Conceptual Multidimensional Models

Chapter in the Book: Multidimensional Databases
Maurizio Rafanelli Ed.

RICCARDO TORLONE
Dip. di Informatica e Automazione
Università Roma Tre
Via della Vasca Navale, 79
00146 Roma – Italy
E-mail: torlone@dia.uniroma3.it
Tel: +39-06-55177053 Fax: +39-06-5573030

1. Introduction

The ability to represent information in an abstract and implementation-independent way is crucial in the life cycle of every information system application, not only in its design but also in its operational phase. This is particularly true in the context of data warehousing and OLAP where, because of the level of complexity, application development and management are usually difficult and error-prone tasks.

In spite of this, conceptual data models for data warehousing have received little attention for a long period in the applicative area. Traditionally, multidimensional applications are modeled in a way that strictly depends on the corresponding implementation. One of the most used formalisms for data representation in this context is the relational model, which is clearly well suited in the case of a ROLAP implementation. In general however, using a logical data model has a number of negative consequences. First, a logical representation is conceived to describe, at the appropriate level of abstraction, how data is stored in a specific DBMS, but it is usually not expressive enough to capture in an effective way the essential, multidimensional aspects of a data warehousing application. Second, it is difficult to define a design methodology that includes a general, conceptual step, independent of any specific system but suitable for all. Finally, in specifying aggregations of data, analysts often need to take care of tedious details that refer to the distribution of the information along the various structures used for its storage. For these reasons, data warehouse developers today understand that conceptual data models and methodologies are fundamental ingredients for the realization of good-quality products and for effective employment of their content.

It is now widely accepted that traditional conceptual data models, such as the Entity-Relationship model, are not appropriate for description of the multidimensional and aggregative nature of OLAP applications. For this reason, a variety of multidimensional data models have recently been proposed by both academic and industry communities, although it should be noted that a consensus on formalism or even a common terminology has not yet emerged.

In this chapter, we first discuss the requirements that an ideal conceptual multidimensional model should fulfill. These requirements are suggested by general information system modeling principles and the specific characteristics of OLAP applications. Building on these requirements, we then present a general conceptual multidimensional data model and show how it can be used to describe the basic aspects of a business application in a way that is easy to understand and independent of the criteria for actual data organization in the various systems. Far from being complete, this model aims at capturing the core of the various proposals of multidimensional data models and the conceptual means adopted by OLAP systems for data representation and manipulation. The model relies on a few agreed concepts. The basic notions are the *dimension* and the *data cube*. A dimension represents a business perspective under which data analysis is to be performed and is organized in a hierarchy of *levels*, which correspond to different ways to group its elements. A *data cube* represents factual data on which the analysis is focused and associates *measures* with *coordinates*, defined over a set of dimension levels.

Starting from the characteristics of the model proposed, we summarize the general features that a multidimensional conceptual model should support. We then survey various multidimensional models proposed and relate their characteristics to these general features Finally, we discuss the main points raised in the chapter and some problems that remain to be solved in this context.

We do not address query languages, which are clearly strictly related to the subject of data models, as they are described in Chapter 10.

2 Background and terminology

2.1 Conceptual data models and data warehousing

A data model is for a database designer what a box of colors is for a painter: it provides a means for drawing representations of reality. Indeed, it has been claimed that “data modeling is an art” [23], even if the product of this activity has the prosaic name of *database scheme*.

When a data model allows the designer to devise schemes that are easy to understand and can be used to build a physical database with any actual software system, it is called *conceptual* [4]. This name comes from the fact that a conceptual model tends to describe *concepts* of the real world, rather than the modalities for representing them in a computer.

Many conceptual data models exist with different features and expressive powers, mainly depending on the application domain for which they are conceived. As we have said in the Introduction, in the context of data warehousing it was soon realized that traditional conceptual models for database modelling, such as the Entity-Relationship model, do not provide a suitable means to describe the fundamental aspects of such applications. The crucial point is that in designing a data warehouse, there is the need to represent explicitly certain important characteristics of the information contained therein, which are not related to the abstract representation of real world concepts, but rather to the final goal of the data warehouse: supporting data analysis oriented to decision making. More specifically, it is widely recognized that there are at least two specific notions that any conceptual data model for data warehousing should include in some form: the *fact* (or its usual representation, the *data cube*) and the *dimension*. A fact is an entity of an application that is the subject of decision-oriented analysis and is usually represented graphically by means of a data cube. A dimension corresponds to a perspective under which facts can be fruitfully analyzed. Thus, for instance, in a retail business, a fact is a sale and possible dimensions are the location of the sale, the type of product sold, and the time of the sale.

Practitioners usually tend to model these notions using structures that refer to the practical implementation of the application. Indeed, a widespread notation used in this context is the "star schema" (and variants thereof) [29] in which facts and dimensions are simply relational tables connected in a specific way. An example is given in Figure 1. Clearly, this low level point of view barely captures the essential aspects of the application. Conversely, in a conceptual model these concepts would be represented in abstract terms which is fundamental for concentration on the basic, multidimensional aspects that can be employed in data analysis, as opposed to getting distracted by the implementation details.

