A MODEL FOR REPRESENTING STATISTICAL OBJECTS

Arie SHOSHANI* , Maurizio RAFANELLI+

* *Lawrence Berkeley Lab. - Univ. of California, 1 Cyclotron Road, Berkeley, CA 94720, USA.*

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

Abstract

In this paper the structure and the semantic properties of the entities stored in databases, whose data are only aggregate-type data, are defined and discussed. This choice is justified by the wide spread use of aggregate data without the corresponding raw data (i.e. micro-data, such as census data). Aggregate data are often derived by applying statistical aggregation (e.g. *sum, count)* and statistical analysis functions over micro-data, so that the relative databases are called "statistical databases". For this reason in this paper the above entities are called *statistical object* and a new representation model based on a graph representation is proposed. Finally representational problems in current models are identified and some solutions are proposed.

1. Introduction

Aggregate data are often derived by applying statistical aggregation (e.g. *sum, count*) and statistical analysis functions over *micro-data* [Won84]. For this reason these databases are often called *statistical databases* (SDBs) [Sho82], [SW85], [Raf89]. In this paper only aggregate-type data will be considered; this is a choice which is justified by the widespread use of aggregate data only, i.e. without the corresponding micro-data. The reason is that often it is too difficult to use the micro-data directly (both in terms of storage space and computation time, for instance, in the case of U.S. census data, which is made up of about ten billion records) and because of reasons of privacy (specially when the user is not the data owner).
Aggregate data are commonly represented and stored as *tables*. In this paper these tables are considered as complex structures which may have many possible representations (e.g. tables, relations, pies, graphs, time series, etc.) and, accordingly, the term *statistical object* (SO) for their conceptual abstraction is used. The rest of the paper defines a *STatistical Object Representation Model* (STORM) which is independent from

the above methodologies. As a consequence, an SO can then have a graphical representation, a 2-dimensional tabular representation, or any other representation preferred by the user (e.g., a relation). Then the concept of and conditions for the *summarizability* of an SO, which guarantee the correct results of summary operations over statistical objects, are developed. Finally some problems which arise with this type of complex data structure are discussed.

2. Basic concepts of statistical databases

This section starts by briefly presenting four basic concepts that are unique to SDBs [RS90a]:
1. Summary attributes - these are attributes that describe the quantitative data being measured or summarized (e.g. "income for socio-economic databases", or "production and consumption of energy data".
2. Category attributes - these are attributes that characterize the summary attributes (e.g. "Sex", "Race" and "Age" characterize "Population counts").
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., 21,373)).
4. Classification hierarchies - a classification relationship often exists between categories. For example specific "Products" (e.g. "Fruits", "Vegetables", "Grains" can be classified as "Agricultural Products").
The phenomenon described always has its "universe of definition" (for example, "Fruit products"). 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 is quantified by a numeric value datum which does not change in time (but new value can eventually be added). Finally, the summary data are always fixed also in place (for example, "in California"). Both the time and the place can be implicit, in the sense that they may appear as descriptive variables (for example, a numeric value datum can be characterized by "Years", whose instances are, for this SO, "1981, 1982,.., 1988", and by "State", whose instances are "California, Texas, Florida". 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.
This paper explore below some of the problems encountered using these methodologies in current models.

3. Problems with current logical models

In this section two kinds of problems will be briefly discussed: the two-dimensional (2D) table representation and the current graph-oriented models.

3.1 Problems with the two-dimentional tabular representation.
The 2D table 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. This paper points out several deficiencies.

3.1.1 The concept of multi-dimensionality is distorted

Let us suppose that you need to squeeze the multi-dimensional space into two dimensions (an action which is typically done by choosing several of the dimensions to be represented as rows and several as columns). For example, suppose you have to represent the "Average Income in California" by "Professional categories", "Sex" and "Year" and "Professional categories" are further classified into "Profession". Figure 1 is an example of a 2D tabular representation, but, obviously, one can choose (according to some other preferred criteria) other combinations by exchanging the dimensions. 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 leads to 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 [RS90a].

3.1.2 The concept of classification relationship is lost

In the 2D representation, classification hierarchies are represented in the same manner as the multi-dimensional categories, while in the former the dimention is only one.

Consider, for example, "Professional Categories" and "Professions" shown in Figure 1. 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, but part of a "classification hierarchy" which includes "Professional Categories" and "Professions". This means that 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.

3.1.3 Lack of metadata level

At present a 2D representation represents both the category names and the category instances together, that is there is no separate description of what the statistical database is about (meta-data). For example, the meta-data for Figure 1 consists only of "Average income in California" by "Sex", "Year" and "Profession", however "Professional Category" is represented together with the data values. Consequently, this representation becomes very large for tables with high dimensionality, or when the categories have a large number of instances and, then, it does not comfortably fit on a page or a screen. In such cases, the representation spreads into multiple pages or screens. 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.

3.1.4 Problems in a relational table representation

In a relational table representation no distinction between category and summary attribute is made; moreover large number of redundent category attribute values are present and the semantics of the relational algebra operators are complitely changed, so that it has to be redefined [RR91].

