Encyclopedia of Data Warehousing and Mining

Second Edition

John Wang
Montclair State University, USA

Volume I
A–Data Pre

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content: Kristin Klinger
Director of Production: Jennifer Neidig
Managing Editor: Jamie Snavely
Assistant Managing Editor: Carole Coulson
Typesetter: Amanda Appicello, Jeff Ash, Mike Brehem, Carole Coulson, Elizabeth Duke, Jen Henderson, Chris Hrobak, Jennifer Neidig, Jamie Snavely, Sean Woznicki
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanbookstore.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of data warehousing and mining / John Wang, editor. -- 2nd ed.
p. cm.
Includes bibliographical references and index.
Summary: "This set offers thorough examination of the issues of importance in the rapidly changing field of data warehousing and mining"--Provided by publisher.
ISBN 978-1-60566-010-3 (hardcover) -- ISBN 978-1-60566-011-0 (ebook)
1. Data mining. 2. Data warehousing. I. Wang, John,
QA76.9.D37E52 2008
005.74--dc22
2008030801

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Conceptual Modeling for Data Warehouse and OLAP Applications

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

The advantages of using conceptual models for database design are well known. In particular, they facilitate the communication between users and designers since they do not require the knowledge of specific features of the underlying implementation platform. Further, schemas developed using conceptual models can be mapped to different logical models, such as the relational, object-relational, or object-oriented models, thus simplifying technological changes. Finally, the logical model is translated into a physical one according to the underlying implementation platform.

Nevertheless, the domain of conceptual modeling for data warehouse applications is still at a research stage. The current state of affairs is that logical models are used for designing data warehouses, i.e., using *star* and *snowflake* schemas in the relational model. These schemas provide a multidimensional view of data where *measures* (e.g., quantity of products sold) are analyzed from different perspectives or *dimensions* (e.g., by product) and at different levels of detail with the help of *hierarchies*. On-line analytical processing (OLAP) systems allow users to perform automatic aggregations of measures while traversing hierarchies: the roll-up operation transforms detailed measures into aggregated values (e.g., daily into monthly sales) while the drill-down operation does the contrary.

Star and snowflake schemas have several disadvantages, such as the inclusion of implementation details and the inadequacy of representing different kinds of hierarchies existing in real-world applications. In order to facilitate users to express their analysis needs, it is necessary to represent data requirements for data warehouses at the conceptual level. A conceptual multidimensional model should provide a graphical support (Rizzi, 2007) and allow representing facts, measures, dimensions, and different kinds of hierarchies.

BACKGROUND

Star and snowflake schemas comprise relational tables termed *fact* and *dimension tables*. An example of star schema is given in Figure 1.

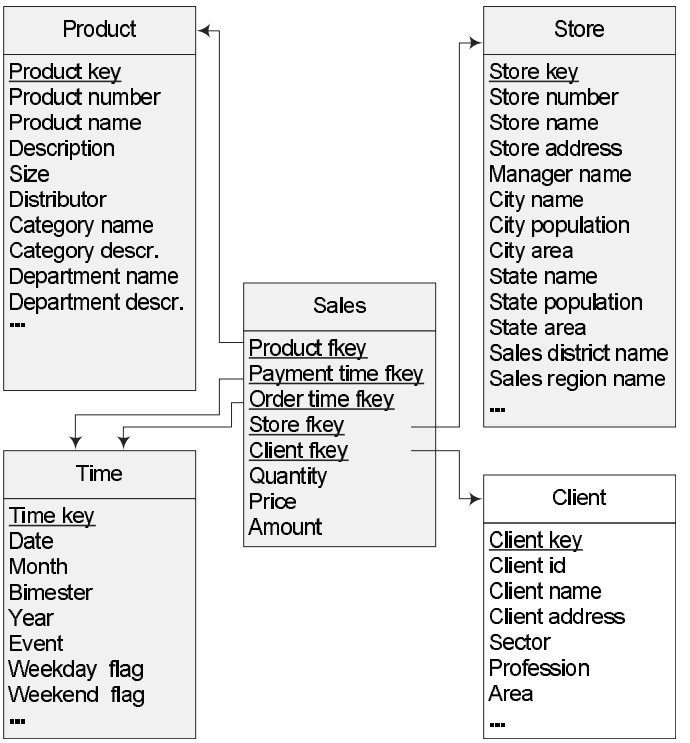
Fact tables, e.g., Sales in Figure 1, represent the focus of analysis, e.g., analysis of sales. They usually contain numeric data called *measures* representing the indicators being analyzed, e.g., Quantity, Price, and Amount in the figure. *Dimensions*, e.g., Time, Product, Store, and Client in Figure 1, are used for exploring the measures from different analysis perspectives. They often include attributes that form *hierarchies*, e.g., Product, Category, and Department in the Product dimension, and may also have descriptive attributes.

