STORM: A STATISTICAL OBJECT REPRESENTATION MODEL

Maurizio RAFANELLI+, Arie SHOSHANI*

+ Istituto di Analisi dei Sistemi ed Informatica
viale Manzoni 30, 00185 Roma, Italy

* Lawrence Berkeley Laboratory
University of California
1 Cyclotron Road, Berkeley, CA 94720, USA.

Abstract. In this paper we explore the structure and semantic properties of the entities stored in statistical databases. We call such entities "statistical objects" (SOs) and propose a new "statistical object representation model", based on a graph representation. We identify a number of SO representational problems in current models and propose a methodology for their solution.

1.0 INTRODUCTION

For the last several years, a number of researchers have been interested in the various problems which arise when modelling aggregate-type data [1st SDBM], [2nd SDBM], [3rd SSDBM], [Rafanelli 89]. Since aggregate data is often derived by applying statistical aggregation (e.g. SUM, COUNT) and statistical analysis functions over *micro-data* [Wong 84] the aggregate data bases are also called "statistical databases" (SDBs) [Shoshani 82], [Shoshani 85].

This paper will consider only aggregate-type data, a choice which is justified by the widespread use of aggregate data only i.e. without the corresponding micro-data. The reason is that it is too difficult to use the micro-data directly (both in terms of storage space and computation time) and because of reasons of privacy (especially when the user is not the data owner).

In SDBs the entities stored are complex data structures (vectors, matrixes, relations, time series, etc.) which are generally called statistical tables. In this paper these complex structures will be called "statistical object" (SO), so as to stress the fact that there may be many possible configurations for that object (e.g. tables, relations, matrixes, graphs).

Each SO is characterized by having a summary attribute, described by a set of modalities (or category attributes); the former is often called *quantitative variable* and the latter *qualitative variable.* The phenomenon described always has its "universe of definition" (for example, "Fruit products in

California"); moreover, the summary data are always fixed in time (or *static* as they are often called). This means that, for example, the production of fruit in California in the years 1981, 1982,.., 1988 is quantified by a numeric value datum which does not change in time, but new value can be added over time.

Various previous papers have dealt with the problem of how to logically represent an aggregate data reality (e.g. [Chan & Shoshani 81, Rafanelli & Ricci 83, Ozsoyoglu et al 85, Su 83]). Starting from those works, this paper will propose a new "statistical object representation model" (STORM), based on a graph representation. In the subsequent sections, after the necessary definitions, the proposed structure for a SO will be discussed and developed.

We follow the definition of the STORM model with an investigation of a well-formed SO, and develop conditions for it. Next, we develop the concept of and conditions for "summarizability" of a SO, which guarantee correct results of summary operations over statistical objects.

2.0 PROBLEMS WITH CURRENT LOGICAL MODELS

2.1 BASIC CONCEPTS

We start this section by briefly presenting four basic concepts that are unique to SDBs, and then discuss deficiencies of currently proposed models.

1. *Summary attributes* -- these are attributes that describe the quantitative data being measured or summarized. For example, "population", or "income for socio-economic databases", or "production and consumption of energy data".

2. *Category attributes* -- these are attributes that characterize the summary attributes. For example, "Race" and "Sex" characterize "Population counts", or "Energy type" and "Year" characterize the "production levels of energy sources".

3. *Multi-dimensionality* -- typically a multidimensional space defined by the category attributes is associated with a single summary attribute. For example, the three- dimensional space defined by "State", "Race" and "Year" can be associated with "Population". The implication is that a combination of values from "State", "Race" and "Year" (e.g. Alabama, Black, 1989) is necessary to characterize a single population value (e.g.10,000).

4. *Classification hierarchies* -- a classification relationship often exists between categories. For example "Cities" can be classified into "States", or specific "Products" (e.g. "Fruits", "Vegetables", "Grains" can be classified as "Agricultural Products").

These basic concepts are addressed in different models currently used to describe statistical data by employing essentially two methodologies: a) 2-dimensional tabular representation and b) graph-oriented representation. We explore below some of the problems encountered using these methodologies in current models.

In this paper, we define a *STatistical Object Representation Model (STORM)* which is independent from the above methodologies. As a consequence, a SO can then have a graphical representation, a 2-dimensional tabular representation, or any other representation preferred by the user (e.g. a "relation").

2.2 PROBLEMS WITH THE TWO-DIMENSIONAL TABULAR REPRESENTATION

The two-dimensional (2D) representation exists historically because statistical data have been presented on paper. This representation, although it continues to be practiced by statisticians today, changes the semantic concepts discussed above. In particular, we point out below several deficiencies.

2.1.1 *The concept of multi-dimensionality is distorted.*

By necessity, we need to squeeze the multi-dimensional space into two dimensions. This is typically done by choosing several of the dimensions to be represented as rows and several as columns. For example, suppose that we need to represent the "Average Income" by "Profession", "Sex" and "Year". Figure 1 is an example of a 2D tabular representation, where two of the dimensions have been represented as rows. Obviously, one can choose (according to some other preferred criteria) other combinations by exchanging the dimensions (e.g., "Year" first, then "Sex"), put two dimensions as columns, or even put all three dimensions as rows or columns.

