OLAP Hierarchies: A Conceptual Perspective*

Elzbieta Malinowski** and Esteban Zimányi

Department of Informatics CP 165/15
Université Libre de Bruxelles, 50 av. F.D. Roosevelt
1050 Brussels, Belgium
{emalinow,ezimanyi}@ulb.ac.be

Abstract. OLAP (On-Line Analytical Processing) tools support the decision-making process by giving users the possibility to dynamically analyze high volumes of historical data using operations such as roll-up and drill-down. These operations need well-defined hierarchies in order to prepare automatic calculations. However, many kinds of complex hierarchies arising in real-world situations are not addressed by current OLAP implementations. Based on an analysis of real-world applications and scientific works related to multidimensional modeling, this paper presents a conceptual classification of hierarchies and proposes graphical notations for them based on the ER model. A conceptual representation of hierarchies allows the designer to properly represent users' requirements in multidimensional modeling and offers a common vision of these hierarchies for conceptual modeling and OLAP tools implementers.

1 Introduction

Decision-making users increasingly rely on Data Warehouses (DWs) as an important platform for data analysis. DW architecture and tools provide the access to historical data with the aim of supporting the strategic decisions of organizations.

The structure of a DW is usually represented using the star/snowflake schema, also called multidimensional schema, consisting of fact tables, dimension tables, and hierarchies [9].

A *fact table* represents the subject orientation and the focus of analysis. It typically contains *measures* that are attributes representing the specific elements of analysis, such as quantity sold, sales. A *dimension table* includes attributes allowing the user to explore the measures from different perspectives of analysis. These attributes are called *dimension levels*[1] [5, 6, 7]. The dimension levels may form a hierarchy, such as City – State – Country, allowing the user to see detailed as well as generalized data. Further, a dimension may also have descriptive attributes called *property attributes*[2][8] that are not used in hierarchy, such as Store number, E-mail address, etc.

[*] This work was funded by a scholarship of the Cooperation Department of the Université Libre de Bruxelles.

[**] Currently on leave from the Universidad de Costa Rica.

[1] Other names, such as *dimension attributes* [5] or *category attributes* [15] are also used.

[2] They also are called *non-dimension attributes* [5].

A. Persson and J. Stirna (Eds.): CAiSE 2004, LNCS 3084, pp. 477–491, 2004.

Property attributes are orthogonal to dimension levels and they complement each other [2].

On-Line Analytical Processing (OLAP) tools allow decision-making users to dynamically manipulate the data contained in a DW. OLAP tools use hierarchies for allowing both a general and a detailed view of data using operations such as drill-down and roll-up.

Although OLAP tools have been successfully used in decision-support systems for many years, they can only manage a limited number of hierarchies comparing to those existing in real-world applications. Usually, OLAP tools only cope with hierarchies that ensure summarizability or that can be transformed so that summarizability conditions hold [10, 11]. Summarizability refers to the correct aggregation of measures in a higher hierarchy level taking into account existing aggregations in a lower hierarchy level.

In this paper, we adopt a conceptual approach and propose an extension of the ER model for representing hierarchies appearing in real-world applications as well as a categorization of such hierarchies. The benefits of using conceptual models in database design have been acknowledged for several decades; however, the domain of conceptual design for multidimensional modeling is still at a research stage. As stated in [17], the analysis of achievements in data warehouse research showed the little interest of the research community in conceptual multidimensional modeling. The proposed conceptual models do not cope with the different kinds of hierarchies existing in real-world applications. Some of these models formally define and offer special notations for commonly-known hierarchies. In many cases, these notations can be extended to manage other kinds of hierarchies proposed in this work. However, there is a lack of a general classification of hierarchies, including their characteristics at the schema and at the instance levels. This situation leads to repeated research efforts in "rediscovering" hierarchies and providing solutions for managing them.

Presenting the different kinds of hierarchies in a systematic way will help OLAP tools implementers to focus on implementation issues, e.g., they can enhance the system performance materializing the aggregated measures of common hierarchy levels or ensure meaningful combination of hierarchies. Further, a DW is in continuous development, coping with new analysis requirements and incorporating new structures. Therefore, a conceptual representation of the hierarchies facilitates this evolution and mitigates the possible technology changes. Furthermore, conceptual hierarchies establish a better communication bridge between decision-making users and designer/implementer.

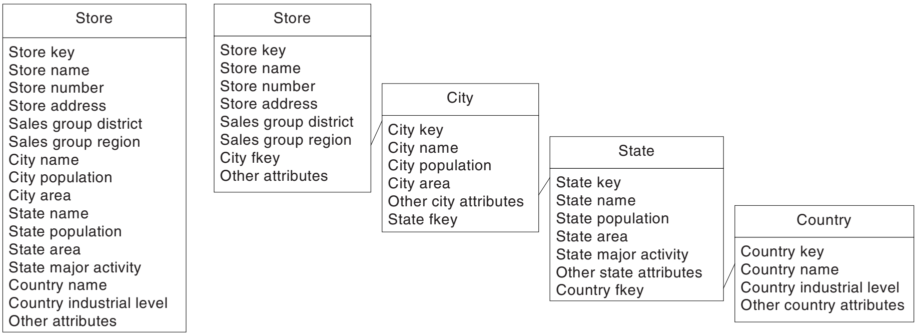
This paper is organized as follows. Section 2 proposes different kinds of hierarchies including their notations. Section 3 refers to works related to representing hierarchies in different multidimensional models, and Section 4 gives conclusions and future perspectives.

