Understanding Analysis Dimensions in a Multidimensional Object-Oriented Model

Alberto Abelló
Universitat Politècnica de Catalunya
Jordi Girona 1-3, E-08034 Barcelona
aabello@lsi.upc.es

José Samos
Universidad de Granada
Avd. Andalucía 38, E-18071 Granada
jsamos@ugr.es

Fèlix Saltor
Universitat Politècnica de Catalunya
Jordi Girona 1-3, E-08034 Barcelona
saltor@lsi.upc.es

Abstract

OLAP defines a set of data warehousing query tools characterized by providing a multidimensional view of data. Information can be shown at different aggregation levels (often called granularities) for each dimension. In this paper, we try to outline the benefits of understanding the relationships between those aggregation levels as Part-Whole relationships, and how it helps to address some semantic problems. Moreover, we propose the usage of other Object-Oriented constructs to keep as much semantics as possible in analysis dimensions.

Key Words: Multidimensional modeling, Analysis dimensions, Mereology, Object-Oriented modeling, On-Line Analytical Processing

1 Introduction

Nowadays, there is a wide interest in information systems that help companies in their decision making processes. At the core of such systems we find the *Data Warehouse*, where we keep all data that could be useful for that purpose. However, storing that data is not enough, we also need query tools. Probably, the most popular of these tools are *On-Line Analytical Processing* (OLAP) applications, firstly identified in [CCS93]. The main characteristic of OLAP tools, besides being fast and easy to use, is that they offer a multidimensional view of the subject of analysis.

Offering a multidimensional view means conceiving the subject of analysis as a *multidimensional space* (also known as *cube* or *hypercube*) containing the measures of the facts we want to analyze. The dimensions of that space are the different points of view we are going to use to analyze it. For instance, if we want to analyze company sales, we could do it attending to four dimensions, i.e Time (when something was sold), Store (where it was sold), Product (what was sold), and Customer (whom it was sold). Benefits of a multidimensional conception are twofold. On the first hand, it helps users to understand data. On the other hand, it helps computers to "understand", in advance, what users want to do, allowing to improve performance.

1.1 Related work

In the last years, lots of efforts have been devoted to multidimensional modeling. Those efforts have been clearly reflected in literature. [ASS01a] contains a survey of some representative multidimensional data models. Here, we emphasize some of them.

Previous to multidimensional models, and even to the definition of OLAP tools, [SR91] presents a statistical model (resembling those multidimensional). The first formal approach to present a multidimensional data model was that in [AGS97]; it proposes a minimal, closed set of operations on the *hypercube*. [Kim96], in a logical phase of design, represents the *hypercube* by means of a relational

Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001)
Interlaken, Switzerland, June 4, 2001
(D. Theodoratos, J. Hammer, M. Jeusfeld, M. Staudt, eds.)
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-39/

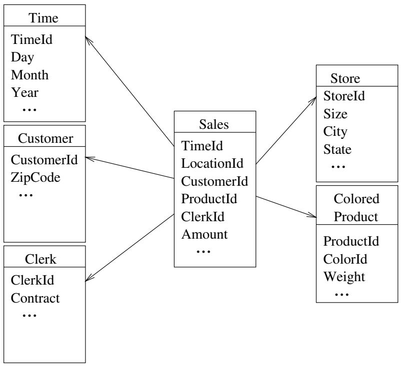

Figure 1: Example of *star schema*

star schema (like that depicted in figure 1), having one central *Fact Table* (containing measures) surrounded by multiple *Dimension Tables* (containing descriptive attributes).