Models using this tabular representation technique improperly consider the different tables to be different statistical objects, while in reality only the 2D representation has changed. In general, the 2D representation of a multi-dimensional statistical object forces a (possibly arbitrary) choice of two hierarchies for the rows and columns. The apparent conclusion is that a proper model should retain the concept of multi-dimensionality and represent it explicitly.

2.2.2 *The classification relationship is lost.*

In the 2D representation, classification hierarchies are represented in the same manner as the multi-dimensional categories. Consider, for example, that "Professions" in Figure 1 are classified into "Professional Categories" as shown in Figure 2.

			Profession				
			Chemical Engineer	....	Executive Secretary	.....	Elementary teacher
Sex	Male	**Year** 80	*1,841*		*2,600*		*1,038*
		81	*2,012*		*2,678*		*1,090*
		82	*2,199*		*2,758*		*1,166*
		..	....		....		....
		88	*3,749*		*3,293*		*1,701*
	Female	**Year** 80	*1,669*		*2,522*		*1,027*
		81	*1,825*		*2,597*		*1,079*
		82	*1,994*		*2,675*		*1,154*
		..	.....		......		......
		88	*3,399*		*3,194*		*1,683*

Figure. 1

Average Income in California			**Professional Category**					
			Engineer		Secretary		Teacher	
			Profession		**Profession**		**Profession**	
			Chemical Engineer	Civil Engineer	Junior Secretary	Executive Secretary	Elementary Teacher	College Teacher
Male	**Year**	80	*1,841*	*2,285*	*1,733*	*2,600*	*1,038*	*1,541*
		81	*2,012*	*2,411*	*1,819*	*2,678*	*1,090*	*1,641*
		82	*2,199*	*2,637*	*1,910*	*2,758*	*1,166*	*1,747*
		. .	....	....	....	......	......	
		88	*3,749*	*4,521*	*2,560*	*3,293*	*1,701*	*2,500*
Sex								
Female	**Year**	80	*1,669*	*1,825*	*1,698*	*2,522*	*1,027*	*1,525*
		81	*1,825*	*1,996*	*1,783*	*2,597*	*1,079*	*1,624*
		82	*1,994*	*2,184*	*1,872*	*2,675*	*1,154*	*1,729*
		. .	.....	......	......	......	......	
		88	*3,399*	*3,744*	*2,508*	*3,194*	*1,683*	*2,524*

Figure 2

As can be seen, there is no difference in the representation of "Sex" and "Year" and the representation of "Profession" and "Professional Category". However, it is obvious from this example that the values of average income are given for specific combinations of "Sex", "Year" and "Profession" only. Thus, “Professional Category” *is not* part of the multi-dimensional space of this statistical object. As can be seen from the above example, there is a fundamental difference between category relationship and multi-dimensionality. Usually, only the low-level elements of the classification relationship participate in the multi-dimensional space. This fundamental difference should be explicitly represented in a semantically correct statistical data model.

2.2.3 *Lack of meta-data level.*

A 2D representation requires that the category names as well as the category instances be represented together. There is no separate description of what the statistical database is about (meta-data). For example, the meta-data for Figure 2 consists only of "Average income" by "Sex", "Year" and "Profession", and professions are classified in "Professional Category", however, it is represented together with the data values. Consequently, the 2D representation becomes very large for tables with high dimensionality, or when the categories have a large number of instances. Such representation cannot comfortably fit on a page or a screen. In such cases, the representation spreads into multiple pages or screens. For example, if we add another dimension, "State" to Figure 2, we may need to represent each state on a separate page. This confuses the global understanding of the statistical object. It is therefore desirable to separate the representation of the categories and the category instances in order to achieve compactness of the semantic description of the database.

2.3 PROBLEMS WITH CURRENT GRAPH-ORIENTED MODELS

An attempt to correct some of the deficiencies of the 2D representation discussed above was made by introducing graph-oriented models. In these models the concepts of multi-dimensionality and classification hierarchies were introduced by having especially designated nodes. For example, in GRASS [Rafanelli 83] (which is based on SUBJECT [Chan 81]) multi-dimensionality is represented by A-nodes (A stands for "association") and C-nodes (C stands for "classification"). Thus, the statistical object of Figure 2 would be represented in GRASS as shown in Figure 3. Note that the node of the type S represents a "summary" attribute.

While this approach has an explicit representation for multi-dimensionality and classification, it left the previously mentioned problem of the lack of meta-data level (section 2.2.3) unresolved. The lack of meta-data level is more subtle in the graph-oriented model and is explained in more detail below.

2.3.1 *Mixing categories and category instances.*

We refer again to Figure 3 and in particular to the classification hierarchy of "Professional Category" and "Profession". Consider the intermediate node "Engineer". It has a dual function. On the one hand, it is an instance of the "Professional Category". On the other hand, it serves as the name of a category that contains "Chemical Engineer", "Civil Engineer", etc. Note that the category "Profession" is missing in this representation. The reason is that after we expand the first level ("Professional Category") into its instances, all the next levels can contain only instances.