2 Hierarchies and Their Categorization

We define a hierarchy as follows. A *hierarchy* is a set of binary relationships between dimension levels. A dimension level participating in a hierarchy is called *hierarchical level* or in short *level*. The sequence of these levels is called a *hierarchical path* or in short *path*. The number of levels forming a path is called the *path length*. The first level of a hierarchical path is called *leaf* and the last is called *root*. The root represents the most generalized view of data. Given two consecutive levels of a hierarchy, the higher level is called *parent* and the lower level is called *child*. Every instance of a level is called *member*. The *cardinality* indicates the minimum and maximum numbers of members in one level that can be related to a member of another level.

Even though in some works the root of a hierarchy is represented using a level called ALL, we consider that its inclusion in a conceptual graphical representation can be ambiguous or meaningless for decision-making users.

Hierarchies are usually presented using a *flat table*, as shown in Figure 1 a), or using a normalized structure called *snowflake scheme*, as shown in Figure 1 b) [9].



a) b)

Fig. 1. Flat table (a) and snowflake (b) representations of hierarchies.

Instead of representing hierarchies at a logical level, we propose to adopt a conceptual perspective and use the ER-like graphical notations shown in Figure 2. Some of these notations are inspired from [18, 19, 20].

A fact relationship represents an n-ary relationship among several dimensions; it can contain attributes called *measures* describing this relationship. Since the roles of a fact relationship always have (0,N) cardinality, we omit such cardinalities to simplify the model.

On the other hand, the relationship linking the levels of a hierarchy is a usual binary relationship and can be represented with the same symbol as the fact relationship. However, since such relationships are only used for traversing from one level to the next one, we propose to represent hierarchies using the graphical notation of Figure 2 c). Notice that if a hierarchy level has specific property attributes, they can be included without ambiguity (Figure 2 c). This enriches the expression power of the model and gives more clarity by grouping the property attributes into their corresponding levels.

Hierarchies can express different structures according to the criteria used for analysis, for example, geographical location, organizational structure, etc. We propose to represent these criteria in the model using a special symbol (Figure 2 e) linking the dimension to be analyzed (leaf level) with the hierarchy. Making this criterion explicit is important since, as will be shown in Section 2.4, the same dimension (e.g., an employee) can be analyzed according to several criteria (e.g., home location and work location), possibly sharing the same hierarchy levels.

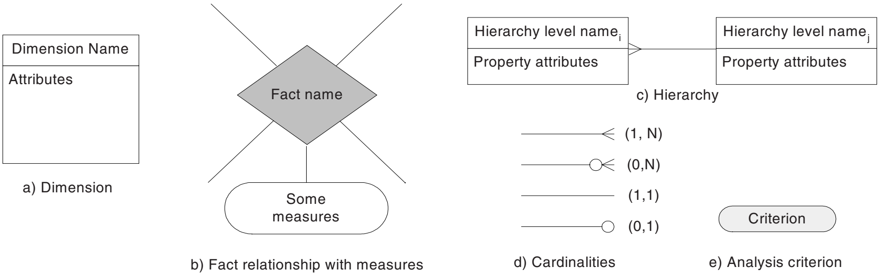


Fig. 2. Notations for multidimensional model: (a) dimension, (b) fact relathiship, (c) hierarchy, (d) cardinalities, and (e) analysis criterion.

2.1 Simple Hierarchies

Simple hierarchies are those hierarchies where the relationship between their members can be represented as a tree. Further, these hierarchies use only one criterion for analysis. Simple hierarchies can be further categorized into symmetric, asymmetric, and generalized hierarchies.

Figure 3 shows the symmetric hierarchy from Figure 1 b) using the proposed notations. A *symmetric hierarchy*[3] has at the schema level only one path where all levels are mandatory (Figure 3 a). At the instance level the members form a tree where all the branches have the same length (Figure 3 b). As implied by the cardinalities, all parent members must have at least one child member and a child member cannot belong to more than one parent member. In commercial systems, this kind of hierarchies is commonly represented at the logical level using a star/snowflake schema.

Another type of simple hierarchy is called an *asymmetric hierarchy*[4] [9]. Such hierarchies have only one path at the schema level (Figure 4 a) but, as implied by the cardinalities, some lower levels of the hierarchy are not mandatory. Thus, at the instance level the members represent a non-balanced tree (Figure 4 b), i.e., the branches of the tree have different lengths since some parent members will not have associated child members. As for symmetric hierarchies, the cardinalities imply that every child member may belong to at most one parent member.

Figure 4 a) shows a hierarchy where a bank consists of several branches: some of them have agencies with ATM, some only agencies, and small branches do not have any organizational divisions. Figure 4 b) illustrates some instances of this hierarchy. As shown in the figure, cardinalities clearly indicate the levels that may be leaves for some branches of the tree, as it is the case for Branch, since it can be related to zero or more Agencies.

