A Data Warehouse Multidimensional Data Models Classification

Alberto Abelló [(a)], José Samos [(b)], and Fèlix Saltor [(a)]

[(a)] U. Politècnica de Catalunya (UPC), Dept. de Llenguatges i Sistemes Informàtics
{aabello,saltor}@lsi.upc.es
[(b)] U. de Granada (UGR), Dept. de Lenguajes y Sistemas Informáticos
jsamos@ugr.es

Abstract. The words On-Line Analytical Processing (OLAP) bring together a set of tools, that use multidimensional modeling in the extraction of information from the Data Warehouse. Lately, a lot of work has been devoted to modeling the multidimensional space. The aim of this paper is twofold. On one hand, it compiles and classifies most of that work. On the other hand, it allows to compare the different terminology used by each author, by placing all the terms in a common framework.

Key Words: Data Warehouse, On-Line Analytical Processing, Multidimensional data modeling

1 Introduction

In the last years, multidimensional modeling has gained attention by the research community. It is a powerful conceptualization technique used in On-Line Analytical Processing (OLAP) applications. As explained in [CD97], OLAP tools, by means of multidimensional modeling, facilitate complex analyses and visualization of the data in the Data Warehouse for decision making processes. Its main advantages are that it is close to the analysts way of thinking, and facilitates improvements in query performance.

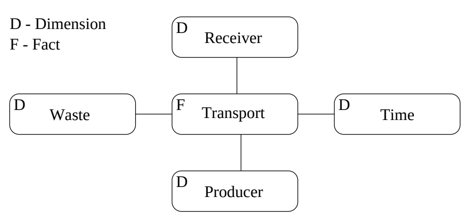

Fig. 1. Multidimensional scheme example

This technique is based on the concept of a hypercube (we will misuse the term "cube" from here on) containing the data cells of interest to the analysis.

Every one of these data cells is identified by a coordinate in each analysis dimension. For instance, as depicted in figure 1, if we are talking about a waste transport business, we could be interested in analyzing *Transports* involving the transported *Waste*, the *Time* the transport takes place, the *Producer* it is transported from, and the *Receiver* it is transported to. Therefore, we would have a 4-dimensional space where each point represents transport data, and is identified by a waste, a point in time, a waste producer, and a waste receiver.

As defined in [EN94], a *data model* is a set of concepts that can be used to describe the structure of a database. In the same book, we find a categorization of *data models* into *High-level or conceptual*, if they provide concepts that are close to the way users perceive data; *Low-level or physical*, if provide concepts that describe the details of how data is stored in the computer; and *Implementation*, if provide concepts that can be understood by end users, but that are not too far removed from the way data is organized within the computer.

Also [BCN92] describe those three levels. Adopting their terminology, we distinguish three different kinds of data models based on the constructs/concepts they provide and the data warehouse design phase they help: Those at *Conceptual* level that are close to the user and independent of the implementation; those at *Logical* level depending on the kind of Database Management System (DBMS) used in the implementation, but still understandable by end users; and finally, those at *Physical* level depending on the specific DBMS used, and conceived to describe how data is actually stored.

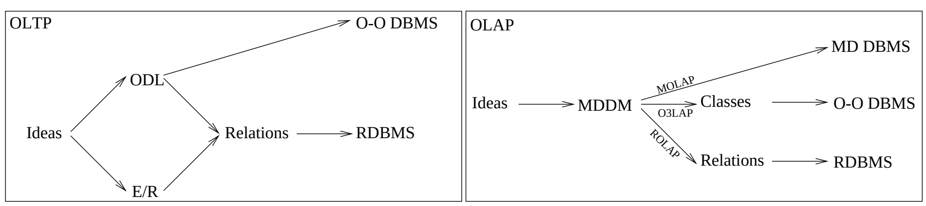

Fig. 2. Modeling and implementation process in OLAP vs OLTP environments

As shown in figure 2 (left), from [UW97], in an On-Line Transactional Processing (OLTP) environment, during the first design step, at conceptual level, we would use Entity-Relationship (E/R) or Object Definition Language (ODL) to represent user ideas; in the next step, at logical level, we would use to use the Relational model, but we could also use Hierarchical, or Network models (not depicted in the figure); and in the last step, at the physical level, the implementation would depend on a specific DBMS (i.e. Oracle, Informix, ObjectStore, etc.). In a similar way, in an OLAP environment, in figure 2 (right), we would have the Multidimensional Data Model (MDDM) at conceptual level, and, depending on the approach (i.e. Relational -ROLAP-, Object-Oriented -O3LAP-, or pure Multidimensional -MOLAP-), we would use a different model at logical level, and a different DBMS for the implementation.

In this paper we want to classify the huge amount of efforts in the area, devoted to modeling the cube. We will divide them into four groups based on the constructs they provide. Besides these three mentioned above, we found another set of models (let us say they are at *Formal* level) whose concepts would not be used at any database design phase, but on giving a theoretical framework, and include an algebra or calculus. In an OLTP environment, a formal model would be the Relational Algebra.

This paper is structured as follows. Section 2 briefly reviews previous work on multidimensional models. Section 3 defines the framework that will allow us to classify the different constructs of each model, in order to be able to relate them. The core of the paper, section 4, contains a short description for each multidimensional model and its constructs. These models are classified in four different subsections based on the modeling level that we think they are in. Finally, section 5 presents some conclusions and a summary table with all the models presented in this paper.

2 Related work

In [BSHD98], a list of requirements for a formal model in order to be suitable for OLAP applications, is used to analyze the seven models found in [AGS97], [LW96], [GL97], [CT97], [Vas98], [Leh98], and [DT97], which are chosen because they contain some kind of formalism. Those requirements (derived from general design principles, and from characteristics of OLAP applications) are the following:

- Explicit separation of cube structure and its contents
- Complex dimensions
  - Level structure
  - Member (i.e. level instance) structure
  - Formalism (mathematical construct) for level structure
  - Dimension attributes (those not defining hierarchies)
- Symmetry of measures and dimension members
- Complex measures
  - Support of structured measures
  - Support of derived measures
  - Additivity of measures
- Query formalism
  - Type of formalism (i.e. algebra or calculus)
  - Ad-hoc hierarchies
  - User defined aggregates

[Ped00] (see [PJ98], and [PJ99] for preliminary work) presents eleven requirements (found in clinical data warehousing) for multidimensional data models, and evaluates twelve preexisting data models against them. Those presented in [AGS97], [Dyr96], [GBLP96], [Kim96], [LW96], [GL97], [CT98b], [DT97], [Leh98], and [Vas98] are among those twelve. An statistical model, and a commercial system are also included. Moreover, it presents a data model which does address all those requirements. The requirements are:

1. Explicit hierarchies in dimensions
2. Symmetric treatment of dimensions and measures
3. Multiple hierarchies in each dimension (different aggregation paths)
4. Support for aggregation semantics (applicability of aggregation functions)
5. Non-strict hierarchies (overlapping classifications)
6. Non-onto hierarchies (non-balanced instances trees)
7. Non-covering hierarchies
8. Many-to-many relationships between facts and dimensions
9. Handling change and time
10. Handling different levels of granularity
11. Handling uncertainty

[Vas00] (see [VS99] for a stand-alone version) gives yet another classification of multidimensional models. It pays attention to [GBLP96], [LW96], [GL97], [GL98], [BPT97], [AGS97], [CT98b], [Leh98], [GJJ97], besides [Vas98], some industrial standards, and a couple of statistical models. The requirements studied in this case are:

- Representation of the multidimensional space
  - Cubes/Tables
  - Explicit/Implicit hierarchies
- Language issues
  - Character of the query language (Procedural/Declarative/Visual)
  - Support of sequences of operations
  - Naturality of OLAP operations modeled
- Mappings offered to
  - Relations
  - Multidimensional arrays

In our approach, we take into account models that were not considered by any of these previous classifications, besides most of those already taken under consideration. Moreover, as explained in previous section, we group those models in four different sets based on the multidimensional database design phase they are conceived for. Some of the publications considered in these three classifications are not included in this one, because they do not fit at any of those sets (i.e. papers about statistical models, or papers whose main subject, despite being devoted to multidimensionality or data warehousing, is not multidimensional modeling).

