MAC: Conceptual Data Modeling for OLAP

Aris Tsois
Knowledge and Database
Systems Laboratory, NTUA
atsois@dblab.ece.ntua.gr

Nikos Karayannidis
Knowledge and Database
Systems Laboratory, NTUA
nikos@dblab.ece.ntua.gr

Timos Sellis
Knowledge and Database Systems Laboratory
Department of Electrical and Computer Engineering
National Technical University of Athens (NTUA)
Zografou 15773, Athens, Greece
timos@dblab.ece.ntua.gr

Abstract

In this paper we address the issue of conceptual modeling of data used in multidimensional analysis. We view the problem from the end-user point of view and we describe a set of requirements for the conceptual modeling of real-world OLAP scenarios. Based on those requirements we then define a new conceptual model that intends to capture the static properties of the involved information. In its definition we use a minimal set of well-understood OLAP concepts like dimensions, levels, hierarchies, measures and cubes. The central concept of the model is the Multidimensional Aggregation Cube (MAC), which gives a broad and flexible definition to the notion of a multidimensional cube. We evaluate our model against other existing multidimensional models and show that MAC offers a unique combination of modeling skills. Our main contribution is the definition of the basic concepts of our model; although the set of requirements and the evaluation of all related models against those requirements represent an additional result.

1 Introduction

In the last years On-Line Analytical Processing (OLAP) [Codd93] has become a major research area in the database community [ChDa97]. The OLAP research is tightly coupled with the research in data warehouses, which are considered to be the information sources based on which On-Line Analytical Processing is performed. The typical data flow path involves the gathering of data from various sources into data warehouse systems and then the usage of those data in the multidimensional analysis process through the use of OLAP applications. Multidimensional analysis mainly involves the computation of aggregated information using a large volume of detailed data. The information is analysed based on its detailed or derived properties (dimensions) using an almost static business model (hierarchies). The reader is referred to [ChDa97] [Inmo96] [Kimb97] [Olap97] for an overview of Data Warehousing and OLAP. In the following we will assume the reader to be familiar with the terminology used in those areas.

A fundamental issue faced by vendors of OLAP applications as well as by researchers in the OLAP domain is the modeling of data. The well-studied conceptual and logical models used in other database areas, like the E/R model or the relational model, do not seam to be sufficient for the OLAP case ([Kimb96][TBC99][S++98][Kimb97]). Vendors have adopted various models, while standardization bodies and researchers have developed and studied additional models. All those models share some common concepts like measures or hierarchies but there is still no formally defined and widely accepted (logical or conceptual) data model. As proved by the history of the relational model a common data model is the key for the collaboration and the rapid progress in an area.

In this paper we address the problem of modeling real-world OLAP scenarios at the conceptual level. The current common practice is to use the well-known E/R model [BCN92] and then to annotate the schema with any additional OLAP specific information. Still, various authors argue that the E/R model is not appropriate for OLAP scenarios since concepts like dimensions, hierarchies and cubes can only be partially represented. As a result, two publications ([TBC99] and [S++98]) already proposed extensions to the E/R model for the multidimensional paradigm. Their approach is mainly suitable for the ODS (operational data store) part of data

Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001)
Interlaken, Switzerland, June 4, 2001
(D. Theodoratos, J. Hammer, M. Jeusfeld, M. Staudt, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-39/

warehouses as they concentrate on the representation of the source-detailed data.

In this paper we consider a slightly different approach, where the information used in multidimensional analysis is the primary target of our modeling concepts. The information used in such an analysis process is mainly aggregated data at various aggregation levels, or combination of such levels. Furthermore, the dimensions, the particular aggregation levels as well as the various hierarchies defined on dimensions represent information used during the analysis.

In order to define a useful conceptual model we first investigate a set of example queries and derive a list of modeling requirements. Based on those requirements we then define the concepts of our model and their semantics. The central concept of our model is the Multidimensional Aggregation Cube (MAC), which is equivalent to an n-way relationship relating measure values to a set of dimension values. A careful definition of dimension values allows a single MAC to represent measure values of arbitrary aggregation levels. This is an essential difference with respect to the other conceptual models and can be used to simplify the schema of the various OLAP scenarios. An additional novelty of our model is the explicit modeling of analysis paths, a feature quite important for OLAP applications.

Generally speaking, the concepts used in multidimensional analysis are mapped directly to corresponding concepts of the MAC model. As a result, the proposed MAC model allows OLAP scenarios to be modeled in a natural and straightforward way. Furthermore, its abilities to model complex dimensions and hierarchies and the broad definition of cubes makes it suitable for highly complicated OLAP applications.

The remainder of this paper is structured as follows: section 2 provides a set of OLAP specific modeling requirements defined through examples. Section 3 defines the basic concepts of the proposed MAC model as well as their semantics. Section 4 provides an overview of related work and describes the results of evaluating 12 multidimensional data models published in research papers. Finally, section 5 concludes the paper and presents our future work intensions.

2 Requirements Through An Example

In this section we will present a set of requirements that we believe to be of key importance for a conceptual model used in multidimensional analysis and OLAP applications. We will present those requirements through the use of an example scenario. The scenario is based on a real Data Warehousing / OLAP project in the development of which we have been involved.

Assume the following example: A chain of stores selling electrical home appliances has built a data warehouse in order to analyze its sales data. The sales data are loaded into the data warehouse from the OLTP system. For each sales transaction the OLTP system records the following information:

- The date of the transaction.
- The cashier ID where the transaction took place.
- The ID of the products being sold.
- The customer ID.
- The sales price for each product being sold.