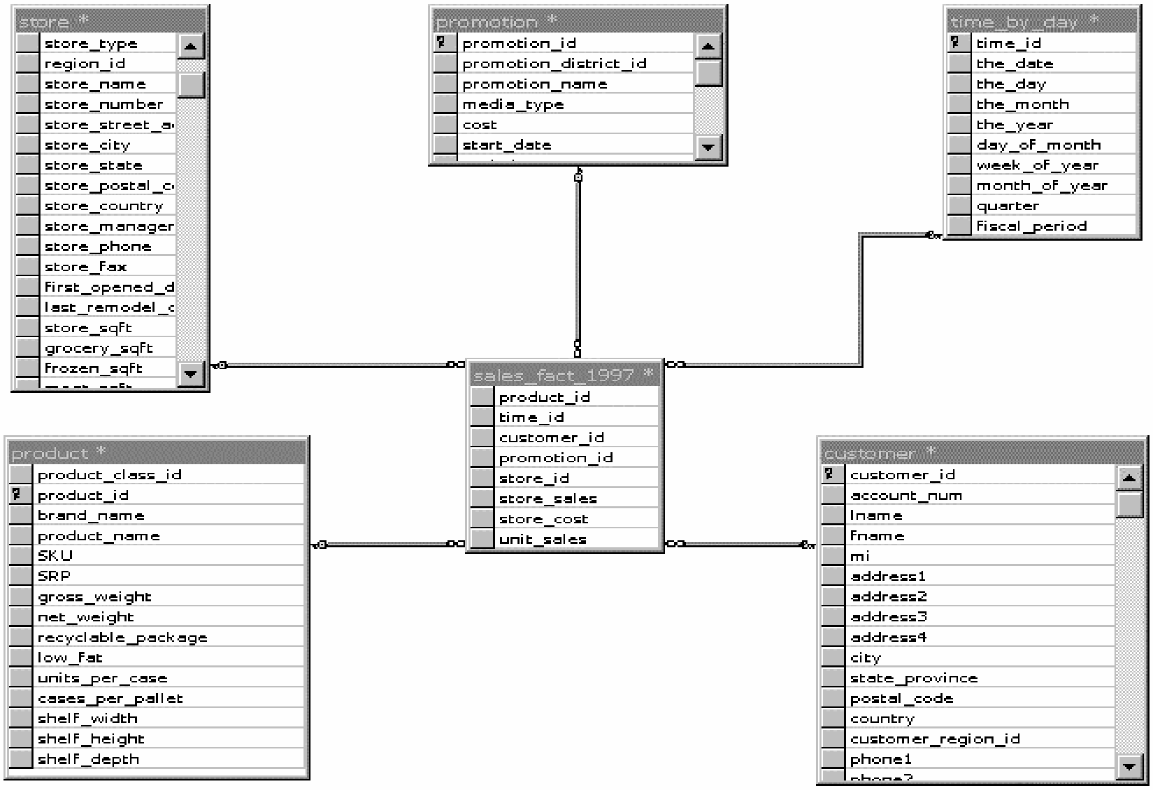

Figure 1: An example of star schema

Before tackling in more detail the characteristics of conceptual models for multidimensional applications, it is worth making two general observations. First, we note that in contrast to other application domains, in this context not only at the physical (and logical) but also at the conceptual level, data representation is largely influenced by the way in which final users need to view the information. Second, we recall that conceptual data models are usually used in the preliminary phase of the design process to analyze the application in the best possible way, without implementation "contaminations". There are however further possible uses of multidimensional conceptual representations. First of all, they can be used for documentation purposes, as they are easily understood by non-specialists. They can also be used to describe in abstract terms the content of a data warehousing application already in existence. Finally, a conceptual scheme provides a description of the contents of the data warehouse which, leaving aside the implementation aspects, is useful as a reference for devising complex analytical queries.

2.2 Modelling Multidimensional Applications

Let us now investigate in more detail, but still informally, the fundamental ingredients of a conceptual data model for data warehousing. We start from the observation made above that the effectiveness of data warehousing modeling strictly depends on the ability to describe factual data according to appropriate *dimensions*, that is, “perspectives” under which data can be analyzed. For instance, in a data warehousing application for a retail company it is useful to organize data along dimensions such as products commercialized by the company, stores selling these products and days on which sales occur. To better support data analysis, it is useful to organize a dimension into a hierarchy of *levels*, obtained by grouping elements of the dimension according to the analysis needs. For instance, we might be interested in grouping products into brands and categories, and days into months and years. When the members of a level *l* can be grouped to members of another level *l’* it is often said that *l rolls up* to *l’*. For instance, the level “product” rolls up to the level “brand”. A level usually has *descriptive attributes* (or simply *descriptions*) associated with it. For instance, descriptions of a store include its name, manager, and address.

Let us consider a more concrete example, which will be used as a simple case study throughout this chapter.

Example 2.1 *The* Toys4All *company produces and sells a large number of products (mainly toys) in a chain of stores, over a wide territory.*

A main business goal for this company could be to understand the impact of promotions on sales, that is, how promotions influence product sales and to what extent promotions are profitable. Another important business goal could be the analysis of the warehouse process, where inventory levels should be measured monthly, for each product and warehouse controlled by the company. It follows that possible dimensions of the Toys4All data warehouse application are Product, Store, Warehouse, Time, *and* Promotion*. The Product dimension may be organized into levels such as* item *(whose members are products such as* Disney’s Dinosaur *and* Duplo Pooh*),* product-line *(containing members like* Mattel’s Disney *and* Lego Duplo*),* brand *(*Mattel *and* Lego*),* category *(*Popular Characters *and* Blocks*), and* department *(*Action Figures *and* Blocks*). The elements of the* Time *dimension describe days over a period of time; this dimension may be organized into the levels* day, month, quarter, year, *and* season*. A member of the level* day *might be Feb 27, 2001. Members of the level* day *can be grouped to members of the level* month, *but also to members of the level* season *(e.g., Carnival). Descriptions of the item* level *might be its name and code.* ■

Traditionally, the entities of an application subject to decision-oriented analysis are called *facts* and the specific and measurable aspects of a fact relevant for the analysis are known as *measures*. A collection of measures for the same fact can be nicely represented by means of a *data cube* (or hypercube) having a “physical” dimension for each “conceptual” dimension of measurement: a coordinate of the data cube specifies a combination of level members and the corresponding cell contains the measure associated with such a combination.

Example 2.2 *For the Toys4All company, a possible fact is the daily sale. This fact can be analyzed with respect to the day of the sale, the product sold, the store of the sale, and the promotion applied to the daily sale. The measurements made for each daily sale could include the number of units sold, the income and the cost. Thus, a data cube* Sales *can be used to describe daily information about the items sold by the stores of the chain. An instance of this data cube can state the fact that on Feb 27, 2001 the store Colosseum has sold 2 pieces of Duplo Pooh, applying a Carnival 2001 Promotion, for a corresponding gross income of 19.98 Euros against a cost of 14.98 Euros.*

In the warehouse process, measurable facts are the inventory levels, to be measured, for instance, monthly, for each product and warehouse. They can be modeled by means of a data cube Inventory. *The measurements made for each monthly inventory could include the inventory level (the quantity in stock at the end of the month), the quantity shipped during the month, and the value at cost of the quantity in stock.* ■

In the next section we will try to formalize the general notions discussed in this section.

