

The consequence of this approach is that from a conceptual point of view, (n-1) dimensions functionally depend on the true dimension, holding the dimensional elements. This means that more than one "dimension" (Brand, VidSys, ...) refers to a single basic object, i.e. a single article in the ongoing example. From an implementation point of view, this approach leads to a high dimensionality and an extremely sparse data cube.

Neither an extended multidimensional model in the modern OLAP community nor the stream of statistical and scientific databases ([12], [19]) has addressed the problem of representing dimensional attributes (or features, properties, etc.) appropriately. Proposals on *multidimensional models* were made to transform cells to dimensions and vice versa ([1]), add complex statistical functions ([10]), or define a sophisticated mapping to the relational model ([7]). Also the historic stream of the *graphically oriented data models* like SUBJECT ([4]), GRASS ([16]), STORM ([17]), STORM+ ([3]) basically only knows classification and cross-product nodes to represent the corresponding data schema. Although these approaches enable the same grouping techniques, the basic problem of a single and therefore high dimensional data cube still remains.

The following consequences can be extracted from the discussion of related work: Firstly, current multidimensional data models are not capable to model dimensions of complex structure, which in reality reflect the user's world. Secondly, beyond classification-oriented analysis ("vertical analysis", drill-down operations), the characteristics or feature descriptions offer a new way of feature-oriented analysis ("horizontal analysis", feature splits). Thirdly, a carefully performed schema design process helps to define a reasonable space for the execution of queries as a basis for efficient query optimization techniques.

3 Dimensional Structures

In the context of multidimensional data models, the notion of a 'dimension' has a large number of different interpretations. Sometimes, a dimension reflects an edge of a multidimensional data cube without further structuring. In most cases, a dimension consists of multiple hierarchically structured classifications based on a set (or list) of basic values. In our approach, a dimension is a very complex entity enabling the proper struc-

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

turing of the world of interest. Furthermore, we propose that dimensions are orthogonal to each other, meaning that no dimension depends on the existence of other dimensions leading to a clean schema design.

3.1 Primary Attribute and Dimensional Elements

Dimensional elements (DE) (or basic objects) are the basic units of a dimensional structure. They are used to address the micro data, i.e. measures or facts. Since in the ongoing example raw data is collected on a single article basis, these single article identifiers reflect the dimensional elements within the product dimension. Furthermore, dimensional elements are instances of the primary attribute (PA) of a dimension. As illustrated in figure 3, for example 'TR-75' is a dimensional element for the primary attribute 'ArticleID'.

3.2 Classification Attributes

Based on the dimensional elements, a balanced tree-structured *classification hierarchy* ([21]) can be defined to identify business terms like product families, groups, and areas in the product dimension (figure 3a) or cities, regions, and countries in a geographical dimension (figure 4). Each *classification node* (C), e.g. 'Video', is an instance of a corresponding *classification attribute* (CA). Thus, on the attribute level, the hierarchy of classification nodes corresponds to a list of classification attributes (CA_i, $i=1,...,\delta$) denoted as *categorization* of that dimension. The root node of the classification hierarchy is a specific 'ALL'-node, covering all dimensional elements. For consistency reasons, this root node is the single instance of the highest classification attribute TOP (CA_δ), which is always member of each categorization.

3.3 Dimensional Attributes

As pointed out earlier, *dimensional attributes* (DA) reflect features or properties of dimensional elements. The important characteristic is that the existence of properties depends on the nodes of the classification hierarchy. For example, only video equipment has a property describing the video system, implying that the property 'VidSys' depends on the classification node 'Video' (figure 3b). Naturally, properties may also be valid for all dimensional elements thus depending on root node 'ALL' of that dimension. During the schema design phase, it may often not be clear which attributes may be used as *classification attributes* and which attributes may be used for further character-

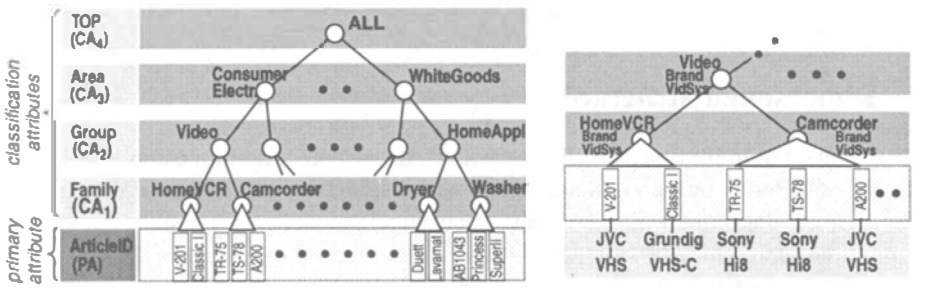


Fig. 3. Classification hierarchy and dimensional attributes of the product dimension

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

izing, i.e. as *dimensional attributes* ([18]). A candidate classification attribute must at least be valid for all dimensional elements and must functionally determine other classification attributes. For example, each article belongs to a product family and the family attribute determines the product group and the product area attributes. If more than one attributes are classification candidates, either the underlying implementation supports multiple classifications or the *most natural* one ([11]) must be selected as the classification attribute.

