WP. 11
ENGLISH ONLY

**UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS**

**EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES (EUROSTAT)**

**ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE**

Joint UNECE/Eurostat/OECD work session on statistical metadata (METIS)
(Luxembourg, 9-11 April 2008)

Topic 2 (ii) Metadata concepts, standards, models and registries

GENERAL REQUIREMENTS FOR THE SOUNDNESS OF METADATA MODELS

Submitted by Statistics Netherlands, The Netherlands[1]

I. INTRODUCTION

1. When is an information model for metadata a good model? What are the general requirements we attribute to metadata models that seem fit for purpose? Is there a way to distinguish the best among a list of models that all apply to a common domain?

2. These questions become particularly relevant when information systems are built that are based on these models. The development of an information system is generally costly, so choosing the 'best' information model in advance makes sense. For obvious reasons, one criterion concerning information models is often dominant during system development. In short, it proclaims that a model should do what it is supposed to do:

(Completeness) *An information model for metadata should allow all instances that are inside the application domain.*

3. For instance, when designing a model upon which an information system for the storage of classification systems is based, this criterion requires that *all* classification systems, even the obscure, must be taken into consideration. This calls for a strategy in which as many 'real world' instances as possible are related to the model, adjusting the model if necessary. Usually one ends up with a model that is more general (i.e., allows more instances) than the one started with.

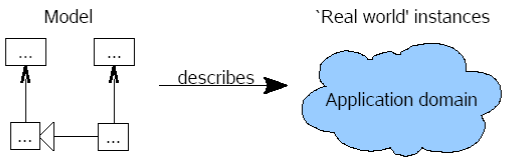

Figure 1: A model and its application domain

[1] Prepared by Tjalling Gelsema (tgsa@cbs.nl).

4. There is an evident drawback in a relentless application of the completeness criterion. Taking it to the extreme, one could end up with the model below (or a similar one) that could be called the *most general metadata model*: every conceivable metadata item and every conceivable association between metadata items is captured in it. Essentially, since it describes the way in which a generalized directed graph is composed, it subsumes every other model. It is also the least expressive, for it lacks the mechanisms (that are usually typical for the application domain) that are needed to guarantee the validity of metadata items. For instance, we could attempt to describe the structure of a classification system with it, treating categories as metadata items and category refinements as associations. Then, just the same, the model would accept the refinement of one category with respect to another, as well as its converse.

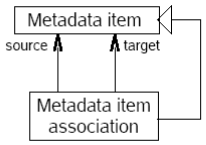

Figure 2: General but useless model for metadata

5. So, it seems that we need other criteria that can counteract the creation of a *most general metadata model*. A sensible goal to pursue is to have a model only just describe the application domain – nothing more (and nothing less).

(No junk) *An information model for metadata should not allow instances that are outside the application domain.*

6. Note that this calls for a different strategy: in order to verify that the 'no junk' criterion is met, every model instance must have a plausible counterpart in the 'real world' application domain. If a model instance is found for which this does not hold, then the model must be *restricted*, as opposed to generalized. For instance, in a first attempt to restrict the general model of Fig.2 above in order to suit the refinement structure of categories in a classification system, one could, cf. [2], add an antisymmetry condition on metadata item associations.[2]

7. Many information systems for statistical metadata lack the mechanisms needed to identify metadata items that are essentially the same. This may lead to inefficiency in storage and to incoherent use of those systems. In some cases, this seems unavoidable. However, we claim that more often than not information models can be fed with the mechanisms – usually rules of logical inference – that are capable of detecting metadata items that are synonyms of one another.

(No confusion) *An information model for metadata should not allow an instance to have more than one representation.*

8. It must be stressed that, in general, these criteria are hard to meet. They may even be nonsensical in some situations. For instance, it does not make sense to demand that a textual definition of some phenomenon of statistical interest is essentially laid down in a single, unique way. We do not advocate such an overly eager obedience of the above criteria. Rather, we seek an application in those situations where confusion or junk is clearly avoidable in an information model.