Star schemas have several limitations. First, since they use de-normalized tables they cannot clearly represent hierarchies: The hierarchy structure must be deduced based on knowledge from the application domain. For example, in Figure 1 is not clear whether some dimensions comprise hierarchies and if they do, what are their structures.

Second, star schemas do not distinguish different kinds of measures, i.e., additive, semi-additive, non-additive, or derived (Kimball & Ross, 2002). For example, Quantity is an additive measure since it can be summarized while traversing the hierarchies in all dimensions; Price is a non-additive measure since it cannot be meaningfully summarized across any dimension; Amount is a derived measure, i.e., calculated based on other measures. Although these measures require different handling during aggregation, they are represented in the same way.

Third, since star schemas are based on the relational model, implementation details (e.g., foreign keys) must be considered during the design process. This requires technical knowledge from users and also makes difficult the process of transforming the logical model to other models, if necessary.

Figure 1. Example of a star schema for analyzing sales



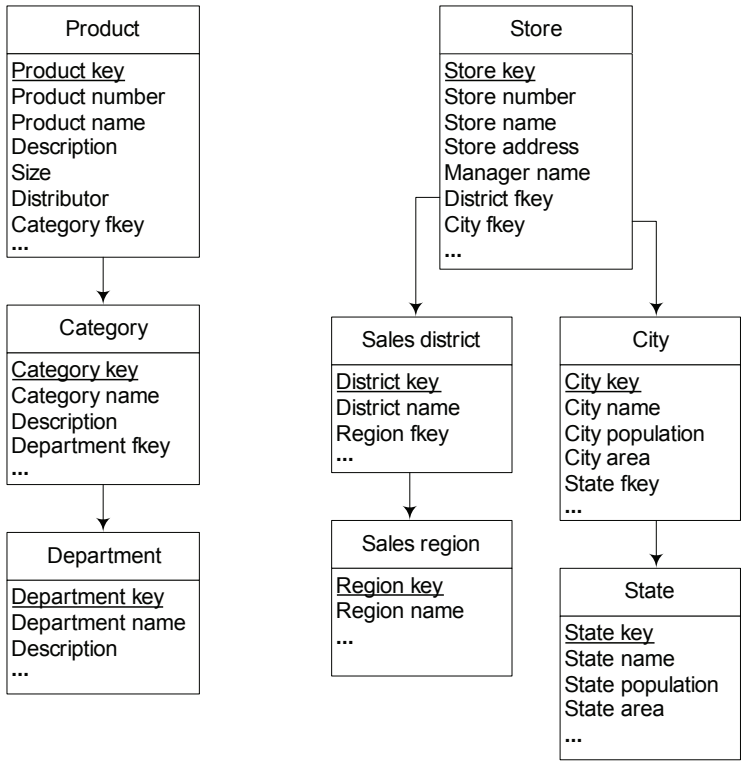
Fourth, dimensions may play different roles in a fact table. For example, the Sales table in Figure 1 is related to the Time dimension through two dates, the order date and the payment date. However, this situation is only expressed as foreign keys in the fact table that can be difficult to understand for non-expert users.