Another consequence of this representation is similar to the problem of large 2D tables mentioned in section 2.2.3 . Here too, a large number of instances of categories produces large graphs that do not fit easily onto a page or a screen.

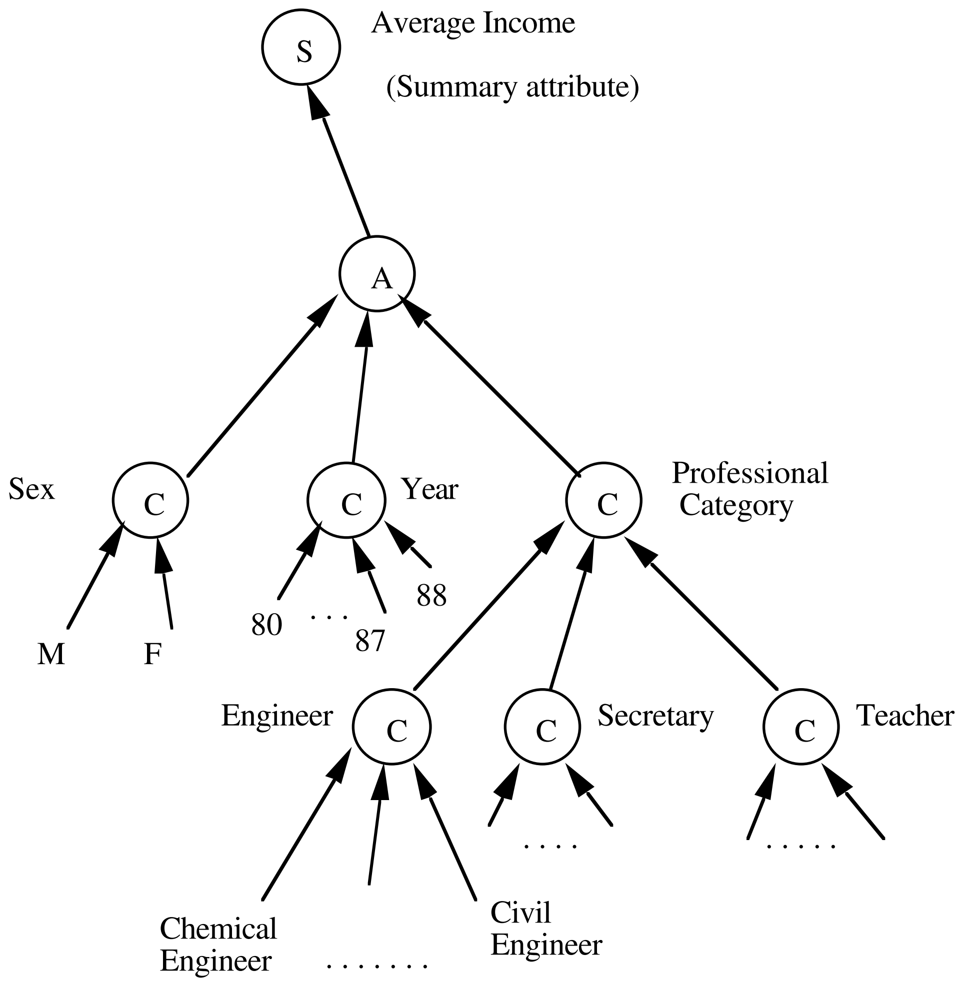


Figure 3

For the above reasons, we have chosen a graph model that separates the categories and their instances into two separate levels. For example, the statistical object of Figure 3 will be represented at the meta-data level (intentional representation) as shown in Figure 4. Underlying this representation the system stores and maintains the instances and their relationship. The instances can become visible to a user by using an appropriate command.