Asymmetrical hierarchies are not easy to manage at the implementation level. A typical solution either transforms an asymmetrical hierarchy into a symmetrical one, introducing placeholders for the shorter branches of the tree [13, 15], or uses the representation of the parent-child relationships existing between the hierarchy members [13, 14].

[3] Such hierarchies are also called *homogenous* [7], *balanced* [13], or *level-based* [14].

[4] Several terms are used: *non-balanced* [13], *non-onto* [15].

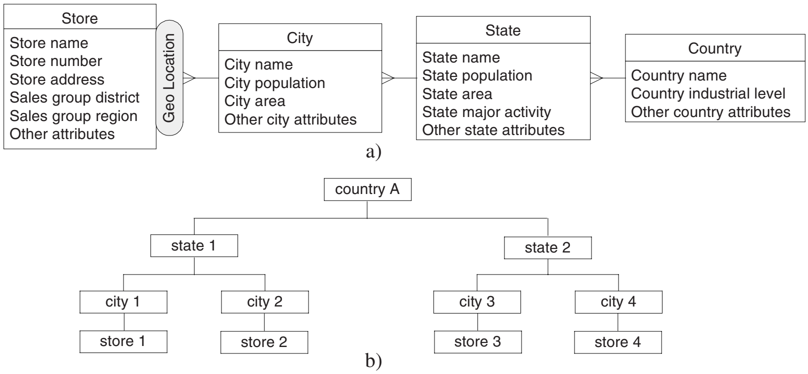


Fig. 3. A symmetric hierarchy (a) model and (b) example of instances.

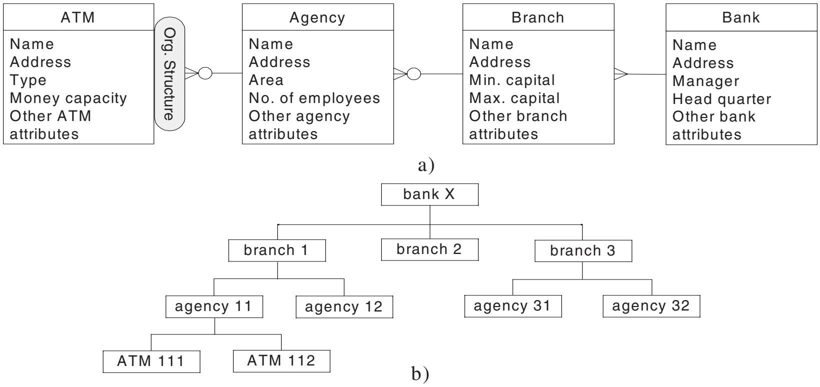


Fig. 4. An asymmetric hierarchy: (a) schema and (b) examples of instances.

Other kinds of hierarchies existing in real-world applications have been explored in the literature, although no commercial system cope with them. Sometimes a dimension includes subtypes that can be represented by the generalization/specialization relationship [1, 12]. Moreover, the specialized subtypes can include their own hierarchy. Figure 5 shows an example where the Customer dimension can be specialized into Company and Person, each of them having its own hierarchy: the hierarchy for the subtype Company consists on the levels Company Type-Sector-Branch-Area while the hierarchy for the Person subtype is formed by the levels Profession Name-Category-Branch-Area. Both subtypes include common levels for Branch and Area.

While both hierarchies can be represented independently repeating the common levels, the complexity of the schema reduces if it is possible to represent shared levels characterized by the same granularity of aggregation. Also, to ensure adequate measure aggregations, the distinction between specific and common hierarchy levels should be clearly represented in the model [6, 7]. However, existing multidimensional conceptual models do not offer this distinction. This work proposes the graphical notation shown in Figure 6 where the common and specific hierarchy levels are represented. We call such hierarchies generalized.

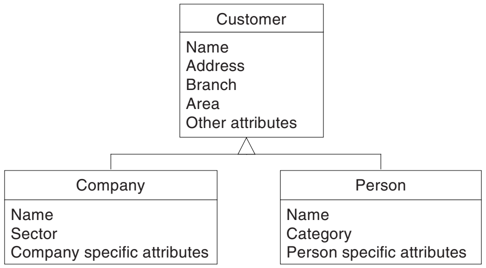
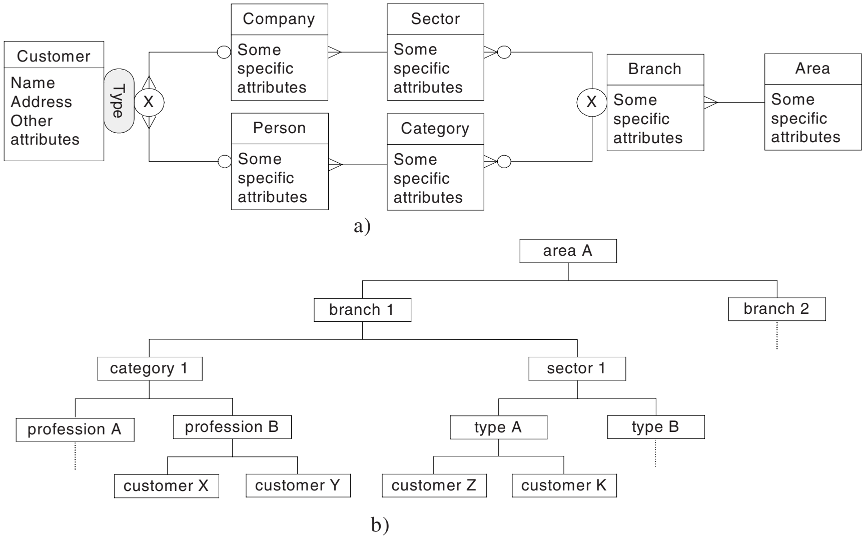