Snowflake schemas have the same problems as star schemas, with the exception that they are able to represent hierarchies. The latter are implemented as separate tables for every hierarchy level as shown in Figure 2 for the Product and Store dimensions. Nevertheless, snowflake schemas only allow representing simple hierarchies. For example, in the hierarchy in Figure 2 a) it is not clear that the same product can belong to several categories but for implementation purposes only the primary category is kept for each product. Furthermore, the hierarchy formed by the Store, Sales district, and Sales region tables does not accurately represent users' requirements: since small sales regions are not divided into sales districts, some stores must be analyzed using the hierarchy composed only of the Store and the Sales region tables.

Several conceptual multidimensional models have been proposed in the literature[1]. These models include the concepts of facts, measures, dimensions, and hierarchies. Some of the proposals provide graphical representations based on the ER model (Sapia, Blaschka, Höfling, & Dinter, 1998; Tryfona, Busborg, & Borch, 1999), on UML (Abelló, Samos, & Saltor, 2006; Luján-Mora, Trujillo, & Song, 2006), or propose new notations (Golfarelli & Rizzi, 1998; Hüsemann, Lechtenbörger, & Vossen, 2000), while other proposals do not refer to graphical representations (Hurtado & Gutierrez, 2007; Pourabbas, & Rafanelli, 2003; Pedersen, Jensen, & Dyreson, 2001; Tsois, Karayannidis, & Sellis, 2001).

Very few models distinguish the different types of measures and refer to role-playing dimensions (Kimball & Ross, 2002, Luján-Mora *et al.*, 2006). Some models do not consider the different kinds of hierarchies existing in real-world applications and only support simple hierarchies (Golfarelli & Rizzi, 1998; Sapia *et al.*, 1998). Other models define some of the hierarchies described in the next section (Abelló *et al.*, 2006; Bauer, Hümmer,

Figure 2. Snowflake schemas for the a) product and b) store dimensions from Figure 1



(a) *(b)*

& Lehner, 2000; Hüsemann *et al.,* 2000; Hurtado & Gutierrez, 2007; Luján-Mora *et al.*, 2006; Pourabbas, & Rafanelli, 2003; Pedersen *et al.*, 2001; Rizzi, 2007). However, there is a lack of a general classification of hierarchies, including their characteristics at the schema and at the instance levels.

MAIN FOCUS

We present next the MultiDim model (Malinowski & Zimányi, 2004; Malinowski & Zimányi, 2008), a conceptual multidimensional model for data warehouse and OLAP applications. The model allows representing different kinds of hierarchies existing in real-world situations.