3 A conceptual multidimensional model

We now present a simple multidimensional data model **"**MD**"** that provides a number of constructs to describe, in an abstract but natural way, the basic notions involved in multidimensional analysis. As is customary in database models, we make a clear distinction between the *scheme* (which specifies the structure of a concept) and the *instance* (that is, the actual values associated with a concept).

3.1 Formal definition of MD

We assume the existence of a finite set of *base types* such as text, integer, decimal, and date. Each base type *t* is associated with a domain of *base values* of that type. We also assume the existence of a countable set of *names* and a countable set of *identifiers* (*ids*). Such ids are values, distinct from base values, that are used to uniquely identify real life objects.

A dimension has three main components: a set of levels, a set of level descriptions and a hierarchy over the levels.

Definition 3.1 [Dimension scheme] *An* MD dimension scheme *D consists of:*

- *a finite set L of names called* levels;
- *a finite set* Δ *of names called* level descriptions*, for each level in L; each description is associated with a base type t;*
- *a partial order* $\leq$ *called* roll up relation *on the levels in L; if* $l_1 \leq l_2$ *we say that* l_1 rolls up to l_2.

There is a natural graphical representation of an MD dimension. Some examples are reported in Figure 2. In this representation, levels are depicted by means of round-cornered boxes and there is a direct arc between the two levels l_1 and l_2 if $l_1 \leq l_2$. Small diamonds depict the descriptions of a level.

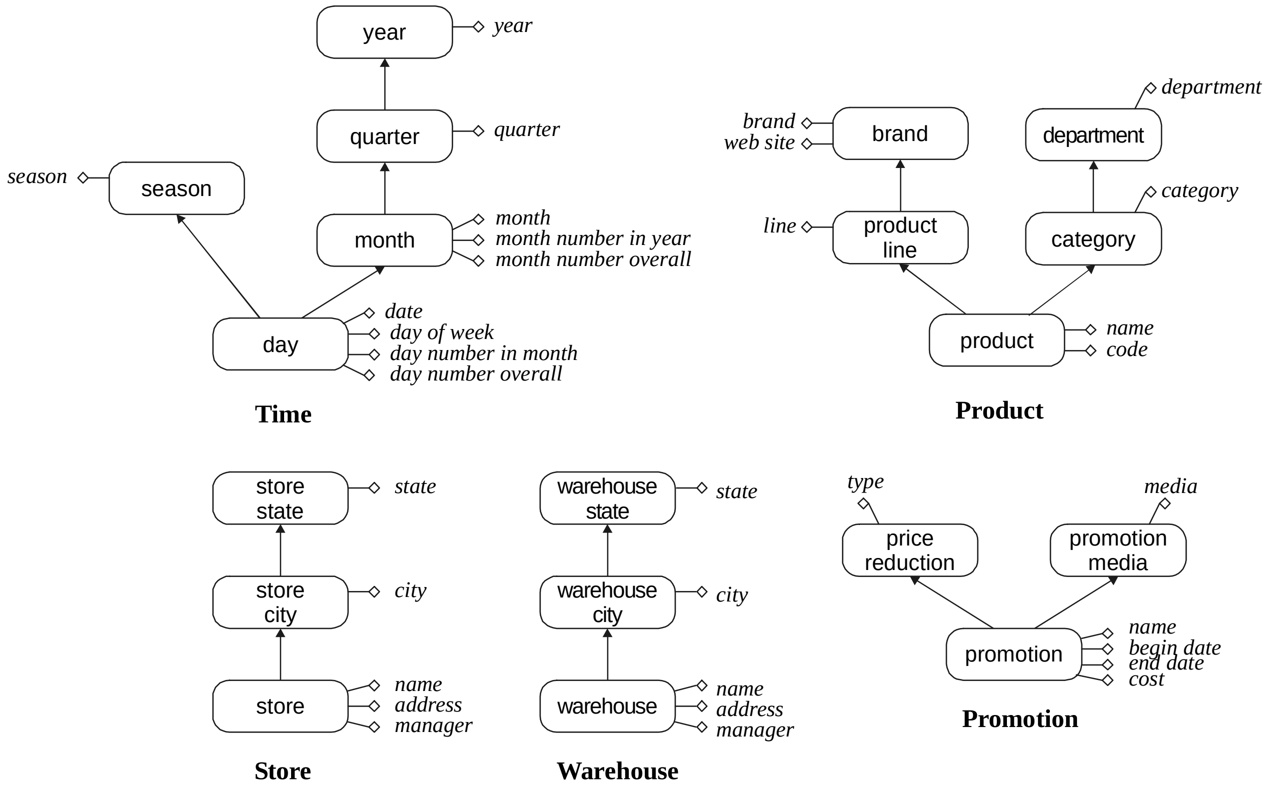

Figure 2: Dimension scheme in the MD model

Example 3.1 *Figure 2 reports the dimensions for the Toys4All company, as described in* Example 2.1 : *Time, Product, Store, Promotion and Warehouse.*

As an example, let us consider in more detail the Time dimension. Its levels are day, month, quarter, year, and season. The roll-up relation on Time is the reflexive and transitive closure of the sets of pairs (day, month), (month, quarter), (quarter, year), and (day, season). Thus, for instance, the level day rolls up to the level month, but also to the level year. Descriptions of the level day are date, day-of-week *(mapping each day to the name of the corresponding day),* day-number-in-month *(mapping each day to the number of the day within its month), and* day-number-overall *(coding days in consecutive day numbers).* ■

Let us now state precisely what is an instance of a dimension scheme.

Definition 3.2 [Dimension instance] *An instance of a dimension $D=(L, \Delta, \leq)$ consists of:*

- *a finite set of (real world) objects, each of which has a unique id associated with it, for each level l in L, called* members *of l;*

- *a function from the members of l to the domain of base type t associated with l, for each level description in Δ;*

- *a roll-up function $\textsc{roll-up}_{l_1 \to l_2}$ from the members of l_1 to the members of l_2, for each pair of levels l_1 and l_2 in L such that $l_1 \leq l_2$; if $m_2 = \textsc{roll-up}_{l_1 \to l_2}(m_1)$ we say that m_1* rolls up to *m_2.*

The roll-up functions of a dimension instance must satisfy the following *consistency conditions.*

Condition 3.1 [Consistency of roll-up] *The family of roll-up functions of a dimension are* consistent *if:*

1. *for each level l, the function $\textsc{roll-up}_{l \to l}$ is the identity on the members of l; and*

2. *if a level l_1 rolls up to l_2 in different ways (e.g., rolling up through either l' or l'') then the members of l_1 roll up to elements of l_2 in a consistent way, that is:*

$$\textsc{roll-up}_{l_1 \to l'}\,(\textsc{roll-up}_{l' \to l_2}\;(m)) = \textsc{roll-up}_{l_1 \to l''}\;(\textsc{roll-up}_{l'' \to l_2}\;(m))$$

   for each member m of l_1.