9. In this paper we will investigate the relevance for statistical metadata of the 'no junk, no confusion' doctrine. In Section II, the 'no confusion' criterion is developed in an example, leading to a new viewpoint of classification systems. In Section III, we do the same for the 'no junk' criterion, which is applied to the construction of data sets. The paper ends with the discussion in Section IV, where the doctrine is placed in its original context (viz. of formal semantics) followed by some concluding remarks. Here the importance for statistical metadata of this style of semantics is stressed.

II. No confusion: Classification systems

[2] An association A between elements a, b is antisymmetric, if $(a,b) \in A$ and $(b,a) \in A$ implies $a = b$.

10. Consider the following example in which the responsibilities for the maintenance and the coordination of classification systems in a statistical office are highlighted. Classification systems are shared among many statisticians. Each statistician can request modifications to a classification that suit his needs. The coordinator's job is to see to it that modifications are reflected in a classification system in such a way that integrity of the system is preserved and that categories that are synonyms are identified as much as possible. Only requests for the addition (and not the removal) of categories are honoured, in order to maintain backward compatibility.

11. Suppose that one of these systems consists of a classification of leisure activities. At the lowest level, there are three: *Reading*, *Biking*, and *Shopping*. Two of them, *Biking* and *Shopping*, are classified as *Outdoor* activities.[3]

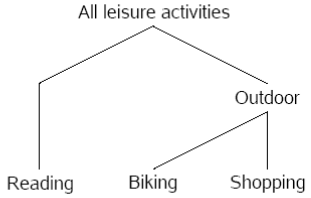

Figure 3: A classification of leisure activities

12. Now what information model is best used for maintenance and coordination? Though at this point the classification exhibits a tree-like structure, in order to prepare for additions, a model that is capable of capturing more complex structures is favoured. Let us take a model that treats a classification system as a set of refinement associations between categories, as pictured below. We assume that refinements are subject to the antisymmetry condition from the Introduction (though we might as well leave it out). The classification of leisure activities is captured in the model by taking all parent-child relationships and storing them as refinements. We agree that the orientation of a parent-child relationship is such that the source of a refinement always corresponds to the parent (though this requirement also is of no further importance).

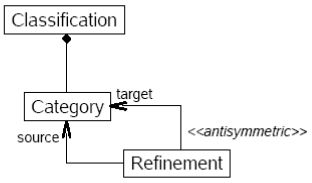

Figure 4: A simple information model for classification structures

13. So, at this point the classification of leisure activities is recorded as a set of four refinements, one for each line in Fig.3, across five categories.

14. Suppose now that a request is submitted for the addition of the category *Economical*, which comprises those activities that do not require any money. Suppose also that another statistician is in need for a category *Relaxing*, for those activities that do not require any effort. The coordinator could for instance position them in the classification structure in the way depicted below.

[3] For illustrative purposes, the classification is kept small. Any realistic classification of leisure activities would of course contain many more categories.

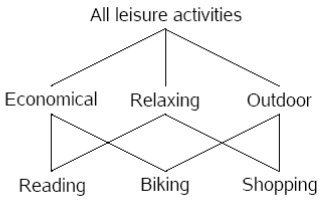

Figure 5: After the addition of two activity classes

15. This change is reflected in the information model by the addition of six refinements and (optionally) the deletion of one, viz. the parent-child relationship between the root category *All leisure activities* and the *Reading* category.[4] To summarize these changes, *Biking* and *Reading* are classified as *Economical* activities (as opposed to *Shopping* and assuming bikes and books need no hiring) and *Reading* and *Shopping* are *Relaxing* (as opposed to *Biking*).

16. The classification of leisure activities now enters a state of 'completion' in a way that will become clear in a minute. There are at least two situations in which the coordinator can be asked to add a category to the classification:[5] as a 'basic' activity (i.e., situated in the lowest level) or as a 'composite' activity (not situated in the lowest level). Suppose first the coordinator receives the request for the addition of the composite category *Group*, of leisure activities that are commonly carried out in groups. In order to position the new category in the system, the coordinator decides that *Biking* and *Shopping* are group activities, and that *Reading* is not. However, this means that with respect to the state depicted in Fig.5, the categories *Group* and *Outdoor* must be synonymous, as they comprise the same basic activities. Moreover, it is clear that *any* new composite category must be synonymous with a category that is already a member of the classification. For instance, the category of *Indoor* activities must be treated as a synonym for *Reading*, as the latter is the only basic activity that is not carried out outdoors.