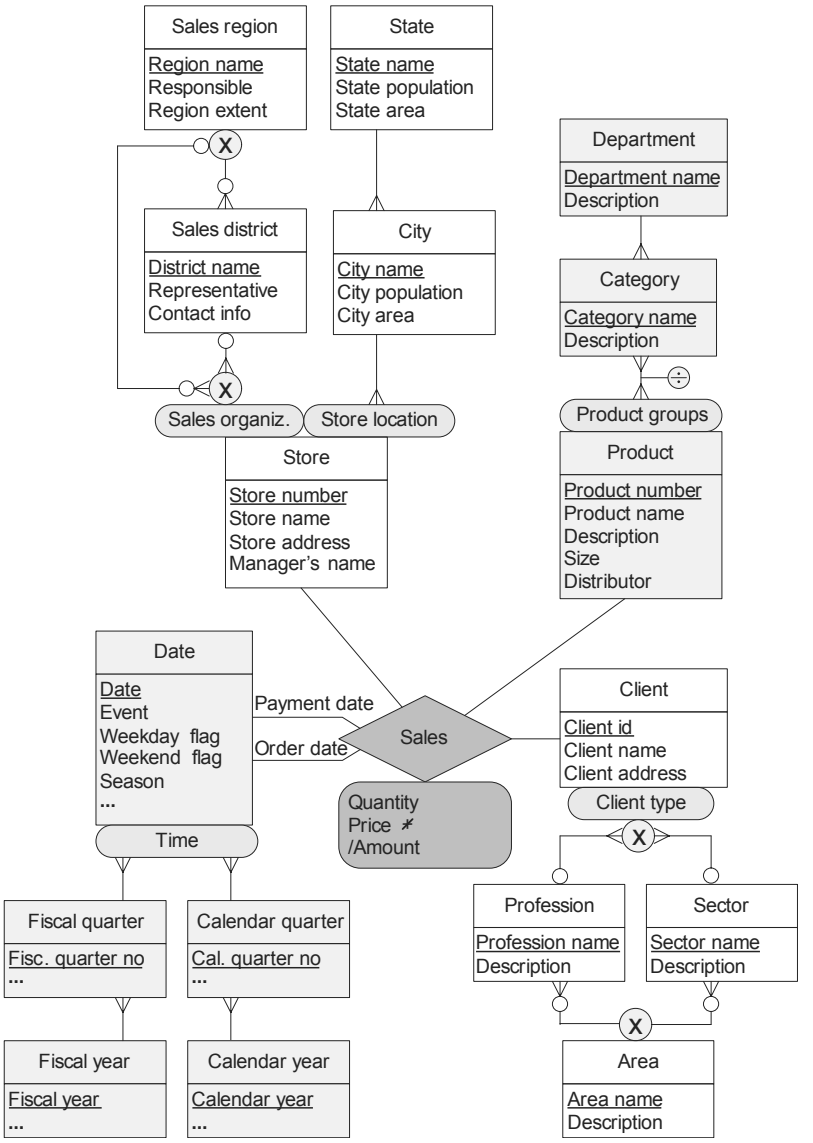
The MultiDim Model

To describe the MultiDim model we use the schema in Figure 3, which corresponds to the relational schemas in Figure 1. This schema also contains different types of hierarchies that are defined in next section.

A *schema* is composed of a set of levels organized into dimensions and a set of fact relationships. A *level* corresponds to an entity type in the ER model; instances of levels are called *members*. A level has a set of attributes describing the characteristics of their members. For example, the Product level in Figure 3 includes the attributes Product number, Product name, etc. In addition, a level has one or several keys (underlined in the figure) identifying uniquely the members of a level.

A *fact relationship* expresses the focus of analysis and represents an n-ary relationship between levels. For example, the Sales fact relationship in Figure 3 relates the Product, Date, Store, and Client levels. Further, the same level can participate several times in a fact relationship playing different roles. Each *role* is identified by a name and is represented by a separate link between the level and the fact relationship, as can be seen for the roles Payment date and Order date relating the Date level to the Sales fact relationship.

Figure 3. A conceptual multidimensional schema of a sales data warehouse



A fact relationship may contain attributes commonly called *measures*, e.g., Quantity, Price, and Amount in Figure 3. Measures are classified as *additive*, *semi-additive*, or *non-additive* (Kimball & Ross, 2002). By default we suppose that measures are additive. For semi-additive and non-additive measures we use, respectively, the symbols +! and ≠ (the latter is shown for the Price measure). For derived measures and attributes we use the symbol / in front of the measure name, as shown for the Amount measure.

A *dimension* is an abstract concept grouping data that shares a common semantic meaning within the domain being modeled. It is composed of either one level or one more hierarchies.

Hierarchies are used for establishing meaningful aggregation paths. A hierarchy comprises several related levels, e.g., the Product, Category, and Department levels. Given two related levels, the lower level is called *child*, the higher level is called *parent*, and the relationship between them is called *child-parent relationship*. Key attributes of a parent level define how

child members are grouped. For example, in Figure 3 the Department name in the Department level is used for grouping different category members during roll-up operations. A level that does not have a child level is called *leaf*; it must be the same for all hierarchies in a dimension. The leaf level name is used for defining the dimension's name. The level that does not have a parent level is called *root.*

Child-parent relationships are characterized by *cardinalities*, indicating the minimum and the maximum number of members in one level that can be related to members in another level. The notations used for representing cardinalities are as follows: —< (1,n), —o< (0,n), —— (1,1), and —o (0,1). For example, in Figure 3 the child level Store is related to the parent level City with a many-to-one cardinality, which means that every store belongs to only one city and each city can have many stores.