Average Income in California			**Professional Category**				
			Engineer		Secretary		Teacher
			Profession		**Profession**		**Profession**
			Chemical Engineer	Civil Engineer	Junior Secretary	Executive Secretary	Elementary Teacher
Sex							
Male	**Year**	81	*1,977*	*2,411*	*5,343*	*1,541*	*12,943*
		82	*2,099*	*2,780*	*5,421*	*1,698*	*13,521*
		. .	....	....	....	......	....
		88	*2,378*	*4,392*	*5,501*	*2,937*	*16,230*
Female	**Year**	81	*258*	*1,120*	*6,673*	*1,623*	*16,071*
		82	*289*	*1,276*	*6,925*	*1,744*	*17,528*
		. .	.....	......	......	......	......
		88	*398*	*1,941*	*8,115*	*2,426*	*21,067*

Figure 1

3.2 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 specially designated nodes.

For example, in GRASS [RR83] multi-dimensionality is represented by A-nodes (A stands for “association”) and C-nodes (C stands for “classification”), while in SUBJECT [CS81] it is represented by X-nodes (X stands for "cross product") and C-nodes (C stands for "clustering"). In [Su83] the Author proposes a semantic association model, SAM*, in which a table is represented graphically by a graph and extensionally by a generalized relation, or G-relation. This term is used because the relation is enlarged to two different types of attributes (category and summary), separated by a double line. The problem of large number of redundencies is not solved. These models also fail to solve some other problems discussed in the following.

4. The STORM model

In this section we propose a new model, called STORM (STatistical Object Representation Model), based on a graph representation. Before discussing this model, we define its data structure: the *statistical object* (SO).

4.1 The statistical object data structure

Definition: A Statistical Object 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, *Average Income in California*). Statistical object characteristics may be associated to each SO (for example, *statistical source*).

C is a finite set of category attributes (sometimes called *characters* by statisticians); each category attribute has a domain associated with it, and a *domain cardinality* which corresponds to the number of instances (sometimes called *modalities* by statisticians) of the domain for that category attribute. Each category attribute has a *primitive domain* (more category attributes can have the same primitive domain) and can also have a property called *unit of measure* , which represents the unit of the domain (we will see later as a domain can have more than one unit of measure). Sometimes this unit of measure is the same of the unit of measure specified in the SO characteristics.

S is a single summary attribute associated with the SO. The summary attribute has a domain and a domain cardinality associated with it. It has different properties, such as *unit of measure* (e.g., *tons*), or *summary type* (e.g., *percentage*), both of them are specified in the above-mentioned SO characteristics. The semantics of this last property will be explained below. Moreover, *marginal values* exist for each SO, i.e. the partial totals of one (or more) category attribute.

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

The following notation to describe an SO is used in this paper:

N (C(1), C(2), ..., C(n) **:** S),

where N and S are respectively the name of the statistical object and the name of the summary type (e.g., *sum*) of the SO, and (C(1), C(2), ..., C(n) are the names of the components of the category attribute set C. The function f is implied by the "**:**" notation. For example, the following describes an SO on various product sales in the USA, where the summary type is SUM and the unit of measure of the summary attribute is DOLLARS:

PRODUCT_SALES_IN_THE_USA (TYPE, PRODUCT, YEAR, CITY, STATE, REGION **:** SUM)

As already mentioned, a statistical object represents a summary of micro-data. This summary involves statistical functions (count, average, etc.), and units of measure of the phenomena of interest (gallons, tons, etc.). Accordingly, the summary attribute has the two properties mentioned above: *summary type* , and *unit of measure* . Note that the above SO is presumed to have been generated micro-data, such as the individual stores where the products were sold. Note also that the name of an 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 intended summary attribute 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 category attributes to the summary attribute.
In the next section, it will be shown that these structural semantics are not sufficient for describing an SO, since it needs 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 is it possible to know if sales figures are given for products, product types, or both? Further, suppose that it is known that figures are given for products, how is it possible to decide whether these figures can be summarized into product type? Similarly, suppose you need to know whether sales figures for cities can be summarized to state levels and to regions.
In order to answer this type of question, it is necessary to capture the structural semantics between category attributes and, in order to solve the above problems, the Authors propose and use the STatistical Object Representation Model (STORM) [RS90a].

4.2 The representation of an SO by STORM

It is best to visualize the STORM representation of an SO in a graphical form as a direct tree. Before discussing the graphical structure of an SO, it is useful to describe the two above-mentioned representation spaces.

4.2.1 The Intentional Space

In this space the STORM model defines three representation levels, called respectively *Topics level* (T-level), *Statistical Objects level* (S-level) and *Base level* (B-level).

T-level.

This is the conceptual level of the model, in the sense that different levels of aggregation (by concepts or topics), represented by a direct, acyclic, connected graph, can be grouped together according to criteria which can vary from designer to designer. The statistical user himself can redesign his *view* according to his own point of view (which, in any case, will be personal, similar to a sub-scheme of a relational DBMS). Only T-type nodes appear in the first level (T-level). A T node represents a topic, that is a subject according to which one or more SOs (or one or more other T nodes) are classified or aggregated. All the lower level T nodes (leaves of the graph) point towards one or more S nodes.