Note that, as is customary in conceptual models, a member of a dimension level is not a value but is the object itself (e.g., a member of the store level is the actual building, not its name and address). In fact, although this object has an id and a number of values (the descriptions) associated with it, its existence and identity are clearly independent of them.

We are now ready to introduce the general notion of *multidimensional database scheme.* This has two main components: a collection of dimensions and a number of *data cube schemes*, which are defined over levels of the dimensions.

Definition 3.3 [Multidimensional Scheme] *A* multidimensional scheme *consists of:*

- *a finite set D of dimension schemes;*

- *a finite set F of* data cube schemes *of the form:*

$$f\,[A_1 : l_1, \ldots , A_n : l_n] \to [M_1 : m , \ldots , M_k : m_k],$$

  where f is a name, each A_i ($1 \leq i \leq n$) is a distinct name called attribute *of f, each l_i is a level of D, each M_j ($1 \leq j \leq k$) is a distinct name called* measure *of f, and each m_j is either a base type or a level of D.*

Note that in MD there is a uniform treatment of measures and dimensions, as a measure can be not only a simple value but also a level of a dimension. This allows the analyst to transform measures into attributes and vice versa [9], an important functionality that any OLAP system should have [39].

Data cube schemes can also be naturally represented by means of diagrams. An example that refers to the dimensions in Figure 2 is given in Figure 3: facts are represented by boxes and measures by circles.

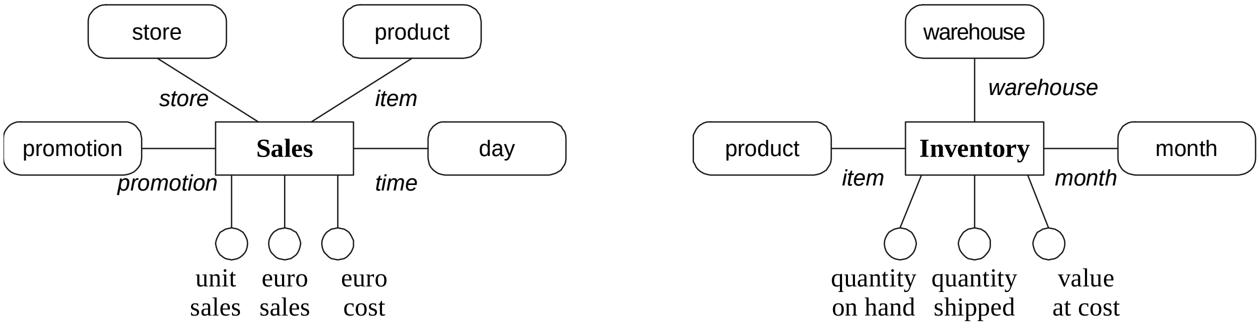

Figure 3: Two data cube schemes over the dimensions in Figure 2

Example 3.2 *A multidimensional scheme for the business processes of the Toys4All Company~~,~~ described in Example 2.1 and Example 2.2 can be defined using the dimension schemes of Example 3.1 . Specifically, two data cubes,* Sales *and* Inventory, *can be used to model the sale process and the warehouse process~~,~~ respectively. The schemes of these data cubes are represented-graphically in Figure 3.*

The data cube Sales *describes daily sales, detailed by item, store and promotion. Its attributes are* time *(at the day level of the time dimension, describing the day in which the sale occurred),* item *(the product sold),* store *(the store having sold the product), and* promotion *(the promotion applied to the sale). Its measures are* unit-sales *(the number of items sold),* euro-sales *(the income of the sale, in Euros), and* euro-cost *(the cost price of the items sold).*

The data cube Inventory *is instead used to represent the inventory levels of the various products, detailed by warehouse and month. Specifically, inventory levels are measured at the end of each month. The measures of this data cube are* quantity-on-hand *(the quantity in stock of a product at the end of the month),* quantity-shipped *(the quantity shipped from the warehouse during the month), and* value-at-cost *(the value of the quantity in stock, at cost price).* ■

Before introducing the notion of instance of a data cube scheme**,** two preliminary notions are needed.

Let $D = (D,F)$ be a multidimensional scheme, $f\,[A_1 : l_1,..., A_n : l_n] \rightarrow [M_1 : m ,..., M_k : m_k]$, be a data cube scheme in F and d be an instance of D.

Definition 3.4 [Conceptual coordinate] *A* (conceptual) coordinate *for f over d is a tuple over the attributes of f, that is, a function mapping each attribute A_i to a member of the level l_i occurring in d.*

Definition 3.5 [Fact] *A* fact *for f over d is a tuple over the measures of f, that is, a function mapping each measure name* M_j *to either a value (if* m_j *is a base type) or a member in d (if* m_j *is a level).*

We are now ready to introduce the notion of instance of a multidimensional scheme.

Definition 3.6 [Instance of multidimensional scheme] *An* instance *of a multidimensional database scheme (D,F) is composed of:*

- *a dimension instance d for each dimension scheme in D;*
- *a partial function called* data cube *mapping coordinates for f over d to facts for f over d, for each data cube scheme f in F.*

An *entry* of a data cube *c* is a coordinate over which the instance of *c* is defined.

SALES

time	*item*	*store*	*promotion*	*unit-sales*	*euro-sales*	*euro-cost*
d_{423}	p_{98}	s_{12}	pr_{111}	2	19.98	14.98
d_{423}	p_{41}	s_{12}	pr_{1}	3	44.94	28.20
d_{423}	p_{56}	s_{21}	pr_{111}	1	2.99	1.10
d_{424}	p_{98}	s_{12}	pr_{1}	1	11.99	7.49
...	...	...	...	...	...	...

INVENTORY

time	*item*	*warehouse*	*quantity-on-hand*	*quantity-shipped*	*value-at-cost*
m_{13}	p_{98}	w_{2}	100	60	749.00
m_{13}	p_{41}	w_{3}	80	100	752.00
m_{14}	p_{98}	w_{2}	50	70	374.50
...	...	...	...	...	...