We assume that all the above information is somehow stored in the data warehouse. We are not going to talk about the design process of the data warehouse, neither about how data is loaded from the OLTP system since our model does not address those issues. The MAC model, which we propose, is mainly suitable for the users of the data warehouse, the persons that analyze the information through the use of an OLAP application.

As described by a plethora of OLAP papers [Mendel], the multidimensional analysis is mainly based on drill-down, roll-up, slice and dice operations that are performed on a multidimensional view of data. Measures values are selected and aggregated using various predefined dimensions, dimension levels and hierarchies. The dimension levels, the aggregation paths defined by hierarchies, the dimensions and the measures are the main concepts used in the analysis. For our example scenario assume that the analysis is performed on the sales price (price of sold items) using the hierarchies defined in Figure 1.

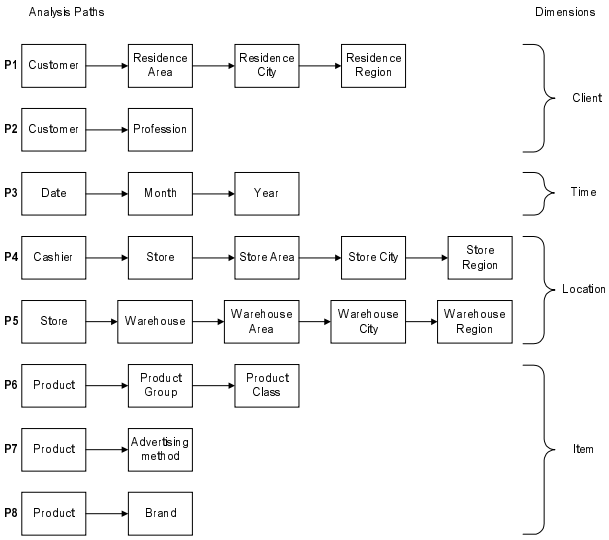

Figure 1: The analysis paths

The paths shown in Figure 1 are grouped into four distinct dimensions. The most detailed level of each dimension corresponds to a basic property of a product's sales price as recorded by the transactions of the OLTP system. For example the customer ID is used for the Customer level of the Client dimension.

Each path is constructed out of two or more levels and the grouping/classification relationships that link those levels. The paths represent sequences of valid roll-up and drill-down operations that can be performed during the analysis. For example the products can be grouped by brand using the grouping/classification relationship defined by the path P8. This relationship links each product in the Product level to some brand in the Brand level.

The designer or the OLAP application defines the schema of the dimensions and hierarchies mostly at design time but an ad-hoc query might need to define its own analysis path. The levels, the grouping/classification relationships and the paths are defined based on the needs of the analysis process.

For example, one possible analysis path is to drill-down from Warehouse Area to Store level. This drill-down operation could reveal the stores supplied by warehouses of a particular area. The designer of the OLAP application has defined this path (P5) but has decided to leave Cashier level out of this path. This decision can mean that when doing roll-up and drill-down operations on this path it is not meaningful to drill-down to the Cashier level.

From the above discussion one can realize that a conceptual data model suitable for multidimensional analysis should provide means to define:
1. dimension levels,
2. grouping/classification relationships (that link those levels) and
3. analysis paths.

Those first-class concepts of the multidimensional analysis must have an appropriate and straightforward representation within a conceptual model. Note that the dimension levels are in fact attributes that can characterize the measure being analyzed and the analysis paths are valid sequences of drill-down/roll-up operations.

According to the first path (P1) of our example scenario, for each customer we have the residence address in the form of area, city, and region. Assume now that for some customers the city attribute is not set (perhaps because their residence area does not belong to any city). In this case the residence area of the customer will be linked by the grouping/classification relationship directly to a residence region and not to a residence city as it happens with most of the other residence areas. Generally speaking a grouping/classification relationship may involve more than two levels since a members of a level may drill-down to members of different levels.

There are also some other aspects of the grouping/classification relationship that we found to be important. In some cases a member of a level is linked to more than one members of the next level. For example a member of the Product level will probably be linked to more than one members of the Advertising Method level since a product can be simultaneously advertised by various methods (newspapers, TV, radio, etc.). Also, some members of a level may have no links to the more detailed level. For example we may have a member of the Store Area level that is not linked to any members of the Store level. This would mean that at the given time there is no store in that area or that the system does not know or does not want to show which stores belong to this area.

The above examples show that a natural model of grouping/classification relationships might involve n-way relationships among levels. Also those relationships might not reference all members of the involved levels.

Let us now consider four queries that the analysts could ask. The first query is:

> *Q1: Give me the sum of sales for the year 2000 per Month, Product Group, Product Class, Store City and Store Region.*

Note that the above question requires aggregation on several levels of the same path. For example it requires the sum of sales for each product group and also the sum of sales for each product class. The query result represented in a grid fashion could look like Figure 2.

		January 2000						Feb...
		Store Region 1			Store Region 2			
		City 1	City 2		City 3	City 4		
Product Class 1	Group 1	1	2	3	4	5	9	
	Group 2	4	5	9	1	3	4	
		5	7	12	5	8	13	
Product Class 2	Group 3	3	4	7	5	5	10	
	Group 4	7	8	15	4	3	7	
		10	12	22	9	8	17	

Figure 2: Example result for query Q1

Now consider the case where the analyst uses two paths of the same dimension for the classification and grouping of sales data.