S-level.

At this level the intentional form of an SO is graphically represented by a tree structure. In it three types of nodes - S, A and C - appear. The root of the tree which represents an SO is always an S node. It represents the summary attribute of such an SO; each S node always has a name. The SO characteristics are linked to this name. An S node has only entering edges, from the top (from one or more T nodes) and from the buttom (from only one A node, or only one C node); if the entering edge is a C node, the SO is

a *vector* , that is, an SO with only one category attribute (only one dimention). The S node can also be *suspended*, in the sense that it has no entered edges; in this case it is called a *scalar* (and it is certain that the SO characteristics are *place* and *time*, for example *Population of the United States in the year 1981*).
An A node represents the aggregation of the category attributes which describe the above mentioned summary attributes; this aggregation is the Cartesian product of the instances of the definition domain of each above category attribute. An A node only points to one S node, but it has at least two entering edges (for the semantic definition of the Cartesian product). Note that an A node does not have a name.
A C node represents a category attribute; a definition domain is always associated to each C node. A C node always has a name, which can be the same in different SOs, while in the same SO different names have to appear (that is, there must not be any ambiguity). A C node can point to an A node, an S node (in this case we have a *vector*), or another C node (in this case we have a *classification hierarchy*). It is also possible to have a *partition* of a C node into different C nodes.
An example of T-level and S-level (in which a scalar, a vector, a classification hierarchy and a partition are present) is shown in Figure 2.

B-level
This level represents the 'primitive' or base category attributes, from which all the S-level category attributes are extracted.
In this level only C and A nodes appear, as both single C nodes, and classification hierarchies, or other simple structures. They are all unconnected, because only primitive concepts are represented, and these concepts are not linked to the SOs of the S-level. Each definition domain of a C node at the S-level is associated with only one base domain of a C node at the base level, and this definition domain is obtained by a mapping (total or partial).
Each C node at the B-level is a parent of at least one C node at the S-level.
For example, if the category attribute "year" appears in three SOs (at S-level), and if the instances of the respective domains are <1980, 1981, 1982>, <1981, 1982, 1983, 1984, 1985> and <1986, 1987, 1988>, then the base domain will have <1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988> as instances.
Note that the names of the category attributes at S level can be different from the name of the primitive domain at B level, from which they are extracted. This is the reason that the links between the primitive C node at B level and the other C nodes (extracted from it) at S level will allow to understand that *year* and *years* represent the same concept (this is important for the manipulation of SOs).
An example of B level and of the mapping between the B and the S levels is shown in the previous Figure 2.
It is important to observe that:
1) it is not possible for one edge to link two nodes of two different SOs; this means that every node C has to be repeated into two different SOs, even if the domain is the same.
2) only one edge can start from every C or A node.
3) an aggregation node has the domain generated by the Cartesian product of its component domains.
Then it is not possible to assign by default a "null" element to the range, and to consider the function to be "onto", where non-existing mappings map to the null element.