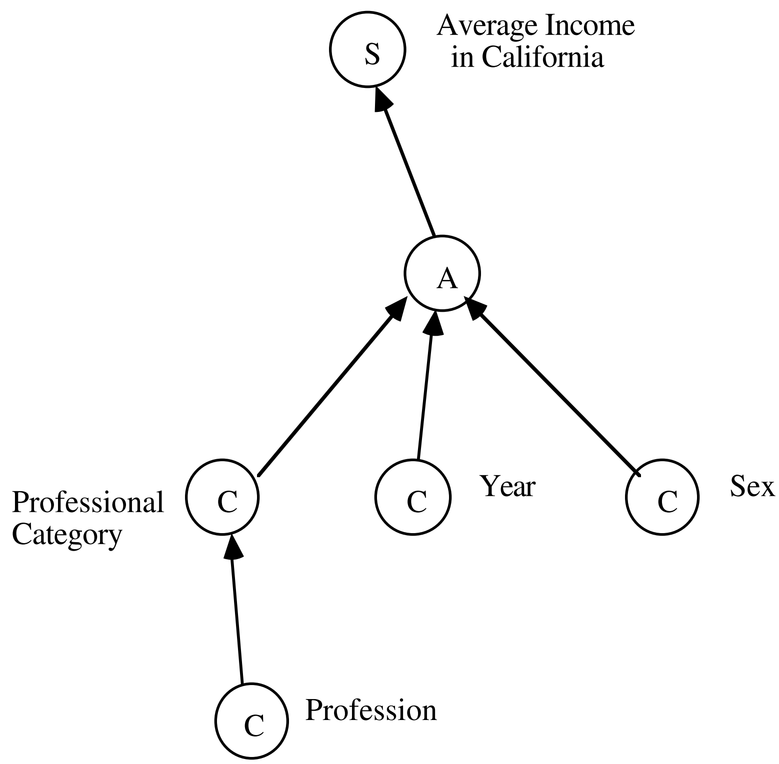


Figure 4

3.0 THE STORM MODEL

Before discussing the representational model, we define the basic components of a Statistical Object.

3.1 STATISTICAL OBJECT

Definition: A Statistical Object (SO) is a logical data structure defined by a quadruple
$< N, C, S, f >$, where:

N is the name of the SO, which describes the universe of the phenomenon of interest (for example, "Gasoline consumption in the USA" is a name that conveys sufficiently the universe of that SO.)

C is a finite set of category attributes; each category attribute has a domain associated with it, and a "domain cardinality" which corresponds to the number of instances of the domain for that category attribute. Each category attribute also has a property called "unit of measure", which represents the unit of the domain.

S is a single summary attribute associated with the SO. The summary attribute also has a domain and a domain cardinality associated with it. In addition to the "unit of measure" property, it has the property "summary type". The semantics of this property will be explained below.

f is a function which maps from the Cartesian product of the category attributes values to the summary attribute values of the SO.

In general, the function *f* is "into", in the usual sense that not every element of the domain (i.e. an element of the cartesian product) maps to an element of the range (i.e. has a summary value associated with it). Alternatively, we can assign by default a "null" element to the range, and consider the function to be "onto", where non-existing mappings map to the null element. However, the issue of the meaning of nulls in Statistical Databases is more complex, and is discussed further in section 6.0 in the context of missing values.

We can use the following notation to describe a SO:

$$N\ (C_{(1)}, C_{(2)}, ..., C_{(n)} : S),$$

where N and S are the name and summary attribute of the SO, and ($C_{(1)}$, $C_{(2)}$, ..., $C_{(n)}$ are the components of the category attribute set C. The function *f* is implied by the ":" notation. For example, the following describes a SO on various product sales in the USA:

PRODUCT SALES (TYPE, PRODUCT, YEAR, CITY, STATE, REGION : AMOUNT)

As mentioned in the introduction, a statistical object SO represents a summary over micro-data. That summary involves some statistical function (count, average, etc.), and some unit of measure of the phenomena of interest (gallon, tons, etc.). Accordingly, the summary attribute has the two properties mentioned above: "summary type", and "unit of measure". In the example above, the summary type is SUM (or TOTAL), and the unit of measure DOLLARS. Note that the above SO is presumed to be generated over some micro-data, such as the individual stores where the products were sold.

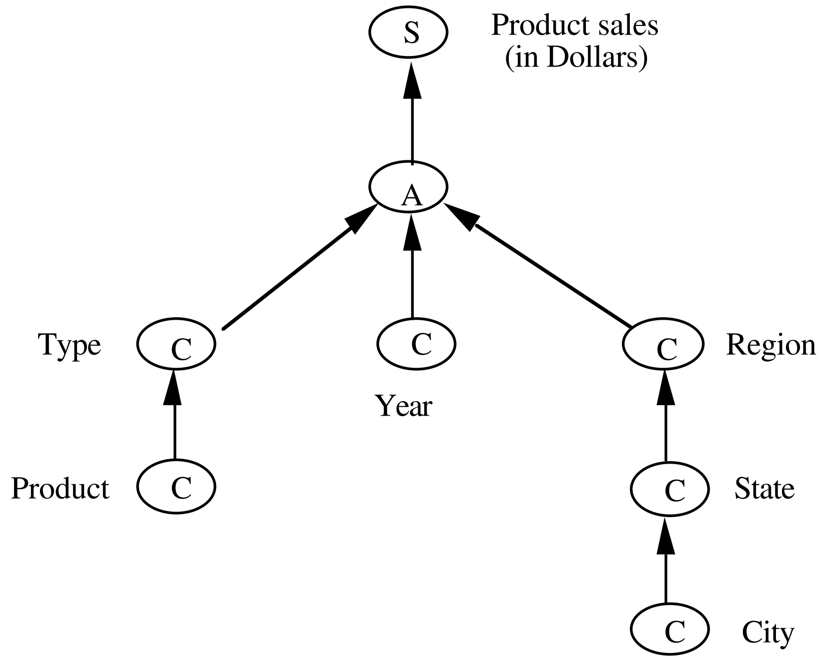
We note that the name of a SO is not necessarily a precise description of the SO universe. In the example given above on "Product Sales", the sales levels are given "by year and by city". Depending on the complexity of the SO, the name may reflect part or all of the category attributes involved. However, it should always reflects the summary attribute intended meaning.