> *Q2: Give me the sum of sales for year 2000 per customer Profession and Residence Region.*

The two levels used for grouping, Profession and Residence Region are in fact independent although they both belong to the same dimension. So, grouping on both of them at the same time defines a two-dimensional space. If the two levels where related, like Store City and Store Region are in the previous query, we would have a one-dimensional space.

The next query shows the necessity of supporting multiple measures defined over the same set of dimensions.

Q3: Give me the sum of sales, the maximum sale value and the number of sales per Store and Month for the year 2000.

In most of the cases some of the measures will be functionally dependent on other more primitive measures. Generally speaking, a set of primitive and derived measures can simultaneously be required for an analysis process. Finally, we present a query with a somehow more complicated selection condition:

Q4: Give me the sum of sales per Month, Store Area and Brand selecting only the store areas that have increased their total sales for the year 2000 by more than 10 percent from the previous year.

This query requires calculation of the total sales value per store area for the years 2000 and 1999 and then the selection of the areas whose sum of sales value for the year 2000 greater than 110% of their sum of sales value for the year 1999. For those store areas the query requires the sum of sales calculated by Month and Brand. This is a typical query that performs selection based on aggregated data at a different level than the data required for its output.

Our experience shows that all the above queries are common OLAP queries. We believe that a conceptual model suitable for multidimensional analysis should accommodate queries like the above. This means that the structure of the query result as well as the structure of any other information involved in the query definition must be easily represented by the concepts of the model. The above requirement comes as a result of our intention of having a conceptual model that can efficiently represent all kinds of information handled by OLAP applications and not only the raw (source - detailed) data. Since we are only talking about a conceptual data model we do not require the model to represent the functional aspects (operators and functions) of the queries but we limit our requirements to the static data involved in those computations.

Based on our example queries we derive the following requirements: a good conceptual model should be able to define aggregations on arbitrary combination of levels (of different paths) even if those levels belong to the same dimension as well as aggregation on a set of levels belonging to a particular path. Furthermore, the model should allow multiple measures to be defined for a given set of dimensions and in some cases represent them in one concept, reflecting the fact that those measures are semantically linked.

3 The Multidimensional Aggregation Cube (MAC) Data Model

In this section we present the Multidimensional Aggregation Cube data model. MAC is a user-centric conceptual data model that attempts to cover the requirements described in the previous section in order to provide a highly expressive and intuitive modeling methodology for the information used in multidimensional analysis.

The MAC model uses concepts that are close to the way OLAP users perceive the information. The model tries to be expressive providing the means to model complicated real-world scenarios while using a minimal set of concepts that remain as simple as possible. The MAC model describes data as *dimension levels*, *drilling relationships*, *dimension paths*, *dimensions*, *cubes* and *attributes*.

Dimension levels represent classes of *dimension members*. Each dimension member represents some instance of a real-world property that an OLAP measure may have. Distinct dimension levels can be related by means of a *drilling relationship*. A drilling relationship indicates that there is a semantic relationship among the involved levels and describes how the dimension members of the children levels can be grouped into sets that correspond to dimension members of the parent level.

A set of drilling relationships can form a *dimension path* if several structural requirements are met. A dimension path defines a meaningful composition of drilling relationships and is used to model a valid sequence of abstraction operations (drill-down/roll-up). One or more dimension paths that share common levels can form a *dimension*.

Finally, we define *multidimensional aggregation cubes* (MACs) as a relationship among the domains of one or more dimensions. A MAC can have one or more measures. Each one of those can be considered as a simple and atomic attribute of the relationship represented by the MAC. An instance of a MAC is called a *MAC cell* or simply a cell. We now give the complete definition of the above terms and provide examples on how they are used. In the following we will use the simple term *cube* to refer to a multidimensional aggregation cube

3.1 Dimension Levels

A dimension level is a set of dimension members. The dimension members are the most detailed modeling concepts of our model and represent instances of real-world properties that OLAP measures may have. In our example scenario, the sale price is one measure of our multidimensional analysis. A property of this measure is the location where the sale took place – the cashier where the sale was recorded. For our example scenario we would define the dimension level Cashier in order to represent the cashiers where the sales transactions take place and each particular cashier of each store would be modeled as a dimension member of this level.

Dimension levels can have one or more attributes. A subset of those attributes always form a *key* for the dimension level. In most of the cases a single attribute acts as the key but multi-attribute keys can also exist. This

is due to the set semantics of dimension levels. The set semantics guaranties that each dimension member represents a uniquely identifiable, within the level, property instance.

Our example scenario requires several dimension levels to be defined. In fact all levels shown in Figure 1 will be modeled as dimension levels of the MAC model. The attributes of each level depend on the available data and the analysis requirements. For example the Residence City level could have the attributes ID, Name and Population where the attribute ID is the key of the level, Name is the real world name of the city and Population stores an estimation for the population of the city.

3.2 Drilling Relationships

A drilling relationship is a special kind of an n-way relationship that relates one dimension level (called the parent of the relationship) to N-1 dimension levels (called the children of the relationship). Formally speaking, a drilling relationship is an n-way relationship, which can relate a member of the parent level to one or more members of the children levels. This means that an instance of a drilling relationship cannot relate only members of the children levels without linking them to exactly one member of the parent level[1]. Furthermore, the parent dimension level of a drilling relationship cannot act as a child level of this relationship. The reasons for imposing the above restrictions will be explained next after defining the semantics of a drilling relationship.