3 Description framework

Each one of the papers described in section 2, begins by defining a list of specific requirements for a multidimensional model in order to evaluate all those models already existing. We are not giving such a list, since we do not want to evaluate the models, but just describe them and keep track of their expressiveness in a common framework.

Each one of the models compiled in this paper uses its own terminology and defines a specific set of design elements. In this sense, we are going to use different detail levels to classify the constructs of the models at the corresponding level, in order to be able to compare them, and examine the expressive power of every model at each one of these levels. For every model, we are going to enumerate the different elements and nexus between them that are provided at each one of the detail levels.

In a multidimensional model, conceptually, we distinguish three different detail levels:

Upper Level (UL): At this level, we find *Dimensions* (D) and *Facts* (F). The dimensions are used to characterize the facts, and show the viewpoints the facts will be analyzed from. By relating a set of dimensions to a fact, we obtain a star shape scheme. The possibility of navigating from a star to another one is shown by their share of dimensions.

Intermediate Level (IL): Dimensions and Facts are decomposed into *Dimension Levels* (DL), and *Fact Cells* (FC) respectively. The different dimension levels in a dimension form a hierarchy. Each FC contains data (a set of measures) at a given DL for each dimension its fact is related to.

Lower Level (LL): The last level shows the attributes of the DL and FC. That is *Classification Attributes* (CA), and *Measures* (M) respectively.

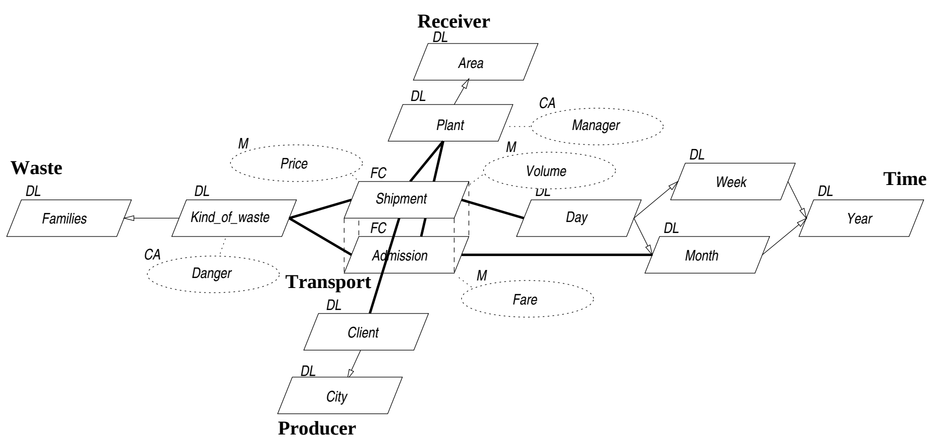

Fig. 3. Further detailed multidimensional scheme example

Going back to our example in figure 1, which represents a multidimensional scheme at UL, as composed by a Fact (*Transport*), and four Dimensions (*Waste*, *Producer*, *Receiver*, and *Time*); figure 3 represents the same multidimensional scheme at lower detail levels. Each one of the Dimensions is further described by a hierarchy of different Dimension Levels. For instance, *Time* Dimension contains *Day*, *Week*, *Month*, and *Year* Dimension Levels. Moreover, if we were interested in the benefits of a transport, we would need to analyze data that would belong to different kinds of cube cells (price of the shipment that we charge to our client, minus admission fare that a processing plant charges to us). On one hand, we can see a *Shipment* Fact Cell containing data about our shipments which depends on the lower level of each one of the four dimensions. On the other hand, data about *Admission* of *Waste* in a *Plant* do not depend on our clients, nor on *Day* level of *Time* dimension (it depends on *Month* level). Therefore, *Admission* and *Shipment* are different kinds of Fact Cells, but belong to the same Fact we want to analyze. Finally, drawn with dotted lines, we can see constructs at LL. Some Dimension Levels have Classification Attributes associated (for instance, a *Plant* has a *Manager*). Besides, Fact Cells have associated Measures (for instance, an *Admission* has a *Fare*) that we want to analyze.

4 The models

This section contains a set of multidimensional data models classified in four different subsections (plus a kind of miscellaneous section). Each one of these four subsections contains proposals at a given design level (i.e. Conceptual, Logical, Physical, and Formal). Inside the subsections, models are chronologically ordered by year. The last subsection contains contributions to multidimensional modeling that were not classified at any of the previous ones.

At the beginning of each one of those subsections, there is a table showing the constructs of each model at the corresponding level with regard to our description framework. As pointed out by [BSHD98], some multidimensional models do not separate cube structure and contents. Only those concepts represented at schema level are considered (relationships among instances are not considered). A tick means something is captured by the model, while a hyphen means that the authors of the model either say something not to be modeled or just do not say how to model it.

A hyphen in the column corresponding to:

Measure (M) means that nothing can be represented in the scheme about measures. Maybe, there are only pure numerical values stored in the cube (without any meaningful domain).

Classification Attribute (CA) means dimensions do not have attributes describing their different entities. All the information is kept in the form of classification hierarchies at the most.

Relationships at LL means there is not any way in the model to represent relationships among Measures and/or Classification Attributes.

Dimension Level (DL) means there are not explicit levels in the dimensions.

Fact Cell (FC) means that either the Measures are not grouped, or they are not related to a specific set of Dimension Levels, but to dimensions as a whole (usually reflected as relating the measures to the lowest level in the classification hierarchy).

Relationships at IL means there is not any way in the model to represent relationships among Fact Cells and/or Dimension Levels. For instance, it implies that there is not the possibility of explicit dimension hierarchies.

Fact (F) means that the different Fact Cells can not be grouped with the intention to relate Measures that, even though are defined at different granularities, are used together in a given decision making process.

Dimension (D) means that either there is only the possibility of modeling one Dimension Level, or if it is possible to model more than one, they cannot be grouped into another element of the model.

Relationships at UL means there is not any way in the model to represent relationships among Fact Cell and/or Dimension Levels.

4.1 At conceptual level

In this section we place those models that contain concepts which are closer to the user than to the actual computer implementation. They try to represent how users perceive a multidimensional cube without paying special attention to formalisms.

At this level, we found the following models: [Leh98] (section 4.1), [CT98b] (section 4.1), [GMR98b] (section 4.1), [TP98] (section 4.1), [SBHD99] (section 4.1), [SCdMM99] (section 4.1), [TBC99] (section 4.1), [NTW00] (section 4.1), and [HLV00] (section 4.1).

Lehner (Nested Multidimensional Data Model) The model in [Leh98] emphasizes the presentation of data at two different ("Nested") levels, and the operations offered to the user in order to accomplish this (i.e. *slicing*, *drill-down*, *roll-up*, *split*, *merge*, *aggregation*, and other cell-oriented operators like *max*, *min*, +, etc.). The existence of two levels is said to improve the power and flexibility of the whole analysis process.

One of the most interesting features of NMDM (besides the existence of two levels) is the way it qualifies dimension instances by means of different sets of attributes. Thus, at the bottom of every classification hierarchy is placed a *primary attribute* (PA) whose instances are called *dimensional elements*. Those *dimensional elements* are the leaf nodes of a balanced tree-structured *classification hierarchy*. Each tree level is called *classification attribute* (CA), whose instances are *classification nodes* (CNs). *dimensional attributes* (DA) are associated to every CN (notice that CNs are instances in the hierarchy).

A *Primary Multidimensional Object* (PMO) consist of an unique cell identifier; a set of CAs and PAs (one per dimension), denoting the granularity of the cell; a set containing one instance per CA/PA, specifying the selection criteria; an aggregation type, describing the aggregation operations applicable; and a data type (i.e. domains over N, Z, or R). In turn, a *Secondary Multidimensional Object* (SMO) consist of a set of CNs; and a set of DAs applicable to them. Thus, a *Multidimensional Object* is a PMO, and a set of DAs for defining the corresponding nested SMOs.