3.4 Node Domains

For a specific classification node, the distinction of classification attributes and dimensional attributes allows to define node domains with regard to the different direction.

Definition: A *classification-oriented node domain* $DOM(C_{|CA})$ holds all classification or basic objects subsumed by the current node C according to a given classification attribute (CA). If the classification attribute (CA) is equal to the classification attribute of the node C, then $DOM(C_{|CA}) := (C)$.

Definition: A *feature-oriented node domain* $DOM(C_{|DA})$ holds all instances of the subsumed dimensional elements for a specific dimensional attribute (DA).

Definition: A node domain without an attribute specification is called the *Null-domain* of that node ($DOM(C_{|})=( )$).

Following the example of figure 3b, the node domain of "Video" according to the product family classification attribute results in

$$DOM(Video_{|Family}) = \begin{pmatrix} Camcorder \\ HomeVCR \end{pmatrix}.$$

Otherwise, the node domain according to the feature 'Brand' results in

$$DOM(Video_{|Brand}) = \begin{pmatrix} Sony \\ JVC \\ Grundig \end{pmatrix}.$$

As final examples, the brand domains of camcorders and HomeVCRs are:

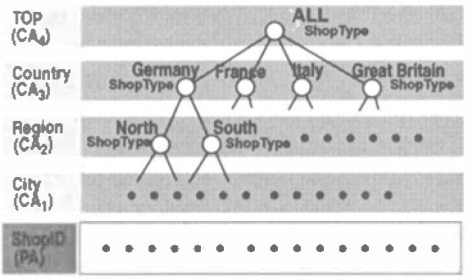
$$DOM(Camcorder_{|Brand}) = \begin{pmatrix} Sony \\ JVC \end{pmatrix} \quad DOM(HomeVCR_{|Brand}) = \begin{pmatrix} JVC \\ Grundig \end{pmatrix}$$

To exemplify the following definitions in the multidimensional context, we extend our ongoing market research example by a second *Shops* dimension. Therefore, we assume the categorization *(Country, Region, City, ShopID)*. At the instance level, we pick the country Germany, which is divided into northern and southern regions. From a possible large list of dimensional attributes (ShopType, PurchaseClass, BranchStore, ...), we pick the feature 'ShopType' with the node domains shown in figure 4).

4 Nested Multidimensional Data Cubes

While the dimensional structures model the business terms of the users' world in a very complex and powerful way, the multidimensional context uses these information for gaining analyses access to the measures or facts. This section starts with the formal definition of "Primary and Secondary Multidimensional Objects" reflecting the multidimensional view of classification and dimensional attributes. Using these mechanisms, this section focuses on the introduction of "Multidimensional Objects" (MOs), representing a consistent and intuitive view to nested multidimensional data cubes.

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983



classification-oriented

$$DOM(Germany_{|Region}) = \begin{pmatrix} North \\ South \end{pmatrix}$$

description-oriented

$$DOM(Germany_{|ShopType}) = \begin{pmatrix} C\&C \\ Retail \\ HyperM. \end{pmatrix}$$

$$DOM(North_{|ShopType}) = \begin{pmatrix} C\&C \\ Retail \end{pmatrix}$$

$$DOM(South_{|ShopType}) = \begin{pmatrix} Retail \\ HyperM. \end{pmatrix}$$

Fig. 4. Sample node domains of the shop dimension

4.1 Granularity and Range Specification

Definition: A *context descriptor schema* (DS) is an n-tuple $(A_1, ..., A_n)$ where each element A_i is either a primary attribute (PA) or a categorization attribute (CA)[2].

Definition: A *context descriptor* (D) is an n-tuple $(c_1, ..., c_n)$ where each c_i is a node of the classification level described by A_i of the corresponding context descriptor schema.