A drilling relationship is used to represent the way in which a member of one level can be *decomposed* into members of some other level. We will explain the semantics of the drilling relationship with a very simple example: assume that a drilling relationship Store_to_Cashier(Store, Cashier) relates the member Store007 of the parent level Store to the members Cashier00701 and Cashier00702 of the unique child level Cashier. The semantics of this relationship is that if we have a measure value characterized by property value Cashier00701 and another, *similar* measure value, characterized by the property value Cashier00702, we can then compute (by applying the proper aggregation function) the measure value with the property Store007. The word *"similar"* in the above definition means that apart from the Cashier00701 and Cashier00702 properties, the two measure values are characterized by the same set of property values.

Based on the above semantics of a drilling relationship, one can easily see that if the parent level is also a child level then we can end up with recursive calculations for measure values. Such a situation is meaningless in OLAP applications so we require that the parent level of a drilling relationship is always different from the children level(s). It can be proved that this requirement does not restrict the modeling power but can only affect the way dimension members are grouped into dimension levels.

Unlike dimension levels the drilling relationships cannot have attributes. This is due to the simple *decomposition* semantics of the drilling relationships. Drilling relationships should not be used to represent general relationships among dimension members and so attributes are not allowed.

In our example scenario we would have a simple drilling relationship linking the level Cashier to the level Store. The Store level would be the parent level of the relationship and the level Cashier would be the only child level. Each member of the Store level would be related to one or more members of the Cashier level indicating the cashiers of each store.

A more complicated example is the relationship among the levels Residence Region, Residence City and Residence Area. The level Residence Region contains as members all the geographical regions of interest. The level Residence City contains all the cities of those regions and the level Residence Area contains individual geographical areas in which cities and regions can be decomposed. Obviously the cities do not cover all the areas of the regions so some areas can only be linked directly to regions. In order to represent the above situation we define two drilling relationships. The first one has the Residence Region as the parent level and the Residence City and Residence Area as child levels. This relationship links to each region the appropriate cities and areas. The second drilling relationship has Residence City as parent level and Residence Area as child level and represents the decomposition of cities into areas.

One can argue that drilling relationships could always be simpler and have only one child. In the above example, dummy cities could be used for grouping areas that are not within a real city. Still, if the cities where not directly related to areas but rather several levels existed among the Residence City and Residence Areas then we would need dummy members on each intermediate level. We believe that such a modeling solution is semantically wrong since we use dummy members that do not correspond to the real world. Furthermore, such a solution would result into unnatural and complicated drill-down operations.

3.3 Dimension Paths

A dimension path is a set of drilling relationships used to model a meaningful sequence of drill-down operations. In its simple form, a dimension path is a sequence of drilling relationships each having only one child level. In this chain each child level of a drilling relationship is also the parent level of the next drilling relationship, except for the last one. The child level of the last drilling relationship is called the *detailed level* [Vass98] of the dimension path.

[1] Note that we do not require the relationship to be strict. Two parent-level members can be related to the same child-level member.

The dimension paths are defined in order to model the paths on which the multidimensional analysis is usually performed. In the field of multidimensional analysis the drill-down and roll-up operations follow pre-designated paths rather than individual drilling relationships. This means that even if a level is a parent of multiple drilling relationships the drill-down operation will be performed based on only one of these relationships – the one that belongs to the dimension path on which the analysis is currently performed.

In order to formally define a dimension path we will first define what is the graph of a set of drilling relationships. Given a set **P** of drilling relationships, the graph of **P** is a directed graph with the following properties:

- Each node of the graph represents a level referenced by some member of **P**. Even if a level is referenced by many drilling relationships, as parent or child level, the graph will contain only one node that represents that particular dimension level.
- For each drilling relationship in **P** and for each child level of this relationship, the graph contains a directed edge from the parent of the drilling relationship to the child level.

We formally define a dimension path to be a non-empty set **P** of drilling relationships that have the following properties:

1. In the graph of **P** exactly one node has no incoming edges.
2. The graph of **P** has no circles – it is a DAG.
3. There are no two drilling relationships in **P** having the same parent level.

The first of the above properties requires that paths always have a unique detailed level. This is required in order to guaranty the aggregation semantics of paths. The detailed level of a path is considered to correspond to the source of information. Based on this detailed data all other aggregation levels can be computed. In other words, through the composition of drilling relationships of a path each member of any level (of this path) is finally linked to a uniquely identifiable set of detailed level members [CaTo98].

Note that, based on the above definition; a dimension path has no build-in support for aggregating all members of its detailed level although this is a very common operation in the multidimensional analysis. For example, our first example query requires aggregation for all customers and our second example query requires aggregation for all products. In order to achieve such an aggregation within a path, a special dimension level (usually called ALL [G++96]) containing a unique member (called all) must be defined. Furthermore, an artificial drilling relationship that will *complete the link* of all detailed level members to the unique member of the ALL level must also be defined. "*Complete the link*" means that either the members of the detailed level are directly related to the all member or indirectly related to it through a sequence of drilling relationships defined by the path. The ALL level is related to the highest available level of the path and not directly to the detailed level.

For our example scenario we need to define several paths. In fact, each path described in Figure 1 will also be a dimension path of the MAC model. To almost each path of Figure 1 we would add the special dimension level ALL. Only for the path P7 we would not add the ALL level since it is not meaningful to aggregate the members of the advertising method. Recall that a product may be related to more than one advertising method. Figure 3 gives a graphical representation of the dimension levels, drilling relationships and dimension paths for the dimensions Item and Location of our example scenario.