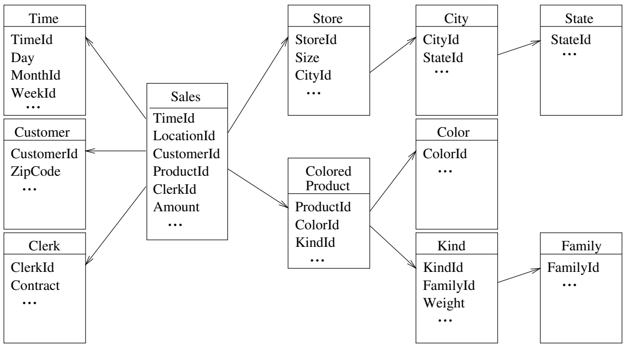

Figure 2: Example of *snowflake schema*

Some authors argue that it is also important to normalize schemas (also known as "snowflaking"). As a side effect, it shows aggregation hierarchies in the dimensions, as can be seen in figure 2. [HS97] presents a description logics model, which describes aggregation hierarchies as partially ordered sets with *Part-Whole* relationship being its strict order. In [TBC99], a multidimensional model, which allows the usage of specialization, aggregation, and membership relationships, is proposed. It is said that dimensions are usually governed by associations of type membership forming hierarchies that specify granularities. [TPG00] also used Object-Oriented (O-O) concepts to model dimensions. Specifically, associations (in UML - Unified Modeling Language - sense) define a directed acyclic graph between aggregation levels, and generalization represents categorization of aggregation levels, allowing to define additional features of the subtypes. There are also some papers specifically related to aggregation hierarchies in analysis dimensions, like [JLS99], and [PR99].

1.2 Aim of this paper

This work is devoted to investigate problems at representing analysis dimensions, and their aggregation hierarchies at conceptual level. The stress is on how to solve those problems by showing aggregation semantics and navigation paths along the dimensions. The importance of semantically rich relationships and their usage in conceptual modeling is outlined in [Sto93]. A first approach to how multidimensional modeling could benefit from O-O semantics was already shown in [ASS01b].

Most of those models mentioned in section 1.1 provide some way to represent aggregation hierarchies. Nevertheless, we argue that those papers treat semantics of conceptual modeling constructs rather superficially, often just pointing to a general idea.

We want to dig into the usage of certain modeling abstractions to solve some well identified semantic problems enumerated in section 2. They are addressed from an O-O point of view in section 3. Specifically, the usage of *Part-Whole*, *Simple-Aggregation*, and *Specialization/Generalization* relationships will be studied. Other O-O concepts that could also be used (for instance attaching methods to aggregation levels) have been left out of the scope of this paper. Section 4 discusses benefits of expliciting analysis dimensions. Conclusions are in section 5, followed by acknowledgements and bibliography.

2 Semantic problems in present multidimensional modeling

This section outlines some problems found in existing multidimensional models. Some of them were already identified in [SR91], [Leh98] and [PJ99]. Even though [SR91] can be found out of place, most of the problems it identifies in statistical modeling are also applicable in multidimensional context. The problems, related to modeling dimensions, are grouped into five sections.

2.1 Aggregation levels graph

At first glance, one could think that aggregation levels graphs are quite simple. Data about Stores is aggregated attending to the City they belong to, data about Cities is aggregated attending to the State they belong to, and so on. It looks linear and simple. However, we just need to look at the ColoredProduct dimension to find that products can be aggregated either by Color or Kind. We can see other examples of multiple aggregation paths in [Tho97].

Some OLAP tools just impose the constraint that an aggregation graph must be connected and show parent-child relationships between attributes. [LAW98] imposes the existence of a common top aggregation level (called *All*), defining a lattice of aggregation levels for every analysis dimension; and identifies relationships between levels as functional dependencies. [PJ99] also identifies multiple aggregation paths in the same dimension, and presents the different aggregation levels forming a lattice, being related by *greater than* relationships (meaning *logical containment* of the elements at one level into those at the other). It could also be the case that our information sources feeding the

Data Warehouse collect data at Month, and Week level, but not at Day level. Therefore, we could define a common aggregation top, but not a common bottom for both aggregation paths.

To the best of our knowledge, there is no justification in literature of the structure of aggregation levels into a dimension and the relationships among them being a lattice, semi-lattice, or just a directed graph. It is necessary to find a wide accepted definition of analysis dimensions. This is the first step to state its structure and properties.

2.2 Relationship cardinalities

Due to one reason or another, almost everybody argues that aggregation hierarchies are formed by "to-one" relationships. It means that an element at a given level is related to exactly one element of the next level in the hierarchy. A Store corresponds to exactly one City; it, in turn, to exactly one State; and so on. As pointed out in [LAW98], this provides nice aggregability properties.

However, we can find examples where hierarchies are not defined by "to-one" relationships in [SR91], [Kim96], and [Tho97]. [PJ99] also presents examples where the dimension hierarchies, besides possibly being "to-many", can be non-covering. In general, the most common (and computationally comfortable) cardinalities are 1..N-1..1 and 1..1-1..1 (meaning minimum..maximum cardinalities at lower-higher aggregation levels).