Fig. 5. Specialization of Customer into Company and Person with common and specific hierarchy levels.



a)

b)

Fig.6. A generalized hierarchy for a sales company: (a) schema and (b) examples of instances.

At the schema level a *generalized hierarchy* can contain 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 can belong to only one of the paths. We propose to include the symbol Ⓧ to indicate that paths are exclusive. Such a notation is equivalent to the {xor} annotation used in UML [3].

Currently, it is not possible to manage generalized hierarchies in commercial tools. If the members differ in hierarchy structure, the common solution is to treat each of the subtypes as a separate dimension with its own hierarchy. This solution has the disadvantages of not allowing to analyze the data according to common hierarchy levels and makes the structure more complex.

Generalized hierarchies include a special case commonly referred as *non-covering hierarchies* [13, 15]. An example of this kind of hierarchy is illustrated in Figure 7 representing a company that has offices in different countries; however, the geographical division in these countries is different, e.g., skipping the division into counties for some countries.

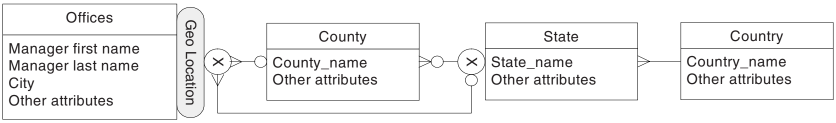


Fig. 7. A non-covering hierarchy.

Thus, a non-covering hierarchy is a special case of a generalized hierarchy with the additional restrictions that at the schema level (1) the root and the leaves are the same for all paths and (2) only one (or several) intermediate level(s) can be skipped without including additional levels.

A logical-level implementation for generalized hierarchies will either (1) ignore the specific levels and use only the common levels for roll-up and drill-down operations, or (2) take into account the existing hierarchy levels for every hierarchy member. Some proposals of logical-level solutions for this kind of hierarchy have been already described [2, 5, 10] and used in commercial systems [13]. For example, Microsoft Analysis Services [13] allows the definition and manipulation of non-covering hierarchies called *ragged hierarchies*. They can be represented in a flat table or in a parent-child structure. It is possible to display such hierarchies in different ways using the dimension properties *Hide Member If* and *Visible*.

2.2 Non-strict Hierarchies

For the simple hierarchies presented before we assumed that each link between a parent and child levels has one-to-many cardinalities, i.e., a child member is related to at most one parent member and a parent member may be related to several child members. However, a many-to-many cardinality between parent and child levels is very common in real-life applications: e.g., an employee working in several departments, a diagnosis belonging to several diagnosis groups [15], a week that may belong to two consecutive months, etc.

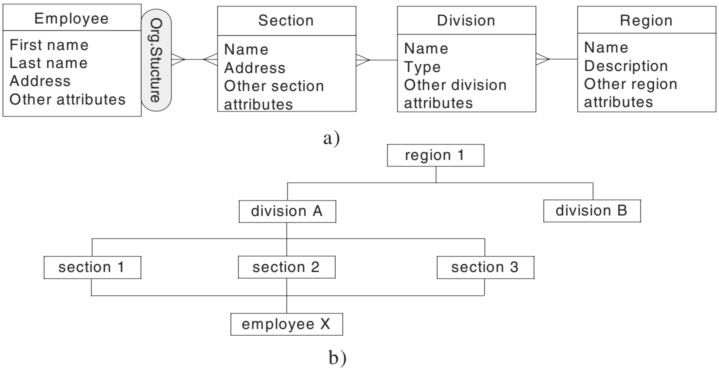
We call a hierarchy *non-strict* if at the schema level it has at least one many-to-many cardinality; it is called *strict* if all cardinalities are one-to-many. The fact that a hierarchy is strict or not is orthogonal to its type. Thus, the different kinds of hierarchies already presented can be either strict or non-strict.

Figure 8 a) shows a symmetric non-strict hierarchy where employees may work in several sections. Since at the instance level, a child member may have more than one parent member, the members of the hierarchy form a graph (Figure 8 b).

There are several possibilities to manage this kind of hierarchy at the logical level. For example, using a *bridge table* [9], transforming the non-strict hierarchy into a strict hierarchy [15], or changing the model into the dimensional normal form proposed by [10].

