A Framework for the Classification and Description of Multidimensional Data Models

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 bring together a set of tools, that use multidimensional modeling in the management of information to improve the decision making process. 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 some of that work, with regard to the design phase they are used in. On the other hand, it allows to compare the different terminology used by each author, by placing all the terms in a common framework.

1 Introduction

In the last years, multidimensional modeling has gained attention of 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 analysis and visualization of data for decision making processes. Its main advantages are its proximity to the analysts way of thinking, and facility to improve query performance.

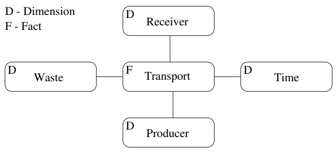

Fig. 1. Multidimensional scheme example

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, in a waste transport business, we would analyze fact "Transport" 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.

In this paper, we classify the huge amount of efforts devoted to modeling the cube into four groups based on the design phase for which they are, from our point of view, more suitable. Moreover, a framework is also defined to be able to compare their modeling constructs. Section 2 briefly reviews previous work on classifying multidimensional models. Sections 3.1 and 3.2 define the framework that will allow us to describe and classify the different models; while section 4 contains a brief comparison of some models. These models are sorted out in four different subsections based on the modeling level that we think they are in. Section 5 presents some conclusions, followed by acknowledgements and bibliography.

2 Related work

In [BSHD98], a list of requirements for a multidimensional model in order to be suitable for OLAP applications, is used to analyze seven models, that are chosen because they contain some kind of formalism. Among those seven we find [AGS97], [GL97], [CT98], [Vas00], and [Leh98]. Those requirements were derived from general design principles, and from characteristics of OLAP applications.

[Ped00] 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], [Kim96], [GL97], [CT98], [Leh98], and [Vas00] are among those twelve. An statistical model, and a commercial system are also included. Moreover, it presents a new data model which does address all those requirements.

[Vas00] gives yet another classification of multidimensional models. Among others, it pays attention to [GL97], [AGS97], [CT98], [Leh98], some industrial standards, and a couple of statistical models.

3 A classification and description framework

The following sections introduce two sets of classification and description levels for multidimensional models. These levels are orthogonal. Thus, a model can be classified as either *Conceptual*, *Logical*, *Physical*, or *Formalism*; and contain constructs at any of the three detail levels, i.e. *Upper*, *Intermediate*, or *Lower*.

3.1 Design levels

A data model is a set of concepts that can be used to describe the structure of a database. We propose to distinguish different multidimensional data models attending to the kind of constructs/concepts they provide and the multidimensional design phase they are used in.

As shown in figure 2 (left), from [UW97], in an On-Line Transactional Processing (OLTP) environment, during the first design step, at *Conceptual* level,

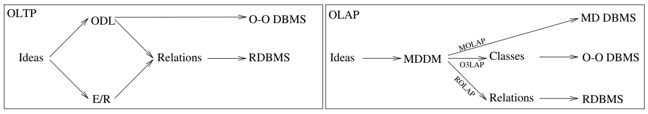

Fig. 2. Modeling and implementation process in OLAP vs OLTP environments

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 usually use the Relational model, but we could also use Hierarchical, or Network models (not depicted in the figure); and in the last step, at *Physical* level, the implementation would depend on a specific DBMS (i.e. Oracle, Informix, ObjectStore, etc.). In a similar way, in our proposal for 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.

We adopted the terminology in [BCN92], that considers models 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. In [EN94], we find a similar 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 they provide concepts that describe the details of how data is stored in the computer; and "Implementation", if they 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.

Besides these three mentioned above, we found another set of models (let us say they are *Formalisms*) whose concepts would not be used at any database design phase, but on giving a theoretical framework. They include an algebra or calculus. In an OLTP environment, a formalism would be the Relational Algebra.

3.2 Detail levels

Each one of the papers mentioned 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. 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, to be able to compare them, and examine the expressive power of every model.

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 relating 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 an aggregation hierarchy. Each *Fact Cell* contains data at a given *Dimension Level* for each *Dimension* its *Fact* is related to.