A difficulty slightly related to this is that of having different path lengths between instances at two aggregation levels in the dimension hierarchy. An instance a at level L_1 is part of b at level L_2, which in turn is part of c at level L_3. However, there is another instance e at level L_1 that is directly part of d at level L_3. This is identified by [PJ99] as non-onto hierarchies.

In general, we could find sixteen different cardinalities between two levels (i.e. two - 0 or 1 for minimum, and 1 or N for maximum - raised to the power of four), most of them presenting summarizability problems. Thus, it is needed to clearly identify meaningless cardinalities to avoid misunderstandings on designing, as well as the meaningful ones to strive to solve problems they generate.

2.3 Heterogeneous aggregation levels

[SR91] detects a problem referred as "non-homogeneous statistical objects". This means having objects at the same aggregation level that have different attributes.

In [Leh98], it is solved by defining the attributes at instance level. However, as pointed out by some authors (see [BSHD98]), explicit separation of cube structure and its contents is a desirable model feature. In this sense, attaching specific attributes to every instance, does not seem a good solution. [LAW98] also tackles the problem, and proposes to solve it by means of attributes with "null" values (showing that a given attribute is non applicable), and restricting the usage of these attributes to selection of instances (forbidding grouping by them). Solution in [BHL00] is much more elegant. It proposes to define different relations for every set of instances sharing the same attributes.

It is not enough to solve this at logical level (by means of relations). Modeling the concepts so that more semantics are captured is also important.

2.4 Reuse of dimensions

Multidimensional cubes are conceived in an isolated manner. However, when we use them, we want to navigate from one cube to another one (known as *drill-across*). This means we are analyzing data in a cube from a given point of view, and want to view data in another cube from the same point of view. Thus, cubes need to have equivalent points of view (dimensions). Moreover, we can also find the same dimension playing different roles in a cube. For instance, in a sale, people dimension plays two different roles (i.e. Clerk and Customer).

Most multidimensional models ignore *drill-across*. If it is considered, like in [Kim96], this operation is restricted to the case that both cubes have common dimension tables. As exemplified in [SBHD99], two cubes could also use the same dimension at different aggregation levels, still allowing *drill-across*.

Multidimensional analysis and research use to be restricted to one cube. Representing inter-dimension relationships would allow more powerful analysis by relating data in different cubes. The more semantically rich these relationships are, the better for the analyst.

2.5 Correlated dimensions

In general, analysis dimensions use to be independent. Thus, the point of view chosen at one of them does not restrict those possible values available at others. However, we can find some cases where there exist meaningless combinations of dimension values (they are correlated). For instance, it may be that all products are not on sale everywhere. Depending on the product characteristics, it is sold in a store or not. Some other examples of this situation can be found in [Kim96], referring the problem as "many-to-many relationships". If values in two dimensions are correlated, we could choose to keep both in the same dimension table.

To the best of our knowledge, there is no multidimensional conceptual model able to capture this kind of relationship. However, we think that it is needed to capture, at conceptual level, the possibility of combining different dimensions to give rise to a new one. Representing both dimensions together, at logical or physical level, would depend on the number of meaningful combinations with regard to the number of elements of the correlated dimensions.

3 How to solve them

We argue that relationships between aggregation levels should be interpreted as *Part-Whole* (also known as composition) relationships. This allows us to use "Classical Extensional Mereology" (CEM) axioms and other concepts in [GP95] to address problems stated in previous section.

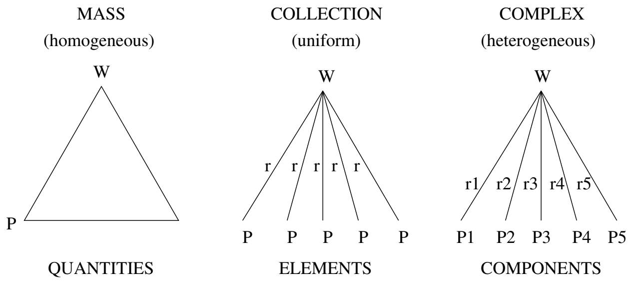

Figure 3: Types of wholes

As depicted in figure 3, we find three different, domain-independent, kinds of *Part-Whole* relations induced by the compositional structure of the whole (i.e. Mass, Collection, or Complex). If there is no compositional structure, the whole is considered homogeneous (ex. an amount of rice). If we take into consideration different elements, it is understood as a collection having a uniform compositional structure (ex. a convoy of trucks). If we see different parts playing different roles, we have a complex with an heterogeneous compositional structure (ex. the pieces in an engine). Mass, Collection, and Complex represent extreme cases on a scale leading from a total lack of compositional structure to wholes with complex internal organization. Different people could conceive a composed element at different points of that scale.