17. Next suppose that the coordinator is asked to add the basic activity of *Piano playing.* This request however, the coordinator must deny for the following reason. Among other objections, adding this category would mean that the notion of total (i.e., *All leisure activities*) is changed in a way conflicting backward compatibility. The fact is, any new use of this root category would then entail a different set of basic activities and thus be incompatible with any prior use.

18. As far as the information model of Fig.4 is concerned, any addition of categories would be welcomed just the same. It should be clear for instance that the *Group* category is treated like any other and is recorded in the information model without the semantic connection it has with the *Outdoor* category. This means that the information model lacks the mechanisms needed to identify synonyms like the ones above. Briefly: it allows confusion.

19. So what are these mechanisms exactly? The arguments of the coordinator in the above example are based upon logical assumptions about the way classification systems work. First, a 'composite' category is identified with the set of 'basic' categories it comprises: any two categories that yield the same set of 'basic' categories must be synonymous. Second, the coordinator uses a notion of complement: he is able to identify *Indoor* and *Reading* because of his knowledge that indoor activities are exactly those activities not carried out outdoors. These assumptions correspond exactly to structures that are well-known in logic, viz. of Boolean algebras [4]. Since this paper is not meant to contain a formal exposition of classification systems as Boolean algebras (but see [6]), we will suffice with an intuitive comparison. The only element that is missing in the system of Fig.5 from the Boolean algebra point of view is the 'lowest' element 0. This may be seen as a harmless 'empty' category in the sense that it answers to, e.g., "which *Shopping* activity is also *Economical*?". In Fig.6 below, a Boolean algebra generated by 3 atoms – i.e., the elements 'just above' the empty category 0 – is shown.

---

[4] Though it is perfectly valid to leave it as it is.

[5] Another situation would be to split up a basic activity into two or more; e.g., when reading a newspaper is treated separately from reading a book. This is not further developed in the example however.

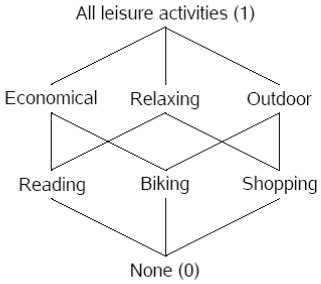

Figure 6: A Boolean algebra generated by three atoms

20. We show how the information model of Fig.4 can be adapted so that it accepts precisely those structures that are Boolean algebras. It is stressed that there are many ways to accomplish this; we just present one that is effective.

21. First, a *bitstring* attribute[6] is added to the *Category* class. This will encode the location of the category in the classification structure. The idea is to reserve a bit location in the bitstring for each atom (each 'basic' activity) in the classification. Thus the length of the bitstring corresponds to the number of atoms. Now the bit position in the string is set to 1 for each atom that is contained in a category. For instance, the *Reading*, *Biking*, and *Economical* categories are encoded 100, 010, and 110, respectively. Next, we derive the refinement association from the bitstring attributes: one category is a refinement of a second, only if the bitstring attribute of the second contains a 1 for every bit position that is set to 1 in the first. Third, we may add a list of synonyms to each category, to prepare for attempts to add a category whose bitstring encoding is already given away.

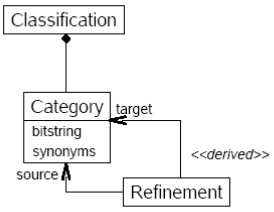

Figure 7: An alternative model for classification structures

22. There is also another way that Boolean algebras make sense as a mechanism underlying classification systems. It is a fact that the set of sub-populations of any given population (i.e., a set of entities) forms a Boolean algebra with union, intersection and set-complement [4]. Now any classification system that serves as a means to identify meaningful sub-populations in a 'reliable' way must satisfy the laws of a Boolean algebra.[7]

III. No junk: Microdata

23. Consider the example domain model below showing the object types of businesses, jobs, and persons. Essentially, the domain model expresses that the role of a business and a person with respect to a job is that of an employer and an employee, respectively. Suppose also that there is an interest in the variables *turnover*, *salary*, and *age*, as indicated in the figure.

[6] This is a string containing just 0's and 1's.

[7] Here 'reliable' means that if two categories correspond to two sub-populations, then their disjunction must correspond to the union of these sub-populations.