Figure 4: A sample instance over the multidimensional scheme of Example 3.2

Example 3.3 *A possible instance for the multidimensional scheme of Example 3.2 is shown in Figure 4. In this example, level members are represented by their ids.*

A coordinate over the data cube scheme Sales is, for example,

$$[time : d_{423}, item : p_{98}, store : s_{12}, promotion : pr_{111}]$$

where d_{423} *is, for instance, the id associated with the physical item at hand.*

The actual instance associates with this entry the value 2 for the measure unit-sales, the value 19.98 for the measure euro-sales, and the value 14.98 for the measure euro-cost. ■

In Figure 4, data cubes are (graphically) represented as a table. This representation suggests how data cubes can be implemented using the relational model: a data cube over a scheme *f* can be represented by a relation over the attributes of *f*, with additional columns for the measures. The attributes of *f* form the key of the relation. In practice, a data cube having *n* attributes and *m* measures can also be represented by means of an *n*-dimensional array in which each (non null) entry corresponds to an entry of *f* and is associated with an *m*-tuple of measures. This representation recalls the way in which multidimensional systems usually store data, thus confirming that the MD is a conceptual model which describes multidimensional data independently of any specific (logical) implementation.

It is apparent that the notation we have used for coordinates resembles subscripting into a multi-dimensional array (although in a non-positional way). However, there is an important difference between data cubes and multi-dimensional arrays. Specifically, in arrays, “physical” coordinates vary over intervals within (linearly-ordered) domains of *values*, whereas domains over which coordinates range in the MD model are *conceptual entities*. In this sense, our notion of coordinate is “conceptual”.

Roll-up functions are a distinctive feature of the model proposed: they describe *intensionally* how members of different levels are related. This description is independent of any effective implementation: roll-up functions can be implemented by means of materialized relations, built-in functions, or external procedures. Moreover, roll-up functions provide a powerful tool for querying multidimensional data, as they can be used to specify how data can be grouped, and how data cubes involving data at different levels of granularity can be joined [7, 9].

3.2 Basic Features of a Multidimensional Model

The MD data model presented in the previous section exhibits those fundamental features that any multidimensional model should include in some form in order to be suitable for OLAP applications. According to Pedersen [39] and Blaschka et al. [6], these “mandatory” features can be summarized as follows.

- *Explicit separation of structure and contents*. This is indeed a basic requirement of database models that make a clear distinction between the *schema*, which describes the structure of data, and the *instances*, which correspond to the actual contents.

- *Explicit notions of dimension and data cube*. These are the basic concepts of multidimensional data representation, as we have discussed in Section 2.

- *Explicit hierarchies in dimensions*. A dimension should be structured into a hierarchy of levels to suggest the modalities in which data can be grouped along dimensions.

- *Multiple hierarchies in each dimension*. In one dimension, there can be more than one path along which to aggregate data. This is captured in MD by having a partial order relationship between the levels of a dimension.

- *Level attributes*. Other descriptive properties of the analysis dimensions**,** independent of the hierarchy relationship among levels**,** should also be representable. Level descriptions are used in MD for this purpose.

- *Measure~~s~~ sets*. This refers to the possibility of defining complex cell structures (grouping more than one measure) related to the same fact. In MD this is implemented by associating several measures to the same cube coordinate.

- *Symmetrical treatment of dimensions and measures*. The data model should allow measures to be treated as dimensions and vice versa. This is important because there are concepts (for instance, the age of customers) that can be measured (for instance, the average age of customers can be of interest) but which can also be used to group facts. This aspect is implemented in MD by allowing measures to be defined over dimension levels. This solution also makes it possible to register factual data at different granularities.

3.3 Advanced Features of a Multidimensional Model

There are a number of further advisable features that a conceptual multidimensional model should support. We have classified these features as "advanced" because they model concepts that~~:~~ either**:** (i) are difficult to represent in a simple way (such as the notion of **"**summarizability**"**), or (ii) serve to capture specific application cases. Adopting once more a terminology inherited from Pedersen [39] and Blaschka et al. [6], these basic features can be summarized as follows.

- *Support for aggregation semantics*. The data model should provide a support for the identification of aggregations whose result is *incorrect,* that is, meaningless to the user. This undesirable situation may occur for two main reasons.

  - A single fact can be counted more than once. Let us consider for instance the data cube Sales of our case study, whose scheme is described in Example 3.2 and reported in Figure 3. If we need the number of sales with respect to a specific media used for their promotion, we should only count a given sale once, even if several promotions have been applied to the sale.

  - Some types of aggregation along certain paths of a dimension can be meaningless for a specific type of measure. For example, it may not be meaningful to add inventory levels of different products together, but calculating their average may make sense. This concept is strictly related to the notion of *summarizability* studied in the context of statistical databases [32, 46], which defines when an aggregation, for instance~~,~~ total sales, can be calculated by directly combining results from lower-level aggregations, for instance, the sales for each store. This problem has been recently investigated by various authors [25, 31].

- *Support for non-standard aggregations of facts*. There are various possible cases.

  - *Non-strict hierarchies*. The hierarchy of levels in a dimension is non-strict if some of the mappings between the members of one level to the members of a higher level are many-to-many rather than one-to-many relationships. In our example, the Product dimension, described in Example 2.1 and represented in Figure 2, becomes non-strict if, for instance, a product can be classified according to different categories. The MD model can be extended to include non-strict hierarchies by assuming that the mappings $\mathit{ROLL\text{-}UP}_{l_1 \rightarrow l_2}$ are simple binary relations over members of levels l_1 and l_2 such that l_1 rolls up to l_2, rather than functions.

  - *Non-onto hierarchies*. A hierarchy in a dimension is "onto" if, for each member *m* of a level there is a member *m*' of a lower level (if any) such that *m*' rolls up to *m*. This property is not satisfied in our case study if, for example, there is a brand in an instance of the Product dimension (see Figure 2) with no associated product. In MD non-onto hierarchies are allowed as no restrictions are posed on the functions $\mathit{ROLL\text{-}UP}_{l_1 \rightarrow l_2}$, which can be therefore non-onto.