In the two-dimensional context stretched by the product and shops dimension, *('Video', 'Germany')* is a valid context descriptor for the context descriptor schema *(Product.Group, Shops.Country)*. With an explicit schema descriptor, the context descriptor is also written as *(Product.Group='Video', Shops.Country='Germany')*.

4.2 Primary Multidimensional Objects

Definition: A *primary multidimensional object* (PMO) is a quintuple (M, DS, D, t_A, t_D) consisting of an unique cell identifier M, a context descriptor schema DS denoting the granularity of the cell, a context descriptor D specifying the selection criteria, an aggregation type $t_A \in \{\Sigma, \phi, c\}$, and a data type $t_D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{R}\}$.

As introduced in [15], the aggregation type describes the aggregation operators which are applicable to the modeled data (Σ: data can be summarized, ϕ: data may be used for average calculations, c: constant data implies no application of aggregation operators). The cardinality of the context descriptor D reflects the dimensionality of the corresponding data cube. Furthermore, the context descriptor D may be compared to the 'where-clause' of a SQL select-statement, thus specifying the partition (or sub-cube) size. The context descriptor schema DS of a PMO may be seen as the list of 'group by' attributes, resulting in a specification of the data granularity.

Examples

If all attributes of the context descriptor schema DS of a PMO are primary attributes from different dimensions, the PMO reflects *micro data*. If at least one attribute of DS is a categorization attribute, the PMO describes *macro data*. For the ongoing example, the PMO

$$P_1 = (SALES, (P.ArticleID, S.ShopID), (P.Group = 'Video', S.Country = 'Germany'), \Sigma, \mathbb{N})$$

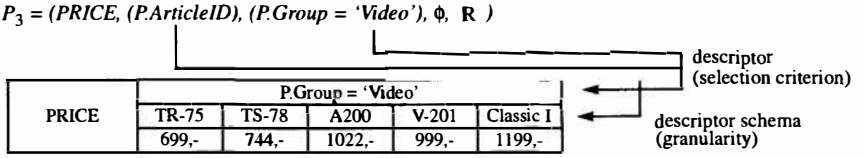
2. In the multidimensional context, each primary or classification attribute is prefixed with the dimension identifier, e.g. "Product.Family" or "Shops.Region".

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

describes a summarizable two-dimensional micro data object for accessing video sales figures of single articles in single shops for all of Germany. Substituting the context descriptor schema of PMO P_1 by the context descriptor schema *(P.Family, S.Region)* leads to the PMO

P_2 = *(SALES, (P.Family, S.Region), (P.Group = 'Video', S.Country = 'Germany'),* Σ, ℕ)

for the description of macro data holding aggregation values of P_1. As a final example, the following PMO P_3 holds one-dimensional price figures for video equipment:



Domain of a PMO

Definition: The *domain of a PMO* is defined by the cartesian product of the classification-oriented node domains of each context descriptor element:
$DOM(P) = \bigotimes_{i=1}^{n} DOM(c_i|_{CA_i})$, where c_i is the i-th component of the context descriptor D and CA_i is the i-th component of the context descriptor schema DS of the PMO.

The following figure 5 details the definition of the domain for the PMO P_2. The classification node 'Video' has 'Camcorder' and 'HomeVCR' as children at the family level. On the geographical dimension, 'Germany' is divided into the regions 'North' and 'South'.

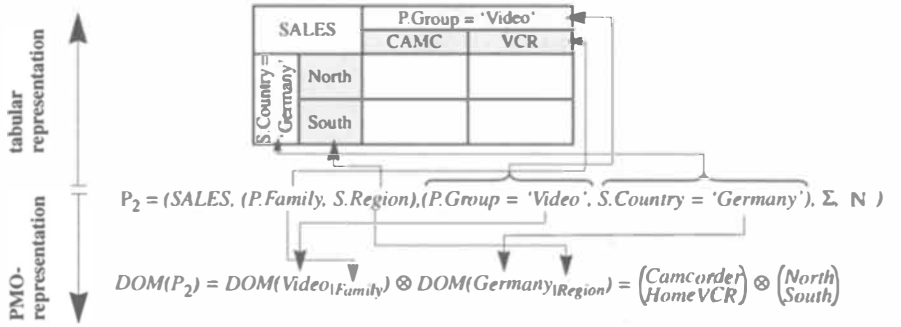


Fig. 5. Domain of the sample PMO P_2

Definition: A *constant primary multidimensional object* is a PMO where the context descriptor schema DS as well as the context descriptor are empty, the aggregation type is 'c', and the data type is $t_D \in \{ \mathbb{N}, \mathbb{Z}, \mathbb{R} \}$.