Lower Level (LL): The last level shows the set attributes of the *Dimension Levels* and *Fact Cells*. That is *Classification Attributes* (CA), and *Measures* (M) respectively.

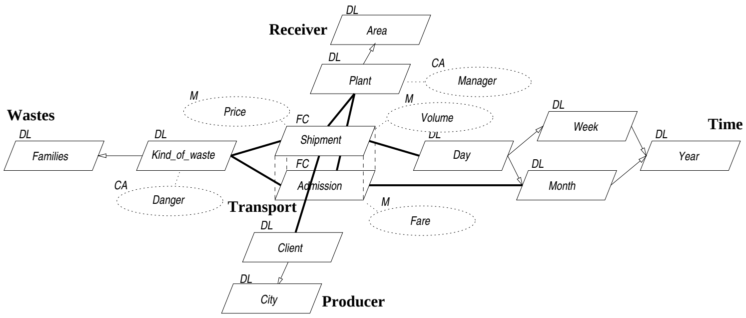

Fig. 3. Further detailed multidimensional scheme example

Going back to our example in figure 1, which represents a multidimensional scheme at *Upper Level*, 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 *Lower Level*. 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 Application of the framework to some relevant models

This section contains a set of multidimensional data models classified in four different subsections. Each one of these four subsections contains proposals at a given design level (i.e. *Conceptual*, *Logical*, *Physical*, and *Formalisms*).

Authors (Model)	Reference	Design Level	Upper Level			Intermediate Level			Lower Level		
			F	D	Rel.	FC	DL	Rel.	M	CA	Rel.
Lehner (NMDM)	[Leh98]	C	–	✓	–	–	✓	✓	–	✓	✓
Cabibbo and Torlone (*MD*)	[CT98]	C/F	✓	✓	✓	–	✓	✓	✓	✓	✓
Golfarelli et al. (DFM)	[GMR98]	C	–	✓	–	✓	✓	✓	✓	✓	✓
Sapia et al. (MERM)	[SBHD99]	C	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tryfona et al. (starER)	[TBC99]	C	–	✓	–	✓	✓	✓	✓	✓	✓
Kimball	[Kim96]	L	–	✓	✓	✓	–	✓	✓	✓	✓
Buzydlowski et al. (O3LAP)	[BSH98]	L	–	✓	✓	✓	✓	✓	✓	✓	✓
Gopalkrishnan et al. (ORV)	[GLK99]	L/P	–	✓	✓	✓	–	✓	✓	✓	✓
Mangisengi et al. (NR)	[MTW99]	L	✓	–	–	✓	–	✓	✓	–	✓
Mangisengi et al. (ER)	[MTW99]	L	–	✓	✓	✓	✓	✓	✓	✓	✓
Dyreson	[Dyr96]	P	–	✓	–	–	✓	✓	–	–	–
Agrawal, Gupta, and Sarawagi	[AGS97]	F	–	–	–	✓	–	–	–	✓	✓
Hacid and Sattler	[HS97]	F	–	✓	–	–	✓	✓	✓	✓	✓
Gyssens and Lakshmanan	[GL97]	F	–	–	–	✓	✓	✓	✓	✓	✓
Vassiliadis	[Vas00]	F	–	✓	✓	✓	✓	✓	✓	–	✓
Pedersen (EMDM)	[Ped00]	F	✓	✓	✓	–	✓	✓	✓	–	–

Table 1. Summary table of the different models presented

Table 1 contains a summary of elements and relationships among them that we found at each model. Notice we are not showing information about neither instances nor instantiation relations. As pointed out by [BSHD98], some multidimensional models do not separate cube structure and contents. Only those concepts represented at schema level are 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.

LL Relationships means there is no 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).

IL Relationships means there is no 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.

UL Relationships means there is no way in the model to represent relationships among *Facts* and/or *Dimensions*.

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.

[Leh98] is rather a presentation-oriented model, conceived to ease navigation through data. It proposes the existence of two levels in order to improve the power and flexibility of the analysis process. The first level (called "Primary Multidimensional Object") contains the *Dimension Levels* at which the facts are, and the second level (called "Secondary Multidimensional Object") shows *Classification Attributes* that select the desired facts. The main characteristic of this model is that different instances in the same class might have different attributes. All schema constructs in this model refer to dimensions.

