Hierarchies in Multidimensional Databases

Elaheh Pourabbas
National Research Council, Rome, Italy

INTRODUCTION

Hierarchies play a fundamental role in knowledge representation and reasoning. They have been considered as the structures created by *abstraction processes*. According to Smith and Smith (1977), an abstraction process is an instinctively known human activity, and abstraction processes and their properties are generally used for multilevel object representation in information systems. An abstraction can be understood as a selection of a set of attributes, objects, or actions from a much larger set of attributes, objects, or actions according to certain criteria. Repeating this selection several times, that is, continuing to choose from each subset of objects, another subset of objects with even more abstract properties, we create other levels of (semantic) details of objects. The complete structure created by the abstraction process is a *hierarchy* and the type of hierarchy depends on the operation used for the abstraction process and the relations. As for the relations, the best known in the literature are *classification*, *generalization*, *association* (or *grouping*), and *aggregation*. Their main characteristics are briefly listed in the following.

Classification is a simple form of data abstraction in which an object type is defined as a set of instances. It introduces *an instance-of* relationship between an object type in a schema and its instances in the database (Brodie, Mylopoulos, & Schmidt, 1984).

Generalization is a form of abstraction in which similar objects are related to a higher level generic object. It forms a new concept by leaving out the properties of an existing concept. With such an abstraction, the similar constituent objects are specializations of the generic objects. At the level of the generic object, the similarities of the specializations are emphasized, while the differences are suppressed (Brodie, et al., 1984).

This introduces an *is-a* relationship between objects. This relation covers a wide range of categories that are used in other frameworks, such as inheritance, implication, and inclusion. It is the most frequent relation resulting from subdividing concepts, called *taxonomies* in lexical semantics. The inverse of the generalization relation, called *specialization*, forms a new concept by adding properties to an existing concept (Borgida, Mylopoulos, & Wong, 1984).

A particular type of generalization hierarchy, named *filter hierarchy*, is defined by the so-called filtering operation. This operation applies a *filter function* to a set of objects on one level and generates a subset of these objects on a higher level. The main difference from the generalization hierarchy is that the objects that do not pass the filter will be suppressed at the higher level (Timpf, 1999).

Association or grouping is a form of abstraction in which a relationship between member objects is considered as a higher level set of objects. With this relationship, the details of member objects are suppressed and properties of the set object are emphasized. This introduces the *member-of* relationship between a member object and a set of objects (Brodie, 1981).

Aggregation is a form of abstraction in which a relationship between objects is considered as a higher level aggregate object (Brodie et al., 1984). Each instance of an aggregate object can be decomposed into instances of the component objects. This introduces a *part-of* relationship between objects. The type of hierarchy constructed by this abstraction is called an *aggregation hierarchy*.

Like data warehousing and OLAP (online analytical processing), the above-mentioned aggregation hierarchies are widely used to support data aggregation (Lenz & Shoshani, 1997). In a simple form, such a hierarchy shows the relationships between domains of values. Each operation on a hierarchy can be viewed as a mapping from one domain to a smaller domain. In the OLAP environment, hierarchies are used to conceptualize the process of generalizing data as a transformation of values from one domain to values of another smaller or bigger domain by means of drill-down or roll-up operators. In the next sections, the roles of aggregation hierarchies in analysis dimensions of a data cube will be analyzed.

BACKGROUND

The core of the aggregation hierarchy revolves around the partial order, a simple and powerful mathematical concept to which a lot of attention has been devoted (see Davey & Priestley, 1993). The partial ordering can be represented as a tree with the vertices denoting the elements of the domains and the edges representing the

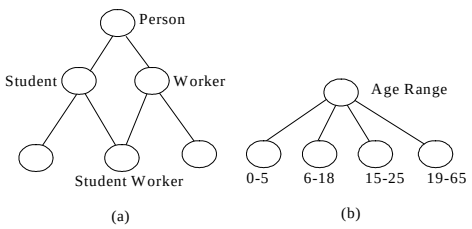
ordering function between elements. The notion of levels has been introduced through the idea that vertices at the same depth in the tree belong to the same level of the hierarchy. Thus, the number of levels in the hierarchy corresponds to the depth of the tree. The highest level is the most abstract of the hierarchy and the lowest level is the most detailed.