So far, we have described the SO in a form that resembles a relation description in a relational model, with the following structural semantics added: there is a single attribute designated as the summary attribute which has a "summary type" and a "unit of measure" associated with it, and there is a function which maps elements of the cartesian product of the rest of the attributes (called category attributes) to the summary attribute. In the next section, we show that these structural semantics are not sufficient for describing a SO, since we need to know the relationship between the category attributes as well. In the example above on “product sales", suppose that product “type” can assume the values: metal, plastic, and wood, and that “product” can assume the values: chair, table, bed. How do we know if sales figures are given for products, product types, or both? Further, suppose that we know that figures are given for products, how do we decide whether these figures can be summarized into product type? Similarly, we need to know whether sales figures for cities can be summarized to state levels and to regions. In order to answer these type of questions, we need to capture the structural semantics between category attributes. For that purpose, we use the STatistical Object Representation Model (STORM).

3.2 THE STORM REPRESENTATION OF A SO

It is best to visualize the STORM representation of a SO in a graphical form as a directed tree. The summary attribute and each of the category attributes are represented as nodes of type S and C, respectively. The root of the tree is always the node S. In addition, another node type is used, denoted an A node, to represent an aggregation of the nodes pointing to it. In most cases the nodes pointing to an A node will be C nodes, but it is possible that an A node will point to another A node. An example of a STORM representation of the SO "product sales" mentioned previously is given in Figure 5a. Another possible representation of the same example is shown in Figure 5b, which illustrates the possibility of an A node pointing to another A node. Note that an aggregation node has the domain generated by the cross product of its component domains. Thus, the node A pointed to nodes "type" and "product" in Figure 5b, represents combinations of type and product.

The two representations of the SO “product sales” given in Figures 5a and 5b have radically different meanings. In Figure 5a the implication is that the sales amounts are given for each product (e.g. chair, table, ...), and that products are grouped into types (e.g. metal, wood, ...). Note that in this example, a product may belong to more than one type. On the other hand, in Figure 5b, the implication is that the sales amounts are given for each type-product combination. Thus, the sales figures are given for "metal-chairs", "plastic-chairs", etc. (These figures could obviously be zero or "non-existing). We would like to emphasize that there is no way of determining which representation is the desired one from the original description of the SO, and therefore, the choice of representation constitutes addtional semantic structure of the SO that should be provided by the database designer.



Figures 5a

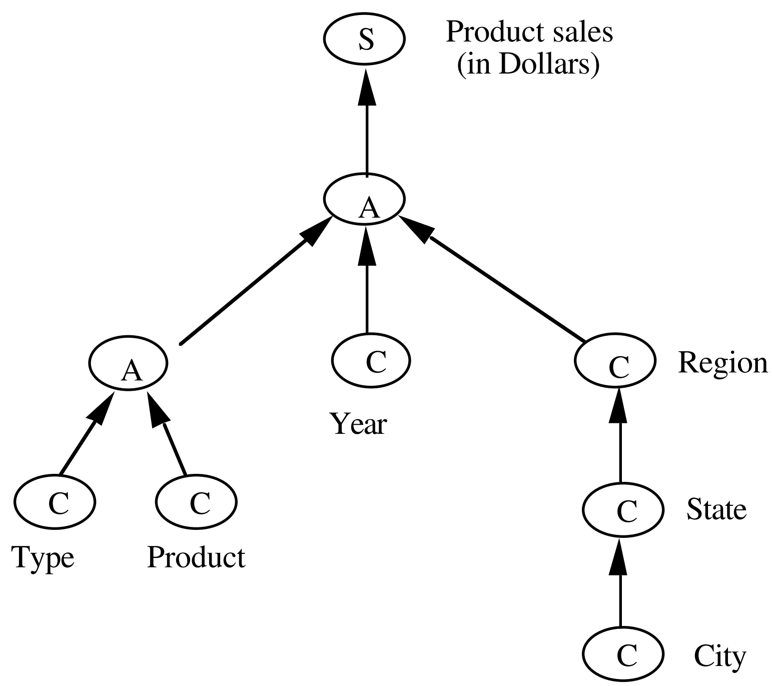


Figure 5b

Other structural limitations of the STORM tree are that the S node has always a single A node pointing to it, and that a C node can have only a single C or A node pointing to it. The reason for the former limitation is that the values of S are defined for the cartesian product of some (relevant) subset of the category attributes. The reason for the latter is that if more that one C node point to another C node, then the semantic intention is that the aggregation of the pointing nodes relate to the other C node; thus, an A node should exist between them. An example is shown in Figure 6, where the aggregation of car model and year maps into the displacement of the corresponding car engine.