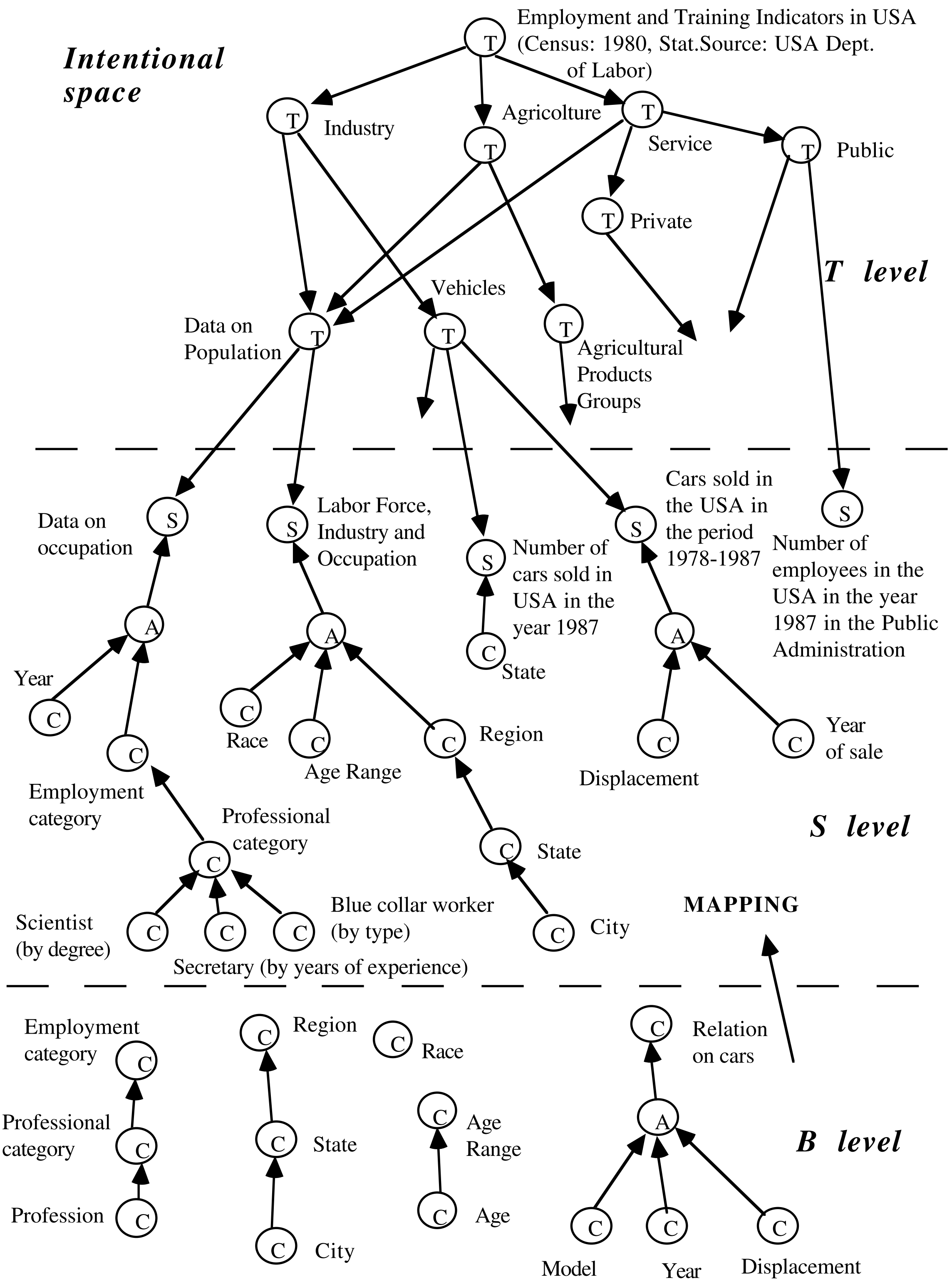

Intentional space
Employment and Training Indicators in USA (Census: 1980, Stat.Source: USA Dept. of Labor)
T
Industry
Agricolture
Service
Public
Private
T level
Vehicles
Data on Population
Agricultural Products Groups
Data on occupation
S
Labor Force, Industry and Occupation
Number of cars sold in USA in the year 1987
Cars sold in the USA in the period 1978-1987
Number of employees in the USA in the year 1987 in the Public Administration
A
C
Year
State
Race
Age Range
Region
Displacement
Year of sale
Employment category
Professional category
S level
State
Blue collar worker (by type)
Scientist (by degree)
City
MAPPING
Secretary (by years of experience)
Employment category
Region
Race
Relation on cars
Age Range
Professional category
State
B level
Profession
Age
City
Model
Year
Displacement

Figure 2

Nevertheless, for the same category attribute domain it is possible to have different units of measure (e.g., *gallons* and *litres*) or different terminologies or languages (e.g., *latte* or *milk*). In short, a STORM representation of an SO from the intentional point of view is a direct tree of nodes of type S, A, and C, with the following structural constraints:
a) There is only one S node and it forms the root of a tree.
b) Only one A (or only one C) node points to the S node.
c) Multiple C nodes point to an A node.
d) One or more C nodes (respectively, a classification hierarchy or a partition) can point to another C node.
In particular, a partition appears when in an SO the instances of a category attribute are, in turn, different classifications (see section 5.5, *Non homogeneous SO*).

4.2.2 The Extensional Space
It is often possible to confuse the concepts of *category attribute* and *instances.* For example, in Figure 2, if exists only the intentional space, *scientist, secretary* and *blue collar worker* should have both the meaning of category attribute names, and of instances of the higher level C node (*Professional category*).
The STORM model solves this ambiguity by introducing the extentional space, which regards both the S-level, and the B-level. The former is called called S Instances Level, the latter B Instances Level.
In Figure 3 an example of these spaces are shown.