As in data warehousing and OLAP, the notion of partial ordering is widely used to organize the hierarchy of different levels of data aggregation along a dimension. Sometimes, hierarchies have been perceived structurally as trees, that is, no generic object is the immediate descendant of two or more generic objects, and where the immediate descendants of any node (supposing any hierarchy is represented by a graph) have classes which are mutually exclusive. A class with a mutually exclusive group of generic objects sharing a common parent is called a *cluster*. Generally speaking, many real cases cannot be modeled by these types of hierarchies (see Figure 1). For this reason, usually, a dimension hierarchy is represented as a directed acyclic graph (DAG). Sometimes, it can be defined with a unique bottom level and a unique top level, denoted by ALL (see Gray, Bosworth, Layman, & Pirahesh, 1996).

One of the most important issues related to the aggregation hierarchy is the *correct aggregation* of data (see Lenz & Shoshani, 1997; Rafanelli & Shoshani, 1990). It is known as *summarizability*, which intuitively means that individual aggregate results can be combined directly to produce new aggregate results.

As subsequently discussed in Lenz and Shoshani (1997), summarizability conditions are the conditions upon which the summarization operation produces the correct result. The authors affirm that three necessary conditions of summarizability have to be satisfied. They are disjointness of levels (or category attributes) in hierarchies, completeness in hierarchies, and correct use of measure (summary attributes) with statistical functions. Disjointness implies that instances of levels in dimensions form disjoint subsets of the elements of a level. Completeness in hierarchies means that all the elements occur in one of the dimensions and every element is assigned to some category on the level above it in the hierarchy. Correct use of measures with statistical functions depends on the type of the measure and the statistical function.

Figure 1. Two typical structures of hierarchies



More recently, the problem of heterogeneity in aggregation hierarchy structures and its effect on data aggregation has attracted the attention of the OLAP database community. The term heterogeneity, as introduced by Kimball (1996), refers to the situation where several dimensions representing the same conceptual entity, but with different categories and attributes, are modeled as a single dimension. According to this description, which has also been called *multiple hierarchy* and recalled in the next section (see Agrawal, Gupta, & Sarawagi, 1997; Pourabbas & Rafanelli, 2003), dimension modeling may require every pair of elements of a given category to have parents in the same set of categories. In other words, the roll-up function between adjacent levels is a total function. The hierarchies with this property are known to be regular or *homogeneous*. For instance, in a homogeneous hierarchy, we cannot have some cities that roll-up to provinces and some to states, that is, the roll-up function between City and State is a partial function.

In order to model these irregular cases, some authors introduced *heterogeneous* dimensions and tackled the summarizability issue by proposing several solutions.

The proposal of Lehner, Albrecht, & Wedekind (1998) consists of transforming heterogeneous dimensions into homogeneous dimensions in order to be in *dimensional normal form* (DNF). This transformation is actually performed by considering categories, which cause the heterogeneity, as attributes for tables outside the hierarchy. On the flattened child-parent relation, summarizability is achieved for dimension instances.

Pederson and Jensen (1999) considered a particular class of heterogeneous hierarchies, for which they proposed their transformation into homogeneous hierarchies by adding null members to represent missing parents. In their opinion, summarizability occurs when the mappings in the dimension hierarchies are *onto* (all paths from the root to a leaf in the hierarchy have equal lengths), *covering* (only immediate parent and child values can be related), and *strict* (each child in a hierarchy has only one parent). The proposed solutions consider a restricted class of heterogeneous dimensions, and null members may cause a waste of memory and increase the computational effort due to the sparsity of the cube views.

Hurtado and Mendelzon (2001) extended the notion of summarizability for homogeneous dimensions in order to tackle summarizability for heterogeneous dimensions. They classified five classes of dimension schemas, which are