The model in [CT98] is qualified as "logical". However, they say that it is independent of any specific implementation, and present a design methodology to obtain a multidimensional scheme from an E/R one. Moreover, it is also said that their model is at a higher level of abstraction than a star scheme consisting of relational tables. Therefore, we classify it as *Conceptual*, in spite of it also provides a strong formal foundation (including a calculus). Each *Dimension* is organized in a hierarchy of *Dimension Levels* corresponding to data domains at different granularity. In turn, a *Dimension Level* can have *Classification Attributes* associated with it. The *Facts* are functions from *Dimension Levels* to *Measures*.

[GMR98] presents a graphical model besides a methodology to obtain a multidimensional scheme from the operational schemes (either E/R or Relational). Contrary to what is said for some *Formalisms*, the authors claim that it is important to clearly distinguish between dimensions and measures. A scheme is composed by a set of *Fact Cells*, and *Dimensions* with hierarchies of *Dimension Levels* forming a "quasi-tree". *Fact Cells* contain *Measures*, while *Dimension Levels* contain two kinds of *Classification Attributes*: those defining aggregation hierarchies, and the others, that are only used to select data. 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 Level*.

[SBHD99] argues that the E/R model is not suited for multidimensional conceptual modeling. Thus, a specialization is defined, and its usage exemplified.

Dimension Level is a specialization of Entity, while Relationship is specialized into *Fact*, and roll-up relations between *Dimension Levels*. *Measures* and *Classification Attributes* are, respectively, "relationship", and "entity" attributes.

[TBC99] is also based on the well known E/R. This model distinguishes three kinds of *Measures*, i.e. "stock", "flow", and "value-per-unit". Moreover, it also allows to represent different kinds of relationships among *Dimension Levels* and *Fact Cells* (i.e. "Specialization/Generalization", "Aggregation", and "Membership"). For instance, aggregation hierarchies in *Dimensions* are defined by means of "Membership" relationships. Moreover, cardinality between a *Dimension Level* and a *Fact Cell* is not restricted to 1:N (as it is in other models), but N:M is also allowed.

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.

Doubtless, the most prominent work at this design level is [Kim96]. It describes the implementation of the multidimensional model on a Relational DBMS. In this book, Ralph Kimball presents some multidimensional design patterns, and describes how they could be tackled. A huge Relational table represents every *Fact Cell*, and smaller tables, corresponding to its *Dimensions*, surround each one of them. *Dimension Levels* are implicit in the values of attributes in *Dimensions*, while foreign keys are defined from each *Fact Cell* to its *Dimensions*. This is known as a "star join schema".

Some efforts have been done to improve Kimball's work. We can see [BSH98], or [GLK99], for two Object-Oriented approaches.

[MTW99] introduces and compares two different approaches to multidimensional modeling (notice that there are two entries in the summary table for these authors). The ideas of those approaches are based on "nested relations" (Non-First Normal Form Relations), and the extension to the Relational model introduced in [Cod79]. By means of "nested relations", it is possible to have *Measures* into *Fact Cells*, and these nested into *Facts*. On the other hand, using Codd's extension allows to introduce "object identifiers" (OIDs), "associations", and "object types". This facilitates to relate *Fact Cells* and *Dimensions*, and define aggregation hierarchies among *Dimension Levels*.

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. This fact could be surprising; however, at *Physical* 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. We did not expect to find any work in this section, however, this one expresses how data should be stored besides some concepts to understand it.

[Dyr96] explains how a sparse cube could be implemented in a MOLAP database by means of disjoint, complete "cubettes". A "cubette" is defined as containing data about a given subset chosen from the domain of interest. An algorithm to retrieve an aggregate value from the incomplete data cube is described, besides another algorithm to remove redundant "cubettes".

4.4 Formalisms

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 can see, in the summary table, that they do not offer as much constructs as other models. However, if we would take into account the expressiveness of the algebras, they could be as semantically rich as conceptual are. We only take into account modeling constructs, studying the expressiveness of the operations is out of the scope of this paper.