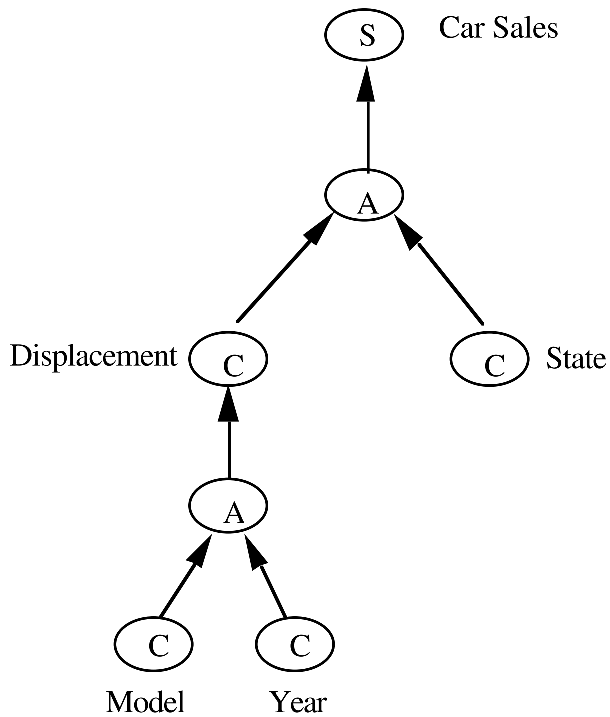


Figure 6

We note that an A node does not have a name. However, one can think of the name as the concatenation of the names of the nodes pointing to it. In the simple case that only leaf nodes point to the A node, the concatenated name can be used. For example, the lower A node in Figure 5b can be named "type/product". In the case that a more complex tree structure is attached to the A node, the name can be generated by concatenating the names of the leaf nodes. For example, the A node in Figure 5a can be named "product/year/city".

Another observation worth making is that in the case of an A node pointing to another A node, it is possible to collapse the structure to a single A node as shown in Figure 7. This can be easily verified by considering the components that make up the cross product elements of an aggregation node. This observation generalizes to multiple number of A nodes by applying the transformation of Figure 7 repeatedly. In spite of this observation, we allow the representation of an A node pointing to another A node, because it helps in presenting the semantics of the SO.

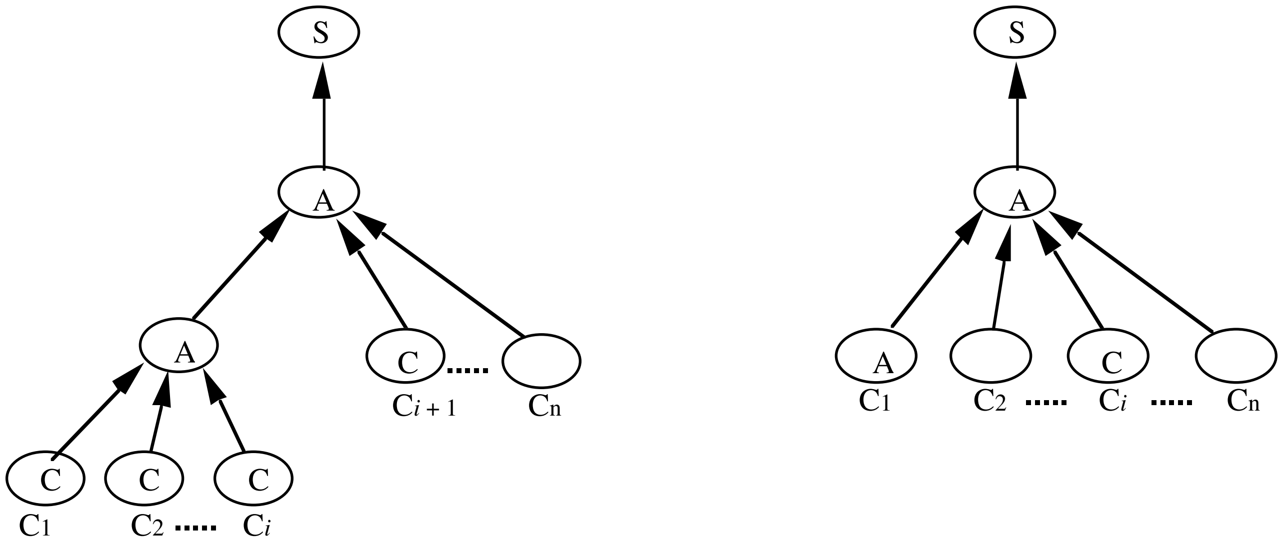


Figure 7

In summary, a STORM representation of a SO is a directed tree of nodes of type S, A, and C, with the following structural constraints:

a) There is only a single S node and it forms the root of the tree.
b) A single A node points to the S node.
c) Multiple C or A nodes can point to an A node.
d) Only a single C or A node can point to another C node.

4.0 MAPPING TYPES

The STORM representation of a SO implies a mapping between the nodes of the directed tree. We explore here the properties of the various possible mapping. We refer again to the example given in Figure 5a.

We already discussed the semantics of the A node as the aggregation of nodes pointing to it. However, as can be observed in Figure 5a, it is not immediately clear what are the components of the A node. For example, in the branch with the nodes "region", "state", and "city", what is actually pointing to A? Is it "region", "state", or "city", or some combination of the these? The answer to this will depend on the type of the mappings between the C nodes.