For example, the PMO (TAX, (), (), c, ℝ) may hold the constant factor for the applicable tax which is invariant according to all dimensions.

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

4.3 Secondary Multidimensional Objects

Definition: A *secondary multidimensional object* (SMO) is a tuple (D, DA), where D is a context descriptor, and DA is a set of dimensional attributes applicable to the context descriptor D = (c_1, ..., c_n). The set of dimensional attributes results in $DA \subseteq \bigcup_{i=1}^{n} DA_i$, where DA_i is the set of dimensional attributes of the classification node c_i.

Definition: The *domain of an SMO* is the cartesian product of the feature-oriented domains according to the classification nodes specified in the context descriptor D of the SMO.

In analogy to the domain of a PMO, the following figure 6 illustrates the domain of an SMO S_1 with regard to the element *('Camcorder', 'North')* from the domain of PMO P_2 and the feature schema *{Brand, ShopType}*.

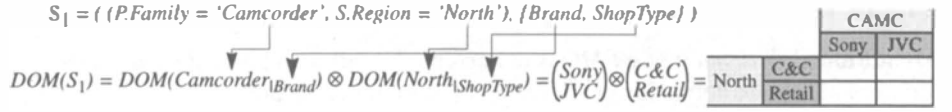


Fig. 6. Domain of the sample SMO S_1

At this point, it is worth to note that in a pure classification-oriented multidimensional environment, the component DA of an SMO would simply result in an empty set of dimensional attributes, resulting in a seamless extension to the classical multidimensional model. Thus, the domain of the SMO

S_2 = ((P.Family = 'Camcorder', S.Region = 'North'), { })

results in

$DOM(S_2) = DOM(Camcorder_{|}) \otimes DOM(North_{|}) = (\ )$.

	CAMC
North	

Furthermore, it is important to realize that the feature schema depends on the context descriptor of the PMO. The instances however depend on the elements of the domain of the PMO! Therefore, the domain of the SMO

S_1' = ((P.Family = 'VCR', S.Region = 'North'), {Brand, ShopType})

would result in

$DOM(S_1') = DOM(VCR_{|Brand}) \otimes DOM(North_{|ShopType}) = \binom{JVC}{Grundig} \otimes \binom{C\&C}{Retail}$,

which is different from DOM(S_1).

4.4 Multidimensional Objects

Definition: A *multidimensional object* (MO) is a tuple (P, DA), where P is a valid PMO and DA is a set of dimensional attributes for defining the corresponding nested SMOs.

The following figure 7 illustrates two multidimensional objects MO_1 and MO_2. Both MOs have the same primary multidimensional object (P_2) but different feature split schema (left: none; right: by Brand and ShopType). Each element of the PMO points to an SMO (left: 0-dimensional, right: 2-dimensional). Each element of an SMO points to

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

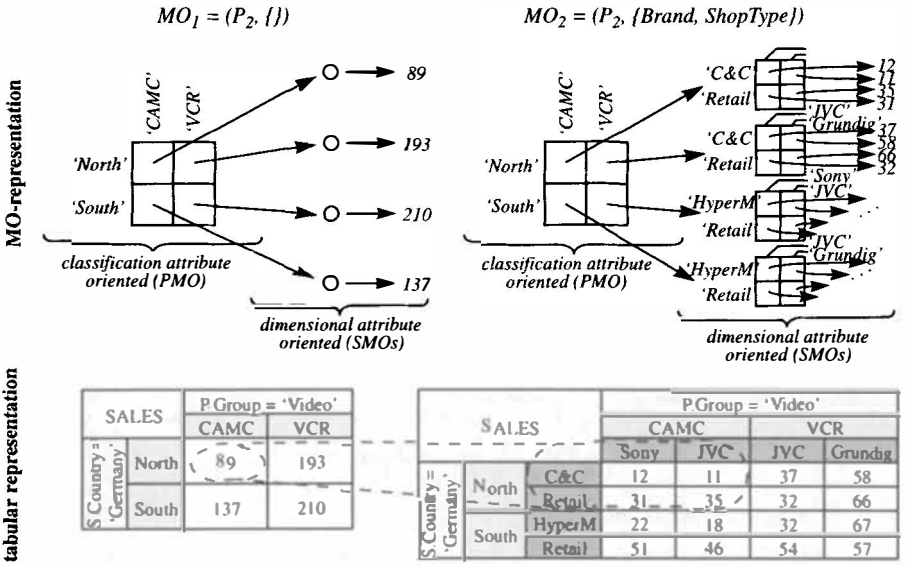


Fig. 7. Sample MOs representing context-sensitive nested data cubes

the real sales figure for the current context. The lower half of figure 7 shows the tabular representation of both MOs (MO_1 and MO_2). As depicted in this representation, each cell of MO_1 (0-dimensional SMO) is expanded by an inner 2-dimensional SMO in MO_2 (dark shaded).

