Some Properties of Statistical Information:

Pragmatics, Semantics, and Syntactics

Bo Sundgren

SCB

R&D Report
Statistics Sweden
Research - Methods - Development
1992:16

INLEDNING

TILL

**R & D report : research, methods, development / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1988-2004. – Nr. 1988:1-2004:2.
Häri ingår Abstracts : sammanfattningar av metodrapporter från SCB med egen numrering.**

Föregångare:

Metodinformation : preliminär rapport från Statistiska centralbyrån. – Stockholm : Statistiska centralbyrån. – 1984-1986. – Nr 1984:1-1986:8.

U/ADB / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1986-1987. – Nr E24-E26

R & D report : research, methods, development, U/STM / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1987. – Nr 29-41.

Efterföljare:

Research and development : methodology reports from Statistics Sweden. – Stockholm : Statistiska centralbyrån. – 2006-. – Nr 2006:1-.

R & D Report 1992:16. Some properties of statistical information: pragmatics, semantics, and syntactics / Bo Sundgren.
Digitaliserad av Statistiska centralbyrån (SCB) 2016.

urn:nbn:se:scb-1992-X101OP9216

Some Properties of Statistical Information:

Pragmatics, Semantics, and Syntactics

Bo Sundgren

SCB

R&D Report
Statistics Sweden
Research - Methods - Development
1992:16

Från trycket	December 1992
Producent	Statistiska centralbyrån, utvecklingsavdelningen
Ansvarig utgivare	Lars Lyberg
Förfrågningar	Bo Sundgren, 08/783 41 48

ISSN 0283-8680

SCB-Tryck, Örebro 1993.01 **Miljövänligt papper**

Bo Sundgren 1991-02-01

SOME PROPERTIES OF STATISTICAL INFORMATION: PRAGMATICS, SEMANTICS, AND SYNTACTICS

0 Introduction

Applying semiotics, the theory of symbols, to information in general, and statistical information in particular, we may distinguish between three aspects, three types of properties (cf Stamper (1973)):

(a) syntactical properties, having to do with the relations between the symbols of information (in a message);

(b) semantical properties, having to do with the relation between the symbols and messages, on the one hand, and the reality that the symbols and messages refer to or represent, on the other;

(c) pragmatical properties, having to do with the relation between the messages and the effects of them on a human receiver;

In more ordinary language we may say that pragmatics deals with the purpose and usage of information, semantics deals with the contents and meaning, and syntactics deals with the physical and technical aspects: how information can be represented and processed.

1 Pragmatical properties of statistical information

An important purpose of information is to facilitate some kind of decision-making. With a wide interpretation of "decision-making", it may even be claimed to be the only purpose of information. However, there are also usages of information that most of us would problably describe in other than decision-oriented terms. For example, in science and research we may collect and analyze information with the primary purpose to get an understanding of how a certain system, or "piece of reality", works.

What is the difference between statistical and non-statistical usage of information? Statistical information is typically used for decisions that may be vaguely described by such terms as "strategical", "management-level", "policy-oriented", if the usage environment is a business or government organization. In a research environment statistical information is typically used for getting an overview of a more or less complex system (sometimes called the universe of discourse or the system of interest), and for formulating and testing hypotheses, and ultimately theories, about this system.

We may describe the typical usages of statistical information as being of a **directive** nature: statistical information gives direction (advice, guidance) to the decision-maker or scientist, but it is usually not the only factor that determines the final conclusions and actions.

As a contrast, non-statistical usage of information is typically much more **operative**. On the basic activity level of a business or government organization, decisions are oriented towards concrete, individual "cases" like the processing of a certain order from a certain customer, or the court trial of a certain person, suspected to have committed a certain crime at a certain time, in a certain place.

To be precise, one should distinguish between *"statistical usage of information"* and *"usage of statistical information"*. Very often, of course, the two go together, but there are also situations, where statistical information is used for non-statistical purposes, that is, information of a nature, which is usually used for directive purposes as those described above, is instead being used for operative, "individual case" oriented decision-making. For example, consider a physician who is treating a patient. He or she may may collect a lot of statistical information about the health condition of the patient. However, the purpose is very operational; the physician should determine (a) from which illness, if any, the patient is suffering, and (b) how to cure the patient. Similarly, in a factory, statistical information may be used for an operational decision whether to accept or reject a certain lot of manufactured products.

There are also many examples of situations where non-statistical information is used for directive decision-making. For example, a politician, who is going to make a decision about how to distribute some government support to different parts of a country, will typically get a lot of statistical information as a basis for the decision. However, most politicians are also likely to be influenced by other types of information, like impressions from having been "on the spot", arguments from lobbyists, and even tactical considerations in view of coming elections.

Thus information, which is of such a nature that we recognize it as statistical information, is typically used for knowledge-formation and decision-making of a nature that we have here labeled as "directive". In other words statistical information is one important type of directive information.

We have also seen that statistical information can be a component in operative decision-making. However, most of the (operative) information used for operative decision-making is of a nature that we recognize as non-statistical.

What then is it in the nature of a piece of information that makes us recognize it as statistical information? This is a question of semantics, which will be addressed in the next section of this paper. Before going there we shall look at some further properties of statistical information, which are connected with its usage.

The activity or process where certain operative information is used is typically also the source of the information. Thus there is a short distance between the birth and usage of operative information. Moreover, the link between the source and usage of operative information is usually very direct and explicit: the information is collected for a very well defined need, and this need is both a necessary and sufficient reason for the information collection. Thus it is both desirable and feasible to tailor the definition of the information in accordance with the one and only need for it.

As a contrast, the need for statistical information (as well as for other types of directive information) is much more negotiable. It is very seldom that a certain piece of statistical information, defined in exactly a certain way, can be claimed to be absolutely necessary for a certain purpose. In fact, directive decisions can be characterized by the fact that they need no information support at all in order to be taken; the decision-maker can always toss a coin, and whatever the outcome, the operations of the organization will probably continue to work, at least for some time. In the long run directive information is naturally expected to improve the performance of the operations, but the links between the information and its effects are much more complicated and less direct and obvious than in the case of operative information and decision-making.

Operative information can be classified as "needed" or "not needed" with respect to a certain operation. Statistical information, on the other hand, can at best be attributed a certain value for the expected improvement of the quality of a certain decision or set of decisions. This value has to be balanced against the costs for the acquisition of the information, that is, the costs for observation, measurement, and processing that is necessary for making the information useful.

Sometimes it is only through the pooling of several needs that the costs for the acquisition of statistical information can be justified. Such situations are called "multi-purpose", since the statistical information collected will be used for several purposes. The conceptual design of multi-purpose information is complicated, since the different information needs are not necessarily compatible. "Common denominators" must be found, and costs must be kept down.

2 Semantical properties of statistical information

Semantics is a research area within informatics and computer science that is receiving growing attention. There are many different names for this subdiscipline: conceptual modelling, infological modelling, semantical modelling, and knowledge representation, to mention some of them.

Problems having to do with the contents and meaning of information have received a lot of attention in statistical organizations, even before computers were introduced. This is not surprising, because it is often more difficult, and at the same time more critical, to define the meaning of statistical information. Why? The answer has to do with what we discussed in the previous section, the usage and purpose of statistical information.

The meaning of operative, non-statistical information is often very obvious and clear for its users. Most statistical products of statistical organization are multi-purpose to some extent, very often to a great extent. This is one reason for statistical organization becoming interested in the semantical aspects of information even before the age of computers. For example, statistical organizations, both national and international, have a long tradition in developing standard definitions, classifications, nomenclatures, code lists, and registers, in order to improve the compatibility, comparability, and usefulness of statistical information.

2.1 OPR/ER methodologies for conceptual modelling

From the definitions in the introduction follows that the semantics of information has to do with the relation between (a) the symbols and messages that constitute information, on the one hand, and (b) the reality and parts of reality that the symbols and messages refer to, on the other.

Thus in order to develop a model of the meaning of a piece of information, we must be able to describe how the piece of information and its parts are related to the piece of reality that the information refers to. One type of conceptual framework, which has become rather popular for doing this is the so-called Entity-Relationship (ER) model.

The Entity-Relationship model is often ascribed to Chen (1976). However, it is a fact that similar conceptual models, originated by several European authors, had been presented and used for at least a decade before the appearance of Chen's paper; for example, see Langefors (1966), Sundgren (1973, 1974), Durchholz and Richter (1974), Lindgreen (1974). The methodology used at Statistics Sweden was called the infological model, or the Object-Property-Relation (OPR) model, later extended to the Object-Property-Relation-Event-Message (OPREM) model; see Malmborg (1982).

Conceptual modelling according to OPR or ER approaches have not particularly emphasized the properties of statistical information. On the contrary most variations of these modelling methodologies have focused on the type of factual information about individual objects that is typical for operative or administrative information systems. This does not mean that OPR/ER approaches cannot be used for analyzing and modelling statistical information. However, there are still needs to refine some of the concepts to make these models even more suitable for cope with statistical information. We shall return to this topic.

All OPR/ER methodologies for describing the semantical aspects of information are based on three fundamental concepts: **objects** (called "entities" in Chen's approach), **properties**, and **relations** (relationships). In some approaches a fourth concept, **time**, can also be regarded as fundamental. The methodologies assume that any piece of reality that is informed about by some collection of information can be conceptualized and modelled in terms of these basic concepts. The piece of reality, which is the object of the conceptualization and modelling for a more or less well-defined purpose, is called the **object system** or, with a term from logic, the **universe of discourse**; see also (ISO-rapporten).