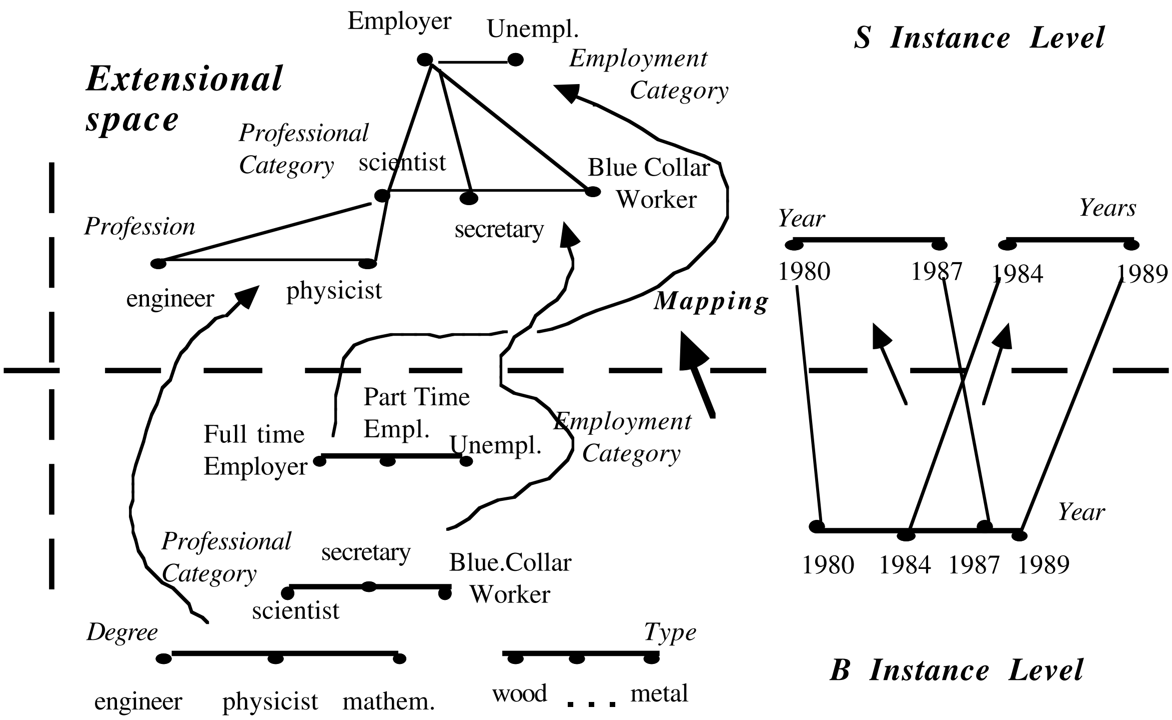


Figure 3

Distinguishing the extensional space from the intentional space, the meaning of *scientist, secretary* and *blue collar worker* is in the latter only *category attribute name,* while in the extensional space it is only *instance of the domain* which refers to the relative C node. Mapping exists between the two levels of the extentional space, in the same way that in the intentional space between the B and S levels. Note that, for instance, at B Instances Level, *Degree, Type*, etc. are not linked to *scientist, blue collar worker* , etc., because this is a particular classification depending on the particular SO. Instead, in Figure 4 an example of mapping between the extensional and the intentional space is shown. Note that at B Instances Level classification hierarchies can exist and, therefore, link between two different domains.