Sect.	F	D	Upper Level Relationships	FC	DL	Intermediate Level Relationships	M	CA	Lower Level Relationships
4.1	–	✓	–	–	✓	Linear hierarchy of DLs $DLs \in cube$	–[1]	✓	CAs associated to DL instances
4.1	✓	✓	F *and* $Ds \in cube$ F is a Mset	–[2]	✓	DLs form a roll-up hierarchy (partial order)	✓	✓	$CA \in DL$ $f : DL^n \rightarrow M$
4.1	–	✓	–	✓	✓	Ds *and* $FC \in cube$ Aggregation hierarchy of DLs Compatibility between FCs	✓	✓	*-to-one* between CA and DL Aggregability between M and D
4.1	✓	✓	Part-whole between F and Ds	–	✓	Classification hierarchy of DLs	✓[3]	✓	Aggregability between M and D
4.1	✓	✓[4]	FCs in F share DLs in some D	✓	✓	*rolls-up to* between DLs *fact* relates n DLs	✓	✓	$M \in FC$ $CA \in DL$
4.1	✓	✓	FCs in F use Ds in F	✓	✓	Aggregation functions between DLs	✓	✓	$M \in FC$ $CA \in DL$
4.1	–	✓	–	✓	✓	Membership DLs hierarchy Aggregation/Specialization between DLs Allows M:N between FC and DL	✓	✓	$M \in FC$ $CA \in DL$
4.1	✓	✓	Ds *and* $FCs \in cube$	✓	✓	Aggregability of FCs Partially ordered set of DLs *groupby* relates FCs and DLs	–[1]	–	–
4.1	–	✓	$Ds \in cube$	✓	✓	$f : DLs \rightarrow FC$ Aggregation path between DLs	✓	✓	$CA \in DL$ $M \in FC$ Aggregability between M and DL

[1] Only domains over N, Z, and R are allowed.
[2] Even though Ms are related to DL, Ms are not grouped into FC.
[3] Possibly derived.
[4] Implicit within the structure of the *rolls-up to* graph.

Table 1. Scheme constructs in the different models at conceptual level

All the schema constructs in this model refer to the dimensions. They are defined as a linear hierarchy of Dimension Levels (called *classification attributes*) at IL, and the instances of each Dimension Level have associated Classification Attributes (called *classification nodes*) at LL.

Cabibbo and Torlone (MD) Cabibbo and Torlone, in [CT98a], [CT98b], and [CT97], qualify their model *MD* as "logical". However, they say that it is independent of any specific implementation, and present a design methodology to obtain an *MD* scheme from an E/R one. Moreover, it is also said that *MD* is at a higher level of abstraction than a star scheme consisting of relational tables. Therefore, we classify it as conceptual, even though it provides a strong formal foundation (including a calculus).

The main constructs in the model are *dimension* and *f-table*. Each *dimension* is organized in a hierarchy of *levels* corresponding to data domains at different granularity. In turn, a *level* can have *descriptors* associated with it. The *f-tables* are functions from *levels* to *measures*.

We can clearly identify the data about dimensions at the three different levels: *dimension* at UL, *levels* at IL, and *descriptors* at LL. About facts, there are *measures* at LL, and a set of *f-tables* at UL. However, *measures* are not grouped regarding the levels they are defined at.

Golfarelli, Maio, and Rizzi (Dimensional Fact Model) [GR98], [GMR98a], [GMR98b], and [GR99] present a graphical conceptual model (DFM) for data warehousing, besides a methodology to obtain a multidimensional scheme from the operational schemes (either E/R or Relational).

Contrary to what is said for some formal models, the authors claim that it is important to clearly distinguish between dimensions and measures. Thus, a *dimensional scheme* consist of a set of *fact schemes*, and each one of these contains a *fact*, *measures*, *dimensions*, and *hierarchies*. A *fact* is a focus of interest, and its attributes are *measures*. The *dimensions* are discrete attributes which determine the minimum level of granularity chosen to represent the *fact*. Finally, a *hierarchy* is a set of dimension attributes linked by *-to-one* relationships (i.e. 1:1, or N:1), which form a *quasi-tree*. Hierarchies may also include *non-dimension attributes* that contain additional information which can not be used for aggregation (but just for selection). Moreover, aggregability can also be expressed by relationships between a *measure* and a *dimension*, tagged by the allowed aggregation functions.

A special relation between two schemes is also defined (called *compatibility* and *strict compatibility*), which indicates and restricts when a query can be formulated including measures in both schemes. Roughly, two schemes are *compatible* when they have, at least, one common *dimension attribute*.

Placing those constructs in our framework, we obtain Classification Attributes (called *non-dimension attributes*) and Measures at LL, grouped respectively into Dimension Levels (called *dimension attributes*) and Fact Cells. The Dimension Levels form dimension hierarchies. Furthermore, Fact Cells are related by compatibility, and Measures and Dimensions by aggregability.

Trujillo, Palomar, and Gómez (GOLD) [TP98], and [TPG00] describe an Object-Oriented conceptual model based on a subset of UML. A query notation is also presented.

A *fact* (represented as a basic class) is described through a set of *fact attributes* (either atomic or derived) representing measures. By mean of whole-part relationships, a *fact* is related to a set of *dimensions* (also represented as basic classes) that show the granularity adopted for representing facts. Those *dimensions* are also described by *dimension attributes*. A *classification hierarchies* is defined as a Directed Acyclic Graph of *level* classes, rooted in the dimension class. Multiple classification hierarchies are allowed; and strictness, and complementariness represented. Aggregability of measures along each dimension can be represented, as well as derived measures.

In this model, information at LL is represented in the form of Measures (called *fact attributes*), and Classification Attributes (called *dimension attributes*). The former are attributes of a Fact at UL, while the later are attributes of a Dimension Level at IL. A Dimension at UL is defined as a classification hierarchy of Dimension Levels.

Sapia, Blaschka, Höfling, and Dinter (Multidimensional Entity Relationship Model) [SBHD99] argues that the E/R model is not suited for multidimensional conceptual modeling. Thus, a specialization is defined, and its usage exemplified.

The design of this model was driven by the following ideas:

- Specialization of the E/R model
- Minimal extension of the E/R model
- Representation of the multidimensional semantics

Following those guidelines, these specializations are introduced:

- A special entity set: *dimension level*
- Two special relationship sets connecting *dimension levels*:
  - *fact* relationship set (n-ary)
  - *rolls-up to* relationship set (binary)

A *rolls-up to* relates two *dimension levels* where the second one represents a higher level of abstraction. This kind of relationships define a Directed Acyclic Graph. Multiple hierarchies, alternative paths, and shared hierarchy levels for different dimensions are allowed. A *fact* relates n different *dimension level* entities. There is not any restriction to different *facts* being related to the same *dimension level*.

Since this model is based on E/R, *dimension levels* and *facts* would have attributes, which we respectively identify as Dimension Attributes and Measures at LL. The *dimension levels* are clearly placed at IL, as well as *facts*. At UL, a Fact would correspond to what is called a *multi-cube model*. At this level we also find implicit dimensions (a hierarchy of *dimension levels*).

Sánchez, Cavero, de Miguel, and Martínez (IDEA) The aim of [SCdMM99] is to present a conceptual multidimensional model allowing to desing multidimensional databases independently of the specific product used in their implementation. Besides the model, a closed algebra is defined with the following operations: *roll-up*, *join*, *destroy dimension*, *slice and dice*, and *select*. A methodology and CASE tool are also mentioned.

A multidimensional scheme is defined as a non empty set of *domains*, set of *domain aggregations*, set of *hierarchies*, and non empty set of *fact schemas*. Three different kinds of *domains* are distinguished (i.e. *dimension domain*, *synthesis domain*, and *description domain*). Furthermore, a *hierarchy* is a set of *domain aggregations* between *category domains* (a subclass on *dimension domain*) linked to shape a directed graph. A *fact schema* is a set of *dimension attributes*, set of *dimensions* (i.e. a subset of that of *dimension attributes*), structure of the cell, and predicate (showing the selected cells). Every *cell structure* is described as a list of *synthesis attributes* plus an attached list of applyable *synthesis functions*.