Since the hierarchies in a dimension may express different conceptual structures used for analysis purposes, we include an *analysis criterion* to differentiate them. For example, the Store dimension includes two hierarchies: Store location and Sales organization.

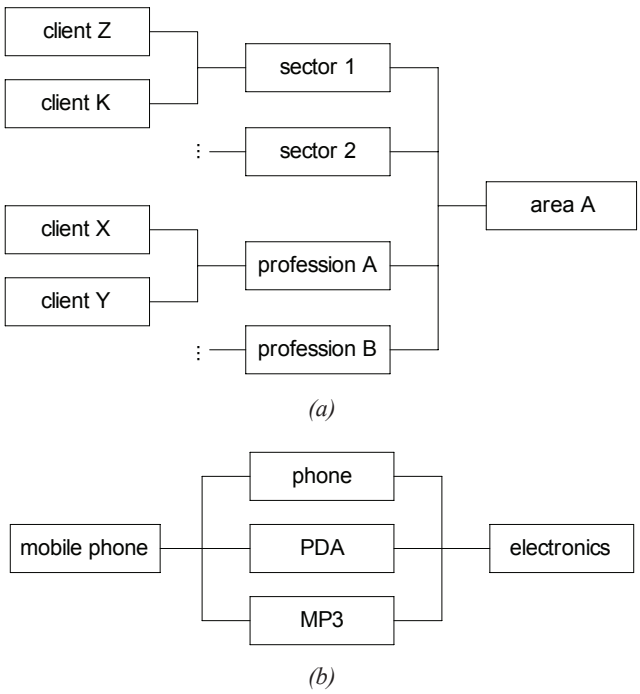
Classification of Hierarchies

We propose next a classification of hierarchies considering their differences at the schema and at the instance levels. Figure 4 shows examples of members for the Customer and Product dimensions from Figure 3. The distinction at the instance level is important since different aggregation procedures are required depending on whether the hierarchy members form a tree or an acyclic graph. This may be deduced from the cardinalities included in the schema.

We distinguish the following types of hierarchies.

- **Simple:** In these hierarchies the relationship between their members can be represented as a tree. They can be of different types.
  - **Balanced:** An example of this hierarchy is the Store location in Figure 3. At the schema level there is only one path where all levels are mandatory. At the instance level the members form a tree where all the branches have the same length. A parent member has one or several child members (at least one)

Figure 4. Examples of members for a) the Customer and b) the Product dimensions



and a child member belongs to only one parent member: the cardinality of child roles is (1,1) and that of parent roles is (1,n).

- **Unbalanced:** This type of hierarchy is not included in Figure 3; however, they are present in many data warehouse applications. For example, a bank may include a hierarchy composed by the levels ATM, Agency, and Branch. However, some agencies may not have ATMs and small branches may not have any organizational division. This kind of hierarchy has only one path at the schema level. However, since at the instance level, some parent members may not have associated child members, the cardinality of the parent role is (0,n).
- **Generalized:** The Client type hierarchy in Figure 3 with instances in Figure 4 a) belongs to this type. In this hierarchy a client can be a person or a company having in common the Client and Area levels. However, the buying behavior of clients can be analyzed according to the specific level Profession for a person type, and Sector for a company type. This kind of hierarchy has at the schema level multiple exclusive paths sharing some levels. All these paths represent one hierarchy and account for the same analysis criterion. At the instance level each member of the hierarchy only belongs to one path. We use the symbol ⊗ for indicating that the paths are exclusive. **Non-covering** hierarchies (the Sales organization hierarchy in Figure 3) are generalized hierarchies with the additional restriction that the alternative paths are obtained by skipping one or several intermediate levels. At the instance level every child member has only one parent member.