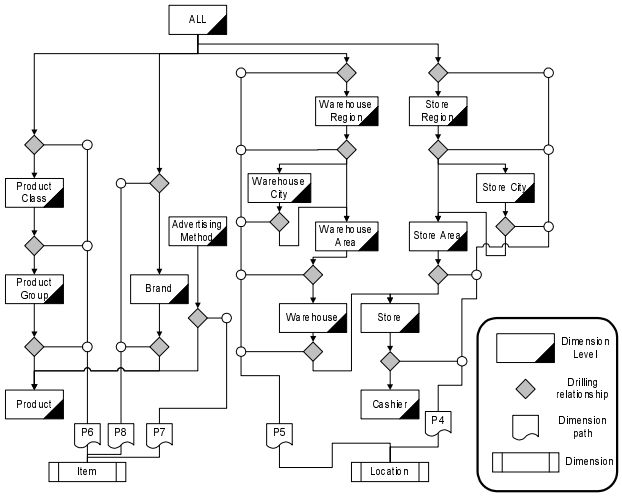

Figure 3: The dimensions Item and Location

3.4 Dimensions

A dimension is a concept used to define meaningful groups of dimension paths. This grouping is essential in order to model the semantic relationships that exist among the various paths and allows powerful OLAP modeling, as we will show in our examples to follow.

The dimensions are complex concepts used in our model to represent the various properties of measures as well as assist the process of multidimensional analysis. With their complex structure, the dimensions can be considered to classify properties into various levels, define the relationship among properties of different levels and describe meaningful drilling paths.

Formally speaking, a dimension is a non-empty set of paths. If a dimension contains more than one path then each path must have at least one common level with at least one other path of the dimension. The common levels must be assigned the same meaning in the various paths where they appear. For example, if we replace in our example both Residence City and Store City levels with a general City level then the paths P1 and P4 end up having

a common level. Those two paths, although they share a common level, cannot be combined into a dimension because they assign different meaning to their common level. The City level in P1 would represent the residence cities of customers while in P4 it would represent the cities in which stores are located.

The reason for grouping dimension paths into dimensions is the semantic relationship that usually exists among various paths. For example, consider the paths P4 and P5 of Figure 3. Both the above paths characterize the sales measures from the store point of view. For both paths, the level Store is used to represent the stores where the sales are recorded so it is meaninful to group these two paths and view them as one Location dimension of the sales measure. Generally speaking we can say that the number of paths involved to define a dimension depend on the content of the dimension and on the complexity of the scenario being modeled. We define the graph of a dimension to be the union of the individual graphs of its dimension paths. Recall that in the graph each dimension level is represented by a unique node so according to the above definitions any dimension graph is a connected graph. Figure 4 gives the graph of the dimension Client.

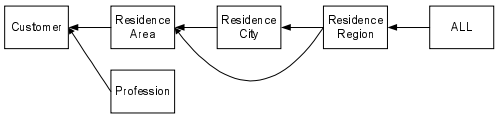

Figure 4: The graph of the dimension Client

The dimensions are of extreme importance in our model because they are used in the definition of cubes. Still, a cube definition does not involve dimensions as sets of paths but rather as sets of what we call *dimension values*[2]. A dimension value can either be a simple dimension member or it can be a set of dimension members. Formally speaking, we define a dimension value of a dimension D to be either a dimension member of a level of D or a set of two or more *compatible* dimension members belonging to *distinct* and *non-related* levels.

Two or more dimension members are called *"compatible"* if it is possible to find - within any level of the dimension – at least one dimension member that can roll-up (using any combination of drilling relationships of the dimension) to each of these members.

The term *"distinct"* means that from each level, at most one dimension member can participate in a dimension value. Furthermore, two or more levels participating in a dimension are called *"non-related"* if there is no path in the directed graph of the dimension linking any two of these levels.

[2] Note the difference among the terms *dimension value* and *dimension member*

Based on the above definition one can realize that the dimension values depend on both the structure of the dimension and on the particular instances of its levels and drilling relationships. The set of all possible dimension values defined for a dimension is called a ***dimension domain***. Note that although the instance of the dimension domain depends on the instances of levels and drilling relationships, the schema of the dimension still determines the structure of the dimension domain by defining all possible combinations of non-related levels.

Take as an example the dimension Client the graph of which is shown in Figure 4. Each member of the levels Customer, Residence Area, Residence City, Residence Region and Profession is a dimension value. If Athens_North, Athens_East are two members of the Residence Area level and doctor, teacher are two members of the Profession level then each of these members is a dimension value. Also, each combination of members: one from the Profession level and one from the Residence Area may be a valid dimension value. If the Customer level contains a member that rolls-up to both the doctor member and the Athens_North member then (and only then) the set {Athens_North, doctor} is a valid dimension value. This customer member would represent a doctor customer living in the North of Athens. Similar constrains must hold in order for the sets {Athens_North, teacher}, {Athens_East, doctor}, and {Athens_East, teacher} to be dimension values. The combination of members is possible only for non-related levels. This means that if Patra is a member of Residence City then the set {Athens_North, Patra} is definitely not a valid dimension value. Either Athens_North is an area of the city Patra and it is redundant to mention Patra or Athens_North is not within Patra and the combination of those members is meaningless.

The intuition behind dimension domains is that *the dimension values represent all possible properties that can be used for multidimensional analysis*. For example a query may ask for customers living in Athens_North or it may ask for customers with the doctor profession that live in the Athens_North area. Still, it is not meaningful to ask for customers that live in the Athens_North area and in Patra city at the same time, since the latter does not include the former. Furthermore if the Customer level includes no members that are doctors living in the Athens East area then the dimension domain will not include such a combination of values making obvious the answer to any such queries. So, the structure and content of the dimension domains can be used as a valuable source of information in a semantic query optimization algorithm.