Explicit modeling of dimensional attributes in dimensional structure, as discussed in section 3 results in nested multidimensional data cubes during the data analysis phase. The next section will explain the different operators to work with such data cubes, specified formally in multidimensional objects.

5 Operators on Multidimensional Objects

In this section, the 'traditional' operators like slicing, aggregation operations and cell-oriented joins of two data cubes are defined on the basis of PMOs. The extension of nested SMOs implies two new operators ("split" and "merge") which dramatically improve the power and flexibility of the whole analysis process.

5.1 Sub-Cube Selection according to Classification Attributes (Slicing)

The result of a *slicing operation* is a MO which inherits all components of the source MO except the context descriptor. The new context descriptor D, given as parameter to the operator is "smaller", i.e. more restrictive than the one from the source MO.

$$MO' := \sigma(D)MO$$

The slice operator has no effect on the source MO when the schema of the new context descriptor is finer than the context descriptor schema (data granularity) of the source MO.

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

Example

To restrict the multidimensional object MO_2 (figure 8) to camcorder sales figures in the northern part of Germany, the slicing operation would be expressed as:

$$MO_2' := \sigma(P.Family = 'Camcorder', S.Region = 'North')MO_2$$

Figure 8 below illustrates this slicing process in the tabular representation.

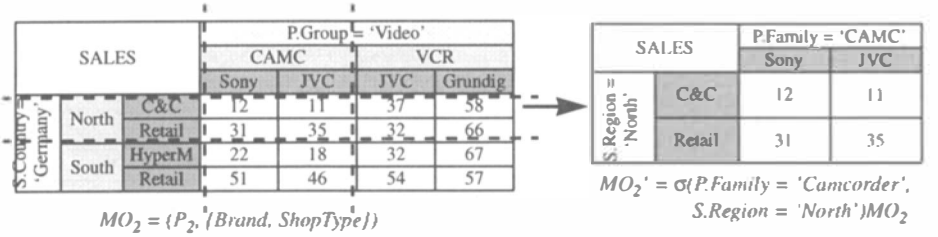


Fig. 8. Example for the slicing operator

5.2 Navigation Operators

Navigation operators are used by the user to explore interactively the multidimensional analysis context. As pointed out in section 2, the classical multidimensional model only supports *classification-oriented navigation* ('vertical analysis') proposing 'drill-down' and 'roll-up' operations. In our data and user interactivity model, these operations are used in a first analysis phase to find an interesting context. These operations are based on the PMO-part of multidimensional objects. The explicit modeling of dimensional attributes yields in an additional analysis phase. Based on the SMO-part of multidimensional objects the selected analysis context can be further and detailed investigated by a dimensional attribute-oriented navigation process ('horizontal analysis'), which is enabled by the new operators 'split' and 'merge'.

PMO-oriented Navigation along the Classification Hierarchy

- The *drill-down* operator ("give details") corresponds to an implicit de-aggregation process according to the aggregation type t_A of the source MO,
  MO' := ↓(DS)MO
  where at least one attribute of the new descriptor schema DS (data granularity) is 'finer' than the attributes of the descriptor schema of the source MO. The 'drill-down' operator has no effect, if all attributes of the descriptor schema of the source MO correspond to primary attributes in their dimensions.
- The *roll-up* operator ("hide details") corresponds to an implicit aggregation process according to t_A of MO,
  MO' := ↑(DS)MO
  where at least one attribute of the new descriptor schema (data granularity) is 'coarser' than the attributes of the descriptor schema of the source MO. Furthermore, the roll-up operator may result in an implicit 'un-slice' operator, because the selection of a parent node within the classification hierarchy may release an earlier

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

performed selection criterion during in the analysis process. If all attributes of the descriptor schema of the source MO correspond to the TOP-attribute, the 'roll-up'-operator has no effect.

Examples

The tabular representation in figure 9 below show a sequence of two drill-down operations (from left to the right) or, from left to the right, the inverse roll-up operations.