In this paper the term "**entity**" will be used in a general and "neutral" sense to refer to all types of components of the object system: primitive components like objects, properties, relations, and time, as well as derived components like variables and values.

2.2 The theory of elementary messages and the principles of entity/reference and instance/type distinction

According to Langefors (1966) the information itself consists of messages. The smallest type of message that conveys meaningful information is called an elementary message, or e-message. There are two types of e-messages: property type e-messages, and relational type e-messages.

An **e-message of property type** tells that a certain object in the universe of discourse has a certain property at a certain time.

An **e-message of relational type** tells that two or more objects are related to each other in a certain way at a certain time.

Entities in the object system are referred to by names or other types of references. A **name** is a direct, explicit reference to an object system entity. A reference is either a name or an expression in terms of other names and references. In this paper a formal language INFOL, based on the conceptual algebra (see Sundgren (1989)) will be used for forming reference expressions. A summary of the formal specification of INFOL is given in an appendix to this paper (section 4).

It should be noted that names and references, being the constituents of messages, and belong to the "information sphere", whereas the entities that they refer to belong to the "object system sphere". It is an extremely important principle of informatics to distinguish between these two spheres; this is something that every professional programmer is painfully aware of, since he has almost certainly many a time caused himself a lot of trouble by mixing up the name of a variable with the variable itself, but it is an equally important principle in the theory of information. We shall refer to this principle as *the entity/reference distinction principle.*

It follows that the structure of an e-message can be described by the following two patterns:

Property type e-message: $<\rho(o), \rho(p), \rho(t)>$;

Relational type e-message: $<<\rho(o_1), ..., \rho(o_n)>, \rho(R^n), \rho(t)>$;

where "o" and "o_i" denote objects, "p" denotes a property, "R^n" denotes an n-ary relation, "t" denotes a time entity (which can be either a point of time or a time interval), and "$\rho(x)$" denotes a reference to entity x.

Properties are often (but not always) thought of as <variable, value> pairs. For example, the property of "being 25 years old" can be thought of as a variable "age" taking a value "25" from its domain of possible values. The structure of this conceptualization of a property type e-message is:

Variable/value type e-message: $<\rho(o), <\rho(V) = \rho(a)>, \rho(t)>$;

where "V" denotes a variable, and "a" denotes a value belonging to the domain of values, or value set, of V.

We formulated above the entity/reference distinction principle. An equally important principle, for informatics in general and for conceptual modelling in particular, is what we may call *the instance/type distinction principle.* Like the first one, this princicipie also has its roots in classical philosophy.

According to all OPR/ER methodologies for modelling the semantics of information, the universe of discourse is conceptualized on two levels of abstraction: the type level and the instance (or occurrence) level.

On the instance level, we think of the real world as a (time-varying) collection of individual object instances, where each object instance is associated with a (time-varying) collection of individual property instances and related in to a (time-varying) collection of other object instances.

On the type level, we recognize that there is often a subcollection of object instances that are "similar" in some sense that makes it justified to regard them as instances of one and the same object type. Usually every object instance of a universe of discourse is considered to belong to at least one object type.

Similarly, most property instances in an object system can be categorized into property types. Within a certain property type the properties are "similar" in some sense that makes it natural to think of them as "values" of one and the same "variable"; cf the variable/value e-message format above. In the literature on conceptual modelling, the term "attribute" is often used with the same meaning as we use "variable" here, that is, to denote property types.

Similarly again, the individual relationships that hold between an object instance of a certain type and one or more instances of other (or even the same) object types can also be classified into "subcollections of similar relationships" or relation types. For example OWN could be a relation type, the instances of which relate instances of the object type PERSON with instances of the object type CAR.

Note. In practice it is often difficult to maintain the instance/type dualism in the terminology used for the concepts. One and the same term may be used to refer to the corresponding instance/type concepts; for example, the term "object" may be use to refer to both "object instance" and "object type". Of course, such ambiguity should only be tolerated if the applicable abstraction level is perfectly clear from the context; otherwize the qualifiers "instance" and "type" must be used.

If we apply the principles of entity/reference distinction and instance/type distinction to the theory of elementary messages, we may define an important type level concept in the sphere of information: the concept of **elementary information kinds**, or e-message types. There are two major categories of e-message types:

Attributive e-message types: $<\rho(O), \rho(V), \rho(T)>$;

Relational e-message types: $<<\rho(O_1), ..., \rho(O_n)>, \rho(R^n), \rho(T)>$;

where "O" and "O_i" denote object types, "V" denotes a variable (an attribute), R^n denotes a relation type, and "T" denotes a **time domain** (a set of time points or time intervals).

In the infological language INFOL an attributive e-message type is referred to by expressions with the following format:

e-message type reference: <object type reference>.<variable reference>;

In the simplest special case, the reference has the form

<object type name>.<variable name>;

Examples: PERSON.name, PERSON.sex, PERSON.age, CAR.registration_number.

In most languages for specification of information, like query languages for databases, the time component is often left out completely in references to information kinds. Leaving out the time component usually does not mean that the time component is irrelevant. It only means that it has not been captured in the formal specification of the information; it is either implicitly understood (by a human) from the context, or it is dwelling somewhere else in the metainformation accompanying the information itself. This could be regarded as an important imperfection in the present state of the art of information specification languages.

In INFOL a time component could easily be added to the format stated above:

e-message type reference:
<object type reference>.<variable reference>(<time reference>);

Examples: PERSON.age(1990-07-01), PERSON.income(1990).

Relational e-message type references can often be seen as implicit components of reference expressions evaluating to variable references. A simple example is "the mother's age of a person", where "person" is an object type, and "the mother's age" is a variable of "person"; of course this attributive e-message type is derivable from the relational e-message type <<"female person", "person">, "mother of", T> and the attributive e-message type <"person", "age", T>. In INFOL this definition would be expressed in the following way:

PERSON.mother_age <--- MOTHER.age;

based on the assumption that "PERSON" has been specified as an object type, "age" has been specified as a variable of "PERSON", and "MOTHER" has been specified as a (binary) relation between "PERSON" and (a subtype of) "PERSON". (The arrow "<---" denotes "is defined as".) "MOTHER.age" is an expression, which according to the syntax rules of INFOL evaluates to a variable reference, where the variable is supposed to relevant for the *current object type*, in this case "PERSON", which is made "current" by the expression to the left of the arrow.

2.3 Object classifications and generic hierarchies

In the original versions of OPR/ER methodologies there were only the two levels of abstraction that result from the instance/type distinction principle. Over the years, starting with Smith & Smith (1977), several additional levels and dimensions of abstraction have been identified, which facilitate a semantically richer analysis and modelling of information.

We have seen that the instance/type abstraction applied to objects is essentially a classification of object instances into object types. There is no reason why a classification process could not be repeated in such a way that we get an n-level **classification hierarchy**. Such classification hierarchies are alternatively called

generalization/specialization hierarchies or **generic hierarchies**. For example, the object type PERSON may be specialized into the *subtypes* MALE_PERSON and FEMALE_PERSON, and the object types BIKE, CAR, and BUS may be generalized into the *supertype* VEHICLE.

In generic hierarchies, variables are **inherited** from supertypes to subtypes, and so are properties, which are common for all objects in a supertype. Thus all variables, which are relevant for the object type PERSON, will also be relevant for the subtypes MALE_PERSON and FEMALE_PERSON, and all instances of both subtypes will automatically inherit all properties which are common for all instances of the PERSON supertype, like "having two eyes". On the other hand, the subtype will have certain variables and properties, which distinguish them from other subtypes on the same level in the generic hierarchy. For example, MALE_PERSON may have a variable "military service done?" and FEMALE_PERSON may have a variable "number of pregnancies". Furthermore MALE_PERSON will have the unique common property *sex="male"*, whereas FEMALE_PERSON will have the unique common property *sex="female"*; this implies also that "sex" is a variable (of the supertype), which serves as a **classification key** for the classification into subtypes.

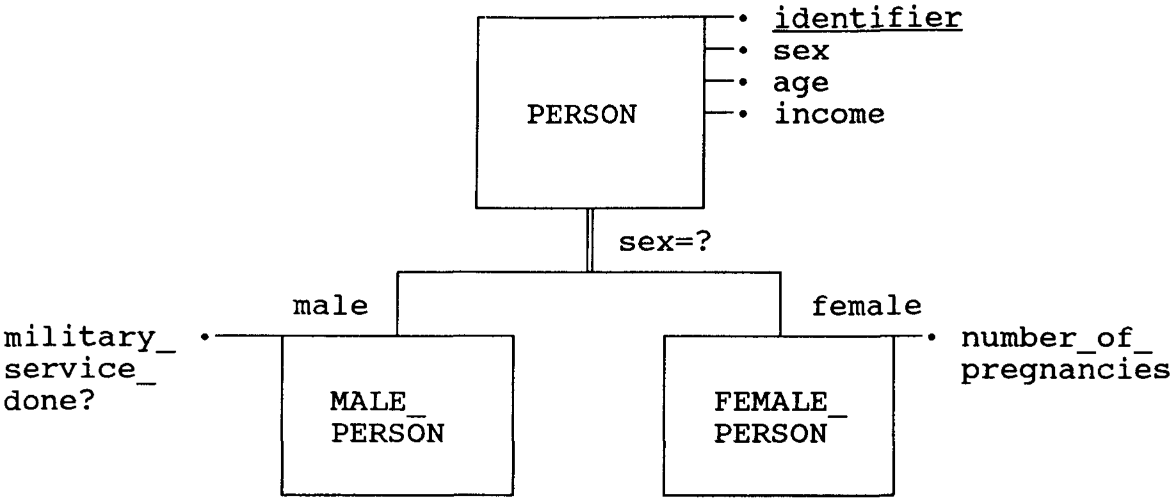
A generic hierarchy may be graphically visualized as in figure 1, which also contains the relevant definitions expressed in the INFOL language.