3.5 Multidimensional Aggregation Cubes

The Multidimensional Aggregation Cube (MAC) is the main and most complex concept of our model. All other concepts previously described are directly or indirectly used in the definition of a MAC. The dimension levels model properties of measures, the drilling relationships define relationships among levels, the dimension paths

group drilling relationships and the dimensions group paths. The MAC is the only concept that associates property values with actual measure values and stresses the complex hierarchical structure defined by dimensions.

Formally speaking, a multidimensional aggregation cube is an n-way relationship relating N dimension domains. This relationship has one or more attributes which represent the measures of the MAC. Each instance of this relationship is called a *cell* and defines a relationship among one dimension value from each of the involved domains. The cell is annotated with the values of the cube attributes – the measure values. The N dimension values that a cell relates are called the coordinates of the cell. Obviously, the measures of a cube are functionally dependant on its coordinates.

Assume the following example: The cube C1 is defined over the domains of the dimensions Location and Item (Figure 3) having only Sum of sales as its measure. C1 contain the cells defined in Table 1 where S represents the Sum of sales measure:

Cell name	Coordinates		M
	Item	Location	S
cell_A	Product =P_A	Cashier =Cashier00701	10
cell_B	Product =P_B	Cashier =Cashier00701	20
cell_C	Brand =B_1	Cashier= Cashier00701	30

Table 1: The cells of the example cube C1

The measure of cell_A represents the sum of sales done at the Cashier00701 for the product P_A. Likewise the measure of cell_B represents the sum of sales done at the Cashier00701 for the product P_B. Finally, cell_C represents the sum of sales done at the Cashier00701 for all products of the brand B_1. Assuming that P_A and P_B are the only products of the brand B_1, the cell_C then represents the aggregation of cell_A and cell_B. This means that the measure of cell_C must be equal to the sum of measures of cell_A and cell_B.

The above example reveals the key property of the MAC model: *the instances of a cube (the cells of a cube) can represent measure values of different granularities even if there is a functional dependency among them*. In our example, cell_A refers to sales measured per Product and Cashier while cell_C refers to sales measured per Brand and Cashier. Those cells, although defined at different levels of granularity, can be part of the same cube. This is due to the definition of the dimension domain, which states that all members of **all** participating levels are valid dimension values.

We believe that the above property of our cubes is crucial for the compact and intuitive representation of multidimensional data. As explained in the section 2, the OLAP users usually handle data defined over various levels of granularity. Queries may impose selection conditions on various dimension levels and may require their result at a completely different granularity. Furthermore, even the granularity of source data may vary. For example, a store may change for a time period the way it records its sales. The store could record only the sum of sales per product for all its cashiers and not per product and cashier as it used to do before. Another example is when a cube contains predicted and actual sales. The predicted values may not be computable at the lowest detail level but only at some higher levels. A final example is when for security reasons the detailed data may not be available for stores of a particular area.

If we had only single-granularity cubes than each time we needed a new combination of levels we would have to add a new cube to the schema. The schema would get complicated and so would the queries. The schema would not only have to include the dimensions and the cubes but also the functional dependencies among the existing cubes. The queries would have to be aware of the available cubes as well as their functional dependencies. They would have to select which ones to use and probably join some of them before defining one or more grouping operations on parts of those cubes.

Our approach allows a single cube to include data of all meaningful granularities. By doing so we simplify the schema making it usable by the users. Also, this approach allows queries to be expressed in a very elegant and straightforward way avoiding any declaration of joins.

Consider the following example: Let C2 be a cube defined over the domain of the dimensions Time, Location, Item, and Client. The structure of those dimensions is illustrated in Figure 1 with the addition of the special level ALL as described earlier. Assume SumOfS is the unique measure of C2 and it represents the sum of sales. Using this schema the query Q4, described in section 2, can be expressed in a very simple manner. We use a QBE notation style and define the query Q4 in Table 2. The **P.** notation defines which coordinates and measures to be returned in the result set.

Time	Location	Item	Client	SumOfS
P.Month	**P.**Store_Area._x	**P.**Brand	ALL	**P.**
Year.2000	Store_Area._x	ALL	ALL	>_s*1.1
Year.1999	Store_Area._x	ALL	ALL	_s

Table 2: The query Q4

An additional advantage of our approach is that the most typical OLAP operations: drill-down, roll-up, slice and dice are translated to simple selection queries on the cube. For example, a drill-down on the results of Q4 to the Product level can be expressed by simply replacing Brand with Product at the Item coordinate of Table 2.

Nevertheless, there is an important argument against this modeling approach. The cube can represent cells with measures that are functionally dependent but it cannot guaranty their consistency. The measure of cell_C could

have been 50 in Table 1 without violating in any way the definition of the cube C1. Still, from a semantically point of view this value would be inconsistent with the values of cell_A and cell_B. This inability to guaranty consistency comes from total absence of the involved aggregation functions. Each measure of a cube is semantically related to some aggregation. In the above examples the measure Sum of sales is obviously related to the SUM aggregation function. Still, the cube definition does not include this relationship making impossible any consistency checking.

We have intentionally chosen not to include the aggregation functions in the MAC model for a number of reasons. We decided not to define operations, not to include aggregation functions and not to cover any other functional aspects of data in the current model because those aspects are orthogonal to the defined concepts. The model can be extended to include aggregation functions, consistency-checking algorithms and operations without changing any of the existing MAC concepts. Furthermore, a separate functional model can cover those needs. It seams that a separate functional model is more suitable because each application domain requires its own specific aggregation functions, operations and consistency semantics.