The main objective of defining relationships between different instances in an analysis dimension is to show how to apply aggregation functions (i.e. sum, min, max, avg, etc.). Since these functions consider instances as equals (playing the same role in the aggregation), we maintain that those relationships should be conceived as collections. From here on, we will refer to Part-Whole relationships between aggregation levels in an analysis dimension assuming they form collections.

In case of having collections, [GP95] considers that the axiomatic system of CEM (as stated in figure 4, that is also explained in [AFGP96]) seems to be ideally suited, except for axiom 6. In our case, axiom 6 also perfectly suits, since a user can always be interested in considering a given set of elements as a whole, in order to apply an aggregation function. Semantically, axiom 5 is not true, since the same collection of elements could compose different wholes (i.e. two clubs, at a given point in time, can have the same set of members). However, in order to apply aggregation functions both collections would give the same result. Thus, we would not be talking about clubs, but just sets of members which would be the same individual.

1. EXISTS. If A is part of B, both A and B exist
2. ANTISYMMETRY. If A is part of B, B is not part of A
3. TRANSITIVITY. If A is part of B and B is part of C, then A is part of C
4. SUPPLEMENTATION. If A is a proper-part-of B, then another individual C exists which is the missing part from B
5. EXTENSIONALITY. A and B have the same parts, if and only if A and B are the same individual
6. SUM. There always exists the individual composed by any two individuals of the theory

Figure 4: Classical Extensional Mereology axioms

[GP95] also explains that there might be more than one way to decompose the same whole, i.e some objects could be understood as collection of different kinds of elements (for instance, a year being a collection of either trimesters or four-month periods).

3.1 Relationships inside an analysis dimension

Some models, like [CT98], and [GMR98], already stated that dimensions contain different levels which represent domains at different granularities. Those granularities show how elements are grouped to apply aggregation functions. Thus, relationships are defined among elements at different levels standing for composition.

We understand by Simple-Aggregation those aggregations that do not give rise to a new instance. Those aggregation relationships that do not reflect composition, are Simple-Aggregations. In this kind of relationship, an instance is related to another just to show a property of the second one. Every instance in an analysis dimension will be related to some instances because of those being its parts, and to other instances because of those simply showing its properties.

We contend that it is essential to distinguish both kinds of aggregation in a multidimensional model, since they will allow to understand what was intended on defining a given schema. Part-Whole relationships will show how different elements are grouped together in a dimension, while Simple-Aggregation will indicate which are the different characteristics available to select instances. Thus, *roll-up* and *drill-down* operations will be performed along Part-Whole relationships, while selection (known as *slice-dice*) will be performed by means of Simple-Aggregation relationships.

In this section, we assume a minimum definition that everybody could agree in roder to deduce some controversial properties of an analysis dimension using CEM axiomatic system. Firstly, on referring to aggregation levels in multidimensional analysis, there is a misuse of language on saying, for instance, "A City decomposes into Stores". The real meaning is easily inferred, but it is important having in mind that it should be said "A set of Stores in a City decomposes into Stores".

We define an analysis dimension as follows:

Definition 1 *An* analysis dimension *is a connected, directed graph. Every vertex in the graph corresponds to an aggregation level containing instances, and an edge reflects that every instance at target level can be decomposed as a collection of elements at source level (i.e. edges reflect Part-Whole relationships between instances in aggregation levels in the dimension).*

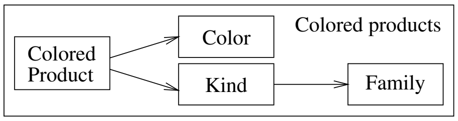

Figure 5: Example of analysis dimension

In O-O terminology, aggregation levels would be classes, and their instances would be objects. Figure 5 shows an example of analysis dimension. It contains a graph with four aggregation levels (i.e. Product, Color, Kind, and Family), and three edges showing that families of products can be decomposed into different kinds of products, and these into colored products which can be grouped by color.