In statistical surveys object classifications and generic hierarchies occur in several forms and for different purposes. For example, within a certain population different subsets of objects, called **domains of interest**, or domains of study, are usually specified. The subsets may be specified one by one (cf INFOL expressions on the form "<object type reference> **with** <property reference>") or as a crossclassification by means of a Cartesian product of variables (cf INFOL expressions on the form "<object type reference> **by** <variable reference>", where the variable reference may be an expression evaluating to a Cartesian product of variables.

<u>**Examples**</u>:

Single subset:	PERSON **with** citizenship = "foreign";
Crossclassification:	PERSON **by** age_group × sex;

Subsets of populations are also specified for **stratification** purposes. Although the purpose is different, the specification of a stratum can formally be expressed in exactly the same ways as specifications of domains of interests.



```
INFOL definitions:

persons_by_sex <--- PERSON(by sex) =

                {MALE_PERSON, FEMALE_PERSON};

MALE_PERSON <--- PERSON(with sex = "male");

FEMALE_PERSON <--- PERSON(with sex = "female");
```

Figure 1. Graphical illustration (with accompanying INFOL definitions of the classification hierarchy "persons_by_sex").

2.4 Abstraction by statistical aggregation

It should be noted that there is only one basic collection of object instances which is classified in a classification hierarchy. This abstraction mechnism should be carefully distinguished from another one, which is very typical for statistical information: the **statistical aggregation hierarchy**. (Unfortunately there is again another type of aggregation, which has nothing to do with the statistical concept of aggregation; see Smith & Smith (1977).)

The meaning of statistical aggregation can be described in two steps. The first step is an object classification of any of the types described in the previous section (including the special cases of subtyping and crossclassification).

In the second step of statistical aggregation the instance/type distinction is "moved one level" with respect to the objects and object classes in the object classification hierarchy defined by the first step. This creates "a higher level of abstraction", where the object classes (types, subtypes, and supertypes) in the classification hierarchy will now be regarded as object instances rather than as object types. These "collective object instances" will be called **aggregated object instances**, and the corresponding "collective object type" will be called an **aggregated object type**.

As an example we may consider the classification hierarchy "persons_by_sex" in the previous section. The individual persons are the object instances in this classification hierarchy, and the object classes (populations and domains of interest) "PERSON", "MALE_PERSON" and "FEMALE_PERSON" are the object types, making up a supertype/subtype hierarchy. Applying the statistical aggregation abstraction mechanism to this hierarchy implies a change of perspective, so that the object classes are no longer looked upon as object types (only), but (also) as object instances on a higher level of abstraction. The object type for these collective instances could be called (in this example) PERSON_GROUP. This object type has three instances according to our definitions:

PERSON_GROUP = {PERSON, MALE_PERSON, FEMALE_PERSON};

= {PERSON **by** sex};

= (PERSON **by** sex).**agg**;

The two last lines above illustrate two alternative INFOL formalisms for expressing derivation of object types by statistical aggregation; one uses the set brackets, {...}, and the other one an operator, **agg**, to indicate the aggregation abstraction.

Like other objects the object instances of PERSON_GROUP may have properties, and these properties may be expressed in terms of variables for PERSON_GROUP. The properties and variables of aggregated objects are very often derivable by some type of **aggregation process** (frequency counting, summarization, computing of averages, percentages, variances, correlations, etc) from properties and variables of the object instances on the next lower aggregation level.

Thus aggregation of object types and aggregation of variables often go hand in hand. This is also reflected in the INFOL formalism. Instead of writing

(PERSON **by** category).**agg.count** (or, equivalently, {PERSON **by** category}.**count**);

it is possible to write

(PERSON **by** category).**count**;

In the latter writing an operator like **count** (or **sum**, **avg**, etc), which aggregates the values of a variable for a (possibly classified) set of objects, *implies* a preceding object aggregation, as expressed by **agg** or {...}.

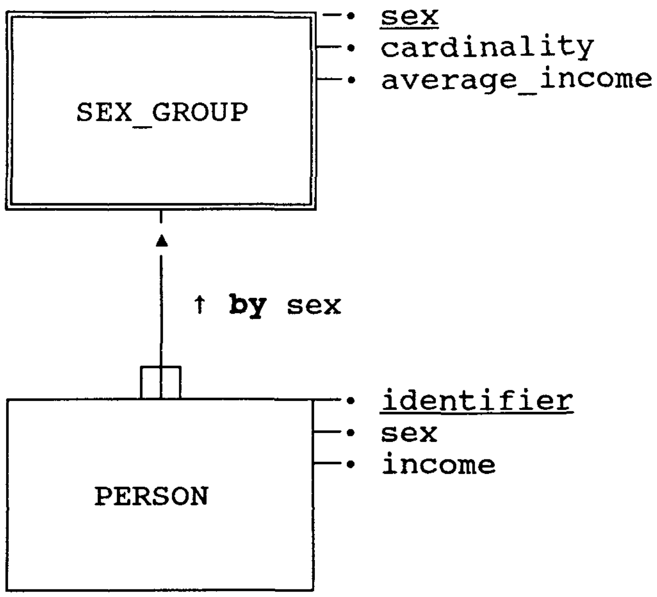
As an example of statistical aggregation we may again consider a group of persons that is classified according to their sex into males and females. As long as we do only this, it is a pure classification hierarchy. As we have seen, the classification can be used for structuring the information into subsets, which are homogeneous with respect to relevant variables and common properties.

However, we may also start to derive properties of the sexgroups as such, for example by simply counting or estimating the number of persons belonging to each group, or by computing or estimating the average income for each group on the basis of the incomes of (a sample of) the respective person instances. On the lowest abstraction level (aggregation level) the object instances will be persons having incomes. On the next level there will be only two instances: "males" and "females" having their respective cardinalities and average incomes as properties.

An aggregation hierarchy may be graphically visualized as in figure 2. The figure also gives some example of the INFOL syntax by showing the definitions for the derivable concepts in the figure.

Abstraction by means of aggregation is one of the most typical semantical features of statistical information. The prefixes *micro-* and *macro-* are often used to distinguish between statistical information before and after aggregation. Sometimes the very term *statistical information* is actually reserved for macroinformation, that is, information that has been subject to some type of aggregation process. However, we shall follow here the more common practice of letting both microinformation and macroinformation be called statistical information, as long as the purpose (or at least one of the most important purposes) of the microinformation is to serve as a basis for aggregation processes and other forms of statistical information processing.

In many statistical information systems the input information consists of unaggregated microinformation, whereas the output information consists of aggregated macroinformation. However, this is a rule with exceptions. For example, if a national statistical office collects economical information from companies, this information is microinformation for the survey processing in the statistical office, but for each one of the companies contributing to the survey, the contributed information is likely to be aggregated from numerous economical transactions inside the company.



INFOL definitions:

```
SEX_GROUP <--- {PERSON(by sex)};

SEX_GROUP.sex <--- some PERSON.sex;

SEX_GROUP.cardinality <--- PERSON.count;

SEX_GROUP.average_income <--- PERSON.avg(income);
```

Figure 2. Graphical illustration of an aggregation hierarchy; with INFOL definition expressions.

Thus the *micro/macro* distinction is relative rather than absolute. If microlevel objects are classified and abstracted into macrolevel objects, these object can again be regarded as microobjects and be subject to classification and abstraction into macroobjects on a yet higher level of abstraction. A special case is if the original classification is a hierarchical classification consisting of more than two levels. An example of statistical aggregation based on a **multilevel classfication hierarchy** is shown in figure 3. The figure also shows how such a **multilevel object aggregation hierarchy** can be graphically represented in a more compact way.