[AGS97] presents 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 consider it a formalism. It does not offer too many conceptual elements to model a multidimensional scheme. Actually, it just provides *Classification Attributes* 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*. An important feature is that allows symmetric treatment of *Measures* and *Classification Attributes* by providing a conversion operation.

[HS97] uses Description Logics to describe the cube. It is represented as a relationship among cells, which keeps coordinates and measures. It is not explicitly said, but *Dimensions* could be modeled as the set of concepts participating in classification hierarchies, which are composed by *Dimension Levels* related through Part-Whole relationships.

[GL97] proposes a Relational approach defining an "n-dimensional table scheme" as a triple containing a dimension names set; an attributes set; and a function from dimension names to attribute set, showing the attributes of each dimension. This allows to represent *Measures*, *Classification Attributes*, and *Dimension Levels*. If we subtract the *Classification Attributes* from the set of all attributes in the cube, we could also consider implicitly defined a *Fact Cell*.

[Vas00] presents a complete and sound algebra. It defines a *Dimension* as a lattice of *Dimension Levels*, that do not contain *Classification Attributes*. *Fact Cells* are implicitly said to be tuples of *Measures*, and always defined at the bottom level of the *Dimensions*.

Finally, [Ped00] provides a formalism and an algebra that is closed and, at least, as strong as Relational algebra with aggregation functions. It defines *Facts*, and *Dimensions* containing different *Dimension Levels*. However, we do not consider the model allows to show *Fact Cells*. Data in the *Facts* can be related to any *Dimension Level*, but this information cannot be shown in the scheme. *Classification Attributes* can neither be related to *Dimension Levels*, while *Dimension Levels* are used as *Measures* (by treating symmetrically dimensions and facts).

5 Conclusions

This paper has presented a framework that allows us to classify and compare multidimensional models. It has been exemplified by studying some representative models. Classification and description of other models has been left out just because of space limitations. 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, and models for absolutely different purposes were put into the same bag. On the contrary, we have classified them in different levels (i.e. *Conceptual*, *Logical*, *Physical*, and *Formalisms*) based on their usage in the multidimensional design process. Furthermore, we have given a framework to compare the terminology used by different authors for the constructs of their models.

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, *Formalisms* are those that offer less conceptual constructs. However, notice they do offer an algebra whose expressiveness was not considered in this paper, because our focus was on modeling constructs. At *Physical* level, we find storage techniques instead of true data models.Thus, just one *Physical* model was found. Moreover, there was not found a great variety of models at *Logical* level.

The more recent the models are, they use to capture more semantics. This can be interpreted as a trend to semantically enrich multidimensional models. However, having models that provide constructs at every detail level does not mean they capture all possible semantics. We have found neither a model encompassing the semantic constructs of the rest, nor a consensus or standard stating what should be represented in a multidimensional scheme.

Acknowledgements

This work has been partially supported by the Spanish Research Program PRONTIC under projects TIC2000-1723-C02-01 and TIC2000-1723-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 *Int. Conf. on Data Engineering (ICDE)*. IEEE Press, 1997.

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

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

[BSHD98] M. Blaschka, C. Sapia, G. Höfling, and B. Dinter. Finding your way through multidimensional data models. In *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), 1997.

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

[CT98] L. Cabibbo and R. Torlone. A logical approach to multidimensional databases. In *Advances in Database Technology - EDBT'98*, number 1377 in LNCS. Springer, 1998.

[Dyr96] C. Dyreson. Information retrieval from an incomplete data cube. In *Int. Conf. on Very Large Data Bases (VLDB)*. Morgan Kaufmann, 1996.

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

[GL97] M. Gyssens and L. V. S. Lakshmanan. A foundation for multi-dimensional databases. In *Int. Conf. on Very Large Data Bases (VLDB)*. Morgan Kaufmann, 1997.

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

[GMR98] 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), 1998.

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

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

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

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

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

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

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

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

[Vas00] P. Vassiliadis. *Data Warehouse Modeling and Quality Issues.* PhD thesis, Department of Electrical and Computer Engineering, 2000.