  - *Non-covering hierarchies*. A hierarchy in a dimension is non-covering if the member of a level rolls up to a member of a higher level in the hierarchy by "skipping" one or more intermediate levels. In the Toys4All example this may happen if, for example, in an instance of the Store dimension (see again Figure 2) there is a member of the Store level that rolls up to a member of the State level, without rolling up to any members of the City level. This would occur if the corresponding store is located not in a city but in a rural area. In MD non-covering hierarchies can be supported by allowing the roll-up functions to be *partial*.

  - *Many-to-many relationships between facts and dimensions*. It may happen that the relationship between a fact and its corresponding dimensions is not a many-to-one mapping. In our case study, it may be the case that a specific sale (a row in the fact cube reported in Figure 4) is actually associated with a combination of promotions rather than just one. This is not strictly forbidden in the model (new rows can be added for this purpose) but can lead to incorrect aggregations (see above). This problem can be solved in many cases with an appropriate instantiation of the dimensions [42].

- *Handling change and time*. Schemes and data change over time, and there may sometimes be an interest in performing analysis across changes. In our example, a category of products might be moved from one department to another and it is wished to analyze the impact of this change on the number of sales. The problem of the management of slowly changing dimensions [29] is related to this aspect. The maintenance of data cubes under dimension updates is also a relevant problem and has been recently investigated [26]. Temporal analysis can also be of interest**;** for instance, the variations in inventory levels over time. Approaches taken in temporal data models [52] could be applied to deal with these cases.

- *Handling imprecision*. Any real application must deal with the intrinsic problem of imprecision in representing and managing information. This problem has been widely studied in conceptual modeling. However, few studies have addressed this interesting and important problem in the context of multidimensional analysis, where imprecise data (for instance, the presence of missing values) can lead to incorrect results in calculating aggregations [16, 41]. A simple way to include a notion of imprecision in the measurement of facts in MD is to allow the presence of null values in data cubes. Conversely, incomplete knowledge of the dimension~~s~~ hierarchies can be taken into account by assuming that the roll-up functions are partial.

4. An overview of Multidimensional Data Models

In this section we briefly report on data models that have been proposed for multidimensional databases, in relation to the requirements reported in the previous section. A more thorough examination and comparison of many such models can be found in several survey papers appearing in the literature [6, 39, 44, 55]. General discussion on OLAP, multidimensional analysis, and data warehousing can be found in [11, 12, 13, 27, 48]. Mendelzon has published a rather comprehensive on-line bibliography on this subject [34]. Further up-to-date information can be found in specialized Web sites, for instance [21, 43].

It should be said that some of the models cited in this section cannot be classified as "conceptual" in the sense specified in Section 2. However, they are mentioned to provide a general overview of the state of the art in both the research community and commercial systems.

According to the classification proposed by Pedersen [39], data warehousing models can be divided into three main categories: *cube models*, *multidimensional models*, and *statistical models*. In the first category are simple models that provide ~~a~~ the notion of cube but in which the concept of dimension is modeled to only a limited extent. Conversely, multidimensional models allow representation of dimensions in structured (although different) ways. With the statistical model we finally denote the large body of work in the area of statistical database modeling, which is strictly related to the multidimensional approach [50].

4.1 Cube models

Simple cube models [14, 20, 22, 29] treat data in the form of *n*-dimensional cubes. They all have a more or less explicit notion of fact, measure and dimension. However, the hierarchy between the various levels of aggregation in a dimension is not explicitly captured by the schema, so the user cannot infer from the schema that, for instance, City rolls up to State and not the opposite. The *star schema* approach [29] (and its variants, like the *snowflake* scheme), in which a central relational table represents the fact on which the analysis is focused, and a number of tables, usually de-normalized, represent the dimensions of analysis, should also be considered a cube model as it is semantically equivalent to these, although at a lower level of abstraction.

The majority of models adopted by commercial systems [43, 38] should also be included in this category. Modelling aspects are covered by commercial systems in a pragmatic way. The representation used in ROLAP (*Relational* OLAP) systems is the star schema [29], whose limit in representing the multidimensional aspects of OLAP applications at the right level of abstraction has already been discussed. In MOLAP (*Multidimensional* OLAP) systems [13]**,** information is represented directly in multidimensional form, but the structure of a dimension is usually hard coded in the physical index structures used to access data.

4.2 Multidimensional models

Multidimensional models [3, 7, 8, 16, 17, 28, 30, 33, 35, 36, 37, 40, 54] capture the hierarchies in the dimensions explicitly, providing a better understanding of the application and a support for easy data cube manipulation. This information may also be useful for query formulation and optimization.

Interestingly, while the basic features are more or less covered by these models, each of them represents the dimension structure very differently; e.g., by using grouping relations [33], dimension merging functions [3], measure graphs [16], roll-up functions [8, 35], level lattices [54], hierarchy schemes and instances [28], or an explicit tree-structured hierarchy as part of the cube [30, 36].

A number of data models have also been defined by extending traditional conceptual data models [49]. Others have used known paradigms (e.g.~~,~~ object-orientation [1] and nested structure models [15]) or specific metaphors (e.g.~~,~~ tapes [18]). Finally, several data models have been proposed with the main goal of studying specific data warehousing application problems, such as incomplete information [16, 41], efficiency issues [24, 28], heterogeneous dimensions [25], dimension updates [26] and temporal OLAP queries [35], and so are well suited for them.

4.3 Statistical models

The last group is statistical database models [5, 44, 45, 46, 50, 53]. A great deal of relevant work has already been done in this area. Shoshani [50] made a very interesting comparison of work done in statistical and multidimensional databases. This revealed that after taking apart the terminology used, the two areas have a lot of overlap, even if each of them has emphasized different aspects. In particular, research in statistical databases has focused on the treatment of complex classification structures, management of certain special dimensions (e.g., spatial and geographic), and on the important issues (especially from the statistical point of view) of privacy and summarizability. On the other hand, OLAP literature has emphasized data warehouse design, query processing and, above all, efficiency issues. It is however clear that, though the emphasis is on different aspects, work done in one area can greatly benefit the other [50].

A statistical data model is usually based on the notions of *summary table*, *summary attribute* and *category attribute*. Actually, there is a close correspondence between these notions and the concepts used in multidimensional data models. Specifically, a summary table corresponds essentially to a data cube, a summary attribute to a measure, and a category attribute to a dimension. As in multidimensional models, a category attribute is always associated with a hierarchy of concepts. A number of operators are usually introduced in statistical models to manipulate, concatenate and aggregate summary tables.