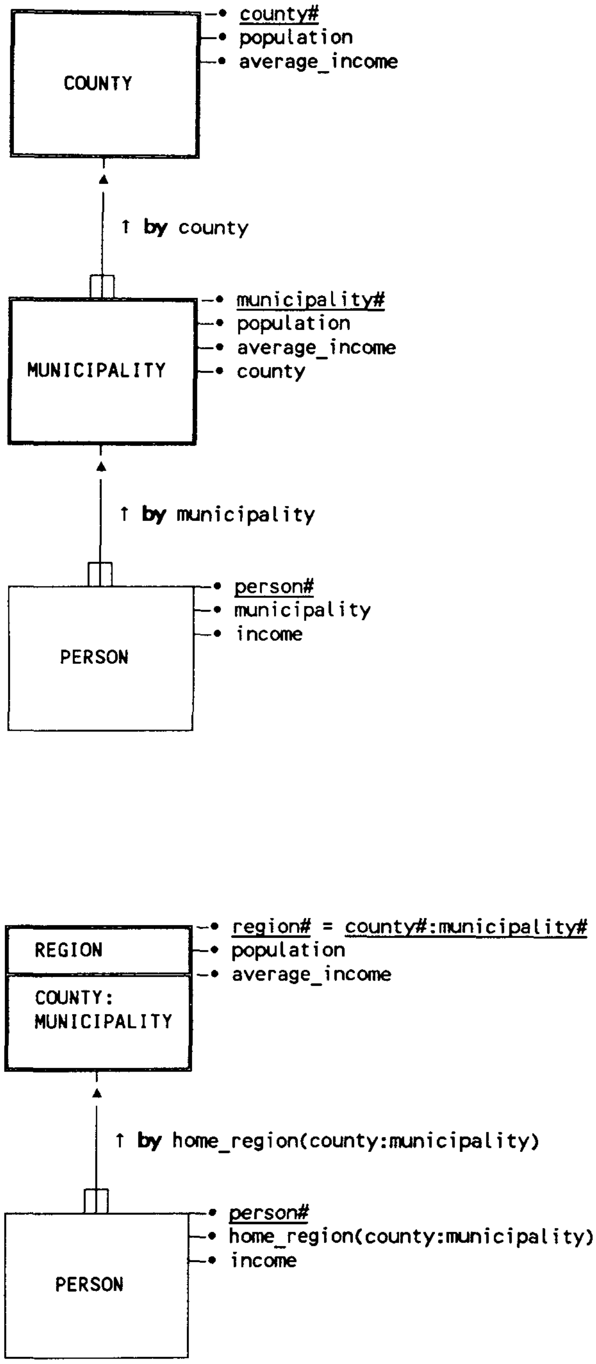


Figure 3. **A multilevel object aggregation hierarchy.**

2.5 Value aggregation and hierarchical variables

As illustrated by the last example above, multilevel object aggregation often goes hand in hand with hierarchical variables like region# = county#:municipality#. A **hierarchical variable** is a variable with a **hierarchically structured value set**, and a hierarchically structured value set is a value set, which can be seen as the result of a **value aggregation** process similary to the object aggregation process described in the previous section.

Object aggregation starts with the set of object instances of an object type. Value aggregation analogously starts with the set of values of a value type. The values are classified into subsets of the value set, and each one of these subset is then, after abstraction, regarded as a value "on a higher level". The higher level values are thus defined as an aggregation (= classification + abstraction) of the lower level values.

For example, suppose that we start with a value type having the value set

V = {1, 2, 3, 4, 5, 6};

and that this value set is classified in the following way:

A = {1, 2, 3}; B = {4}; C = {5, 6};

Now we may abstract the subsets A, B, and C into higher level values of a higher level value set

W = {A, B, C};

which we may write

W = {A{1, 2, 3}, B{4}, C{5, 6}};

in order to indicate the definitions of the higher lever values in terms of the lower level values. By taking the union of V and W we finally get a complete, hierarchically structured value set of two levels:

U = {A{1, 2, 3}, 1, 2, 3, B{4}, 4, C{5, 6}, 5, 6};

We can see that each reference to a higher level value includes the definition of the value in terms of the lower level values. Thus the specification of the new, hierarchical value set includes both the values and the structure between the values. However, the naming convention used here is not very practical. A more practical way of referring to the values, which still retains the structure visible, would be the following one:

U = {A, A:1, A:2, A:3, B, B:4, C, C:5, C:6};

The lower level values are referred to by a "family name", indicating the "parent value" on the next higher level in the hierarchy, and a "first name", indicating the "member of the family". An even more common practice is to rename the low level values, using the fact that the "first names" need be unique only "within the family":

U = {A, A:1, A:2, A:3, B, B:1, C, C:1, C:2};

Now we have finally arrived at the typical pattern for naming values in hierarchical value sets. It is also customary to give a name to a hierarchical value set that indicates the hierarchical structure by having a name component for each level in the hierarchy. Example:

region = county : municipality;

Like in this example, the hierarchically structured name need not be the only name of the value set. Note also that the naming conventions just described actually imply some ambiguity. In the just given example, it is easy to conclude that a region value name consists of two parts: a county value name, and a municipality value name. However, this is in a way both true and not true. As described above, the municipality name would consist of two parts: a "family name", and a "first name". The family name of a municipality value is actually the same as the (complete) name of a county value. Thus the complete municipality value name consists of a complete county value name together with the first name of the municipality; the latter is only unique "within the family". If we call the complete name of a hierarchical value "the long name", and the first name part "the short name", we get:

region value name = county value name : short municipality value name;

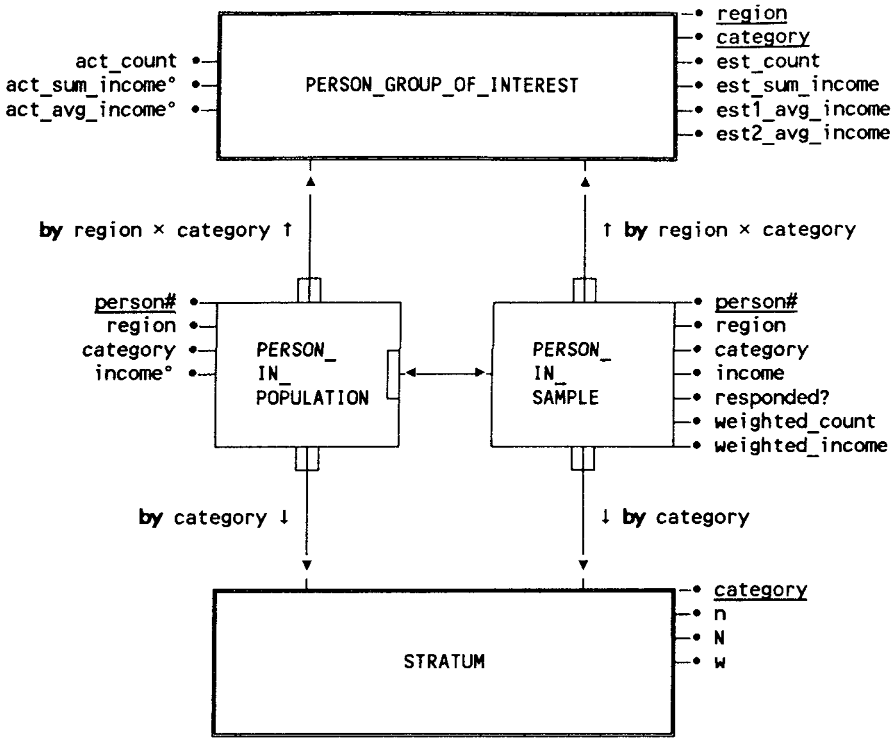
long municipality value name = county value name : short municipality name;

However, in practice the term "municipality value name" would often be used ambiguously to denote both the long and the short municipality name. To make it even more concrete: if "region code" consisted of four digits, where the first two digits would identify a county, and the last two would identify a municipality within a given county, the last two digits would probably often be referred to as "the municipality code", although it would only be the short name part of the complete municipality code. This ambiguity is a source of some confusion when discussing hierarchical variables and hierarchically structured value sets.

The discussion here about two-level hierarchical variables and value sets can easily be generalized to n levels.

2.6 Sampling and estimation

Beside aggregation, sampling is a typical process in many statistical information systems, as well as "the twin process" of estimation based on sampled information. How can we model the semantics of sampled statistical information and of the processes of sampling and estimation? Figure 4 illustrates one possible way of tackling these problems, using some of the extensions to ordinary OPR modelling that have been introduced in this paper.



```
Derivable object types:

PERSON_GROUP_OF_INTEREST = PERSON_IN_POPULATION(by region × category).agg;

PERSON_GROUP_OF_INTEREST = PERSON_IN_SAMPLE(by region × category).agg;

STRATUM = PERSON_IN_POPULATION(by category).agg;

STRATUM = PERSON_IN_SAMPLE(by category).agg;

Derivable variables for STRATUM:

n = PERSON_IN_SAMPLE(with responded="yes").count;

N = PERSON_IN_POPULATION.count;

w = N/n;

Derivable variables for PERSON IN SAMPLE:

weighted_count = STRATUM.w;

weighted_income = weighted_count * income;

Derivable actual variables for PERSON GROUP OF INTEREST:

act_count = PERSON_IN_POPULATION.count;

act_sum_income° = PERSON_IN_POPULATION.sum(income°);

act_avg_income° = PERSON_IN_POPULATION.avg(income°);

Derivable estimated variables for PERSON GROUP OF INTEREST:

est_count = PERSON_IN_SAMPLE.sum(weighted_count);

est_sum_income = PERSON_IN_SAMPLE.sum(weighted_income);

est1_avg = est_sum_income/est_count;

est2_avg = est_sum_income/act_count;
```

Figure 4. An object graph - with accompanying INFOL definitions - corresponding to a sample survey.

The example used in figure 4 is a hypothetical sample survey, where the population is a set of object instances belonging to the object type PERSON. We know the values of some variables for all the instances in the population: *person#*, *region*, and *category*. Population characteristics (parameters) that are functions of these variables can be estimated (computed) by evaluting the function over the object instances in the population. On the other hand *income* is a variable which is assumed to be relevant but not known for the object instances of the PERSON population. Instead it should be estimated after observing a sample of PERSON objects. The sample is supposed to be taken on the basis of random sampling from subsets of the population formed by stratification. Every object instance within a certain stratum has equal selection probability n/N, where n is the number of instances to be selected from the stratum, and N is the total number of instances in the stratum; n/N varies between strata.