2.3 Multiple Hierarchies

Multiple hierarchies represent the situation where at the schema level there are several non-exclusive simple hierarchies sharing some levels. However, all these hierar-



a)

b)

Fig. 8. A symmetric non-strict hierarchy: (a) model and (b) example of instances.

chies account for the same analysis criterion[5]. 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. The nodes at which the parallel hierarchies split and join are called, respectively, *splitting level* and *joining level.*

Multiple hierarchies can be inclusive and alternative. In a *multiple inclusive hierarchy* all members of the splitting level participate simultaneously in parent levels belonging to different hierarchies. Thus, when traversing the hierarchy (e.g., during roll-up operation), the measure presented in a fact relationship must be distributed between several hierarchies. We use the symbol ⨸ to represent the requirement of measure distribution.

An example of a multiple hierarchy is presented in Figure 9 where sport club' activities are classified into association and recreation programs. Here, both hierarchies are simple and share the level of Regional Committee. They represent the same analysis criterion of sport activity type. Currently, multiple inclusive hierarchies are not considered in OLAP tools.

In *multiple alternative* hierarchies it is not semantically correct to simultaneously traverse the different composing hierarchies. The user must choose one of the alternative hierarchies for analysis. An example is given in Figure 10 where the Time dimension includes two hierarchies corresponding to calendar periods: Month-Quarter-Year and Week-Year. As can be seen in the example it is meaningless to combine both hierarchies.

Multiple alternative hierarchies can be implemented in commercial tools, for example in Microsoft Analysis Services [13]. However, meaningless intersections may appear since there are no restrictions for simultaneously combining the simple hierarchies composing a multiple hierarchy.

Notice that even though generalized and multiple hierarchies share some levels and use only one analysis criterion, they can be easily distinguished from each other.

[5] As we will see in Section 2.4, hierarchies with multiple paths accounting for different criteria are called parallel hierarchies.

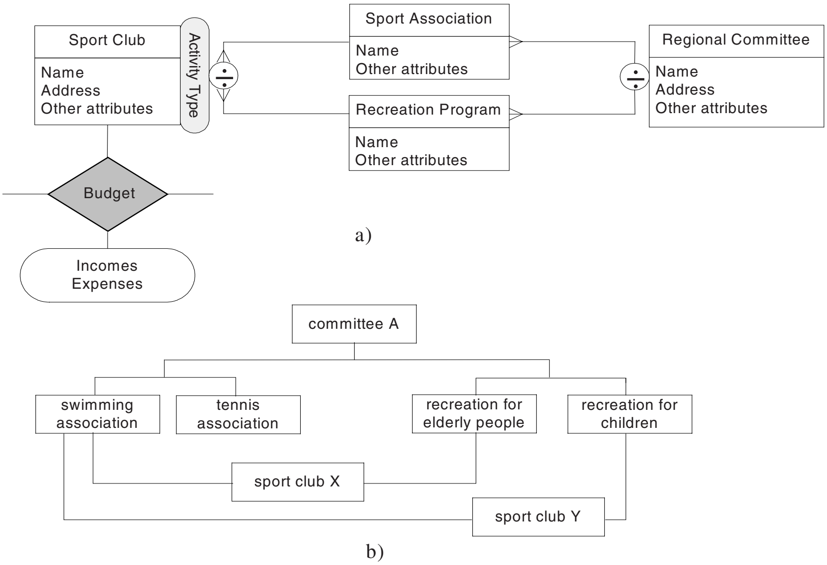


Fig. 9. Example of a multiple inclusive hierarchy: (a) model and (b) instances.

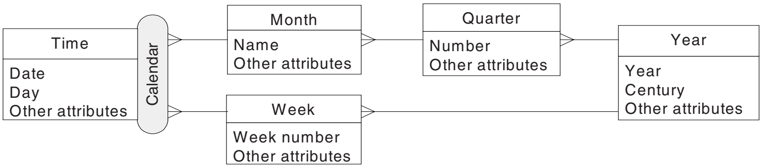


Fig. 10. A multiple alternative hierarchy.

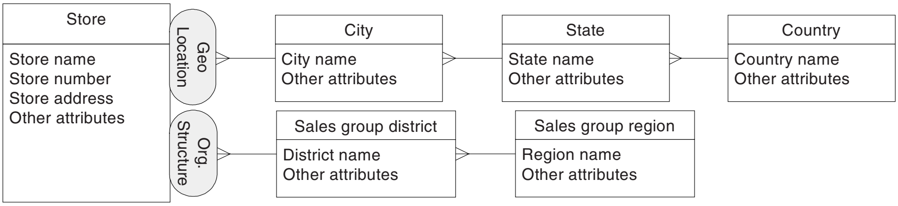


Fig. 11. Parallel independent hierarchies associated to one dimension.

2.4 Parallel Hierarchies

Parallel hierarchies arise when a dimension has associated several hierarchies accounting for different analysis criteria. These hierarchies can be composed of different kinds of hierarchies presented before. Such hierarchies can be independent or dependent.

In a *parallel independent hierarchy,* the different hierarchies do not share levels, i.e., they represent non-overlapping sets of hierarchies. An example is given in Figure 11.