Notable examples of conceptual statistical models are *STORM* [46] and *Mefisto* [45]. In particular, Mefisto introduces the important notion of *statistical entity,* the conceptual counterpart of the notion of summary table.

In statistical models, a structured classification hierarchy is almost always coupled with an explicit aggregation function on a single measure to produce a sort of pre-defined object capable of answering a specific set of queries. This approach is sometimes less flexible than the approaches usually taken by multidimensional models, but unlike most of these, it can provide an effective way to avoid incorrect results from queries.

5. Future Trends and Conclusions

In this chapter, we have discussed the requirements that an ideal conceptual multidimensional model should fulfill. These are suggested by general information system modeling principles and by the specific characteristics of OLAP applications. Starting from these requirements, we have presented a simple conceptual multidimensional data model, called MD, which can be used to describe the basic aspects of a business application in a way that is easy to understand and ~~is~~ independent of the criteria for actual data organisation in the various systems. With this model, we have tried to capture both the conceptual means used in business applications to describe information and the core of the various multidimensional data models proposed in the scientific literature or adopted by commercial systems. The model relies on two principal, agreed concepts: the *dimension* and the *data cube*. A dimension represents a business perspective under which data analysis is to be performed and is organized in a hierarchy of *levels*. The levels of a dimension correspond to different ways of grouping dimension~~s~~ members. A *data cube* represents the factual data on which the analysis is focused and associates *measures* with *coordinates*, defined over a set of dimension levels. Using these concepts as a reference, we have summarized the general features that a multidimensional conceptual model should support and mentioned the various multidimensional models which have been proposed.

Clearly, much work remains to be done in this area. First of all, the use of conceptual data models has still difficulties to overcome in the applicative area and the research community should clearly demonstrate the benefits ~~in~~ to be gained by adopting them. Moreover, with such a proliferation of data models, a commonly accepted formalism is strongly advisable. This is fundamental for support of interoperability and standardization. Another problem that still needs to be solved is the definition of an effective and general methodology for the development of OLAP applications, an important aspect which has received little attention [8, 19, 29]. This would also lead to the development of CASE tools which, in contrast to the present situation, were not strictly related to a specific OLAP system. Devising a common standard declarative language is also of high importance and the use of a conceptual multidimensional model (independent of the underlying physical model) could give useful results in the area of logical optimization and caching rules (in order to exploit the possibility of reusing existing data cubes for the computation of new ones). Finally, there are a number of specific problems, such as the characterization of summarizability, for which a definitive solution has not yet been given.

References

[1] A. Abello, J. Samos, and F. Saltor. Benefits of an Object-Oriented Multidimensional Data Model. In *14th European Conference on Object-Oriented Programming (ECOOP 2000), Lecture Notes in Computer Science 1944, Springer-Verlag*, pages 141–152, 2000.

[2] S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In *22nd Int. Conf. on Very Large Data Bases, Bombay,* pages 506–521, 1996.

[3] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In *13th Int. Conf. on Data Engineering*, pages 232–243, 1997.

[4] C. Batini, S. Ceri, and S. Navathe. *Conceptual Database Design: an Entity-Relationship Approach*. Benjamin/Cummings, 1992.

[5] A. Bezenchek, M. Rafanelli, and L. Tininini A Data Structure for Representing Aggregate Data. In *8th Int. Conference on Scientific and Statistical Database Management (SSDBM'96),* IEEE Computer Society Press, pages 22–31, 1996.

[6] M. Blaschka, C. Sapia, G. Höfling, and B. Dinter. Finding your way through multidimensional data models. In *9th Int. Conf. on Database and Expert Systems Applications (DEXA), Lecture Notes in Computer Science 1460, Springer-Verlag*, pages 198–203, 1998.

[7] L. Cabibbo and R. Torlone. Querying multidimensional databases. In *6th Int. Workshop on Database Programming Languages (DBPL'97),* 1997.

[8] L. Cabibbo and R. Torlone. A logical approach to multidimensional databases. In *6th Int. Conference on Extending Database Technology (EDBT'98), Springer-Verlag*, pages 183–197, 1998.

[9] L. Cabibbo and R. Torlone. From a procedural to a visual query language for OLAP. In *10th Int. Conference on Scientific and Statistical Database Management (SSDBM'98),* IEEE Computer Society Press, pages 74–83, 1998.

[10] D. Chatziantoniou and K. Ross. Querying multiple features of groups in relational databases. In *22nd Int. Conf. on Very Large Data Bases, Bombay*, pages 295–306, 1996.

[11] S. Chaudhuri and U. Dayal. An overview of Data Warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1): 65–74, March 1997.

[12] E. F. Codd, S. B. Codd, and C. T. Salley. Providing OLAP (On Line Analytical Processing) to user-analysts: an IT mandate. Arbor Software White Paper, *http: //www.arborsoft.com*.

[13] G. Colliat. OLAP, relational, and multidimensional database systems. *ACM SIGMOD Record*, 25(3): 64–69, September 1996.

[14] A. Datta and H. Thomas. A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases. In *Int. Seventh Annual Workshop on Information Technologies and Systems (WITS 1997)*, pages 91–100, 1997.

[15] S. Dekeyser, B. Kuijpers, J. Paredaens, and J. Wijsen. Nested data cubes for OLAP. In *Int. Workshop on Data Warehousing & Data Mining, Singapore,* pages 129–140, 1998.

[16] C. E. Dyreson Information Retrieval from an Incomplete Data Cube. In *22nd Int. Conf. on Very Large Data Bases, Bombay,* pages 532–543, 1996.

[17] E. Franconi and U Sattler. A data warehouse conceptual data model for multidimensional aggregation. In *Int. Workshop on Design and Management of Data Warehauses (DMDW),* 1999.

[18] M. Gebhardt, M. Jarke, and S. Jacobs. A toolkit for negotiation support interfaces to multi-dimensional data. In *ACM SIGMOD Int. Conf. on Manag. of Data*, pages 348–356, 1997.

[19] M. Golfarelli, D. Maio, and S. Rizzi. Conceptual design of data warehouses from E/R schemes. In *31st Hawaii Intl. Conf. on System Sciences*, 1998.