The OPR-model for the sample survey contains two object types corresponding to the (generic) object type PERSON: PERSON_IN_POPULATION and PERSON_IN_SAMPLE; there is a partial one-to-one relation between the two object types. The two other object types in the model, STRATUM and PERSON_GROUP_OF_INTEREST, are formed by statistical aggregation of (any one of) the PERSON object types. The formal definitions, expressed in INFOL, can be found in the text under the object graph. The meaning of the object type STRATUM is obvious from the name. The object type PERSON_GROUP_OF_INTEREST is an object type, whose instances are **domains of interest** or **domains of study** in the sense of Marriott (1990), that is, subgroups of the population (including the population as a whole) which are of particular interest for the users of the statistical results derived from the survey.

Many of the variables for the object types are derivable from other variables; once again the definitions are stated in INFOL below the object graph. Variables for which data are not available (like *income* for PERSON_IN_POPULATION) are indicated by a small ring (°) after the variable name.

3 Syntactical properties of statistical information

3.1 Input-oriented structures of statistical data: questionnaires and forms

A questionnaire or form for eliciting statistical data from an informant (respondent) has often a hierarchical structure. The root node in the hierarchy corresponds to an object, which is either the respondent or an object that the respondent can inform about. The other non-leaf nodes in the hierarchy usually correspond to objects that are dependent upon the root-node object. The leaf nodes, finally, are the individual questions (or groups of questions) of the questionnaire, and they correspond to variables (or groups of variables) of the objects in the non-leaf nodes.

Like many other types of hierarchical structures, the structure of a questionnaire can be described by means of a **structure diagram**, containing the three structure elements known from structured programming: sequence, selection, and iteration (repetition); see for example Dahl, Dijkstra, and Hoare (1972) and Jackson (1975).

Figure 5 illustrates the structure diagram technique applied to an imaginary questionnaire concerning a person's educational background. The diagram should be read as follows. The data collected by the questionnaire consists of four major parts, corresponding to four major sets of questions: first the person is asked for some background information (name, date_of_birth, sex etc), then there are some questions about the basic education that every person is supposed to have, then those people who have undergone some vocational training are asked to supply some information about this, and finally those who have completed one or more programmes of higher education are asked to supply some details about these. Thus there are two mandatory and two optional parts of the questionnaire. A small ring or zero in the upper right part of a box indicates an optional part that will apply once, if it applies at all; an asterisk in the same position indicates an optional part that may be repeated. In this particular questionnaire, the respondent is assumed to have zero or one vocational trainings to inform about, where as the number of higher educations may be zero, one or more. If we look at the details of the main parts of the questionnaire, we can see, for example, that both a basic school education and a higher education programme are supposed to consist of a (variable) number of education component, where each component consists of a subject (course) and an optional judgement (score).

The questionnaire in this example, like many a real-world statistical questionnaire, has a rather complex structure. Nevertheless, it can relatively easily be mapped into the typical flat file structure of a relational database. (As a matter of fact, any state-of-the-art relational database management system would supply a forms-oriented user interface, where the user could define a hierarchical questionnaire like the one in the example, and have the data in it automatically mapped into a specified relational structure.)

One way of designing the mapping between a statistical questionnaire and a relational database is to proceed in the following three steps:

Step 1. Model the hierarchical structure of the questionnaire using the diagram technique in figure 5 or some equivalent formalism (for example the PASCAL-based methodology used in the Dutch system BLAISE; see Bethlehem et al (1987).

Step 2. Develop the OPR-model underlying the questionnaire, unless it already exists; it is always possible to find a conceptual OPR-model, which will make it possible to define the hierarchical questionnaire structure as a **view**, or **external schema**, in database terminology.

Step 3. Transform the OPR-model into a relational data model. This can always be done by applying some simple transformation rules; See Sundgren (1984) and Elmasri and Navathe (1989).

Figure 5 illustrated the result of step 1 for a simple example. Figures 6 and 7 illustrate the results of step 2 and 3 for the same example.

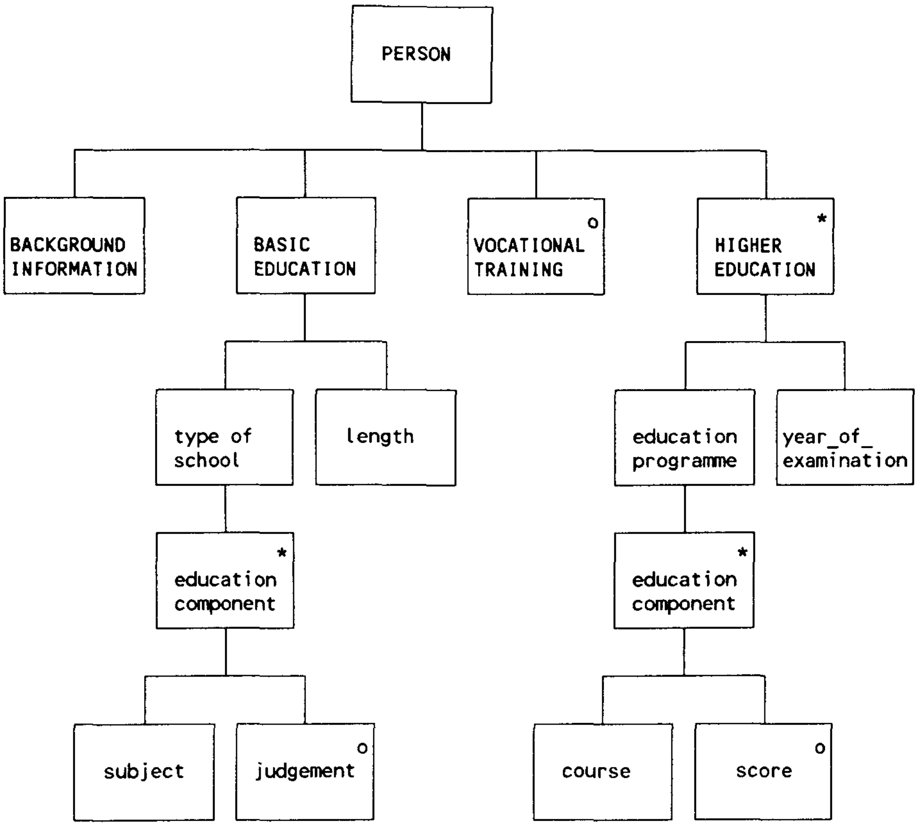


Figure 5. **Structure diagram illustrating the typical, hierarchical structure of a statistical questionnaire.** (The diagram technique follows Jackson (1975), but some conventions have been adopted, which make the diagram more compact.)

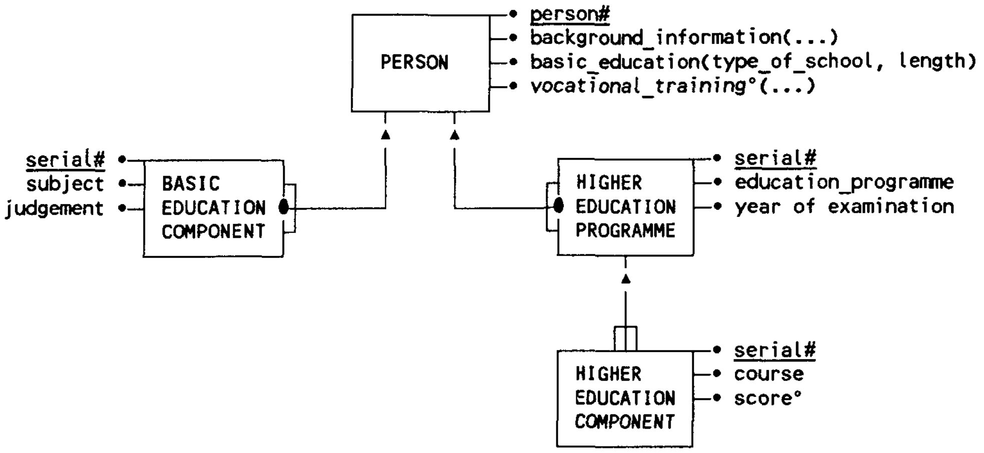

PERSON
person#
background_information(...)
basic_education(type_of_school, length)
vocational_training°(...)
serial#
subject
judgement
BASIC EDUCATION COMPONENT
HIGHER EDUCATION PROGRAMME
serial#
education_programme
year of examination
HIGHER EDUCATION COMPONENT
serial#
course
score°

Figure 6. An object graph showing the OPR-model underlying the hierarchical structure of the statistical questionnaire in figur 5.