From definition 1 and CEM axioms, some properties can be deduced with regard to analysis dimensions:

Property 1 *A dimension does not contain cycles.*

Proof 1 *Let us suppose that a cycle in the dimension graph exists. By successively considering axiom 3 on any instance A of a level forming the cycle, we would obtain that exists another instance B of another level forming the cycle so that A is part of B and B is part of A. This contradicts axiom 2, then a cycle can not exist in the graph of a dimension.*

Property 2 *For every dimension, there exists a unique aggregation level* Atomic *which contains elementary (i.e. that can not be broken down) instances. Notice that elementary instances could be unknown in a given database.*

Proof 2 *By property 1, there is at least a level whose instances do not have parts. If there is more than one of those* Atomic *levels, since a dimension is connected and axiom 3, there will exist an instance E conceived as composition of elementary instances at each one of the* Atomic *levels. By axiom 5, all those collections of elementary instances composing E must be the same collection of elements. Therefore, there exists only one* Atomic *level.*

Property 3 *For every dimension, there might exist a level* All *containing instances composed by all elementary instances in the dimension. If this level exist, a) Its instances are not collected by instances at any other aggregation level; b) This aggregation level has exactly one instance; and c) It is unique in the dimension.*

Proof 3 *By successively considering axiom 6 we can construct an instance E composed by all elementary instances in the dimension. a) If E would be a proper-part-of an E', by axiom 4 there would be an elementary instance that is not in E. Therefore, E is at a level whose instances are not part of any other instance in the dimension. b) If this level would contain two instances, both containing all elementary instances, by axiom 5 they would be the same instance. c) This level is unique, since if there were another level whose instances collect all elementary instances, they would be the same instance we already have in* All *level (by axiom 5).*

Property 4 *Those levels whose instances are not collected by instances of any level (i.e. they are not source of edges in the dimension graph) can be connected with an edge to level* All.

Proof 4 *The instance of level* All *can be decomposed into instances at any level covering* Atomic *level. If there is a level not covering* Atomic *level, a collection can be added to it, by axiom 6, collecting every elementary instance missing.*

Property 5 *Every instance in an aggregation level, that is not* Atomic, *has at least a part.*

Proof 5 *An instance without parts is elementary, and all elementary instances are at* Atomic *level, by property 2.*

Property 6 *Every instance in an aggregation level that is not* Atomic *might have more than one part.*

Proof 6 *If the part-of relationship between two instances is a proper-part-of, by axiom 4 the collection will have more than one part.*

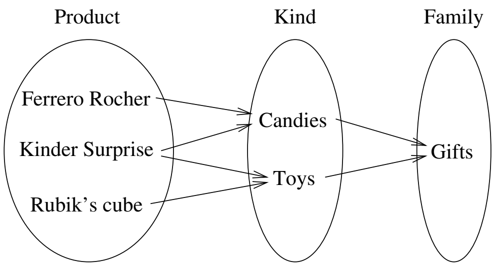

Figure 6: Example of overlapping classifications

Property 7 *An element might be part of several collections at the same time.*

Proof 7 *There is no mereological axiom forbidding the sharing of elements among several collections, in spite of* *it is a necessary condition to ensure summarizability (as shown in [LS97]). We argue that allowing this case is not a conceptual, but a computational problem (addressed as so in [PJ99]). If, as depicted in figure 6, a given product (at level Product) is allowed to belong to two different kinds of products at the same level Kind, some derived attributes of instances of level Family (which are composed by elements at level Kind) must be calculated from elements at level Product (ex: $card(Gifts) \neq card(Candies) + card(Toys)$).*

Property 8 *If level* All *exists in the dimension, the graph is a lattice, and collections in each level are disjoint; then for every level S, every instance in it is part of a collection at each and every other level T being target of edges leaving level S.*

Proof 8 *A lattice with* All *level at top, by axiom 3, implies that every elementary instance is collected in at least one instance of any other level. By imposing that collections in a level are disjoint, we obtain that every element in S must be collected exactly in a collection in T. If elements were not disjoint, there could be an instance of S overlapping several collections in T, so that it would not be completely contained into any of them.*