[20] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data Cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In *12th IEEE International Conference on Data Engineering, Vienna*, pages 152–159, 1996.

[21] L. Greenfield. Data Warehousing Information Center. *http: //www.dwinfocenter.org/*.

[22] M. Gyssens and L.V.S. Lakshmanan. A foundation for multi-dimensional databases. In *23rd Int. Conf. on Very Large Data Bases*, pages 106–115, 1997.

[23] R. Hull. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. In *16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems*, pages 51–61, 1997.

[24] V. Harinarayan, A. Rajaraman, and J. Ullman. Implementing data cubes efficiently. In *ACM SIGMOD International Conf. on Management of Data,* pages 205–216, 1996.

[25] C. Hurtado and A. Mendelzon. Reasoning about Summarizability in Heterogeneous Multidimensional Schemas. In *8th International Conference on Database Theory (ICDT), Lecture Notes in Computer Science 1973, Springer-Verlag*, pages 375–389, 2001.

[26] C. Hurtado, A. Mendelzon, and A. Vaisman. Maintaining data cubes under dimension updates. In *15th Int. Conf. on Data Engineering*, pages 346–355, 1999.

[27] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 2nd ed., 1996.

[28] H. V. Jagadish, L. V. S. Lakshmanan, and D. Srivastava What can Hierarchies do for Data Warehouses? In *25nd Int. Conf. on Very Large Data Bases, Bombay*, pages 530–541, 1999.

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

[30] W. Lehner. Modeling large scale OLAP scenarios. In *6th International Conference on Extending Database Technology (EDBT), Lecture Notes in Computer Science 1377, Springer-Verlag*, pages 153–167, 1998.

[31] W. Lehner, J. Albrecht, and H. Wedekind. Normal Forms for Multidimensional Databases. In *10th Int. Conference on Scientific and Statistical Database Management (SSDBM'98)*, IEEE Computer Society Press, pages 63–72, 1998.

[32] H. J. Lenz and A. Shoshani. Summarizability in OLAP and statistical data bases. In *9th Int. Conf. on Scientific and Statistical Database Management (SSDBM)*, pages 132–143, 1997.

[33] C. Li and X. S. Wang. A data model for supporting on-line analytical processing. In *Proceedings of Conference on Information and Knowledge Management*, pages 81–88, 1996.

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

[35] A. O. Mendelzon and A. A. Vaisman Temporal Queries in OLAP. In *26nd Int. Conf. on Very Large Data Bases, Cairo, Egypt*, pages 242–253, 2000.

[36] Microsoft Corporation. OLE DB for OLAP 2.0. Microsoft Technical Document, 2000.

[37] T. B. Nguyen, A. M. Tjoa, and R. Wagner. Conceptual Multidimensional Data Model Based on MetaCube In *1st International Conference on Advances in Information Systems (ADVIS)*, pages 24–33, 2000.

[38] Oracle Corporation. Oracle OLAP products: adding value to a data warehouse. Oracle Technical Document, 1998.

[39] T. B. Pedersen. Aspects of Data Modeling and Query Processing for Complex Multidimensional Data. *PhD thesis, Faculty of Engineering & Science, Aalborg University*, 2000.

[40] T. B. Pedersen and C. S. Jensen. Multidimensional Data Modeling for Complex Data. In *15th International Conference on Data Engineering (ICDE)*, IEEE Computer Society Press, pages 336–345, 1999.

[41] T. B. Pedersen, C. S. Jensen, and C. E. Dyreson. Supporting Imprecision in Multidimensional Databases Using Granularities. In *11th Int. Conference on Scientific and Statistical Database Management (SSDBM'99)*, IEEE Computer Society Press, pages 90–101, 1999.

[42] T. B. Pedersen, C. S. Jensen, and C. E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Information Systems*, 26(5): 383–423, 2001.

[43] N. Pendse and R. Creeth. The OLAP Report. *http: //www.olapreport.com*.

[44] M. Rafanelli. Aggregate statistical data: models for their representation. *Statistics and Computing*, 5(1): 3–24, 1995.

[45] M. Rafanelli and F.L. Ricci. Mefisto: A Functional Model for Statistical Entities. *IEEE Transactions on Knowledge and Data Engineering*, 5(4): 670–681, 1993.

[46] M. Rafanelli and A. Shoshani. STORM: A statistical object representation model. In *5th Int. Conf. on Scientific and Statistical Database Management (SSDBM), Lecture Notes in Computer Science 420, Springer-Verlag*, pages 14–29, 1990.

[47] S. Rao, A. Badia, and D. Van Gucht. Providing better support for a class of decision support queries. In *ACM SIGMOD International Conf. on Management of Data*, pages 217–227, 1996.

[48] J. Samos, F. Saltor, J. Sistac, and A. Bardés. Database architecture for data warehousing: an evolutionary approach. In *9th Int. Conf. on Database and Expert Systems Applications (DEXA), Lecture Notes in Computer Science 1460, Springer-Verlag*, pages 746–756, 1998.

[49] C. Sapia, M. Blaschka, G. Höfling, and B. Dinter. Extending the E/R Model for the Multidimensional Paradigm. In *Advances in Database Technologies, ER'98 Workshops, Lecture Notes in Computer Science 1552*, pages 105–116, 1998.

[50] A. Shoshani. OLAP and statistical databases: similarities and differences. In *16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems*, pages 185–196, 1997.

[51] Stanford Technology Group, Inc. Designing the data warehouse on relational databases, 1995. Unpublished manuscript.

[52] A. U. Tansel, J. Clifford, S. K. Gadia, S. Jajodia, A. Segev, and R. T. Snodgrass. *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings, 1993.

[53] L. Tininini, A. Bezenchek, and M. Rafanelli. A System for the Management of Aggregate Data. In *7th Int. Conf. on Database and Expert Systems Applications (DEXA), Lecture Notes in Computer Science 1134, Springer-Verlag*, pages 531–543, 1996.

[54] P. Vassiliadis. Modeling multidimensional databases, cubes and cube operations. In *10th Int. Conference on Scientific and Statistical Database Management (SSDBM'98)*, IEEE Computer Society Press, pages 53–62, 1998.

[55] P. Vassiliadis and T. K. Sellis. A Survey of Logical Models for OLAP Databases. *SIGMOD Record*, 28(4): 64–69, 1999.