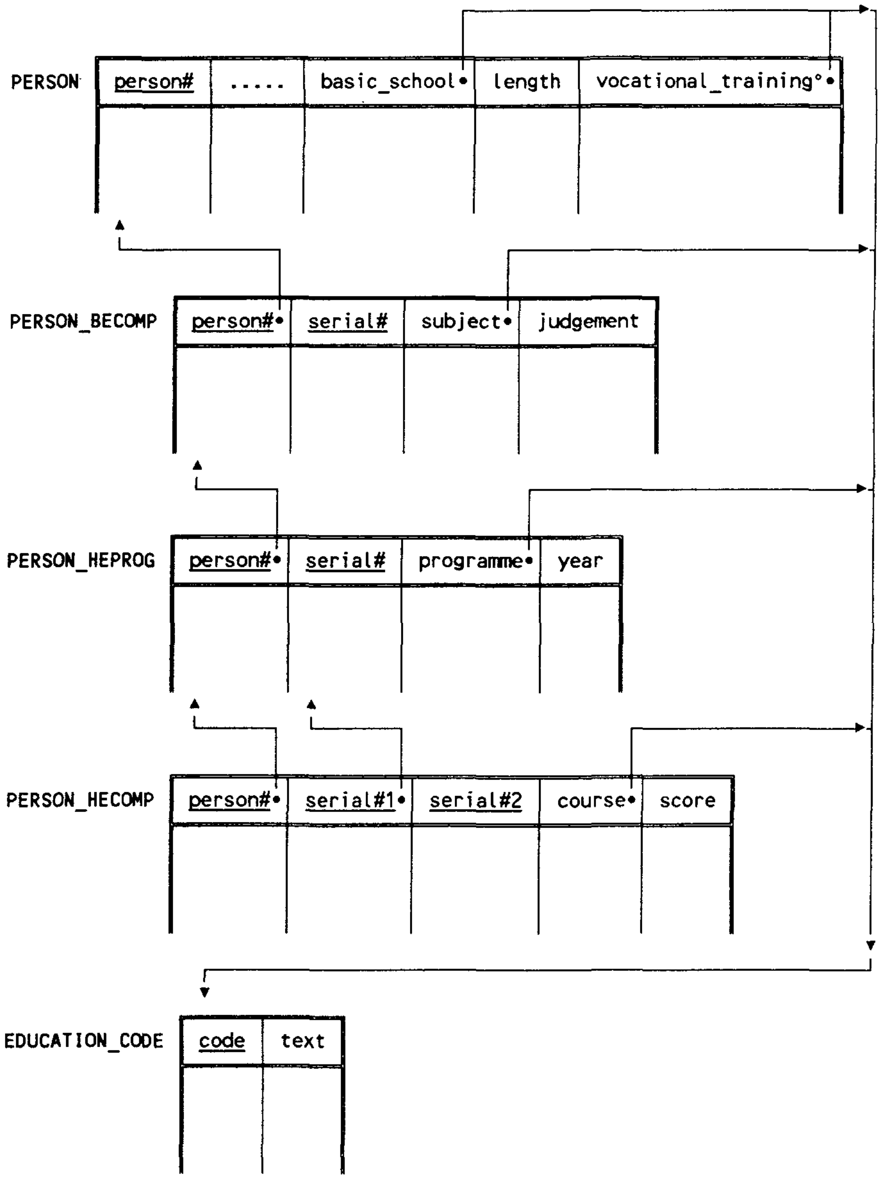


Figure 7. A relational data model corresponding to the conceptual OPR-model in figure 6 and the questionnaire in figure 5.

3.2 Output-oriented structures of statistical data: statistical tables

There are several techniques, known from the literature, for describing the semantical and syntactical structure of statistical tables; see for example Shoshani (1982) and Sato (1988). One problem is to which extent semantical and syntactical aspects of the table structure should be treated separately, and to which extent they should be mixed together. From a purely syntactical point of view, the most fundamental structure is the two dimensions of the paper or screen upon which the table is usually presented. On the other hand, from a purely semantical point of view, the basic structure is determined by (a) the principal logical parts of the definition of the information contents of a table, and (b) the components of each one of these logical parts.

In my opinion it is unnecessarily complex to describe and understand the syntactical structure of a statistical table, without having first a good semantical analysis of the table.

A statistical table is a structured representation of a structured set of statistical messages. Let us consider a cell in a typical statistical table. It usually contains a number, and it is further characterized by its place in the table structure, together with the textual information associated with this place (head and stub texts etc). The data and the associated metadata of a cell in a table represents a piece of statistical information that we can call a **statistical e-message.** Referring back to the general definition of an e-message in section ..... we will identify most statistical e-messages associated with table cells a special case of property type e-messages, consisting of an object part, a property part, and a time part.

The object part of a statistical e-message refers an object group, which is a population of objects of interest, or a subset of such a population, a so-called domain of interest, or domain of study. With the modelling perspective of section ....., the object group is a macro-object resulting from an abstraction by aggregation. Thus the macro-level object can be defined in terms of a micro-level object type and a property giving a restriction to a subtype:

<object type> **with** <property>;

Example:

PERSON **with** sex="female" **and** age<20;

The property part of a statistical e-message typically refers to a <variable, value>-pair, where the variable is a so-called parameter or characteristic of the domain of interest referred to by the object part of the message, and where the value is the value referred to by the number in the table cell. Example:

estimated_average_income = 15000;

The time part of a statistical e-message can be a point of time (if the e-message informs about a state in the domain of interest) or a time interval (if the e-message informs about a change in the domain of interest. The time part is often the same for all the statistical e-messages in a table, and then it can be mentioned in the table head only. However, if the table contains statistical information that is organized as one or more time series, common time parts for subcollections of e-messages (cells) may appear in column heads or in the stub.

Thus an example of a complete statistical e-message could be:

<PERSON **with** sex="f" **and** age<20, est_avg_income=15000, year=1990>;

In the syntax of INFOL this could be expressed as:

{PERSON(**with** sex="f" **and** age<20)}.est_avg_income(year=1990)=15000;

In the introduction to this section we defined a statistical table as a structured representation of a structured set of statistical messages. Many tables appearing in statistical presentations have a very regular structure, at least from a semantical point of view. In fact it could be argued that even tables, which are not quite so regular, could and should be thought of as being built up from components that are regular. The regular structure that we are now going to discuss has been referred to in the literature as the **box structure** or **matrix format** of statistical tables.

A **box** is an n-dimensional structure defined or "spanned" by n variables, called γ-variables. The Cartesian product of the value sets of the n γ-variables defines the **cells** of the box. Each cell contains a vector of m values of m variables, called ß-variables. The ß-variables are the same for all cells in a box, and they are supposed to be statistics or estimated parameters for domains of interest corresponding to the respective cells. The domain of interest associated with a particular cell is defined by (a) the population property, sometimes called the α-property, which is the same for all the domains of interest corresponding to all the cells in the box, and (b) the property distinguishing the particular domain of interest, associated with the particular cell, from the domains of interest of the other cells in the box; the latter property is called the γ-property, since it is defined by a logical **and**-combination of n <γ-variable, value>-pairs:

$$P_\gamma = (\gamma_1 = v_1) \textbf{ and } (\gamma_2 = v_2) \textbf{ and } \ldots \textbf{ and } (\gamma_n = v_n)$$

For example, the information defined by the INFOL-expression

{PERSON(**with** nationality="foreign")(**by** sex × region)}.est_avg_income(1990)

can be organized as a two-dimensional box spanned by the γ-variables "sex" and "region". Each cell would contain a value of the ß-variable "est_avg_income(1990)" for a domain of interest defined by the α-property "PERSON(**with** nationality= "foreign")" **and** a γ-property defined by a certain <sex value, region value> combination.

In terms of the relational data model, a box can always be represented by a relational table, having a column combination corresponding to the n γ-variables as its primary key, and columns corresponding to the m ß-variables as additional columns. In the example above, we would get the following relational table, if we assume that "region" has three values: A, B, and C;

FOREIGNERS =

SEX	REGION	EST_AVG_INCOME(1990)
female	A	.....
female	B	.....
female	C	.....
male	A	.....
male	B	.....
male	C	.....

This mapping of a box structure to a relational table could be regarded as a **normal representation** of statistical tables. Naturally, in many situations one like a representation for presentation purposes, which looks quite different from this normal representation, but with contemporary computer technology and software tools, it should not be difficult for a user to transform a table, stored according to the normal representation, into his or her preferred format in a particular usage situation. On the other hand, for facilitating automatic exchange of statistical data and metadata, it would be extremely valuable if a standard representation format could be agreed upon. Work in this direction is presently going on within international organizations; see ......

Of course the example used above for illustration purposes is extremely simplified. There are many qualifications that need to be added. Consider the following list of problem areas:

- sums on different levels;
- hierarchical variables;
- sparse tables;
- null values;
- time;

Derivable data like sums can be represented explicitly or implicitly. Implicit representation means that we indicate in our metadata description of a table that we want the sums to be computed, whenever the table is presented to the user. (Naturally different users, and even the same user, can have different definitions of the same table for different purposes.) Explicit representation of derivable data implies redundance but may speed up retrieval. Generally speaking, redundance means additional storage costs and updating problems; however the updating problems need not be too severe in statistical databases, since they are often quite statical by nature, and updated incrementally only.

In a specification and query language like INFOL, we may indicate by a suitable symbolism, which total and partial sums that we would like to specify. Example:

{FOREIGNER(**by** sex(*, Σ(*)) × region(A, B, Σ(A, B), C, Σ(*))}.
est_avg_income(1990)

First of all, we have here introduced the name "FOREIGNER" to mean the same as "PERSON(**with** nationality="foreign")". Then we have indicated, variable by variable, which values and (total and partial) sums that we want to be available for a particular usage of aggregated data. For example, an asterisk (*) indicates the selection of all values in the value set for a particular variable, and Σ(...) indicates that the ß-variables should be aggregated over the listed values of the γ-variable (according to the proper formula for each variable, depending on whether it is a total, an average, a percentage, or whatever).