In contrast, *parallel dependent hierarchies*, have different hierarchies sharing some levels. An example is given in Figure 12. It represents an international company that sells products and analyzes the achievements of their employees. The dimension Sales Employee contains two hierarchies: one symmetric that represents the sales organization structure and other one, non-covering, that represents the geographic division of the address of the sales employee. Both hierarchies share the common levels of State and Country.

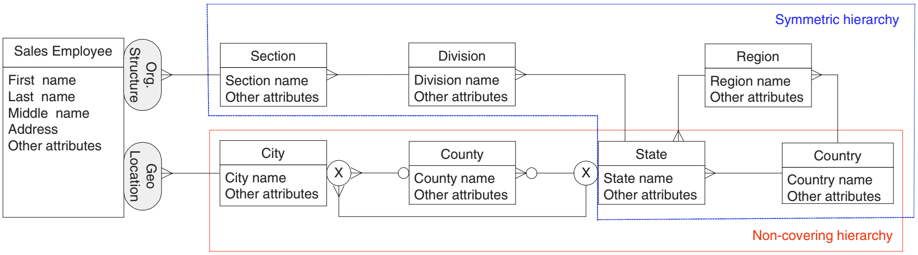


Fig. 12. Parallel dependent hierarchies including symmetric and non-covering hierarchies.

Notice that both multiple and parallel dependent hierarchies share some level(s). However, the main distinction between them is that parallel dependent hierarchies account for different analysis criteria. Thus, the user can use them independently during the analysis process. Moreover, for multiple alternative hierarchies as well as for parallel dependent hierarchies the fact that the shared level(s) are explicitly represented is important since, at the implementation level, the aggregated measures for these levels can be reused. For example, Microsoft OLAP Server [13] already adopted this solution for their multiple alternative hierarchies.

2.5 Metamodel

This section summarizes the proposed classification of hierarchies using the metamodel of Figure 13.

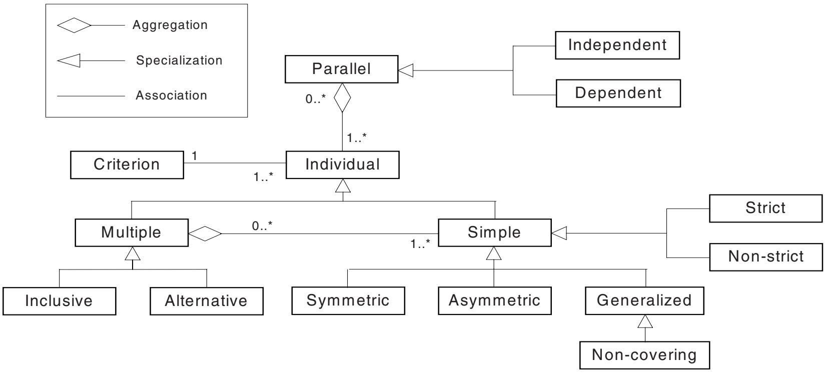


Fig. 13. Metamodel of hierarchy classification.

Parallel hierarchies can be specialized into independent and dependent hierarchies. A parallel hierarchy is an aggregation of individual hierarchies. Each individual hierarchy is associated with one analysis criterion, while the same criterion can be used for many individual hierarchies. An individual hierarchy can be simple or multiple.

Simple hierarchies are divided into symmetric, asymmetric, and generalized hierarchies. Also, generalized hierarchies include the special case of non-covering hierarchies. Moreover, for each of these simple hierarchies, another specialization can be applied, making them strict or non-strict. Multiple hierarchies are composed of one or more simple hierarchies. Additionally, multiple hierarchies can be specialized into inclusive and alternative.

3 Related Work

Table 1 compares conceptual multidimensional models that, at the best of our knowledge, cope with hierarchies. We use four symbols for their comparison: blank space indicates no reference to the hierarchy exists, ± a definition of the hierarchy is presented, □ the hierarchy can be represented in the model, and √ a definition and a graphical representation are given. In case that a different name for a hierarchy is proposed, it is included in the table. All the models include explicitly or implicitly the strict and parallel independent hierarchies. On the other hand, none of the models refer to multiple inclusive hierarchies. Thus, we omit them in Table 1. However, none of the models take into account different analysis criteria applied for hierarchies; in consequence, the multiple alternative and parallel dependent hierarchies cannot be distinguished.

Pedersen et al. [15] presents different types of hierarchies appearing in a healthcare application. They distinguish some of the hierarchies presented in this work as can be seen in Table 1. Other kinds of hierarchies (such as generalized and parallel) are not considered in his work.

Hüsemann et al. [8] adopt a conceptual model approach based on functional dependencies and a multidimensional normal form [10] to ensure correctness of schemas. Their model represents two kinds of hierarchies: symmetric and what they called multiple. The latter category is further divided into optional and alternative hierarchies corresponding to, respectively, generalized and multiple alternative hierarchies.

Golfarelli et al. [5] propose a conceptual model called DFM based on directed, acyclic, and weakly-connected graphs. This graphical model allows to represent symmetric and parallel independent hierarchies. The model is also able to represent other hierarchies (Table 1) although it is difficult to distinguish them.