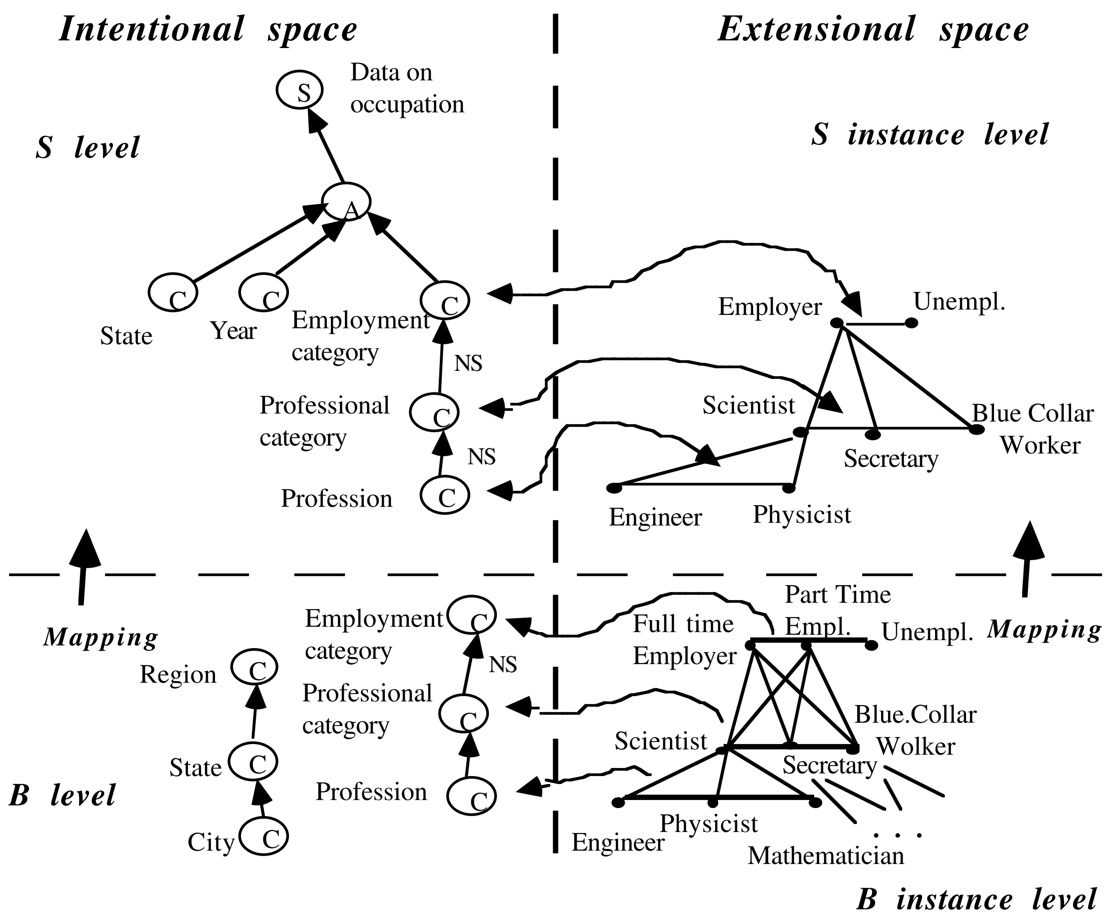


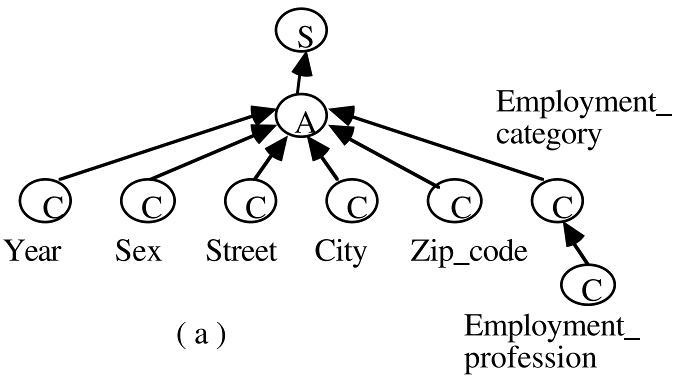
Figure 4

4.3 Canonical form of a statistical object

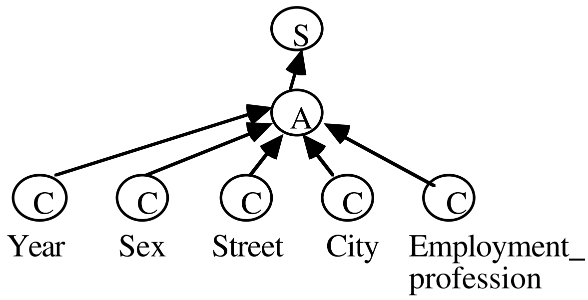
We define now the *key* of an SO:

Definition: The *key* of a statistical object is the n-ple (with n ≥ 2) consisting of the minimum set of the names of category attributes which: a) identifies in a unique way each summary data value at the lowest level of disaggregation; b) is able to produce all the other category attributes of the SO; c) has no redundancy.

Definition: The *canonical form* of an SO is the graphical representation of the SO key.
An example regarding the SO of Figure 5-a is shown in Figure 5-b. In particular, this means that only one C nodes level appears in the canonical form (therefore without classification hierarchies). Note also that *Zip code* does not appear in the canonical form because is available from *street* and *city* , that is, from the canonical form point of view this is a redundant information.



(a)



(b)

Figure 5

5. Irregular Statistical Object and Marginals

In this section some important problems, which arise when different levels of classification (the problem of graph *symmetry*), or different *types* of classification (the problem of graph *homogeneity*) appear, are discussed. Also identification and multi-valued dependencies are discussed, as well as non-summarizability (and knowledge of marginals). For all these particular situations *irregular* SO are defined; in these SO almost a labeled edge appears, which is inherited toward or from the B level. In the next sections we will discuss five tiypes of labels, distinguishing between labels which can appear only at S-level, only at B-level or both.

5.1 Identification (ID) dependency

It can happen that, in a classification hierarchy one instance of a category attribute is not sufficient to recognize exactly the object characterized by it; for example, suppose you have the SO of Figure 6, in which the classification hierarchy *region - state - city*

appears, and suppose that one instance of the category attribute *city* refers to different instances of the category attribute *state*.
This means that we have two cities having the same name which are in different states. Therefore, we need to know not only the name of the city, but also the name of the state to which this city refers. We call this *identification dependency* (or ID dependency). Obviously, such a dependency can appear in both the S and the B level, but if it exists at B level, it might not exist at S level, while if it exists at S level, must exist at B level.
This means that the ID dependency is inherited from S to B level, but the contrary is not true. In STORM this dependency is represented by a label *ID* on the edge which links the two category attributes (in the case of Figure 6, the edge between *city* and *state*, even if other labels refer to this edge).

5.2 Multi-valued (MV) dependency

The multi-valued dependencies are already known and studied in the relational theory. Here we wish to discuss how STORM resolves the problem (and represents this situation) when this dependency exists in an aggregate data structure such as the SO.
First of all, this dependency does not appear at S level, but only at B level.
In fact, the mapping between the B level and the S level transforms the MV dependency in a Cartesian product between the two category attributes. Multi-valued dependencies can exist between two (or more) C nodes, for example, *type* (instances of which are, for example, *wood* and *metal and plastic*) and *product* (e.g., *chair* and *table*) .
In Figure 6 an example of this dependency is shown (note the edge without arrow for the MV dependency, as well as the label MV on the same edge, at B level).