Measures correspond to attributes defined on *synthesis domains*, while Classification Attributes are those attributes defined on *description domains*. At IL, we find that every Dimension Level corresponds to a *dimension attribute*, and Fact Cells are called *cell structure*. Different *dimension attributes* are related by *aggregation functions* giving rise to Dimensions. Each *fact schema* contains exactly one *cell structure*. However, different *fact schemas* are related. We would identify a *multidimensional schema* containig a set of related *fact schemas* sharing Dimensions as a Fact.

Tryfona, Bushorg, and Christiansen (starER) In [TBC99], firstly a set of user requirements for a data warehouse conceptual model is listed. Then, a data model (based on the well known E/R model) addressing those requirements is defined. The requirements are:

1. Represent *facts* and their *properties*. Three different kinds or *properties* are considered (i.e. *stock*, *flow*, and *value-per-unit*)
2. Connect the temporal dimension to *facts*
3. Represent *objects*, capture their *properties* and *associations* among them. Three different kinds of associations are highlighted:
   (a) Specialization/Generalization
   (b) Aggregation
   (c) Membership, characterized by strictness (or, not) and completeness (or, not)
4. Record the *associations* between *objects* and *facts*
5. Distinguish *dimensions* and categorize them into *hierarchies* (*dimensions* are those *objects* connected by an *association* relationship to a *fact*)

Based on those requirements, the constructs of the model are *Fact set* that represents a set of real-world facts sharing the same characteristics or properties; *Entity set* which represents a set of real-world objects with similar properties; *Relationship set* represents a set of associations (of any kind out of the three aforementioned) among *entity sets* and *fact sets* (any cardinality is allowed - i.e. 1:N, N:1, and N:M); and *Attribute* which represents a static property of *entity sets*, *relationship sets*, or *facts sets*, which can be of any of the three kinds mentioned above.

Placing those constructs in our three levels, we can see implicitly defined a Dimension at UL as a set of related *entity sets*. Those *entity sets*, besides *fact sets*, would respectively play Dimension Levels and Fact Cells roles at IL. Finally, their *attributes* would be Measures and Classification Attributes at LL. Three different kinds of relationships are allowed between Dimension Levels at IL: Specialization, Aggregation, and Membership. Moreover, any cardinality is allowed for the relationship between a Dimension Level and a Fact Cell.

Nguyen, Tjoa, and Wagner conceptual multidimensional data model This multidimensional model, presented in [NTW00], uses the Object-Oriented paradigm to represent the metamodel. Specifically, UML is used in a schema modeling multidimensional data and metadata all together.

The *dimension members* form a *hierarchical domain* which partitions them into *dimension levels*, that belong to a *dimension*. In turn, *measures* are integer or float values grouped into cells, grouped into *groupbys*, where every cell conforms with a *groupby schema*. Each *groupby schema* refers to a set of *measure schemas*, and *dimension levels*; where *measure schemes* indicate aggregability of measures, and *dimension levels* show the granularity of the measures.

Data and metadata are defined at the same level. Thus, multidimensional schemes have a predefined structure, and Measure domains and Classification Attributes cannot be defined. Therefore, we consider that this model does not allow the representation of any kind of information at LL. However, at IL, we find Dimension Levels and Fact Cells (i.e. *groupbys*); and at UL we find Facts (as the *groupby schemas* associated to a cube schema) and Dimensions (called *dimension schemas*).

Hüsemann, Lechtenbörger, and Vossen conceptual warehouse design [HLV00] presents a phase-oriented Data Warehouse design methodology, which systematically derives schemes in *generalized multidimensional normal form*.

Those schemes contain *dimensions* structured in terms of one or more *aggregation paths* (which could be *alternative* or *optional*) that share the same terminal *dimension level*; and a *fact*, which is a set of *measures* determined by terminal *dimension levels* (*measures* functionally depend on *dimension levels*). The sets of *dimension levels* of different *dimensions* are assumed to be disjoint. Each one of those levels has a set of *property attributes* associated. A *fact scheme* represents the dimensional context for a set of *facts* that share the same terminal *dimension levels*. Summarizability is also shown by relating measures and *dimension levels* to a *restriction level* indicating the aggregation functions allowed.

This model has Measures and Classification Attributes at LL, which are respectively grouped into Fact Cells and Dimension Levels at IL. However, while Dimension Levels are grouped into Dimensions based on the meaningful *aggregation paths*, Fact Cells are not grouped if they are not sharing the same terminal Dimension Levels. It is important to remark that summarizability is shown at LL (for each Measure).

4.2 At logical level

This section contains those papers describing a model neither conceptual, nor physical. Their constructs are clearly oriented to a given kind of DBMS. Nevertheless, they are not that far from users conceptions.

At this level, we found the following models: [Kim96] (section 4.2), [BSH98] (section 4.2), [MTW99] (section 4.2), [GLK99] (section 4.2), and [MK00] (section 4.2).

Kimball multidimensional model [Kim96] and [KRRT98] describe the implementation of the multidimensional model on a Relational DBMS. In these

	Upper Level			Intermediate Level			Lower Level		
Sect.	F	D	Relationships	FC	DL	Relationships	M	CA	Relationships
4.2	–	✓	Ds shared by FCs	✓	–[1]	FK between FC and Ds	✓	✓	$CA \in D$ $M \in FC$
4.2	–	✓	Ds shared by FCs	✓	✓	FK between FC and Ds FK between DLs	✓	✓	$CA \in DL$ $M \in FC$
4.2 (NR)	✓	–	–	✓	–	$FC \in F$	✓	–	$M \in FC$
4.2 (ER)	–	✓	$Ds \in FC$	✓	✓	$DLs \in D$	✓	✓	$CA \in DL$ $M \in F$
4.2	–	✓	Ds shared by FCs	✓	✓	Pointers from FC to Ds Pointers between DLs	✓	✓	$CA \in DL$ $M \in FC$
4.2	✓[2]	✓[2]	Star schemes share Ds	✓	✓	DLs form hierarchies FC form hierarchies *one-to-many* between FCs and DLs	✓	✓	$CA \in DL$ $M \in FC$

[1] They are implicitly defined by CAs in each D.
[2] Implicitly defined by existing hierarchies.

Table 2. Scheme constructs in the different models at logical level

books, Ralph Kimball presents some multidimensional design patterns, and describes how they could be tackled.

The *star join schema* is defined as composed by a huge central *fact table*, and a set of usually smaller *dimension tables* surrounding it. The primary key of the *fact table* is composed by a foreign key to each one of the primary keys of the *dimension tables*. The *fact table* contains *numerical measures* (usually continuously valued, and additive), while *dimension tables* have *attributes* (usually textual, and discrete). The *dimension tables* can be shared by different *fact tables* giving rise to a *data warehouse bus* architecture.

The possibility of normalizing the *dimension tables* (obtaining an *snowflake schema*) is presented as an option that should be avoided. It would allow to explicit dimension hierarchies. However, the saved space is irrelevant, while query performance is really worsened (a series of joins become necessary), and browsing into dimension attribute values is more difficult.

Kimball's model does not define any explicit aggregation hierarchy or Dimension Levels, but they are implicit in the Dimension Attributes. Moreover, the *fact table* represent a given Fact Cell related to its Dimensions by foreign keys. At LL, we find Classification Attributes, as well as Measures.

Buzydlowski, Song, and Hassell (O3LAP) [BSH98] draws the advantages of an O3LAP approach as opposed to ROLAP and MOLAP. It presents a direct translation from Kimball's model into the Object-Oriented paradigm. Instead of using relational tables, the usage of object classes is proposed. Only two new concepts are introduced (i.e. *dimension non-associative classes*, and *dimension associative classes*) in order to distinguish those dimension with and without an explicit hierarchy, respectively.

Thus, its constructs are those of Kimball's model plus the possibility of expliciting Dimension Levels, at IL, within a Dimension.