Tryfona et al. [19] propose the StarER model, an extension of the ER model. They refer to different kinds of hierarchies based on Pedersen's work and showed that their graphical model is able to represent hierarchies as shown in Table 1. Since they do not refer to generalized hierarchies as discussed in this paper, their model does not make the difference between this and multiple hierarchies.

Sapia et al. [18] develop an extension of the ER model called ME/R model. With respect to hierarchy modeling, they allow to represent only symmetric and multiple alternative hierarchies or parallel dependent. Similarly to the previous model, ME/R can be extended to manage some of the hierarchies presented in this work (Table 1).

Table 1. Comparison of the models with different kinds of hierarchies.

Hierarchy /Model	Symmetric	Asymmetric	Generalized	Non-covering	Non-strict	Multiple alternative or Parallel dependent
[15]	Onto ±	Non-onto ±		Non-covering ±	Non-strict ±	Multiple ±
[8]	Simple √		Multiple Optional √	±		Multiple alternative √
[5]	√		□	□		√
[19]	√	□			Non-strict √	Multiple √
[18]	√					√
[16]	Classification ±	Aggregation[6] ±	Aggregation[6], Multiple[6] ±	Aggregation[6] ±		Multiple[6] Multiplicity ±
[6]	Strictly homogenous ±	Homogenous ±	Heterogeneous ±	Heterogeneous ±		Heterogeneous ±
[1]	√		□		√	√
[12]	√		Specialization □	√	√	Multiple, Alternative √
[20]	√			√	±	□

Purabbas et al. [16] define a classification hierarchy that corresponds to symmetric hierarchies. Their definition of aggregation hierarchies includes asymmetric, generalized, and non-covering hierarchies without making clear distinction between them. They do not allow the existence of non-strict hierarchies. Moreover, their definition of multiple hierarchies can be applied to generalized, multiple alternative, and parallel dependent hierarchies.

Hurtado et al. [6, 7] define homogenous and heterogeneous hierarchies. The former include symmetric and asymmetric hierarchies. The latter can be used for generalized, non-covering, and multiple alternative or parallel dependent hierarchies. They extended the notion of summarizability to heterogeneous hierarchies by including constraints that can be applied for the logical-level implementation of the hierarchies proposed in this paper. They do not focus on conceptual modeling and a graphical representation of them.

Abelló et al. [1] proposed a conceptual multidimensional model called YAM^2 based on UML. According to hierarchies, their model allows to include symmetric, non-strict, and multiple alternative hierarchies using the part-whole relationship.

Luján-Mora et al. [12] represent hierarchies using the UML model. They consider hierarchies as specified in Table 1. They distinguish a categorizing dimension based on the specialization/generalization relationship. Using this representation, they do not focus on hierarchies that include common and specific levels as proposed in this work for generalized hierarchy.

[6] No distinction can be made for different kinds of hierarchies.

Tsois et al. [20] propose the MAC model, which is used for modeling OLAP scenarios. Regarding to hierarchy specification, even if they refer informally to some of hierarchies proposed by this work; they do not propose a graphical representation for them.

Current commercial OLAP products do not allow conceptual modeling of different kinds of hierarchies. They usually represent multidimensional model limited to star or snowflake view of data without distinguishing different kinds of hierarchies. Further, ADAPT (Application Design for Analytical Processing Technologies) [4] even though introduces new features for multidimensional modeling such as dimension scope, dimension context, it does not include different kinds of hierarchies limiting the graphical representation to hierarchies commonly-used in OLAP tools. These implementation-level tools allow to manage symmetric and parallel hierarchies, some of them extends their product functionalities and include asymmetric, and non-covering hierarchies [13, 14].

4 Conclusions

Nowadays many organizations use data warehouses to support the decision-making process. Data warehouses are defined using a multidimensional model that includes measures, dimensions, and hierarchies. OLAP tools allow interactively querying and reporting a multidimensional database using operations such as drill-down and roll-up. OLAP tools exploit the defined hierarchies in order to automatically aggregate the measures to be analyzed. However, although many kinds of hierarchies can be found in real-world applications, current OLAP tools manage only a limited number of them.

Therefore, designers must apply different “tricks” at the implementation level to transform some hierarchy types into simple ones. Moreover, there is not a common classification and representation of the different kinds of hierarchies. Thus, when the user requires complex multidimensional analysis including several kinds of hierarchies, this situation is difficult to model and therefore it is not clear how to implement it.

In this paper, we took a conceptual approach and studied in a systematic way the different kinds of hierarchies referring them to a common multidimensional model. We also proposed a graphical notation for representing such hierarchies in conceptual models.

We distinguished simple and multiple hierarchies. The latter are composed of one or more simple hierarchies accounting for the same analysis criterion and include the inclusive and alternative hierarchies. This distinction is important since simple hierarchies generate tree structures for instances whereas multiple hierarchies generate acyclic graph structures. Moreover, simple hierarchies include further types: symmetrical, asymmetrical, generalized, and non-covering hierarchies. We also analyzed the case where the usual one-to-many cardinality linking a child level to a parent level in the hierarchy is relaxed to many-to-many leading to non-strict hierarchies. Finally, we discussed the situation where several hierarchies accounting for different analysis criteria may be attached to the same dimension. Depending on whether they share or not common levels, we called them parallel dependent and parallel independent hierarchies, respectively.