5.3 Non-symmetric (NS) Statistical Object

The problem of the non-symmetry of an SO arises when one category attribute is classified in a classification hierarchy, in which, in the extensional space, the number of levels is different, depending on the category attribute to which we refer [RS90b]. This means that the descriptive part is not decomposed symmetrically.
We distinguish two different types of non-symmetry: *structural,* (for example, the classification hierarchy of Figure 4, formed by *Employment category*, whose instances are *employed* and *unemployed*, and *Professional category*, which obviously refers only to *employed*, and whose instances are *Scientist, Secretary* and *blue collar worker*), and *occasional,* which appears only at S level and means *not available summary value* at a lower level for that particular SO (such as the case of Figure 6, where *data on production* for the cities of Arizona and Nevada are *not available*). The former is inherited from S level to B level (the contrary is not always true), the latter is never inherited. In STORM this situation is represented by a label *NS* on the edge which presents this non-symmetry (without any distinction between structural or occasional non-symmetry).

5.4 Non summarizability (N+) and describing marginals (M and Mo)

Let us now the define marginal values, or *marginals*.
Definition: We define *marginal* the correct value for existing instances of category attributes.

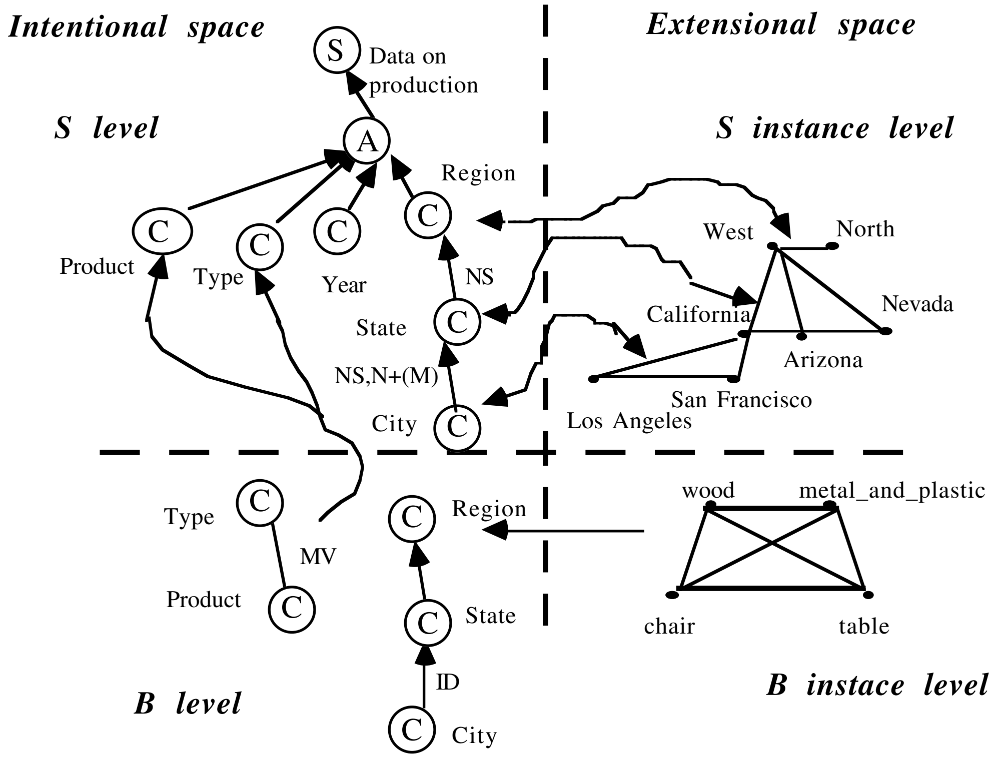


Figure 6

Often it is not possible to summarize one or more category attributes with regard to a given statistical object, both because it has no sense, and because the sum of the summary values carries out an uncorrect value.
The former happens some time in the case of the time series: for example, the population of the United States, counted every year, is not summable along the category attribute *year*. In this case a label *N+* appears near the edge outcoming from the C node which presents this non-summarizability .
The latter refers, instead, to two different situations:
a) when an overlap exists in the summary attribute depending on the category attribute not summarizable (for example, suppose that the phenomenon represented is *Number of diseases in the USA,* classified by *type of disease*; *state* and *year*). Obviously, a person can have two or more diseases, so that the marginal referring to the total diseases is lower than the sum of each number of a given disease ; if we know the marginal, this is a *recoverable* non-summarizability and, because we have an overlapping, the edge will be flagged by the label *N+ (Mo)* .
b) when the summary attribute does not represent the total of the population observed in the phenomenon described by the SO, with regard to one category attribute.
For example, suppose that the population of the C node *city* refers only to the cities *San Francisco* and *Los Angeles*, which are the only instances of the relative definition domain of *city* , while the population, which refers to the instance *California* of the C node *state* at the higher level of the classification hierarchy, regards *all* the population of

this state. The total of the population is, obviously, different at the different levels of aggregation.
If we have the marginal values of the higher level (i.e., we know the total population of the California), the edge outcoming from the C node at the lower level will be flagged by a label *N+ (M)* . This means that, in our case, it is possible to summarize with regard to *city* .