The normal form representation of this selection would be:

FOREIGNERS =

SEX	REGION	EST_AVG_INCOME(1990)
female	A	.....
female	B	.....
female	A, B	.....
female	C	.....
female	*	.....
male	A	.....
male	B	.....
male	A, B	.....
male	C	.....
male	*	.....
*	A	.....
*	B	.....
*	A, B	.....
*	C	.....
*	*	.....

As can be seen from this example, the introduction of a summary level for a γ-variable is equivalent to extending the value set with one element. With one summary level introduced for "sex" and two for "region", the cardinalities of the value sets will grow from 2 to 3, and from 3 to 5, respectively, and the number of rows in the relational table will grow from 2 × 3 = 6 to (2 + 1) × (3 + 2) = 15, that is, by 150% in this case.

Null values can be treated in much the same way as summary values.

One or more of the γ -variables can be **hierarchical variable.** A hierarchical variable of k levels will be represented by k columns in the normal format relational table. For each level in the hierarchy, one or more summary values may be introduced. However, the summary values for different levels in the hierarchy cannot be chosen as independently of each other as they can for the variables in a crossclassification.

As an example, we may assume that "region" in the example above is a two level hierarchy "county:municipality", where county A consists of the municipalities A1 and A2, county B consists of the only municipality B1, and county C consists of the municipalities C1, C2, and C3. The information specified by the INFOL expression

{FOREIGNER(**by** sex(*) × county(*, Σ(*)) : municipality(*, Σ(*))}. est_avg_income(1990)

would have the normal form representation (only the first part of it is shown):

FOREIGNER =

SEX	COUNTY	MPLTY	EST_AVG_INCOME (1990)
fem	A	1	.....
fem	A	2	.....
fem	A	*	.....
fem	B	1	.....
fem	B	*	.....
fem	C	1	.....
fem	C	2	.....
fem	C	3	.....
fem	C	*	.....
fem	*	*	.....

So far we have assumed without discussion that the time component of the statistical table should be associated with the ß-variable(s) and the column(s) representing the ß-variables. However, there are several alternatives, corresponding to slightly different semantical interpretations of the data, and with different performance characteristics, if implemented in a relational database.

If the table is a typical snapshot representation of the object system, there is only one time involved, and this time could be indicated in the metadata accompanying the table as a whole. In the example above, the relational table could for example be named "FOREIGNER(1990)".

If there are repeated snapshots, there could be uniform tables for different times, with names containing a time parameter. Example: "FOREIGNER(t)", where t = 1980, 1985, 1990. Another alternative is to put time as a parameter in the names of the ß-variables. Example: "est_avg_income(t)", where t = 1980, 1985, 1990.

A third alternative is to regard time as a γ-variable, which is crossclassified with the other γ-variables. This implies a slight change in our conceptualization of the population of interest. If the "snapshot version" of our population of interest is assumed to contain objects of a certain type OBJ, a corresponding population with extension in time would constist of <OBJ, time> pairs, and interest groups would be formed by crossclassifying this population by means of the Cartesian product of the original γ-variables with an additional γ-variable that is based on time.

Example:

{<PERSON, TIME>
(**with** PERSON.nationality="foreign" **and** TIME.year = 1985 **or** 1990)
(**by** PERSON.sex × PERSON.region × TIME.year)}. est_avg_income;

The normal relational representation of this conceptualization would be:

FOREIGNERS =

SEX	REGION	TIME	EST_AVG_INCOME
female	A	1985	.....
female	A	1990	.....
female	B	1985	.....
female	B	1990	.....
female	C	1985	.....
female	C	1990	.....
male	A	1985	.....
male	A	1990	.....
male	B	1985	.....
male	B	1990	.....
male	C	1985	.....
male	C	1990	.....

Summary values can be defined as in previous examples. However, it should be noted that the meaning of summary values formed over the time γ-variable may not be obvious. In fact it would very often not be meaningful at all.

Yet another modelling of time will be necessary for **event-based statistical information.** In contrast to the snapshot-based statistical information that typically emanates from a statistical survey of traditional type, event-based statistical information often comes from other than statistical sources, for example administrative registers and other administrative information systems. Such systems are more or less directly updated, when events of certain types occur in the object system. The flow of such events (and consequent updates) is more or less continuous, and the updating transactions must be **time-stamped.** As a matter of fact the events will be a basic object type, forming at least one of the populations of interest, in the conceptual model for this type of statistical information. The time of the event will be a variable of the event object, and if the value set of the time variable is properly classified (grouped), it can serve as a γ-variable very much like other γ-variables in the aggregation and tabular presentation of statistical information. For this type of time γ-variable, the formation of summary values is usually meaningful and often useful, since it is meaningful to count events over different periods of time, and to summarize other variables for these events.

Apart from events, **processes** is another type of object, which sometimes occur in event-based statistical systems. A process is characterized in the time dimension by a starting-point (associated with a process birth event) and a completion-point (associated with a process death event). Processes can be treated similarly as events in the aggregation of statistical information.

So far we have discussed the semantical structure of statistical macroinformation, and how it can be represented by relational tables in a kind of canonical form for statistical macrodata. However, we still have to discuss desirable presentation structures for statistical macrodata, as well as operators needed to transform the statistical macrodata from the normal representation form to other desirable formats. In this paper I shall only give a few hints about these topics.

The most straightforward presentation of a box structure of statistical data that is stored in its normal relational form is a listing of the relational table, row by row. Such a presentation would not be satisfactory in many situations, even if one made some cosmetical improvements, such as suppressing γ-variable values whenever they are identical with the corresponding values in the previous row. A transformation of a slightly more complicated nature, which is often desirable, is to move one or more of the γ-variables from the stub of the table to the column heads. Example:

		Estimated average income 1990		
County	Municipality	Men	Women	Both sexes
A		.....	.....	.....
	1	.....	.....	.....
	2	.....	.....	.....
B		.....	.....	.....
	1	.....	.....	.....
C		.....	.....	.....
	1	.....	.....	.....
	2	.....	.....	.....
	3	.....	.....	.....
All counties		.....	.....	.....

The moving of the γ-variable "sex" from the stub to the column headings in this example can be done by operations in an extended relational algebra. However, it is a relatively complex operation, and it implies a non-uniform handling of the value names for γ-variables in the stub and γ-variables in the column headings; the former are normal data in the relational table, whereas the latter must be part of the column names.

A much more attractive solution to this problem (and similar ones) is to define a **box algebra**, that is, an algebra the operators of which transform boxes into boxes of another structure. Such algebras have been proposed and implemented; see for example Nilsson (1984).

Another type of problem arises when the user wants to have the aggregated statistical data presented in a non-regular form, that is, a form which is not compatible with the box structure as such. Usually, however, such non-regular presentation structures can be constructed from a small number of regular boxes. A relatively common situation is when the user wants to present in the same table the contents of two boxes that have all γ-variables except one (more general: k) in common. Example:

		Estimated average income 1990				
		By sex		By marital status		
County	Munici-pality	Male	Female	Never married	Married now	Married before
A		.....	.....	.....	.....	.....
	1	.....	.....	.....	.....	.....
	2	.....	.....	.....	.....	.....
B		.....	.....	.....	.....	.....
	1	.....	.....	.....	.....	.....
C		.....	.....	.....	.....	.....
	1	.....	.....	.....	.....	.....
	2	.....	.....	.....	.....	.....
	3	.....	.....	.....	.....	.....
All		.....	.....	.....	.....	.....

4 Appendix: A formal specification of the INFOL language

4.1 The metalanguage of the formal specification

[...] The brackets embrace something that may be omitted.

[...]* The construction within the brackets may be repeated a variable number of times (zero, one or more), with a comma between the repetitions; in other words the whole construction is a list (an ordered set) of constructions of the same kind.

<...> Indicates that at this place in a construction there should be an element of the type mentioned within the broken brackets.

... <--- ... The structure of an INFOL definition. The construction to the left of the arrow is what is defined by the definition. The construction to the right of the arrow is the construction which defines the construction to the left.

... ::= ...; The structure of a metalanguage definition.

| Denotes logical "or" in metalanguage definitions, that is, indicates alternative construction possibilities.

obj	object
prop	property
rel	relation
var	variable
val	value
ref	reference
relop	relational operator (<, =, >, etc)

4.2 INFOL constructions

```
(1)   <obj type ref> ::=   <obj type name> |
                           <obj type ref> [with <prop ref>] |
                           <obj type ref> [by <var ref>].agg;

(2)   <prop ref> ::=       <prop name> |
                           <var ref> <relop> <var ref> |
                           not <prop ref> |
                           <prop ref> and <prop ref> |
                           <prop ref> or <prop ref>;

(3)   <var ref> ::=        <var name> |
                           <path ref>.<var ref> |
                               <function>([<var ref>]*) |
                               <quantifier> <var ref>;

(4)   <val set ref> ::=        <val set name> |
                               <function>([<val set ref>]*);

(5)   <val ref> ::=            <val name> |
                               <val set ref>.<val name>;

(6)   <function> ::=           <arithmetical function> |
                               <string function> |
                               <set function> |
                               <logical function> |
                               <aggregation function>;

(7)   <path ref> ::=           [<rel ref>.[<quantifier>] <obj type ref>]*;

(8)   <quantifier> ::=         some | all;

(9)   <rel ref> ::=            <rel name> |
                               [<path ref>]*;

(10)  <definition> ::=         object type <obj type name> [with variables]
                                     [.[<var name>[(<val set name>)]]*]
                                     [<--- <obj type ref>][.[<var ref>]*]] |
                               property <prop name> [<--- <prop ref>] |
                               variable [<obj type ref>.]
                                     <var name>[(<val set name>)]
                                     [<--- [<var ref>[, if <prop ref>]]*] |
                               value set <val set name> with values
                                     [<val name>]* |
                               value set <val set name>(<val set name>)
                                     <--- [<val name>, if [<val name>]*]* |
                               value [<val set name>.]<val name>
                                     [<--- <val ref>] |
                               relation <rel name> between objects
                                     [<obj type name>(<cardinality>)
                                     [alias <role name>]]*
                                     [<--- <rel ref>];

(11)  <query> ::=              <obj type ref>.[<var ref>]*;

(12)  <view> ::=               <obj type ref>.[<var ref>]* |
                               [object type <obj type name> [with variables]
                                     .[<var name>]*]*,
                               [relation <rel name>]*;
```