For example in some applications it might be acceptable to view and analyze data that includes inconsistent parts (maybe due to missing, incomplete or wrong information). A separate functional model can be tailored to cope with operations on such data. By forcing consistency at the data modeling level we would make our model inappropriate for those applications.

4 Related Work

Modeling multidimensional data is not an OLAP specific issue. In the database community, several research areas like statistical databases, scientific databases, geographical databases and temporal databases deal with multidimensional data. Still, each of these areas has particular modeling needs and has developed specialized multidimensional data models. The area closest to data warehouses and OLAP is the statistical database area [Shos97] where several multidimensional models have been proposed [OOM85], [RR91]. In fact those models where proposed long before the appearance of the term "OLAP" [Codd93].

In the data warehouse and OLAP area the first multidimensional data models where developed by product vendors as the research in the OLAP domain has followed the evolution of industrial products. Vendors as still using and developing their own data models. Also, various standardization bodies have defined their own models [Meta97] [Olap97] [TPC99]. Due to space limitations we are not going to discuss any of the previously referenced models but refer the reader to [VaSe99] for an overview and comparison of those models.

During the last few years a plethora of multidimensional data models for data warehouses and OLAP have been proposed. A comparison of some of them can be found in [VaSe99] and [SBH99]. We are currently aware of 12 models that have been published in research papers. Most of them are logical data models and only few ([TBC99] [S++98]) can be considered as purely conceptual. Each of those models has taken a somehow different modeling approach ranging from a simple global table to sophisticated object classes.

In order to demonstrate that our model is not 'yet another' multidimensional model we evaluated all 12 published models against the requirements described in section 2. This may not be *fair* for the purely logical models since the requirements represent conceptual modeling needs. Still, the evaluation is done only to demonstrate that none of the models published so far has the expressive power of MAC. In fact the evaluation shows that one requirement is not satisfied by any of the models and even for the remaining requirements there is no model satisfying all of them. Since our model can satisfy all the requirements of the evaluation we argue that our proposal is an improvement to the existing status.

The requirements of our evaluation are presented in the following list. Each requirement states what the model should be capable of representing within a schema.

1. Levels within dimension (even in the form of simple attributes).
2. Grouping/classification relationships among levels.
3. Many-to-many type of grouping/classification relationships.
4. N-way grouping/classification relationships that relate n dimension levels.
5. Grouping/classification relationships that do not require total participation of the involved levels.
6. Analysis paths.
7. Multiple measures as part of one concept.
8. Measures defined at any granularity level – for each involved dimension.
9. Measure values defined over various granularity levels as part of one concept.
10. Measure values characterized, for some of its dimensions, by more than one dimension level members.

Note that the aggregation level of a measure value is the lowest dimension level that can be used to characterize this value. Also, an analysis path is a lattice of grouping/classification relationships defined on a set of levels. This lattice prevents the user from performing a meaningless (according to the schema designer) drill-down or roll-up operation to an arbitrary -outside the lattice- level of the dimension.

	1	2	3	4	5	6	7	8	9	10
[AGS97]							✓	✓		
[CaTo98]	✓	✓						✓		✓
[DaTh97]	✓						✓	✓		
[GoRi98]	✓	✓			✓		✓	✓		✓
[GyLa97]	✓						✓	✓	✓	✓
[Lehn98]	✓						✓	✓		✓
[LiWa96]	✓							✓	✓	✓
[PeJe99]	✓	✓	✓	✓	✓			✓	✓	✓
[S++98]	✓	✓	✓		✓		✓			
[TBC99]	✓	✓	✓	✓	✓		✓			
[Truj99]	✓						✓	✓		
[Vass98]	✓	✓						✓		
MAC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3: Evaluation of multidimensional models

The result of our evaluation is shown in Table 3. Note that some of the models ([GyLa97], [Truj99], [AGS97]) represent relationships among levels using user-defined functions, which are then used in operations. Also, other models ([LiWa96], [Lehn98], [DaTh97]) leave the relationships to be defined by the particular data instances and provide no schema definition for them. In both cases we considered that the requirements 2,3,4,5 involving grouping/classification relationships as part of the schema are not met.

The requirement not met by any of the models is the concept of an analysis path. We believe that this information is an important structural part of the dimension design and it should be represented at the conceptual level.

Although our model seams to be able to model a broader range of OLAP scenario than other proposed models, there are a few requirements mentioned in several papers ([Codd93], [PeJe99], [TBC99] [GyLa97]), which are not satisfied by MAC. In our opinion, the most important of such requirements is the support for correct aggregation of data. As described in [LeSh97] the measures cannot always be consistently aggregated by an arbitrary aggregation function. In order to provide support for correct aggregations the model must include additional information regarding measures and grouping/classification relationships. Our model does not include such additional information since we believe that this kind of information, as well as information about aggregation functions and derived measures, can be described by an independent functional model which will supplement MAC.

A second important requirement stated by various papers ([GyLa97], [PeJe99], [AGS97]) is the need for *symmetric treatment* of dimensions and measures. It is important to note that what the authors finally mean by *symmetric treatment* is the ability to transform a measure into a dimension and the other way around. All models claiming to support this requirement ([GyLa97], [PeJe99], [AGS97]) do so by providing the appropriate transformation operations. So, this requirement does not mean that dimensions and measures are represented in the same manner by the model. We believe that our model can easily support this requirement through the definition of the proper transformation operations (initially called Push and Pull by [AGS97]).