5.5 Non-homogeneous Statistical Object

The problem of the non-homogeneity of an SO arises when the instances of a category attribute are, in turn, different classifications [RS90b].
An example of SO non-homogeneity is shown in Figure 2 with regard to the category attribute *Professional category.*
This attribute is partitioned in three category attributes, *Scientist, Secretary* and *Blue collar wolker*, each of them classified by different criteria (in this case, respectively by *degree, years of experience* and *type*).
The solution proposed for the representation of this situation is shown in Figure 2, S Level, where the type of classification (for example, *by degree*) is specified inside the name of the C node. Instead, at the S Instances Level, three different domains exist (see Figure 3), which can also be linked between them (see Figure 4). Each of them have complitely different type of instances (e.g., *1-3, ... >9* for *Years of experience*, or *engineer, ... , physician* for *Degree*, etc.).
Note that this is only a representation problem, because in reality the population described by *Years of experience* or by *Degree* is the same, *Number of employees*, and they have to be read as *Scientist with degree* or *Secretary with years of experience.*

6. Representing and interpreting nulls in Statistical Objects

The presence of scarse data or null values is frequent in the statistical objects. In the case of statistical databases, the refined subdivision made in the theory of null values is not necessary. In fact, the statisticians basically use only two types of null values: Unknown (or *non-available,* NA) and *non-existing* , - (the latter are often called *structural zero entries* [Bis78]). In this model we also assume, for the same reasons of semplicity, the two above-mentioned types of null values.
An example of *non-available* null value is an SO relative to the production of fruit in the USA, in which only the data regarding the states of California, Texas and Oregon are present, while these relative to the other states are missing.
An example of *non-existing* data, instead, is an SO where the data relative to the reports of diseases subdivided into *type of disease* and *sex* are represented; in this case the missing data regard *type of disease = cancer of the prostate gland* for *sex = female* or *type of disease = breast cancer* for *sex = male* (the value will be zero, but it will be a structural zero, in that it can never assume a value which is different from zero). From the summarizability point of view, these are regular zeroes and therefore can be summable.
Note that there is an important difference between the two types of null values, specially with reference to the relative marginals. In particular, in the former (non-available) it is not possible to computate the total, which begins *non-available.* In the latter (non-existing), the total is the sum of zero (structural) with the other numbers, although, for example, the total of the prostate cancer cases without referring to sex is the total

reported in the marginal, but in reality it represent the total of males alone and not the entire population.

7. Conclusions and future works

The work described here was motivated by limitations of current models for describing Statistical (aggregate) Databases. We have defined a new model, called STORM (STatistical Object Representation Model), and showed how it overcomes these limitations. In addition, we discussed some anomalous situations (non-simmetricity, non-summarizability, non-homogeneity, etc.) and proposed the suitable solutions by the above model.
The future evolution of this model can be resumed in the following points:
- enlargement of the model in order to represent generalized objects (i.e., not only aggregate data, but also conventional disaggregate data, represented by relations);
- definition of a DDL and its implementation;
- definition of a query language, able to browse and to query the database defined by the previous DDL, and its implementation.

BIBLIOGRAPHY

[Bis78] Bishop Y. "Discrete Multivariate Analysis. Theory and Pratice" MIT Press, 1978

[CS81] 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.

[Raf89] Rafanelli M. "Statistical and Scientific Database Management Systems" Encyclopedia of Computer Science and Technology, Kent A. & Williams J.G. Eds, M.Dekker Publ., 1989

[RR83] Rafanelli M., Ricci F.L. "Proposal of a Logical Model for Statistical Database" Proceed. of the second International Workshop on Statistical Database Management, Los Altos, CA, 1983

[RR91] Rafanelli M., Ricci F.L. "A functional model for macro-databases" ACM-Sigmod Recors Vol. 20, N°1, 1991

[RS90a] M.Rafanelli, A.Shoshani :"STORM: a statistical object representation model" in "Statistical and Scientific Database Management" Lecture Notes in Computer Science, N. 420, Springer-Verlag Pub., 1990
and
IEEE Data Engineering, Vol. 13, N°3, Sept. 1990

[RS90b] M.Rafanelli, A.Shoshani :"On the representation problems of statistical object", International Conference on Engineering Information in Data Base and Knowledge Based Systems, Berlin, Dec. 4-6, 1990

[Sho82] Shoshani A. Statistical Databases: Characteristics, Problems and Solutions" Proc. of the 7th Intern. Confer. on Very Large Data Bases (VLDB), Mexico city, 1982.

[SW85] Shoshani A., Wong H.K.T. "Statistical and Scientific Database Issues" IEEE Transactions on Software Engineering, Vol.SE-11, N.10, October 1985.

[Su83] Su S.Y.W. "SAM*: A Semantic Association Model for Corporate and Scientific/Statistical Databases" Inform-ation Sciences, Vol. 29, N. 2 and 3, May and June 1983.

[Won84] 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.