With regard to problems stated in section 2.1, from definition 1 and properties 1 and 2 we ensure that, in general, those aggregation levels in a dimension form a semi-lattice. Moreover, properties 3 and 4 show that *All* level can always be defined in order to obtain a lattice. Those problems about relationships cardinalities, in section 2.2, are explained by the other properties. Properties 5 and 6 imply that the relationships between two levels will involve 1..N parts for every whole. Property 7 explains that a part could participate in more than one whole or not. Property 8 shows that if we have a lattice with level *All*, and parts do not participate in more than one whole; there is a whole for every part (i.e. we have cardinality 1..N-1..1). If the same part can participate in more than one whole at the same level we can not guarantee that there is a whole for every part (even if *All* exists in the dimension, we have cardinality 1..N-0..N). In any case, axiom 6 shows that the needed instances could be obtained to have 1..N wholes for every part (so that we have 1..N-1..N).

3.2 Relationships between dimensions

It is not enough showing relationships inside a dimension or aggregation level. It is also important to analyze relationships between elements analysis dimensions in different cubes or even in the same one. In this section we are going to consider two kinds of relationship i.e. Specialization, and Aggregation.

Specialization

The usage of specialization relationships between aggregation levels is proposed in [TBC99], and [TPG00]. We completely agree that specialization is an essential relationship to be shown in multidimensional schemas. Nevertheless, we argue that isolated aggregation levels can not be specialized. They must be considered inside a dimension.

Property 9 *In general, a level and its specialization can not belong to the same analysis dimension.*

Proof 9 *Let us assume that both a level L and its specialization L_S are in the same dimension. In order to define a lattice with level* All*, since in this case L_S must cover* Atomic *level, we could be forced to have some instances in L_S. Those instances we are forced to have in L_S, could not fulfill specialization criterion. Therefore, it is not always possible to have both aggregation levels in the same dimension.*

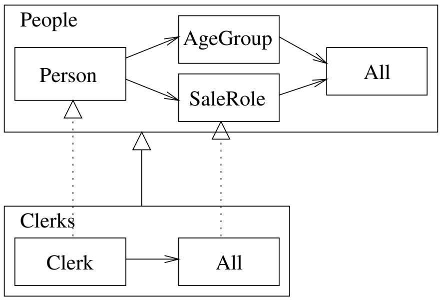

Figure 7: Example of dimension specialization

Figure 7 shows an example where People dimension is specialized at SaleRole level (solid arrow) to have a Clerks dimension. This specialization contains a level with all people acting as clerk, and another one with only one element representing the set of all clerks. Dashed arrows show that a level is specialization of another one. AgeGroups aggregation level is not of interest in Clerks dimension. Notice that if it would, it would not be specialization of the same level in People dimension since its instances would be different (they would collect less people).

Definition 2 *If D_S is the specialized dimension of D at level L, D_S contains at least the aggregation level L_S (specialization of L), and a specialization of every level in D containing parts of instances of L_S. These specialized levels contain exactly those instances of the corresponding level of D being part of any collection in L_S. Besides those mandatory levels in D_S, it is also possible that D_S contain other levels (that are not specialization of any level in D) with elements not in D.*

All instances of an aggregation level will have common properties, since it represents a given class of objects able to play the same role in a collection. By specializing an analysis dimension, we will be able to show attributes common only to a subset of instances, besides their specific Part-Whole relationships, which solves problems stated in section 2.3. Simple-Aggregations, as well as Part-Whole relationships are inherited along specializations. Therefore, it also addresses problems stated in section 2.4. It is not only possible to *drill-across* from a *cube* C_1 to a *cube* C_2 when both share dimensions, but also when the dimensions of C_1 are specialization of those in C_2.

Aggregation

Another interesting relationship to be shown is that of elementary instances in a dimension being aggregated in elementary instances in another dimension. This means expressing aggregation relationships between dimensions.