- $MO_1 = \downarrow(P.Family, S.Region)MO'_1$
  $MO'_1 = \downarrow(P.ArticleID, S.Region)MO_1$
- $MO_1 = \uparrow(P.Family, S.Region)MO''_1$
  $MO'_1 = \uparrow(P.Group, S.Region)MO_1$

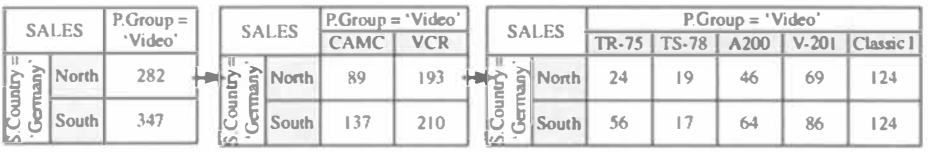


Fig. 9. Example of the drill-down operator

SMO-oriented Navigation along dimensional attributes

- The *split operator* adds a valid feature (DA_i) to the set of dimensional attributes, thus incrementing the dimensionality of the nested SMOs.
  $MO' := \rightarrow(DA_i)MO$ where $DA(MO') = DA(MO) \cup \{DA_i\}$.
- The *merge operator* removes a specific dimensional attribute (DA_i) from the MO's dimensional attribute set, thus decrementing the dimensionality of the nested SMO.
  $MO' := \leftarrow(DA_i)MO$ where $DA(MO') = DA(MO) \setminus \{DA_i\}$.

Examples

Once MO_1 is selected by use of PMO-oriented operations, it serves as the basis for SMO-oriented analysis. As shown in the figure 10 below, the first split operation is made according to the dimensional attribute 'ShopType' of the geographical dimension. Further split operations to increase or merge operations to undo former split operations are also seen in figure 10. The necessary transformations from MO_1 to MO_2 (figure 7) and vice versa are specified as follows:

- $MO_2 = \rightarrow(Brand)(\rightarrow(ShopType)MO_1)$ // expand the nested SMOs
- $MO_1 = \leftarrow(ShopType)(\leftarrow(Brand)MO_2)$ // collapse the nested SMOs

5.3 Implicit Aggregation

The use of navigation operations also implicitly performs an aggregation process corresponding to the default aggregation type (t_A) of the source MO. Consider two MOs (MO' and MO") with the same descriptor $D = (c_1,...,c_n)$ where MO' is coarser (either classification-oriented or dimensional attribute-oriented) than MO". Furthermore, let x

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

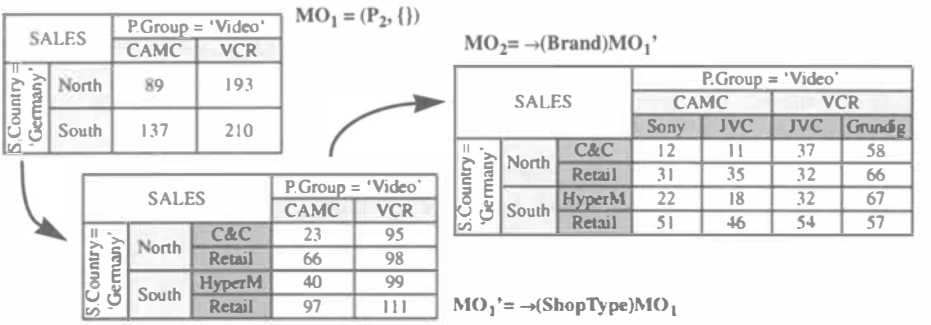


Fig. 10. Example for the split operator

denote a value in a single cell of the coarse MO" and let x_j (j=1,...,κ) denote the values of the finer MO' which are directly derived from x, then according to the different aggregation types t_A, the following properties hold:

- $t_A(MO) = \Sigma: \Rightarrow \quad x = \sum_{j=1}^{\kappa} x_j$
- $t_A(MO) = \phi: \Rightarrow \quad x = \frac{1}{\kappa}\left(\sum_{j=1}^{\kappa} x_j\right)$
- $t_A(MO) = c: \Rightarrow \quad x = x_1 = \ldots = x_\kappa$

In the case of a classification-based relationship, κ corresponds to the size of the domain, which is computed by the cartesian product of the node domains according to the finer descriptor schema $DS(MO) = (A_1,...,A_n)$.

$$\kappa = \prod_{i=1}^{n} \left| DOM(c_i|_{A_i}) \right|$$

In the case of a dimensional attribute-based relationship, the cartesian product of the node domains according to the set of dimensional attributes $DA(MO) = \{DA_1, ..., DA_m\}$ determines the number of the new cells, i.e. the size of the SMO of the finer MO.

$$\kappa = \prod_{k=1}^{m} \left| DOM(c_i|_{DA_k}) \right| \; (\forall c_i \in D)$$

Example

Referring to the definition of MO_1 and MO_2, the following figure 11 shows that summing up all partial sums generated by two successive split operations, is equal to the total sales figure of MO_1.