Let us first examine the mapping between "city" and "state". We assume that city names are unique within states, that is, each state can map into a single state. (We will show later what are the consequences of relaxing this assumption). This mapping is therefore "single-valued", or in other words a function. Similarly, if we assume that states are unique within regions, then the mapping between the corresponding nodes will also be single-valued. In this case, the node that should be considered as relevant to the aggregation node A is only "city", because the product sales amounts are given for cities. However, the nodes "state" and "region" exist in that structure to indicate that the two single-valued mappings (city --> state, and state --> region) are also specified as part of the SO description, and therefore the sales amounts for states and regions can potentially be calculated. We call the ability for such summary type calculation "summarizability". As we will see in the next section, single valued mappings are one of the conditions for summarizability.

Now, let us consider the branch in Figure 5a that includes "type" and "product". As mentioned above, a product (such as "chair") can be of several types (such as "metal" or "wood"). Such a mapping is called multi-valued (it is obviously not a function). Here again, as was the case with single-valued mappings, the node relevant to the aggregation node A is the leaf node "product", because sales amounts are given by product and regardless of their type. However, as will be shown in the next section it is not possible, in general, to summarize sales amounts to the "type" level.

Finally, we consider a special case of a multi-valued mapping that can and should be treated as a single-valued mapping. Consider, again, the case of the mapping between "city" and "state", and relax the condition that each city has a unique name within a state. This example is quite common in many countries. For example, Manhattan exists both in New York state, and in Kansas. Since city names can be the same in multiple states, the mapping can be considered multi-valued. However, this case is misleading, because there really exist different instances of cities, and thus there should be sales figures associated with each of these cities even if their names are the same. In such a case, we will consider the mapping to be of type ID (identification - the term is borrowed from Entity-Relationship modeling terminology). Accordingly, each city needs its associated state for unique identification, and the mapping can be considered single-valued.

In the case of an ID mapping both nodes involved participate in the aggregation node A. Thus, in the example above both "city" and "state" are essential to the aggregation node. In addition, ID conditions can propagate to the next levels. Consider, for example, that there is an ID mapping between "state" and "region" as well. Then, all three nodes "city", "state", and "region" are essential to the aggregation node. However, what if the mapping between "state" and "region" is ID, but the mapping between "city" and "state" is not? In that case only "city" is essential to the aggregation node, since cities have unique names.

5.0 SUMMARIZABILITY AND WELL-FORMEDNESS OF A SO

We mentioned above the concept of summarizability in the context of a single SO. However, the structural semantics of a SO are not only necessary for the interpretation of a single SO, but also for combining information from multiple SOs. As an example, consider the two SOs that represent population by cities by year by sex, and area by state. Suppose that we wish to derive population densities by state by year. In order to achieve the desired result, we need to summarize the population from cities to states, as well as over sex, and then divide the corresponding results into areas of states. The question arises under what conditions can we be sure that the summarization is done correctly. First, we define the term summarizability of a mapping.

Definition: Given the summary values for a C-node (or A-node) X and a mapping from node X to a C-node Y, the mapping is summarizable if using this mapping yields the correct summary values for Y.

Theorem: A multi-valued mapping is not summarizable.

Proof: Given a multi-valued mapping from X to Y, then a summary value for an X instance may be shared by more than one instance of Y. In general, there is no way of determining how the shares are divided. Since the summary value for an instance of Y has to be determined from multiple component shares, it is impossible to correctly calculate the summary values for Y.

To illustrate the above theorem, let us consider the previous example of the multi-valued mapping between products and types. Suppose that sales amounts for products are as follows: chairs - $1000, tables - $500. Suppose that we know from the mapping that some of the chairs and tables were made of metal and some of wood. It is obvious that without additional information there is no way of calculating the actual sales amount for metal and wood (although bounds can be found).

Definition: A SO is summarizable if all of its mappings are summarizable.

Obviously, a summarizable SO cannot contain multi-valued mapping. Usually, if a multi-valued mapping has been defined, then the designer of that SO should consider making the two nodes involved components of the same A node, as was shown in Figure 5b for the example above.

Summarizability also occurs for C-nodes under the same A-node. Consider for example, Figure 5b again. In that example, the nodes type and product are under the same A-node. We wish to summarize correctly sales by type as well as sales by product. This can be done because we can summarize over all products for a each type, and similarly summarize over all types for a each product. However, if there was a single-valued dependency between the C-nodes, summarizability will not be possible. To see this point, consider an A-node that has "city" and "state" pointing to it. If the mapping is of type ID, then summarizing over all states for the same city name will yield the wrong result. If the mapping is single-valued, summarizing over states to get city values is unnecessary since there is only one state for each city.

Putting C-nodes that have single-valued (or ID) dependency between them under the same A node can easily occur in the trivial case that all the C-nodes of a SO are put under a single A-node. Such will be the case for our example if we put "type", "product", "year", "city", "state", and "region" under the same A-node. In general, the situation can be more subtle. As an example, we refer again to Figure 6. In this case there is a single-valued mapping between the combination (aggregation) of nodes model and year, and the node displacement. If all three nodes were put under a single A-node, it may be more difficult to detect the single-valued dependency. Accordingly, we define the following.