5 Conclusions

In this paper we addressed the problem of conceptual modeling of data used in multidimensional analysis. We presented a set of modeling requirements through the use of examples and with those requirements in mind we defined a new conceptual data model, named MAC. The proposed model uses concepts familiar to OLAP users, like dimensions, levels, paths, measures and cubes. Those concepts are properly defined in order to allow modeling of complicated real-world scenarios. Our evaluation and comparison to previously published models showed that MAC offers a unique combination of modeling skills. Our model is the first user-centric conceptual model to define cubes as multi-granularity relationships making both schemas and queries much more simple and intuitive. The model defines dimension levels, drilling relationships, dimension paths and dimensions as first-class and standalone concepts, making it possible to share those concepts among multiple cubes. Furthermore, the complexity of drilling relationships and the usage of analysis paths in the definition of dimensions are additional novelties of our model taking a step beyond the classical multiple hierarchies. Finally, note that the definition of dimension domains implicitly represents a straightforward method for semantic query optimization at both the schema and the instance level.

Future work includes the definition of MAC as an extension to the E/R model and the research of a suitable logical model on which concepts of our model can be mapped. We also plan to define a functional model that will include aggregation functions, derived measures, and operations and will define the summarizability [LeSh97] of measures as well as other consistency rules.

Acknowledgements

This work has been partially funded by the European's Union Information Society Technologies Programme (IST) under project EDITH (IST-1999-20722).

References

[AGS97] R. Agrawal, A. Gupta, S. Sarawagi: *Modeling Multidimensional Databases*. Proc. of the ICDE 1997.

[BCN92] C. Batini, S. Ceri, S. Navathe: *Conceptual Database Design*. Benjamin/Cummings, 1992.

[CaTo98] L. Cabibbo, R. Torlone: *A Logical Approach to Multidimensional Databases.* Proc. of the EDBT 1998.

[ChDa97] S. Chaudhuri, U. Dayal: *An overview of Data Warehousing and OLAP technology*. ACM SIGMOD Record, 26(1), March 1997.

[Codd93] E. F. Codd: *Providing OLAP to user-analysts: An IT mandate*. E.F. Codd and Associates, 1993.

[DaTh97] A. Datta, H. Thomas: *A conceptual Model and an algebra for On-Line Analytical Processing in Data Warehouse*. Proc. of the WITS 1997.

[G++96] J. Gray et al.: *Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab and Sub-Totals*. Proc. of the ICDE 1996.

[GoRi98] M. Golfarelli, S. Rizzi: *A Methodological Framework for Data Warehouse Design*. Proc. of the DOLAP 1998

[GyLa97] M. Gyssens, L.V.S. Lakshmanan: *A Foundation for Multi-Dimensional Databases*. Proc. of the VLDB 1997.

[Inmo96] W.H. Inmon: *Building the Data Warehouse*. John Wiley & Sons, 1996.

[Kimb96] R. Kimball: *The Data Warehouse Toolkit*. John Wiley & Sons, 1996.

[Kimb97] R. Kimball: *A Dimensional Modeling Manifesto*. DBMS, August 1997.

[Lehn98] W. Lehner: *Modeling Large Scale OLAP Scenarios*. Proc. of the EDBT 1998

[LeSh97] H. Lenz, A. Shoshani: *Summarizability in OLAP and Statistical Databases*. In Proc. of the SSDBM 1997.

[LiWa96] C. Li, X. S. Wang: *A Data Model for Supporting On-Line Analytical Processing*. In Proc. of the CIKM 1996.

[Mendel] A. O. Mendelzon: *Data warehousing and OLAP: a researchoriented bibliography* . http://www.cs.toronto.edu/~mendel/dwbib.html.

[Meta91] Metadata Coalition: *Meta Data Interchange Specification*. (MDIS Version 1.1), August 1997.

[Olap97] OLAP Council: *OLAP and OLAP Server Definitions*. 1997 http://www.olapcouncil.org/reasearch/glossary.htm

[OOM85] G. Ozsoyoglu, M. Ozsoyoglu, F. Mata: A Language and a Physical Organization Technique for Summary Tables. Proc. of the SIGMOD 1985.

[PeJe99] T. B. Pedersen, C. S. Jensen: *Multidimensional Data Modeling of Complex Data*. Proc. of the ICDE 1999.

[RR91] M. Rafanelli, F.L. Ricci: A functional model for macro-databases. SIGMOD Record, 20(1), March 1991.

[SBH99] C. Sapia, M. Blaschka, G. Höfling: *An Overview of Multidimensional Data Models for OLAP*. Technical Report 1999. http://www.forwiss.tu-muenchen.de/

[S++98] C. Sapia, M. Blaschka, G. Höfling, B. Dinter: Extending the E/R model for the Multidimensional Paradigm. Proc. of the DWDM 1998.

[Shos97] A. Shoshani: *OLAP and statistical databases: Similarities and differences*. Proc. of the PODS 1997.

[TBC99] N. Tryfona, F. Busborg, J. G. B. Chistiansen: *starER: A Conceptual Model for Data Warehouse Design*. Proc. of the DOLAP 1999.

[TPC99] TPC: *TPC Benchmark H* and *TPC Benchmark R*. Transaction Processing Council. June 1999. http://www.tpc.org/

[Truj99] J. Trujillo: *The GOLD model: An Object Oriented multidimensional data model for multidimensional databases*. Proc. of the ECOOP 1999.

[VaSe99] P. Vassiliadis, T. Sellis: *A Survey of Logical Models for OLAP Databases*. SIGMOD Record 28(4), Dec. 1999.

[Vass98] P. Vassiliadis: *Modeling Multidimensional Databases, cube and cube operations*. Proc. of the SSDBM 1998.