Mangisengi, Tjoa, and Wagner (Nested Relations and Extended Relational) [MTW99] introduce and compare two different approaches to multidimensional modeling (notice that there are two entries in the summary tables for these authors). The ideas of those approaches are based on *nested relations* (Non-First Normal Form Relations), and an extension of the Relational model introduced by [Cod79].

A *nested relation* is a relation whose attributes may be other relations. By nesting relations, we can reflect the different detail levels in the fact measures. Therefore, we will obtain a Fact, at UL, as a relation; different nested relations corresponding to Fact Cells at different detail levels, at IL; and finally, the Measures for each Fact Cell at LL.

On the other hand, Codd's extension to the Relational model uses concepts like "object identifiers" (OIDs), "associations", or "object types". Moreover, it allows new operations like "patt", which partitions a relation based on a given attribute. A *fact relation* can be modeled as an association relation with participating *dimension relation* types, containing OIDs of dimension tuples. Each *dimension relation* type could further be refined by other characteristics (expliciting the aggregation hierarchy) in the same way (having OIDs as attributes). Thus, at UL, we would have the Dimensions. At IL, each Dimension, contains identifiers of its Dimension Levels, which contain identifiers of finer levels, and so on. A *fact relation* would correspond to a Fact Cell at this level. Finally, every Dimension Level contains Classification Attributes, and every Fact Cell contains Measures, at LL.

Gopalkrishnan, Li, and Karlapalem (Object-Relational View) [GLK99] also presents an Object-Oriented approach to multidimensional modeling. It not only describes a data model, but a methodology to build a Data Warehouse from Relational data sources.

A translation from Kimball's *snowflake schemes* to an Object-Oriented model is provided. The poor browsing performance in this kind of schemes, outlined by Kimball, is avoided here by using a *Structural Join Index Hierarchy* mechanism. A one-to-one mapping from Kimball's tables to object classes is defined. Foreign keys are translated to Object Identifier pointers.

By these means, we obtain Dimensions as a hierarchy of Dimension Levels related by object pointers. At IL, we also have Fact Cells, related to Dimensions by object pointers, too. Fact Cells as well as Dimension Levels contain attributes (i.e. Measures, and Classification Attributes respectively).

Moody, and Kortink design methodology [MK00] describe a methodology to develop multidimensional models from E/R models. The idea behind this work is to have the multidimensional design benefit from the information already in the operational schemes.

Different kinds of schemes can be obtained as result of the different steps (i.e. *flat*, *terraced*, *star*, *constellation*, *galaxy*, *snowflake*, or *star cluster*). All those kinds of schemes contain relational tables and are based upon the duality fact-dimension. They are characterized by different levels of denormalization in either fact or dimension tables. Thus, they offer different topologies:

- *Flat schemes* contain the minimum number of *fact tables*. They do not have any *dimension table*, because they are collapsed (denormalized) into the corresponding *fact table*. Moreover, some *fact tables* are also collapsed into more detailed ones if possible. They keep all possible joins precalculated.
- *Terraced schemes* contain all the *fact tables* without any *dimension table* (all them are collapsed). These schemes only precalculate star joins (i.e. those involving a *fact table* and a *dimension table*).
- *Star schemes* contain *fact* as well as *dimension tables*. However, they do not explicit dimension hierarchies, since they are collapsed into a single *dimension table*.
- *Constellation schemes* consists of a set of *star schemes* with hierarchically linked *fact tables*.
- *Galaxy schemes* consist of *star schemes* sharing dimensions.
- *Snowflake schemes* are *star schemes* with explicit dimension hierarchies obtained by dimension normalization.
- *Star cluster* schemes are *snowflake schemes* were we collapse those *dimension tables* that do not have a multiple hierarchy.

This methodology, in addition to Measures and Classification Attributes (at LL) being members of Dimension Levels and Fact Cells (at IL) respectively, considers constructs to relate those Dimension Levels and Fact Cells. Different Dimension Levels can be related to form possibly multiple dimension hierarchies. Moreover, Fact Cells can be related to show fact hierarchies (i.e. different levels of detail). It is not explicitly said in the methodology, but, at UL, we can identify a Dimension as a set of *dimension tables* in the same dimension hierarchy; and a Fact as a set of *fact tables* in the same fact hierarchy. It is also explained what to do with *many-to-many* relationships, and *subtypes*, since they could be found in a E/R model, but can not exist in a multidimensional one.

4.3 At physical level

In this section, we place those proposals that explain how a data cube could be implemented (i.e. stored, and/or retrieved). The proposals at this level do not only depend on the kind of DBMS, but also present which specific mechanisms it should implement.

At this level, we found only one paper about modeling: [Dyr96] (section 4.3). It could be surprising that there is only one paper in this section. However, at this level, proposals must be devoted to specific storage techniques instead of providing a true data model. Since modeling is a conceptualization by means of a given set of constructs, it is more suitable the closer to the user we are. Thus, we did not expect to find any work in this section, but this one expresses how data should be stored besides some concepts to understand it.

Dyreson [Dyr96] explains how a sparse cube could be implemented in a MOLAP database by means of disjoint, complete *cubettes*. An algorithm to retrieve

	Upper Level			Intermediate Level			Lower Level		
Sect.	F	D	Relationships	FC	DL	Relationships	M	CA	Relationships
4.3	–	✓	–	–	✓	*finer that* between DLs	–	–	–

Table 3. Scheme constructs in the different models at physical level

an aggregate value from the incomplete data cube is described, besides another algorithm to remove redundant *cubettes*.

A *measure* is defined as a system of measurement, and a *unit* as a subset chosen from the domain of interest. Thus, a set of disjoint *units*, chosen from the same domain, form a *measure*. A partial order is defined among *measures* based on their granularity or precision. A *cubette* is defined as containing data about a given *unit*, at a given detail level (i.e. *measure*).

Dimension Levels (called *measures*) and hierarchies (defined as graphs of *finer that* relationships between *measures*) are the only constructs provided in this framework, both at IL. There is nothing said about fact information.

4.4 At a formal level

In this section, we place those models mainly devoted to the definition of a multidimensional algebra and/or calculus. They do not pay too much attention to facilitate the capture of the specific user concepts. Since their focus is not in conceptualizing users ideas, we will see, in the summary tables, that they do not offer as much constructs as other models. However, if we would take into account the expressiveness of the algebras, things might be evenly matched.

At this level, we found the following models: [AGS97] (section 4.4), [LW96] (section 4.4), [DT97] (section 4.4), [HS97] (section 4.4), [GL97] (section 4.4), [Vas00] (section 4.4), and [Ped00] (section 4.4).

	Upper Level			Intermediate Level			Lower Level		
Sect.	F	D	Relationships	FC	DL	Relationships	M	CA	Relationships
4.4	–	–	–	✓	–	–	–[1]	✓	$FC = CAtuple$ or $FC = bool$
4.4	–	✓	$cube =< D, CAset >^n$ Cubes share Ds	–	–	Aggregation hierarchies defined at query time	–[2]	✓	–
4.4	–	–	–	✓[3]	✓	$Mset\ and\ Dset\ \in cube$	✓	✓	$f : D \rightarrow CAset$
4.4	–	✓	–	–	✓	Part-whole between DLs	✓	✓	$f : CA^n \rightarrow M$ Aggregated concepts
4.4	–	–	–	✓[4]	✓	$DLset \in cube$	✓	✓	$f : D \rightarrow CAset$
4.4	–	✓	*basic cube* uses Ds	✓	✓	DLs form a lattice	✓	–	$FC = Mtuple$
4.4	✓	✓	$F\ and\ Ds \in cube$ Cubes sharing dimensions form a *multidimensional object family*	–	✓	DLs form a lattice Applicability of aggregation functions per DL	✓[5]	–	–

[1] Due to the desired symmetry fact-dimension, everything is considered a dimension, and the function from the cartesian product of the dimension domains is defined on the booleans rather than on measures.
[2] There is not any information about measures in the scheme. A function is defined from dimensions to a set of scalar values.
[3] Implicit on defining a cube as containing a set of measures.
[4] Those attributes that are not at any DL, must be in the FC.
[5] Dimensions are treated as Measures.