Property 10 *If elementary instances in a dimension D are part of elementary instances in dimension D_A, the graph of D will be a subgraph of D_A. Notice that instances in D will not be those in D_A, but part of them.*

Proof 10 *Elementary instances in D_A can be grouped so that the same elementary instance in D is part of every element in each collection. By axiom 6, these collections can become instances in D_A. Then, instances in D_A can be grouped by the same criteria used on grouping elements in D.*

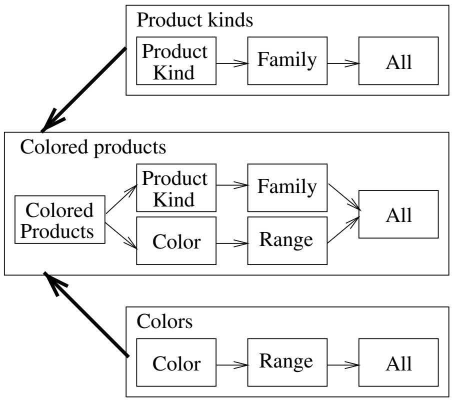

Figure 8: Example of dimension aggregation

Besides having Sales by ColoredProduct, we could obtain data in another star by Color, or ProductKind. Instances in these dimensions would be aggregated to show the kind of product sold, and the color of that product. As depicted in figure 8, the composed dimension would contain, at least, the graph of each one of the parts, joining *All* levels, plus a common *Atomic*.

By means of aggregation relationships between analysis dimensions, we address the problem found in section 2.5. Two dimensions aggregated to generate a new one mean that there is a relationship between them that should be considered, at design and query time.

4 Discussion

[Kim96], as well as other authors (like [Gio00]), argue that normalizing dimension tables is a serious mistake, but in a reduced set of specific cases. From their point of view, even though it saves some (negligible) storage space, it intimidates users by unnecessarily complicating the schema, and slow down most forms of browsing among dimensional attributes (joins are slower and less intuitive than selections). The point is that normalization explicits aggregation hierarchies, which show how measures can be summarized (known as *roll-up*) or decomposed (known as *drill-down*). Nevertheless, they argue that hierarchies are necessary neither to *roll-up*, nor to *drill-down*, since they are implicit in attribute values.

However, some people disagree with those ideas (see [PJ99] or [LAW98], for instance), and contend that aggregation hierarchies should be explicit; since they provide basis for defining aggregate data, and show navigation paths in analysis tasks.

From our point of view, the context makes the difference. If we are at a logical or physical design phase, as in [Kim96], it is possible to obtain better performance or understandability by denormalizing some tables. However, at a conceptual level, we must represent aggregation paths besides their different semantics. If this puts obstacles in the way of non-expert users understanding schemas, the user interface can hide as much information as necessary to make it understandable to a given user. Performance problems of the system will be addressed at further design phases (i.e. logical and physical).

As already stated in literature, it is important to separate conceptual and physical components. Logical or physical models are semantically poorer than conceptual ones. That is why conceptual models are so important. They give to the user much more information about the modeled reality, and are closer to his/her way of thinking. This is specially necessary in analysis tasks, because of the unpredictable nature of user queries in these environments. This kind of users can not be restricted to a small set of predefined queries. Indeed, they need to generate their own queries, most of times based on metadata. Thus, it is essential for a conceptual model to provide means to show aggregation hierarchies, and as much semantics as possible. For instance, showing that two analysis dimensions are specialization of another one, means that their instances (i.e. customers and clerks) can be compared.

Semantics are not only useful for users, but they can also improve query performance. In our example, Clerks and Customers are specialization of the same class, i.e. People. Comparing instances in those dimensions if the specialization is disjoint means they will always be different. Just knowing whether it is covering or not, would allow to obtain thresholds of aggregation results. The specialization being covering and disjoint also suggest parallel computing.

5 Conclusions

There is some controversy about whether aggregation hierarchies must be implicit or explicit. In this paper we argue that, at conceptual level, it is essential to explicit aggregation hierarchies, and as much information as possible about analysis dimensions. That information will ease the user to understand data, and pose ad-hoc queries. Users could classify and group data sets in an appropriate manner.

We identified some problems on explicitly modeling aggregation hierarchies. We contend that those problems can be addresses by providing Part-Whole semantics to relationships between aggregation levels, and considering mereology axioms. Thus, we defined an analysis dimension as a connected, directed graph of aggregation levels, and for each one of the problems, some mereological properties were inferred to solve it. To the best of our knowledge, this is the first work deducing properties of analysis dimensions instead of just imposing them.

Not only Part-Whole, but other kinds of relationship were found interesting for analysis dimensions (i.e. Specialization/Generalization, and Simple-Aggregation). It was also shown how different dimension can be related and the consequences that relationships have in aggregation hierarchies.