4.3 Informal comments, qualifications, and explanations

First a general comment. The first object type referred to in an INFOL expressions will be the first **current object type** when the expression is processed (by a human being or otherwize). The current object type will be changed as a result of certain constructions that appear in INFOL expression. For example, a path reference will "move" the current object type from one object type to another, via still other object types, which are "nodes" on the path. The properties, variables, and relations, referred to at a certain place in an INFOL expression, must be *compatible with the current object type* at that place. For example, if a variable is referred to, it must be relevant for the current object type at the place of the variable reference. Now will follow a numbered sequence of further comments related to the formal definitions with the corresponding numbers in the previous section:

(1) A practical convention is to let the object aggregation operator **agg** be implied, if it is followed by a variable reference that starts with an aggregation function like **count**, **sum**, or **avg**. Another possibility, which has been illustrated in the paper, is to use set brackets {...} around the classification expression, instead of writing **agg** after it.

(2) Note that <var ref> includes a reference to a constant value as a special case; a constant value can be regardes as a special case of a function.

(6) The arithmetical functions are the usual ones. String functions include concatenation and substring operations. Set functions include Cartesian product (×) and hierarchical combination (:). Aggregation functions include functions that operate on the values of zero, one, or more variables for a *set of object instances* and produce a single value as an outcome. In addition to the functions mentioned above, and a other statistical operators like those which compute variance and correlation, there are functions like **max** and **min.**

(7) Intuitively speaking, a path connects two object type nodes in an object graph. The connection consists of a chain of segments, where the start node and the end node of every segment are object types that are *directly related* in the object graph. If there is only one direct relation between two nodes, a practical convention may be to omit the relation reference in the path description. Alternatively, it is always possible to omit a reference to an object type at the end of a segment corresponding to a binary relation, since the object type is uniquely determined anyhow.

(8) The quantifiers **some** and **all** correspond to the *existence quantifier* (∃) and the *universal quantifier* (∀) known from predicate logic.

(10) The definition of an INFOL entity consists in general of two parts. The first part is a **declaration**, which (a) declares the new entity to belong to a specified category (for example "object type"), and (b) gives a name to the entity. The second part of the definition is separated from the first one by an arrow (<---), and it contains a **derivation expression**, that is, an expression for deriving the new entity from existing ones. Of course, the second part appears only if the new entity is derivable. If the first part of the definition is missing, it is an **implicit definition**; the category of an implicitly defined entity is implied by the context, and the name, if needed, will have to be automatically generated.

(11) This seemingly simple expression actually covers the whole query language of INFOL, including so-called $\alpha\beta$-queries and $\alpha\beta\gamma$-queries (see Sundgren

(1973)), and it can be regarded as a conceptual level counterpart to database query languages like SQL.

(12) The first type of view definition creates a **single-object view**, that is, a view which arranges all information around one single object. The other type of view definition creates a **multi-object view**, containing several object types and relations. The single-object view is often useful as a basis for statistical tabulations.

5 References

Bethlehem J. G., Denteneer D., Hundepool A. J., and Keller W. J. (1987) *The BLAISE System for Computer-Assisted Survey Processing*, The Hague: Central Bureau of Statistics of the Netherlands.

Chen P. (1976) The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1:1.

Dahl O.-J., Dijkstra E. W. and Hoare C. A. R. (1972) *Structured Programming*, London: Academic Press.

Durchholz R. and Richter G. (1974) Concepts for Data Base Management. In Klimbie J. W. and Koffeman K. L. (eds) *Data Base Management*, Amsterdam: North-Holland.

Elmasri R. and Navathe S. B. (1989) *Fundamentals of Database Systems*, Redwood City: Benjamin/Cummings Publishing Company.

Jackson M. A. (1975) *Principles of Program Design*, London: Academic Press.

Langefors B. (1966) *Theoretical Analysis of Information Systems*, Lund: Studentlitteratur.

Lindgreen P. (1974) Basic operations on information as a basis for data base design. In Rosenfeld J. L. (editor) *Information Processing 74*, Proceedings of IFIP Congress 74, Amsterdam: North-Holland.

Malmborg E. (1982) *The OPREM-approach - An extension of an OPR-approach to include dynamics and classification*, Stockholm: Statistics Sweden.

Marriott F. H. C. (1990) *A Dictionary of Statistical Terms*, 5th ed, Longman Scientific & Technical.

Nilsson G. (1984) *TBE - Table By Example*, Örebro: Statistics Sweden.

Sato H. (1988) A Data Model, Knowledge Base, and Natural Language Processing for Sharing a Large Statistical Database. In *Proceedings of the 4th International Workshop on Statistical and Scientific Database Management*, Rome.

Shoshani A. (1982) Statistical Databases: Characteristics, Problems, and Some Solutions. In *Proceedings of the 8th International Conference on Very Large Data Bases*, Mexico City.

Smith J. M. and Smith D. C. P. (1977) Database Abstractions: Aggregation and Generalization. *ACM Transactions on Database Systems*, 2:2.

Stamper R. (1973) *Information in Business and Administrative Systems*, London: B. T. Batsford.

Sundgren B. (1973) *An Infological Approach to Data Bases*, Stockholm: Statistics Sweden.

Sundgren B. (1974) Conceptual Foundation of the Infological Approach to Data Bases. In Klimbie J. W. and Koffeman K. L. (eds) *Data Base Management*, Amsterdam: North-Holland.

Sundgren B. (1984) *Conceptual Design of Data Bases and Information Systems*, Stockholm: Statistics Sweden.

Sundgren B. (1989) Conceptual Modelling as an Instrument for Formal Specification of Statistical Information Systems. In *Proceedings of the 47th session of the International Statistical Institute*. Paris: INSEE.

R & D Reports är en för U/ADB och U/STM gemensam publikationsserie, som fr o m 1988-01-01 ersätter de tidigare "gula" och "gröna" serierna. I serien ingår även **Abstracts** (sammanfattning av metodrapporter från SCB).

R & D Reports Statistics Sweden are published by the Department of Research & Development within Statistics Sweden. Reports dealing with statistical methods have green (grön) covers. Reports dealing with EDP methods have yellow (gul) covers. In addition, abstracts are published three times a year (light brown/beige covers).

Reports published during 1992:

1992:1 (grön) Industrins konkurrenskraft och produktivitet i fokus - en utvärdering av statistiken **(Margareta Ringquist)**

1992:2 (grön) Automated Coding of Survey Responses: An International Review **(Lars Lyberg and Pat Dean)**

1992:3 (grön) TABELLER ,... TABELLER ,... TABELLER ,... - Variation och Förnyelse **(Per Nilsson)**

1992:4 (grön) Basurval vid SCB? Studier av reskostnadseffekter vid övergång till basurval **(Elisabet Berglund)**

1992:5 (beige) Abstracts I - sammanfattning av metodrapporter från SCB

1992:6 (grön) Utvärdering av framskrivningsförfarande för UVAV-statistik **(Kerstin Forssén & Bengt Rosén)**

1992:7 (grön) Cross-Classified Sampling for the Consumer Price Index **(Esbjörn Ohlsson)**

1992:8 (grön) Bortfallsbarometern nr 7 **(Mats Bergdahl, Pär Brundell, Anders Lindberg, Håkan Lindén, Peter Lundquist, Monica Rennermalm)**

1992:9 (beige) Abstracts II - sammanfattning av metodrapporter från SCB

1992:10 (gul) Organizing the Metainformation Systems of a Statistical Office **(Bo Sundgren)**

1992:11 (grön) CLAN - ett SAS-program för skattningar av medelfel **(Claes Andersson, Lennart Nordberg)**

1992:12 (grön) KVALITETSRAPPORTEN - Utveckling av kvaliteten för SCBs statistikproduktion **(Jan Eklöf, Per Nilsson)**

1992:13 (grön) The Use of Registers as Auxiliary Information in the Swedish Labour Force Survey **(Jan Hörngren)**

1992:14 (beige) Abstracts III - sammanfattning av metodrapporter från SCB

1992:15 (grön) Operationalising a Hedonic Index in an Official Price Index Program: personal computers in the Swedish import price index **(Jörgen Dalén)**

Kvarvarande **beige** och **gröna** exemplar av ovanstående promemorior kan rekvireras från Inga-Lill Pettersson, U/LEDN, SCB, 115 81 STOCKHOLM, eller per telefon 08-783 49 56.

Kvarvarande **gula** exemplar kan rekvireras från Ingvar Andersson, U/LEDN, SCB, 115 81 STOCKHOLM, eller per telefon 08-783 41 47.