The proposed hierarchy classification will help designers to build conceptual models of multidimensional databases used in decision-support systems. This will give decision-making 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. Moreover, the proposed notation allows a clear distinction of each type of hierarchy taking into account their differences at the schema as well as at the instance levels. Most of the existing conceptual multidimensional models do not distinguish between the different kinds of hierarchies proposed in this paper, although some of these models can be extended to take into account the proposed hierarchy classification. Further, the proposed hierarchy classification provides OLAP tool implementers the requirements needed by business users for extending the functionality offered by current OLAP tools.

The present work belongs to a larger project aiming at developing a conceptual methodology for data warehouses. We are currently developed mappings for transforming the different kinds of hierarchies into relational model. This logical level also includes the analysis of summarizability as well as dimensional and multidimensional normal forms. We are also working on the inclusion of spatial and temporal features into the model.

References

1. Aballó A., Samos J., Saltor F. YAM^2 (Yet Another Multidimensional Model): An extension of UML. In *Proc. of the Int. Database Engineering and Application Symposium*, pp. 172-181, 2002.
2. Bauer A., Hümmer W., Lehner W. An Alternative Relational OLAP Modeling Approach. In *Proc. of the 2nd Int. Conf. on Data Warehousing and Knowledge Discovery*, pp. 189-198, 2000.
3. Booch G., Jacobson I., Rumbaugh J. *The Unified Modeling Language: User Guide*. Addison-Wesley, 1998.
4. Bulos D., Forsman S. Getting Started with ADAPT™ OLAP Database Design. *White Paper*. http://www.symcorp.com/download/ADAPT%20white%20paper.pdf, 1998.
5. Golfarelli M., Rizzi S. A Methodological Framework for Data Warehouse Design. In *Proc. of the 1st ACM Int. Workshop on Data Warehousing and OLAP*, pp. 3-9, 1998.
6. Hurtado C., Mendelzon A. Reasoning about Summarizabiliy in Heterogeneous Multidimensional Schemas. In *Proc. of the 8th Int. Conf. on Database Theory*, pp. 375-389, 2001.
7. Hurtado C., Mendelzon A. OLAP Dimension Contraints. In *Proc. of the 21st ACM Int. Conf. on Management of Data and Symposium on Principle of Databases Systems*, pp. 169-179, 2002.
8. Hüsemann B., Lechtenbörger J., Vossen G. Conceptual Data Warehouse Design. In *Proc. of the Int. Workshop on Design and Management of Data Warehouses*, p. 6, 2000.
9. Kimball R., Ross M., Merz R. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, 2002.
10. Lehner, W., Albrecht J., Wedekind H. Normal Forms for Multidimensional Databases. In *Proc. of the 10th Int. Conf. on Scientific and Statistical Database Management*, pp. 63-72, 1998.
11. Lenz H. and Shoshani A. Summarizability in OLAP and Statistical Databases. In *Proc. of the 9th Int. Conf. on Scientific and Statistical Database Management,* pp. 132-143, 1997.

12. Luján-Mora S., Trujillo J., Song I. Multidimensional Modeling with UML Package Diagrams. In *Proc. of the 21st Int. Conf. on Conceptual Modeling,* pp. 199-213, 2002.
13. Microsoft Corporation. *SQL Server 2000. Books Online.* (Updated – Service Pack 3). http://www.microsoft.com/sql/techinfo/productdoc/2000/books.asp, 2003.
14. Oracle Company. Oracle 9i OLAP User's Guide, http://otn.oracle.com/products/bi/pdf/ Userguide.pdf, 2002.
15. Pedersen T., Jensen Ch., Dyreson C. A foundation for Capturing and Querying Complex Multidimensional Data. *Information Systems*, 26(5): 383-423, 2001.
16. Pourabbas E., Rafanelli M. Characterization of Hierarchies and Some Operators in OLAP Environment. In *Proc. of the 2nd ACM Int. Workshop on Data Warehousing and OLAP,* pp. 54-59, 1999.
17. Rizzi S. Open Problems in Data Warehousing: 8 Years Later. In *Proc. of the 5th Int. Workshop on Design and Management of Data Warehouses*, http://sunsite.informatik.rwth-aachen.de/ Publications/CEUR-WS/Vol-77/keynote.pdf, 2003.
18. Sapia C., Blaschka M., Höfling G., Dinter B. Extending the E/R Model for Multidimensional Paradigms. In *Proc. of the 17th ER Int. Workshop*, pp. 105-116, 1998.
19. Tryfona N., Busborg F., Borch J. StarER: A Conceptual Model for Data Warehouse Design. In *Proc. of the 2nd ACM Int. Workshop on Data Warehousing and OLAP*, pp. 3-8, 1999.
20. Tsois A., Karayannidis N., Sellis T. MAC: Conceptual Data Modeling for OLAP. In *Proc. of the Int. Workshop on Design and Management of Data Warehouses*, p. 5, 2001.