Definition: A well-formed SO contains no multi-valued mappings along the branches of its tree, and no single-valued mappings between nodes that point to the same A-node.

Corollary: A well-formed SO is a necessary condition for summarizability.

6.0 ADDITIONAL CONDITIONS FOR SUMMARIZABILITY

We identified above two necessary conditions for summarizability, i.e. the two conditions for well-formedness. There are two additional conditions for a SO to be summarizable. Together, all four conditions are sufficient to ensure that a SO is summarizable.

To visualize the first additional condition, consider the mapping between cities and states. In order to summarize correctly over cities we need to know that there are no missing values. However, it is reasonable to assume that some small towns or other villages were not included in the list of cities, and therefore sales figures for them are not included. If we summarize to the state level, we will get incorrect results.

To compensate for such situations, database designers often include another value for cities, which we will label "other". If a sales figure for "other" was available, then we could claim that the summary can be done correctly. We will call a mapping that satisfies this condition a "full" mapping. Obviously, this is a semantic condition that depends on the specific mapping. Some mappings may be naturally full. For example, the mapping between states and regions (e.g. west, mid-west, ...) can be expected to be full because *all* the states will be partitioned into disjoint sets that belong to regions.

The second additional condition has to do with missing values. We make the distinction between non-existent and "unknown" or "missing" values. A non-existent value is one for which no valid category attribute combination exists. For example, in a database on cancer rates, a value for breast cancer for males (regardless of any other category attributes) does not exist. On the other hand missing values can occur for many other reasons.

It is not possible to get correct results when missing values exist. The condition that there are no missing values is a global condition of the SO, and not unique to each mapping. We say that the SO is "complete" if it has no missing values. Although this condition is obviously required for summarizability, it could be tolerated if the information on the missing values is added to the response to the summary operation. Thus, in the case that only a small number of values are missing, most of the results will be correct or near correct.

To summarize, the four conditions for summarizability are:

a) The SO has no multi-valued mappings between nodes in the branches of the SO tree.
b) The SO has no single-valued (or ID) mappings between nodes that point to the same aggregation A-node.
c) All the single-valued (and ID) mappings are full.
d) The SO is complete.

We will call a SO that fulfills all four conditions “summarizable”, and a SO that fulfills only the first three conditions “weakly summarizable”. The first two conditions are sufficient for a SO to be considered well-formed.

7.0 CONCLUSIONS

The work described here was motivated by limitations of current models for describing Statistical Databases. We have defined a new model, called the STatistical Object Representation Model (STORM), and showed how it overcomes these limitations. In addition, we have defined the conditions for a well-formed Statistical Object (SO), and the conditions for "summarizability", which are necessary to ensure that the results of statistical summaries are correct.

BIBLIOGRAPHY

[1st SDBM] Proc. of the first LBL Workshop on Statistical Database Management, Menlo Park, CA, 1981.

[2nd SDBM] Proc. of the second Workshop on Statistical Database Management, Los Altos, CA, 1983.

[3rd SSDBM] Proc. of the Third Workshop on Statistical and Scientific Database Management, Grand Duchy of Luxembourg, 1986.

[Chan & Shoshani 81] Chan P., Shoshani A. "SUBJECT: A Directory Driven System for Organizing and Accessing Large Statistical Databases" Proc. of the 7th Intern. Confer. on Very Large Data Bases (VLDB), 1981.

[Ozsoyoglu et al 85] Ozsoyoglu, G., Ozsoyoglu, Z.M., and Mata, F., "A Language and a Physical Organization Technique for Summary Tables", Proc. ACM SIGMOD Conf., 1985.

[Rafanelli & Ricci 83] Rafanelli M., Ricci F.L. "Proposal of a Logical Model for Statistical Data" Base" in [1st SDBM]

[Rafanelli et al. 89] Rafanelli M., Klensin C.J., Svensson P. Eds. "Statistical and Scientific Database Management" Lecture Notes in Computer Science, N. 339, Springer-Verlag Pub., 1989.

[Shoshani 82] Shoshani A. Statistical Databases: Characteristics, Problems and Solutions" Proc. of the 7th Intern. Confer. on Very Large Data Bases (VLDB), Mexico city, Mexico , 1982.

[Shoshani & Wong 85] Shoshani A., Wong H.K.T. "Statistical and Scientific Database Issues" IEEE Transactions on Software Engineering, Vol.SE-11, N.10, October 1985.

[Su 83] Su S.Y.W. "SAM*: A Semantic Association Model for Corporate and Scientific/Statistical Databases" Information Sciences, Vol. 29, N. 2 and 3, May and June 1983.

[Wong 84] Wong H.K.T. "Micro and Macro Statistical/Scientific Database Management" Proc. of the 1st IEEE Intern. Confer. on Data Engineering, Los Angeles, CA, 1984.