Table 4. Scheme constructs in the different models at formal level

Agrawal, Gupta, and Sarawagi logical model [AGS97] presented one of the first multidimensional models, and probably, one of the most referenced ones. In spite of its qualification as "logical" by the authors, since its focus is on presenting an algebra as powerful as relational algebra, we are going to place it at formal level. The main characteristics of this model are the following:

- Symmetric treatment of measures and dimensions by providing conversion operations from one to another.
- A minimal, closed set of operations (i.e. *push*, *pull*, *destroy dimension*, *restriction*, and *join*) which can be directly translated to SQL.
- Support for multiple (non-explicit) hierarchies along each dimension.

This model distinguishes a *cube* composed by k dimensions, a function from k parameters to the booleans (or tuple of values), and a name for each dimension. It does not provide any means to explicit dimension hierarchies, and the only relation between different values in the cube is they are in a tuple. However, the model does allow to show which tuple of values is available depending on the selected dimension values.

This does not offer too many conceptual elements to model a multidimensional scheme. Actually, it just provides Classification Attributes (in the form of dimension values) without any possibility of even grouping them into different Dimensions. At most, we could consider that it allows to group measures into tuples giving rise to Fact Cells.

Li and Wang (Multidimensional Data model) In [LW96], its authors define a "Formal Multidimensional Data (MDD) model for OLAP systems". At the center of their approach is the notion of a multidimensional cube. They also define a Multidimensional Database (MDDB) as a set of multidimensional cubes and a finite set of relations.

A cube scheme is a set of pairs *dimension name*, *set of attribute names*. Thus, a cube is a cube scheme and a mapping from a combination of tuples (one for each dimension) containing the attribute values to a scalar value. There is not any kind of information in the scheme at IL, and aggregation hierarchies are not explicitly defined, but dynamically fixed at query time by means of ordering operations. However, cubes in the same MDDB share dimension relations. This means that, if two cubes have the same dimension name, they are using the same dimension relation.

Besides a formalism for multidimensional cubes, they also present a *grouping algebra*, which is used to query the MDDB; and a multidimensional cube algebra, used to query a MDDB and generate views. A novel feature of the *grouping algebra* is that it includes order-related operations. The set of operations provided by this algebra are those of the relational algebra, plus some order-oriented operations, and an aggregation operation. The multidimensional cube algebra offers six operations that are mappings from cubes to cubes (i.e. *add dimension*, *transfer*, *union*, *cube aggregation*, *rc-join*, and *construct*).

We can see the conceptual elements provided by this model as Dimensions at UL, stating a implicit relation among different cubes possibly sharing a dimension; and deaggregating dimensions into Classification Attributes at LL. Since the mapping function between dimensions and measures is defined on a scalar value (without any kind of semantic domain), we can say it does not provide any means to represent Measures.

Datta and Thomas The model of [DT97] resembles that of [AGS97]. The three goals of the authors on offering their model are to:

1. Allow symmetric treatment of dimensions and measures
2. Separate structure and content
3. Provide comprehensive OLAP functionality

They define a *data cube* as a set of dimensions; a set of measures; a set of attributes; and a mapping function corresponding to each dimension, a set of attributes. So, they neither define explicit hierarchies, nor the set of measures available at each level.

By defining *cube-instances*, they accomplish their second goal. A *cube-instance* is a *data cube* plus a set of values, plus a mapping from the cartesian product of the dimension domains to the values. Moreover, a set of operations (i.e. *restriction*, *aggregation*, *cartesian product*, *join*, *union*, *difference*, *pull*, and *push*) is defined on *cube-instances*. Operations *push* and *pull* are used to accomplish the first goal.

In this case, we can clearly see elements at different levels. At LL, we find Classification Attributes and Measures. While at IL, we find the set of Dimensions (as sets of Classification Attributes), each corresponding to exactly one level; and the set of Measures in the cube (implicitly, the Fact Cell corresponding to the unique level in the cube).

Hacid and Sattler description logics framework [HS97] propose an object-centered, logical framework (i.e. Description Logics) for multidimensional data models. Their aim is to facilitate comparison or evaluation of different multidimensional models, provide well defined semantics, and allow precise definition of problems. A translation between an Extended E/R diagram and Description Logics is given in [FS99].

By means of Description Logics, they represent a cube as a relationship among cells, which keep the coordinates and measures. Every cell in a cube must have the same structure. The functional dependency between coordinates and measures is explicitly shown. Beside cubes, dimension hierarchies can also be modeled. A hierarchically structured dimension is a set of objects interrelated by part-whole relationships. Thus, a hierarchy is represented as a finite partially ordered set.

Moreover, a set of operations on cubes is also defined. In this case, those operations are *restrict*, *destroy*, *join*, *rename*, *Join* (which offers more parameters than *join*), *aggr*, and *roll-up*. Furthermore, a whole section is devoted to the problems of the *drill-down* operation.

This is a semantically powerful model. However, some multidimensional modeling mechanisms are not explicitly explained or exemplified (i.e. the participation of dimension hierarchies in the definition of a cube, or the usage of complex concepts). At UL, Dimensions could be modeled as the set of concepts participating in classification hierarchies, in spite of it is not explicitly said. At IL, those hierarchies are decomposed into different aggregation concepts at different levels, related by part-whole relationships. Finally, at LL, we find Measures and Classification Attributes that can be aggregated into more complex concepts (maybe at IL or UL, it is not clarified by the explanation).

Gyssens, and Lakshmanan multi-dimensional database model As some models before, [GL97] also define some required functionalities, and drive their model to fulfill them:

1. Ability to pose powerful ad-hoc queries through a simple and declarative interface
2. Ability to restructure information
3. Ability to classify or group data sets
4. Ability to summarize values

To accomplish these goals, the authors propose a relational approach, and define an n-dimensional table scheme, as a triple containing a dimension name set; an attribute set; and a function from dimension names to attribute set, showing the attributes of each dimension. From this definition, they develop an algebra based on the Relational Algebra. They add, to the redefinition of classical relational operators, *fold* and *unfold* (in order to remove and add a dimension to the scheme, respectively); and a summarization and aggregation functions. Operators *fold* and *unfold* allow to convert measures into dimension attributes and vice versa, since the attributes of the disappeared dimension remain in the cube as measures. Therefore, the model allows a symmetric treatment of both of them. Moreover, it shows that every multidimensional table can be represented by a classical relation and vice-versa.

With regard to the modeling elements provided, we can distinguish Classification Attributes and Measures at LL. At IL, the Classification Attributes form a Dimension Level. If we subtract the Classification Attributes from the set of all attributes in the cube, we could also consider implicitly defined a Fact Cell.

Vassiliadis [Vas98], and [Vas00] present another formal model for multidimensional data, besides its mapping to ROLAP and MOLAP databases. Here, the cube algebra (demonstrated to be complete and sound) consists of just three operations (i.e. *navigate*, *selection*, and *split measures*).

For each dimension of analysis, a set of levels is defined, forming a lattice (bounded by "All" at top, and the *detailed level* at bottom). A dimension consists of a set of dimension paths, which are totally ordered lists of levels. A dimension level belongs exactly to one dimension, and has an associated space of values. The dimension levels can be monovalued or multivalued, whether their domain is a set or a power set of the space of values.

A MDDB is defined as a set of dimensions, dimension levels and a *basic cube*. A *basic cube* contains the data cells at the maximum level of detail. Over this, by mean of the cube algebra, other cubes (we could call views) are defined. The existence of the basic cube is justified by the impossibility of performing the drill-down operation without it.

In this model, a Dimension (at UL) is composed by a lattice of Dimension Levels (at IL). However, the dimension levels do not contain further details at LL. If we look at Facts Cells (at IL), which are not explicitly defined, we see that they are a tuple of Measures identified by the bottom levels of the different dimension lattices.