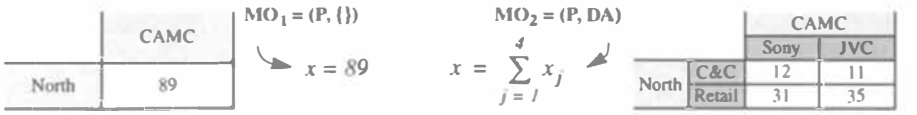


Fig. 11. Implicit aggregation when performing navigation operators

This implicit aggregation process holds for classification-oriented navigation operators as well.

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

5.4 Explicit Aggregation Operator

The result of an *aggregation operation* reflects the original MO with a new context descriptor schema DS, which is "coarser" than the original one (like 'roll-up'). The single numeric values are aggregated according to the explicitly specified operator.

$MO' := \theta(DS)MO$

where $\theta \in \{ SUM, AVG, MIN, MAX, COUNT \}$, if $t_A(MO) = \Sigma$.
where $\theta \in \{ AVG, MIN, MAX \}$, if $t_A(MO) = \phi$.
where $\theta = ID$, if $t_A(MO) = c$.

Example

To sum up all video equipment sales to the family level in the product dimension and upto different regions specified in the geographic dimension, i.e. generate MO_1 of the ongoing example, the following expression must be stated:

$MO_1 := SUM(P.Family, S.Region)\ (P_1, \{\})$

5.5 Cell-Oriented Operators

The objectives of a *cell-oriented operator* are the cells themselves, thus resulting in a MO with the same context descriptor D and the same context descriptor schema DS.

- unary operators: $MO' := \theta(MO)$ where $\theta \in \{ -, abs, sign \}$
- binary operators: $MO' := \theta(MO_1, MO_2)$ where $\theta \in \{ *, /, +, -, min, max \}$
  before performing any binary cell-oriented operator, the dimensionality as well as the context descriptor schema of each MO are aligned. Furthermore, if the MOs obtain different aggregation types, the resulting MO gets the strongest type according to the ordering ($c < \phi < \Sigma$).

Example

To calculate the purchase of each article, the MOs M'=(P1,{}) and M"=(P3,{}) are joined by the multiplication operator. Since M" is one-dimensional, it is expanded to the second geographical dimension of M'. Furthermore, the resulting MO M has aggregation type Σ because $t_A(M')=\Sigma$ is stronger than $t_A(M")=\phi$. Thus, $M := *(M', M")$, where

$M = ((PURCHASE, (P.ArticleID, S.ShopID), (P.Group = 'Video', S.Country = 'Germany'), \Sigma, \mathbb{N}), \{\})$.

As seen in this section, on the one hand the defined set of operators on multidimensional objects enables a simple navigation process through the multidimensional analysis context performing implicit aggregation operations. Furthermore, this navigation process is extended through split/merge operators hiding the nested cube model of the conceptual layer for the user, which, of course, is only faced with the tabular representation at the external layer. On the other hand, explicit aggregation and cell-oriented operators allow the specification of sophisticated statistical analysis queries against the multidimensional cube.

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

6 Summary and Conclusion

This paper motivates and formally describes an extended multidimensional data model. The extension is based on the fundamental distinction between classification attributes and dimensional attributes within a single dimension. This approach leads to functionally independent dimensions at the schema design level and to low-dimensional but nested data cubes during the analysis process. The described model is currently being implemented within the CubeStar-project at our department (http://www6.informatik.uni-erlangen.de/Research/cubestar/index.html) on top of a relational model. In opposite to the well-known Star-/Snowflake schema, the nested multidimensional model requires an alternative cube-to-relation mapping approach.

Although the proposed nested multidimensional data model allows the flexible modeling of statistical and scientific application scenarios, there still remain some open issues which must be solved: From a modeling point of view, versioning of dimensional structures must be supported by the data model. Since all multidimensional models handle the time dimension equally to other dimensions, the special characteristics of time must be considered and mechanisms like 'transaction/valid time' must be transferred from different temporal models. From an implementation point of view, redundancy-based query optimization, i.e. supporting of pre-aggregates reflects a major requirement for an efficient ad-hoc analysis process. Therefore, already existing work must be extended to the model of context-sensitive and nested data cubes. Nevertheless, we believe that our modeling approach is an adequate basis for solving all these problems.