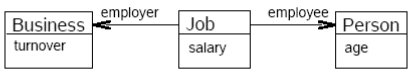

Figure 8: Businesses, jobs, and persons

24. In this section we will study the (logical) mechanisms that are used in the construction of datasets from variables. We start off with an information model that expresses that in order to describe a dataset containing microdata, one simply lists the variables that correspond with its columns. The model in Fig.9 below uses an aggregation (the diamond shaped arrow) between the *Microdataset* and *Variable* classes to express this.

25. As is usual for microdata, each row of the dataset corresponds to a single entity of a single object type (and conversely, each entity is represented by just one row). For this purpose, the *Microdataset* class contains the *object type* attribute. The fact that a variable holds the result of a measurement of a property of a single object type is indicated by the *object type* attribute of the *Variable* class. In the information model, we assume that there is no prior connection between both *object type* attributes. Of course, in a realistic setting both classes would contain many more attributes (such as a name that can be given to the dataset or the variable). We leave those out because they have no part in the argument.

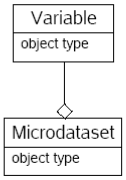

Figure 9: Information model of a dataset and its variables

26. Now according to the domain model in Fig.8, there are nine (more or less meaningful) datasets that can be composed from the variables that are shown in it. For completeness sake, we list them all in the table below.

Dataset no.	Object type of dataset	List of variables
1	Person	age
2	Business	turnover
3	Job	salary
4	Job	salary, age of employee
5	Job	salary, turnover of employer
6	Job	salary, age of employee, turnover of employer
7	Job	age of employee
8	Job	turnover of employer
9	Job	age of employee, turnover of employer

27. For instance, a dataset in which the rows represent a population of jobs and the columns represent the job's salary and the employee's age respectively, is listed fourth. However, according to the information model of Fig.9 many more combinations are possible, all of which are meaningless. For instance, the information model would not object to the combination below. That combination however does not make sense, since a person can have more than one employer. Put differently, a dataset that contains both the variables *age* and *turnover* is necessarily a dataset of jobs, i.e., in which each single row represents a single job rather than a single person.

Dataset no.	Object type of dataset	List of variables
10	Person	age, turnover of employer

28. Thus, the information model of Fig.9 allows junk. It is clear that the object type relationships between *Job* and *Person*, and between *Job* and *Business* somehow play a role in the construction of datasets, and that the information model does not account for that.

29. To see what logical steps are required for the construction of datasets, we present the domain model of Fig.8 in a different way in Fig.10 below. First, we treat variables as arrows, in much the same way as we treat object type relationships. To account for value domains, we add the types *Amount* and *No. of years*. Next, we observe that arrows can be composed: the dashed arrow labelled *age ◦ employee* in Fig.10 represents the *employee* arrow followed by the *age* arrow. This indicates that the *age* of an *employee* is a meaningful attribute of a *Job*, cf. the seventh dataset in the list of datasets above. Finally, we observe that two (or more) arrows with a common origin (but possibly different targets) can be combined to form a dataset. Let us call this combination the *product* of two arrows. In Fig.10 the arrow labelled *‹salary, age ◦ employee›* is such a product; it corresponds to the fourth dataset in the list. The target value domain *Amount × No. of years* indicates the domain of all pairs of values taken from *Amount* and *No. of years*, respectively. Finally, the origin of the combined arrow, viz. *Job*, yields the object type of the dataset. We now claim that all datasets in the list above are constructed out of proper compositions and products[8] of arrows, and that these are the only ones that can be constructed in this way.

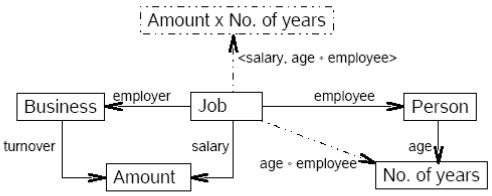

Figure 10: Logical constructs for datasets

IV. Discussion

30. The arguments of Sections II and III are meant to illustrate that the logic of classification systems is essentially that of Boolean algebras, and the logic used in the construction of datasets is essentially one of ‘arrows’, respectively. The initial information models in both sections were chosen as typical examples of models that lack such logic.