It is important to notice that, as can be read in [AFGP96], Part-Whole and Simple-Aggregation relationships are closely related. Namely, there are some properties that the whole inherits from its parts (ex. being defective), others that the parts inherit from the whole they are part of (ex. location), and some properties in the parts which are systematically related to properties of the whole (ex. weight of parts being less than weight of the whole). This has implications on the aggregability of measures, as well as the inheritance of properties between parts and wholes. However, this was left out of the scope of this paper, and will be tackled as future work.

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

[AFGP96] A. Artale, E. Franconi, N. Guarino, and C. Pazzi. Part-Whole relations in Object-centered systems: an overview. *Data and Knowledge Engineering (DKE)*, 20, 1996.

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

[ASS01a] A. Abelló, J. Samos, and F. Saltor. A Framework for the Classification and Description of Multidimensional Data Models. In *Proc. of the 12th Int. Conf. on Database and Expert Systems Applications (DEXA)*. Springer, 2001. To be published.

[ASS01b] A. Abelló, J. Samos, and F. Saltor. Benefits of an Object-Oriented Multidimensional Data Model. In *Objects and Databases - Int. Symposium -*, volume 1944 of *LNCS*, pages 141–152. Springer, 2001. ECOOP'00, Sophia Antipolis (France).

[BHL00] A. Bauer, W. Hümmer, and W. Lehner. An Alternative Relational OLAP Modeling Approach. In *Prc. of the 2nd Int. Conf. on Data Warehousing and Knowledge Discovery*, volume 1944 of *LNCS*, pages 189–198. Springer, 2000.

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

[CCS93] E. F. Codd, S. B. Codd, and C. T. Salley. Providing OLAP to user-analysts: An IT mandate. Technical report, E. F. Codd & Associates, 1993.

[CT98] L. Cabibbo and R. Torlone. A Logical Approach to Multidimensional Databases. In *Advances in Database Technology - EDBT'98*, volume 1377 of *LNCS*, pages 183–197. Springer, 1998.

[Gio00] W. A. Giovinazzo. *Object-Oriented Data Warehouse Design*. Prentice Hall, 2000.

[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.

[GP95] P. Gerstl and S. Pribbenow. Midwinters, end games, and body parts: A classification of part-whole relations. *Int. Journal of Human-Computer Studies*, 43(5,6), 1995.

[HS97] M.-S. Hacid and U. Sattler. An Object-Centered Multi-dimensional Data Model with Hierarchically Structured Dimensions. In *Proc. of the IEEE Knowledge and Data Engineering Workshop*. IEEE Computer Society, 1997.

[JLS99] H. V. Jagadish, L. V. S. Lakshmanan, and D. Srivastava. What can Hierarchies do for Data Warehouses? In *Proc. of 25th Int. Conf. on Very Large Data Bases (VLDB)*, pages 530–541. Morgan Kaufmann, 1999.

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

[LAW98] W. Lehner, J. Albrecht, and H. Wedekind. Normal Forms for Multidimensional Databases. In *Proc. of 8th Int. Conf. on Statistical and Scientific Database Management (SSDBM)*. IEEE Computer Society, 1998.

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

[LS97] H.-J. Lenz and A. Shoshani. Summarizability in OLAP and Statistical Data Bases. In *Proc. of the 9th Int. Conf. on Scientific and Statistical Database Management (SSDBM)*. IEEE Computer Society, 1997.

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

[PR99] E. Pourabbas and M. Rafanelli. Characterization of Hierarchies and Some Operators in OLAP Environment. In *Proc. of ACM 2nd Int. Workshop on Data Warehousing and OLAP (DOLAP)*, pages 54–59, 1999.

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

[SR91] A. Shoshani and M. Rafanelli. A Model for Representing Statistical Objects. In *Proc. of the 3rd Int. Conf. on Information Systems and Management of Data (COMAD)*. McGraw-Hill, 1991.

[Sto93] V. C. Storey. Understanding semantic relationships. *VLDB Journal: Very Large Data Bases*, 2(4), October 1993.

[TBC99] N. Tryfona, F. Busborg, and J. G. B. Christiansen. starER: A conceptual model for data warehouse design. In *Proc. of ACM 2nd Int. Workshop on Data Warehousing and OLAP (DOLAP)*, pages 3–8, 1999.

[Tho97] E. Thomsen. *OLAP Solutions*. John Wiley & Sons, 1997.

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