Acknowledgements

I would like to thank Prof. Shoshani for the valuable discussions. Moreover, I acknowledge Prof. Wedekind, J. Albrecht, and H. Günzel for helpful comments on a draft of the paper.

References

1. Agrawal, R.; Gupta, A.; Sarawagi, S.: Modeling Multidimensional Databases, in: *13th International Conference on Data Engineering*, (ICDE'97, Birmingham, U.K., April 7-11), 1997, pp. 232-243
2. *Essbase Analysis Server - Bringing Dynamic Data Access and Analysis to Workgroups Across the Enterprise*, Product Information, Arbor Software Corporation, 1995
3. Bezenchek, A.; Massari, F.; Rafanelli, M.: "STORM+: statistical data storage and manipulation system", in: *11th Symposium on Computational Statistics*, (Compstat '94, Vienna, Austria, Aug. 22-26), 1994
4. Chan, P.; Shoshani, A.: SUBJECT: A Directory Driven System for Organizing and Accessing Large Statistical Data Bases, in: *7th International Conference on Very Large Data Bases* (VLDB'81, Cannes, France, Sept. 9-11), 1981, pp. 553-563
5. Codd, E.F.: Codd, S.B.; Salley, C.T.: *Providing OLAP (On-line Analytical Processing) to User Analysts: An IT Mandate*, White Paper, Arbor Software Corporation, 1993
6. Gray, J.; Bosworth A.; Layman A.; Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total, in: *12th IEEE International Conference on Data Engineering* (ICDE'96, New Orleans, Louisiana, Feb. 26 -Mar. 1), 1996, pp. 152-159

Final edited form was published in "Advances in Database Technology (EDBT '98). Valencia 1998", S. 151–167,
ISBN 978-3-540-69709-1
https://doi.org/10.1007/BFb0100983

7. Gyssens, M.; Lakshmanan, L.V.S.: A Foundation for Multi-Dimensional Databases, in: *23th International Conference on Very Large Data Bases* (VLDB'97, Athen, Greece, Aug. 25-29), 1997, pp. 106-115
8. *The INFORMIX-MetaCube Approach*, Product Information, Informix Software, Inc., 1996
9. Lehner, W.; Ruf, T.; Teschke, M.: CROSS-DB: A Feature-extended Multi-dimensional Data Model for Statistical and Scientific Databases, in: *5th International Conference on Information and Knowledge Management,* (CIKM'96, Rockville, Maryland, Nov. 12-16), 1996, pp. 253-260
10. Li, C; Wang, X.S.: A Data Model for Supporting On-Line Analytical Processing, in: *5th International Conference on Information and Knowledge Management,* (CIKM'96, Rockville, Maryland, Nov. 12-16), 1996, pp. 81-88
11. Lorenzen, P.; *Constructive Philosophy*, Amherst, Univ. of Massachusetts Press, 1987
12. Michalewicz, Z. (Ed.): *Statistical and Scientific Databases*, New York, Ellis Horwood, 1991
13. *The Case for Relational OLAP*, White Paper, MicroStrategy, Inc., 1995
14. *Personal Express Language Reference Manual, Volumes I / II,* Oracle Cooperation, 1996
15. Rafanelli, M.; Ricci, F.: Proposal of a Logical Model for Statistical Databases, in: *2nd International Workshop on Statistical Database Management*, Los Altos, CA, 1983
16. Rafanelli, M.; Ricci, F.: A Graphical Approach for Statistical Summaries: The GRASS Model, in: *ISMM International Symposium on Microcomputers and their Applications*, 1987
17. Rafanelli, M.; Shoshani, A.: STORM: A Statistical Object Representation Model, in: *5th International Conference on Statistical and Scientific Database Management* (5SSDBM, Charlotte, NC, April 3-5), 1990, pp. 14-29
18. Shoshani, A.; Lehner, W: *Are classifications and attributes orthogonal to each other?*, personal communication, 1997
19. Shoshani, A.: Statistical Databases: Characteristics, Problems, and Some Solutions, in: *8th International Conference on Very Large Data Bases* (VLDB´82, Mexico City, Mexico, Sept. 8-10), 1982, pp. 208-222
20. Shoshani, A.: OLAP and Statistical Databases: Similarities and Differences, in: *16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, (PODS'97, Tucson, Arizona, 13-15 May), 1997, pp. 185-196
21. Smith, J.M.; Smith, D.C.P.: Database Abstractions: Aggregation and Generalization, *ACM Transactions on Database Systems 2(1977)2*, pp. 105-133