- **Non-strict:** An example is the Product groups hierarchy in Figure 3 with members in Figure 4 b). This hierarchy models the situation when mobile phones can be classified in different products categories, e.g., phone, PDA, and MP3 player. Non-strict hierarchies have at the schema level at least one many-to-many cardinality, e.g., between the Product and Category levels in Figure 3. A hierarchy is called *strict* if all cardinalities are many-to-one. Since at the instance level, a child member may have more than one parent member, the members form an acyclic graph. To indicate how the measures are distributed between several parent members, we include a distributing factor symbol ⊜. The different kinds of hierarchies previously presented can be either strict or non-strict.
- **Alternative:** The Time hierarchy in the Date dimension is a alternative hierarchy. At the schema level there are several non-exclusive simple hierarchies sharing at least the leaf level, all these hierarchies accounting for the same analysis criterion. At the instance level such hierarchies form a graph since a child member can be associated with more than one parent member belonging to different levels. In such hierarchies it is not semantically correct to simultaneously traverse the different composing hierarchies. Users must choose one of the alternative hierarchies for analysis, e.g., either the hierarchy composed by Date, Fiscal quarter, and Fiscal year or the one composed by Date, Calendar quarter, and Calendar year.
- **Parallel:** A dimension has associated several hierarchies accounting for different analysis criteria. Parallel hierarchies can be of two types. They are *independent,* if the composing hierarchies do not share levels; otherwise, they are *dependent*. The Store dimension includes two parallel independent hierarchies: Sales organization and Store location. They allow analyzing measures according to different criteria.

The schema in Figure 3 clearly indicates users' requirements concerning the focus of analysis and the aggregation levels represented by the hierarchies. It also preserves the characteristics of star or snowflake schemas providing at the same time a more abstract conceptual representation. Notice that even though the schemas in Figures 1 and 3 include the same hierarchies, they can be easily distinguished in the Figure 3 while this distinction cannot be made in Figure 1.

Existing multidimensional models do not consider all types of hierarchies described above. Some models give only a description and/or a definition of some of the hierarchies, without a graphical representation. Further, for the same type of hierarchy different names are used in different models. Strict hierarchies are included explicitly or implicitly in all proposed models.

Since none of the models takes into account different analysis criteria, alternative and parallel hierarchies cannot be distinguished. Further, very few models propose a graphical representation for the different hierarchies that facilitate their distinction at the schema and instance levels.

To show the feasibility of implementing the different types of hierarchies, we present in Malinowski and Zimányi, (2006, 2008) their mappings to the relational model and give examples of their implementation in Analysis Services and Oracle OLAP.

FUTURE TRENDS

Even though some of the presented hierarchies are considered as an advanced feature of multidimensional models (Torlone, 2003), there is a growing interest in having them in the research community and in commercial products.

Nevertheless, several issues have yet to be addressed. It is necessary to develop aggregation procedures for all types of hierarchies defined in this paper. Some proposals for managing them exist (e.g., Pedersen *et al.*, 2001; Abelló *et al.*, 2006; Hurtado & Gutierrez, 2007). However, not all hierarchies are considered and some of the proposed solutions may not be satisfactory for users since they transform complex hierarchies into simpler ones to facilitate their manipulation.

Another issue is to consider the inclusion of other ER constructs in the multidimensional model, such as weak entity types, multivalued or composite attributes. The inclusion of these features is not straightforward and requires analysis of their usefulness in multidimensional modeling.

CONCLUSION

Data warehouse and OLAP applications use a multidimensional view of data including dimensions, hierarchies, facts, and measures. In particular, hierarchies are important in order to automatically aggregate the measures for analysis purposes.

We advocated that it is necessary to represent data requirements for data warehouse and OLAP applications at a conceptual level. We proposed the MultiDim model, which includes graphical notations for the different elements of a multidimensional model. These notations allow a clear distinction of each hierarchy type taking into account their differences at the schema and at the instance levels.

We also provided a classification of hierarchies. This classification will help designers to build conceptual models of multidimensional databases. It will also give users a better understanding of the data to be analyzed, and provide a better vehicle for studying how to implement such hierarchies using current OLAP tools. Further, the proposed hierarchy classification provides OLAP tool implementers the requirements needed by business users for extending the functionality offered by current tools.

REFERENCES

Abelló, A., Samos, J., & Saltor, F. (2006). YAM^2: a multidimensional conceptual model extending UML. *Information Systems*, *32*(6), pp. 541-567.