31. The viewpoint of a classification system as a Boolean algebra is developed in [6]. It is shown there that this viewpoint leads to concise characterizations of notions commonly associated with classification systems. For instance, a level is characterized as the set of atoms in a sub-algebra, and a hierarchy is a linear sequence of sub-algebra inclusions. It seems that the Boolean algebraic view of classifications is new. However, a related view on ontological systems [3] appears to point in that direction also.

32. The logic of ‘arrows’ mentioned in Section III is category-theoretical [1,9] in nature. In category theory, the composition of arrows is the primitive construct. The product of arrows developed in Section III is precisely the product construction from category theory. The ‘arrow’ view of a variable is due to [5,6]. There it is argued that a variable really is a function $v : s \to x$, where s represents a set of objects of statistical interest (a population) and x is a value domain. Thus, in the category of sets and functions, v is an arrow.

33. It seems that a category-theoretic foundation of statistical metadata is a study worth undertaking. Since this paper is not meant to give a full category-theoretic foundation, the constructs of Section III may appear to be rather superficial. They are not, however. For a start, if the ‘arrow’ view (or rather: the ‘function’ view) of a variable is taken as a point of departure, then the product of two variables *is* the dataset composed from them (rather than that the product merely *represents* the dataset in some way). Other constructs from category theory seem to give a precise understanding of common operations seen in the production of statistics. For instance,

[8] To be precise, products must be generalized to account for one arrow (instead of two or more) in order to represent, e.g., the first and second datasets in the list.

we claim that the row-wise combination of datasets is given by the coproduct [1,9], and that the construction of a family of variables (indexed, e.g., by points in time, as seen in time series) is explained by exponentiation [1].

34. The 'no junk, no confusion' paradigm originated from formal semantics [8]. The initial algebra style of semantics has had applications in many fields and has been very successful in the field of programming languages semantics [7] in particular. Also, it is very closely related to the notion of sketches in category theory (see [1]). In a very precise way, initial algebra semantics describe the connection between a(n information) model (e.g., a computer program, or the metadata of a dataset) and its 'real world' instance (the outcome of its execution and the dataset itself, respectively). It seems that this style of semantics can also be productive in the field of statistical metadata (see, e.g., [5]).

35. Having read Sections II and III, the reader may argue that the author is trying to solve problems that were due to himself primarily. It is true that in both sections the 'naïve' information model taken as a starting point is solely the author's choice. However, we claim that, essentially, the arguments remain valid were a different information model (e.g. that of [10]) taken as a point of departure. In particular we claim that [10] (as well as many similar models) lacks the logical constructs explained in the paper.

V. Conclusion

36. We presented two examples of information models that lack the logical mechanisms needed for the identification of synonymous information and the exclusion of nonsensical information. We also gave indications for repairing these models. The search for models that adhere to the 'no junk, no confusion' paradigm is based on theoretical underpinnings from formal semantics. It appears that formal semantics can play a constructive role in the development of sound models for statistical metadata.

VI. References

1. M. Barr and C. Wells, *Category Theory for Computing Science* (Les Publications CRM, Montréal, 1999)
2. E. van Bracht, CRISTAL, a Model for the Description of Statistics (CBS internal report, 2002)
3. R.E. Clay, Relation of Lesniewski's Mereology to Boolean Algebra, *The Journal of Symbolic Logic* **39**, No 4 (1974)
4. B.A. Davey and H.A. Priestly, *Introduction to Lattices and Order* (Cambridge University Press, 1990)
5. T. Gelsema, A statistical machine, part B: Types, sub-types, variables, and aggregates (CBS internal report, October 2006)
6. T. Gelsema, A statistical machine, part C: The nature of classification (CBS internal report, June 2007)
7. J.A. Goguen and G. Malcolm, *Algebraic Semantics of Imperative Programs* (The MIT Press, Cambridge Massachusetts, 1996)
8. J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright, Initial Algebra Semantics and Continuous Algebras, *Journal of the ACM* **24**, No 1 (1977)
9. S. Mac Lane, *Categories for the Working Mathematician* (Springer-Verlag, New York, 1998)
10. Statistical Data and Metadata Exchange Initiative, SDMX Information Model: UML Conceptual Design (Version 2.0, November 2005)