Pedersen (Extended Multidimensional Data Model) Besides a classification of multidimensional models, [PJ98], [PJ99], and [Ped00], already referenced in section 2, also present an *Extended Multidimensional Data Model* (EMDM) is presented. After the definition of the requirements (most of them refer to semantics) for the usage of a multidimensional model in a clinical context, and the verification that none of the existing models addresses all of them, this new model is defined.

EMDM provides a formalism and algebra that is closed and, at least, as strong as relational algebra with aggregation functions. The operations in the algebra are *selection*, *projection*, *rename*, *union*, *difference*, *identity-based join*, *aggregate formation*, *value-base join*, *duplicate removal*, *SQL-like aggregation*, *star-join*, *drill-down*, and *roll-up*. How to implement the model using relational databases is also explained.

An *n-dimensional fact scheme* consists of a *fact type*, and n *dimension types*. In turn, a *dimension type* consists of a set of partially ordered *category types* forming a lattice. To each *category type*, an *aggregation type* has been associated, indicating the aggregate functions applicable at that level. The model treats dimensions and measures symmetrically. Multiple hierarchies per dimension, non-strict hierarchies, non-onto hierarchies, non-covering hierarchies, or many-to-many relations between facts and dimensions are allowed. However, there is no way to reflect such information in the scheme. Instead, it is deduced from data instances. Moreover, relating values that represent the "same" concept along time is also possible thanks to temporal constructs.

The semantic constructs offered by the model are Dimensions and Facts at UL, and Dimension Levels at IL. We do not consider the model allows to show Fact Cells at IL. Data in the Facts can be related to any Dimension Level, however this information cannot be shown in the scheme. At LL, we find that Classification Attributes do not exist. However, in spite of Fact Cells do not have attributes neither, dimension values are used as Measures.

4.5 Other work

There are other papers about multidimensional interfaces, multidimensional query languages, etc. ([GJJ97], [GL98], [GBLP96], and [BPT97] among others) that also treat, as a minor subject, some kind of multidimensional model. We left these out of the classification, because the models have not any new or improved characteristics, neither was it the aim of the authors to present a multidimensional model.

Moreover, there is a lot of literature devoted to either ROLAP or MOLAP implementation ([HRU96], and [TS97] among others). For instance, they present different kinds of indexing techniques or partition strategies. We did not include any of them in this survey (in the section about models at physical level), because they do not model the multidimensional data, but just give useful hints to obtain good storage or query performance.

5 Conclusions

This paper has reviwed and classified the (to the best of our knowledge) proposed multidimensional models. There exist previous studies comparing different multidimensional models (see section 2). However, those studies did it just to show their lacks against a given list of requirements. Models for absolutely different purposes were put into the same bag. On the contrary, we have classified them in different sets based on their usage in the data warehouse design process. Furthermore, we have given a framework to compare the terminology used by different authors for the constructs of their models.

Table 5 contains a summary of elements and relationships among them that we found at the scheme level of each model (see section 3 and the beginning of section 4 for the meaning of the different columns). Notice we are not showing information about neither instances nor instantiation relations. As outlined in [BSHD98], some models do not separate cube structure and contents. In these cases, we only have taken into account that information contained in the scheme. A cell containing a hyphen means the corresponding model does not provide any construct in that context, while a tick implies the model does provide some kind of construct.

At first glance, it seems that conceptual models offer the possibility of representing much more semantics that models at other levels. Indeed, conceptual models do have to provide a rich set of semantic constructs in order to capture user ideas. In turn, models at formal level are those that offer less conceptual constructs. However, notice they do offer an algebra whose expressiveness was not considered along this work. Only one physical model was found (without too many constructs), as expected. At physical level, we find storage techniques instead of true data models.

Looking at the table we can appreciate that the more recent the models are (they are ordered chronologically into each design level), they use to more capture semantics. This can be interpreted as a trend to semantically enrich

Authors (Model)	Design Level	Upper Level			Intermediate Level			Lower Level		
		F	D	Rel.	FC	DL	Rel.	M	CA	Rel.
Lehner (NMDM)	C	–	√	–	–	√	√	–	√	√
Cabibbo and Torlone (*MD*)	C/F	√	√	√	–	√	√	√	√	√
Golfarelli et al. (DFM)	C	–	√	–	√	√	√	√	√	√
Trujillo et al. (GOLD)	C	√	√	√	–	√	√	√	√	√
Sapia et al. (MERM)	C	√	√	√	√	√	√	√	√	√
Sánchez et al. (IDEA)	C	√	√	√	√	√	√	√	√	√
Tryfona et al. (starER)	C	–	√	–	√	√	√	√	√	√
Nguyen et al.	C	√	√	√	√	√	√	–	–	–
Hüsemann et al.	C	–	√	√	√	√	√	√	√	√
Kimball	L	–	√	√	√	–	√	√	√	√
Buzydlowski et al. (O3LAP)	L	–	√	√	√	√	√	√	√	√
Mangisengi et al. (NR)	L	√	–	–	√	–	√	√	–	√
Mangisengi et al. (ER)	L	–	√	√	√	√	√	√	√	√
Gopalkrishnan et al. (ORV)	L/P	–	√	√	√	–	√	√	√	√
Moody and Kortink	L	√	√	√	√	√	√	√	√	√
Dyreson	P	–	√	–	–	√	√	–	–	–
Agrawal, Gupta, and Sarawagi	F	–	–	–	√	–	–	–	√	√
Li and Wang (MDD)	F	–	√	√	–	–	√	–	√	–
Datta and Thomas	F	–	–	–	√	√	√	√	√	√
Hacid and Sattler	F	–	√	–	–	√	√	√	√	√
Gyssens and Lakshmanan	F	–	–	–	√	√	√	√	√	√
Vassiliadis	F	–	√	√	√	√	√	√	–	√
Pedersen (EMDM)	F	√	√	√	–	√	√	√	–	–

Table 5. Summary table of the different models presented above

multidimensional models. However, having models that provide constructs at every heading does not mean they capture all possible semantics. There is neither a model encompassing the semantic constructs of the rest, nor a consensus or standard stating what should be represented in a multidimensional scheme. We think that more work needs to be done in this area.

Acknowledgements

This work has been partially supported by the Spanish Research Program PRONTIC under projects TIC99-1078-C02-01 and TIC99-1078-C02-02, as well as the grant 1998FI-00228 from the Generalitat de Catalunya.

References

[AGS97] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In *Proc. of 13th. Int. Conf. on Data Engineering (ICDE)*, pages 232–243. IEEE Press, 1997.

[BCN92] C. Batini, S. Ceri, and S. Navathe. *Conceptual Database Design- an Entity-Relationship Approach*. Benjamin/Cummings, 1992.

[BPT97] E. Baralis, S. Paraboschi, and E. Teniente. Materialized views selection in a multidimensional database. In *Proc. of 23rd Int. Conf. on Very Large Data Bases (VLDB)*, pages 156–165, 1997.

[BSH98] J. W. Buzydlowski, I. Song, and L. Hassell. A Framework for Object-Oriented On-line Analytical Processing. In *Proc. of the ACM 1st Int. Workshop on Data Warehousing and OLAP (DOLAP)*, Washington D.C. (USA), 1998.

[BSHD98] M. Blaschka, C. Sapia, G. Höfling, and B. Dinter. Finding your way through multidimensional data models. In *Proc. of 9th Int. Conf. on Database and Expert Systems Applications (DEXA)*, number 1460 in LNCS. Springer, 1998.

[CD97] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1):65–74, 1997.

[Cod79] E. F. Codd. Extending the relational model to capture more meaning. *ACM Transactions on Database Systems*, 4(4):394–434, 1979.

[CT97] L. Cabibbo and R. Torlone. Querying multidimensional databases. In *Proc. of 6th Int. Workshop on Database Programming Languages (DBPL6)*, Estes Park (USA), 1997.

[CT98a] L. Cabibbo and R. Torlone. From a procedural to a visual query language for olap. In *Proc. 10th Int. Conf. on Scientific and Statistical Database Management (SSDBM)*. IEEE Computer Society, 1998.