Bauer, A., Hümmer, W., & Lehner, W. (2000). An alternative relational OLAP modeling approach. *Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery*, pp. 189-198. Lectures Notes in Computer Sciences, N° 1874. Springer.

Golfarelli, M., & Rizzi, S. (1998). A methodological framework for data warehouse design. *Proceedings of the 1st ACM International Workshop on Data Warehousing and OLAP*, pp. 3-9.

Hurtado, C., & Gutierrez, C. (2007). Handling structural heterogeneity in OLAP. In R. Wrembel & Ch. Koncilia (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, pp. 27-57. Idea Group Publishing.

Hüsemann, B., Lechtenbörger, J., & Vossen, G. (2000). Conceptual data warehouse design. *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouses*, p. 6.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit*. John Wiley & Sons Publishers.

Luján-Mora, S., Trujillo, J. & Song, I. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering*, *59*(3), pp. 725-769.

Malinowski, E. & Zimányi, E. (2004). OLAP hierarchies: A conceptual perspective. *Proceedings of the 16th International Conference on Advanced Information Systems Engineering*, pp. 477-491. Lectures Notes in Computer Sciences, N° 3084. Springer.

Malinowski, E. & Zimányi, E. (2006). Hierarchies in a multidimensional model: from conceptual modeling to logical representation. *Data & Knowledge Engineering*, 59(2), pp. 348-377.

Malinowski, E. & Zimányi, E. (2008). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer.

Pedersen, T., Jensen, C.S., & Dyreson, C. (2001). A foundation for capturing and querying complex multidimensional data. *Information Systems, 26*(5), pp. 383-423.

Pourabbas, E., & Rafanelli, M. (2003). Hierarchies. In Rafanelli, M. (Ed.) *Multidimensional Databases: Problems and Solutions,* pp. 91-115. Idea Group Publishing.

Rizzi, S. (2007). Conceptual modeling solutions for the data warehouse. In R. Wrembel & Ch. Koncilia (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, pp. 1-26. Idea Group Publishing.

Sapia, C., Blaschka, M., Höfling, G., & Dinter, B. (1998). Extending the E/R model for multidimensional paradigms. *Proceedings of the 17th International Conference on Conceptual Modeling*, pp. 105-116. Lectures Notes in Computer Sciences, N° 1552. Springer.

Torlone, R. (2003). Conceptual multidimensional models. In Rafanelli, M. (Ed.) *Multidimensional Databases: Problems and Solutions*, pp. 91-115. Idea Group Publishing.

Tryfona, N., Busborg, F., & Borch, J. (1999). StarER: A Conceptual model for data warehouse design. *Proceedings of the 2nd ACM International Workshop on Data Warehousing and OLAP*, pp. 3-8.

Tsois, A., Karayannidis, N., & Sellis, T. (2001). MAC: Conceptual data modeling for OLAP. *Proceedings of the 3rd International Workshop on Design and Management of Data Warehouses*, p. 5.

KEY TERMS

Conceptual Model: A model for representing schemas that are designed to be as close as possible to users' perception, not taking into account any implementation considerations.

MultiDim Model: A conceptual multidimensional model used for specifying data requirements for data warehouse and OLAP applications. It allows one to represent dimensions, different types of hierarchies, and facts with associated measures.

Dimension: An abstract concept for grouping data sharing a common semantic meaning within the domain being modeled.

Hierarchy: A sequence of levels required for establishing meaningful paths for roll-up and drill-down operations.

Level: A type belonging to a dimension. A level defines a set of attributes and is typically related to other levels for defining hierarchies.

Multidimensional Model: A model for representing the information requirements of analytical applications. A multidimensional model comprises facts, measures, dimensions, and hierarchies.

Star Schema: A relational schema representing multidimensional data using de-normalized relations.

Snowflake Schema: A relational schema representing multidimensional data using normalized relations.

ENDNOTE

1 A detailed description of proposals for multidimensional modeling can be found in Torlone (2003).