[CT98b] L. Cabibbo and R. Torlone. A logical approach to multidimensional databases. In *Advances in Database Technology - EDBT'98*, number 1377 in LNCS, pages 183–197. Springer, 1998.

[DT97] A. Datta and H. Thomas. A Conceptual Model and an algebra for On-Line Analytical Processing in Data Warehouses. In *Proc. of Workshop on Information Technologies and Systems (WITS)*, Atlanta (USA), 1997.

[Dyr96] C. Dyreson. Information retrieval from an incomplete data cube. In *Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB)*, pages 532–543. Morgan Kaufmann Publishers, 1996.

[EN94] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems, 2nd Ed.* Benjamin Cummings, 1994.

[FS99] E. Franconi and U. Sattler. A data warehouse conceptual data model for multidimensional aggregation. In *Proc. of 1st Int. Workshop on Design and Management of Data Warehouses (DMDW)*, Heidelberg (Germany), 1999.

[GBLP96] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In Stanley Y. Wu, editor, *Proc. 12th Int. Conf. on Data Engineering (ICDE)*, pages 152–159. IEEE Computer Society, 1996.

[GJJ97] M. Gebhardt, M. Jarke, and S. Jacobs. A toolkit for negotiation support interfaces to multi-dimensional data. *SIGMOD Record*, 26(2):348–356, 1997.

[GL97] M. Gyssens and L. V. S. Lakshmanan. A foundation for multi-dimensional databases. In *Proc. of 23rd Int. Conf. on Very Large Data Bases (VLDB)*, pages 106–115. Morgan Kaufmann Publishers, 1997.

[GL98] F. Gingras and L. V. S. Lakshmanan. nD-SQL: A multi-dimensional language for interoperability and OLAP. In *Proc. 24th Int. Conf. on Very Large Data Bases (VLDB)*, pages 134–145, 1998.

[GLK99] V. Gopalkrishnan, Q. Li, and K. Karlapalem. Star/snow-flake schema driven object-relational data warehouse design and query processing strategies. In *Proc. of 1st Int. Workshop on Data Warehousing and Knowledge Discovery (DaWaK)*, number 1676 in LNCS, pages 11–22. Springer, 1999.

[GMR98a] M. Golfarelli, D. Maio, and S Rizzi. Conceptual design of data warehousing from E/R schemes. In *Proc. of 31st Hawaii Int. Conf. on System Sciences*, 1998.

[GMR98b] M. Golfarelli, D. Maio, and S Rizzi. The Dimensional Fact Model: a Conceptual Model for Data Warehouses. *Int. Journal of Cooperative Information Systems*, 7(2&3):215–247, 1998.

[GR98] M. Golfarelli and S. Rizzi. A Methodological Framework for Data Warehouse Design. In *Proc. of the ACM 1st Int. Workshop on Data Warehousing and OLAP (DOLAP)*, Washington D.C. (USA), 1998.

[GR99] M. Golfarelli and S Rizzi. Designing the data warehouse: key steps and crucial issues. *Journal of Computer Science and Information Management*, 2(1):1–14, 1999.

[HLV00] B. Hüsemann, J. Lechtenbörger, and G. Vossen. Conceptual data warehouse design. In *Proc. of 2nd Int. Workshop on Design and Management of Data Warehouses (DMDW)*, Stockholm (Sweden), 2000.

[HRU96] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 25(2):205–216, 1996.

[HS97] M.-S. Hacid and U. Sattler. An object-centered multi-dimensional data model with hierarchically structured dimensions. In *Proc. of the IEEE Knowledge and Data Exchange Workshop (KDEX)*, pages 65–72. IEEE Computer Society, 1997.

[Kim96] R. Kimball. *The Data Warehouse toolkit.* John Wiley & Sons, 1996.

[KRRT98] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. *The Data Warehouse lifecycle toolkit.* John Willey & Sons, 1998.

[Leh98] W. Lehner. Modeling large scale OLAP scenarios. In *Advances in Database Technology - EDBT'98*, volume 1377 of *LNCS*, pages 153–167. Springer, 1998.

[LW96] C. Li and X. S. Wang. A data model for supporting on-line analytical processing. In *Proc. of 5th Int. Conf. on Information and Knowledge Management (CIKM)*, 1996.

[MK00] D. L. Moody and M. A. R. Kortink. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. In *Proc. of 2nd Int. Workshop on Design and Management of Data Warehouses (DMDW)*, Stockholm (Sweden), 2000.

[MTW99] O. Mangisengi, A M. Tjoa, and R. R. Wagner. Multidimensional Modeling Approaches for OLAP Based on Extended Relational Concepts. In *Proc. of the 9th Int. Database Conf. on Heterogeneous and Internet Databases (IDC)*, Hong Kong, 1999.

[NTW00] T. B. Nguyen, A M. Tjoa, and R. R. Wagner. An object oriented multidimensional data model for olap. In *Proc. of 1st Int. Conf. on Web-Age Information Management (WAIM)*, number 1846 in LNCS, pages 69–82. Springer, 2000.

[Ped00] T. B. Pedersen. *Aspects of Data Modeling and Query Processing for Complex Multidimensional Data.* PhD thesis, Faculty of Engineering and Science, Aalborg University (Denmark), 2000.

[PJ98] T. B. Pedersen and C. S. Jensen. Research issues in clinical data warehousing. In *Proc. of 10th Int. Conf on Statistical and Scientific Database Management (SSDBM)*, pages 43–52. IEEE Computer Society, 1998.

[PJ99] T. B. Pedersen and C. S. Jensen. Multidimensional data modeling for complex data. In *Proc. of 15th Int. Conf. on Data Engineering (ICDE)*, pages 336–345. IEEE Computer Society, 1999.

[SBHD99] C. Sapia, M. Blaschka, G. Höfling, and B. Dinter. Extending the E/R model for the multidimensional paradigm. In *Proc. Int. Workshop on Data Warehouse and Data Mining (DWDM) in conjunction with the ER'98*, number 1552 in LNCS, pages 105–116. Springer, 1999.

[SCdMM99] A. Sánchez, J. M. Cavero, A. de Miguel, and P. Martínez. IDEA: A Conceptual Multidimensional Data Model and Some Methodological Implications. In *Proc. of VI Congreso Int. de Investigación en Ciencias Computacionales*, Cancún (Mexico), 1999.

[TBC99] N. Tryfona, F. Busborg, and J. G. B. Christiansen. starER: A conceptual model for data warehouse design. In *Proc. of ACM 2nd Int. Workshop on Data Warehousing and OLAP (DOLAP)*, Kansas City (USA), 1999.

[TP98] J. C. Trujillo and M. Palomar. An Object-Oriented Approach to Multidimensional Database Conceptual Modeling. In *Proc. of the ACM 1st Int. Workshop on Data Warehousing and OLAP (DOLAP)*, Washington D.C. (USA), 1998.

[TPG00] J. C. Trujillo, M. Palomar, and J. Gómez. Applying Object-Oriented Conceptual Modeling Techniques to the Design of Multidimensional Databases and OLAP applications. In *Proc. of 1st Int. Conf. on Web-Age Information Management (WAIM)*, number 1846 in LNCS, pages 83–94. Springer, 2000.

[TS97] D. Theodoratos and T. K. Sellis. Data warehouse configuration. In *Proc. of 23rd Int. Conf. on Very Large Data Bases (VLDB)*, pages 126–135, 1997.

[UW97] J. Ullman and J. Widom. *A First Course in Database Systems*. Prentice-Hall, 1997.

[Vas98] P. Vassiliadis. Modeling Multidimensional Databases, Cubes and Cube operations. In *Proc. of 10th Int. Conf. on Scientific and Statistical Database Management (SSDBM)*. IEEE Computer Society, 1998.

[Vas00] P. Vassiliadis. *Data Warehouse Modeling and Quality Issues*. PhD thesis, Department of Electrical and Computer Engineering, National Technical University of Athens (Greece), 2000.

[VS99] P. Vassiliadis and T. Sellis. A survey of logical models for olap databases. *SIGMOD Record*, 28(4), December 1999.