

A Classification Scheme of Validation Rules Applied to Statistical Data Bases

George PETRAKOS, Kostas KALOGEROPOULOS, Gregory FARMAKIS,
Photis STAVROPOULOS

Liaison Systems S.A.
Akadimias 77 - Athens - 106 77 GR
e-mail: {FirstName.LastName}@liaison.gr

Abstract: The introduction of a new logical structure of data validation rules, presented in this paper, aims at the enrichment and further refinement of the domain-based rules classification framework, adopted as a design standard by Eurostat. The new scheme introduces three types of generic validation rules using Boolean Algebra expressions, achieving a level of detail, which supports the direct translation of the rules into relational algebra. These expressions that can be constructed and evaluated at run-time, using a logical schema of generic data domains, maintained in a relational repository of rules. This approach, already been elaborated and tested vertically in all statistical sub-systems of the National Statistical Service of Greece, is illustrated here, using a list of validation rules from various classification cells of our design.

Keywords: Data Quality, Statistical Data Validation, Data Editing, Boolean Algebra

1. Introduction

Statistical agencies are producers of statistical information, which is reported to a variety of users, ranging from international organizations and governing bodies to the general public. The information needs to be of high quality and, therefore, the same holds for the data from which it is extracted. This has led to the adoption of data editing as a necessary step between data collection and data analysis. It has gained more prominence today as statistical agencies devote their efforts to the improvement of the whole range of their operations. It is not viewed anymore simply as a process of “cleaning up” data sets but mainly as a way to measure the quality of collected data and as a source of information for the more efficient design of surveys; see, for example, Granquist (1997), Granquist and Kovar (1997), Engstrom and Granquist (1999) and Nordbotten (1999).

The importance of editing was recognized early on and it attracted the research efforts of many statisticians (Nordbotten (1963, 1965), Pritzker et al (1965), Freund and Hartley (1967), Naus et al (1972)). As computing power increased, becoming cheaper at the same

time, the potential of the use of computers for data editing became apparent. Not only can computers perform the task quickly and consistently but they may also check the edit rules (hereafter, the edits) themselves for logical inconsistencies. Stored in a computer, the data take the form of database records. In their landmark work, Fellegi and Holt (1976) laid the principles of data editing (change as few fields as possible to make a record satisfy all edits) and the foundations for its implementation with computers. Even today however their algorithms pose a computing challenge; see Winkler (1999) for an overview of efforts to speed the algorithms up. Many national statistical services use sophisticated software and explore new methods for editing; see, for example, U.S. Bureau of the Census (1996), Winkler and Draper (1997), Chen et al (2000), Todaro (1998), Statistics Canada (1998), de Waal (1996, 2000a, 2000b), Engstrom (1999), Luzi and Pallara (1999), Revilla and Rey (1999, 2000) and Vickers (1999).

The authors of the present paper are in the process of formulating an alternative approach to the implementation of editing. The aim of the paper is to present a component of this approach, namely a systematic description and classification of all possible edits that can be used on one or multiple data sets. An important part of editing is the “cleaning up” of the data set either by “recontacting” the responding units or by automatic imputation. However, the present paper deals mainly with edits and we will not elaborate on imputation any further. The organization of the paper is as follows: in section 2 we present our proposed classification of edits. In section 3 our approach to editing is outlined. Section 4 is concerned with the architecture of a system that will implement our approach. Finally, we conclude, in section 5, with some remarks about the potential of our approach and we point to areas where further research is needed.

2. Classification of edits

Two of the authors have previously undertaken work on the classification of edits, see Petrakos & Farmakis (2000), which is complementary to the classification we will present here. For this reason we begin the section with a brief summary of this previous work. The edits are grouped into more comprehensive and meaningful categories, by identifying their targets. These are the *data elements* (variables), which are elementary data building blocks and have values, the *entities*, which are groups of logically related elements that have instances and the *data schemas*, which are groups of related entities. The classification scheme proposed is based on four axes, namely the target of the edits, the direction of coherence, the type of the variable(s) involved in the edits, and the target attribute checked by the edits. Therefore a primary L_{ijkl} classification is constructed where $i=\{0,1,2,3\}$ denotes the target of the edits, with $i=0$ consisting of preliminary edits applied to the metadata elements of the whole data set. Furthermore level 1 ($i=1$) consists of edits applied to a single data element, level 2 ($i=2$) of edits applied to an entity, while level 3 ($i=3$) edits are validating a group of entities (data schema). In a group of logically related variables, controls coded as L_{21++} are referred to as vertical coherence ($j=1$) edits (e.g. edits involving the sample mean of a variable), while controls coded as L_{22++} are referred to as horizontal coherence ($j=2$) edits (e.g. an association rule between fields of a record). The differentiation according to the type of the variable involved is denoted by $k=1$ for qualitative and $k=2$ for quantitative variables and finally $l=1,2$ denote edits checking the

data type-length and domain of the variables respectively. Therefore the classification scheme for the various edits applied (or that need to be applied) to statistical databases takes the form of a 4×2^3 table.

Edits may also be classified according to the type of the implied logical relationship between the variables they involve. This is what the new classification we propose here is based upon. This classification was developed in response to the requirements of the National Statistical Service of Greece (NSSG) for an efficient way of programming edits. NSSG plans to implement editing as part of its Integrated Information System, currently under development. A computer program for editing, will be written specifically for each survey. Our classification provides a formal language for expressing edits, which facilitates their transformation into computer code.

According to this classification, any edit, irrespective of whether it refers to data or to metadata can either be a range rule or an association rule. For example, an invalid blank due to non-response may be considered as a value outside the permissible range for the corresponding data element. For another example, consider the comparison of a data set with historical data. It amounts to checking whether one or more association rules connecting the schemas “data set” and “historical data” hold. By further dividing association rules into two categories we argue that any edit may be classified into one of the following categories:

Type I edit: $\{f(Va) \in A\}$

Type II edit: $\{g(Vb) \in B\} \Rightarrow \{f(Va) \in A\}$

Type III edit: $\{g(Vb) \in B\} \Leftrightarrow \{f(Va) \in A\}$

In other words, all edits are logical statements taking the value TRUE or FALSE in any particular situation. If a statement is a validity rule, it must take the value FALSE in order to consider that the corresponding edit is failed. In case of a conflict rule the contrary holds. In the following discussion we consider that the edits are validity rules.

In our notation, Va and Vb may be data elements, data entities, or data schemas. The use of the same notation for all kinds of data reflects the fact that conceptually we treat them all as the same thing. The term we use for Va and Vb is ‘variable’. The functions f and g may have any form. A very common case is the identity function (i.e. when the edit refers to variable Va itself). A simple example of a different than identity f is the case where Va is the vector of sample observations on a single characteristic and $f(Va)$ is the sample mean. Finally A and B are subsets of $f(D_A)$ and $g(D_B)$ respectively, where D_A and D_B are the domains (the ranges of allowable values) of Va and Vb respectively.

Type I edits are obviously range edits. Type II edits represent situations where if Vb satisfies a condition then Va must satisfy some other condition. Finally, type III edits represent situations where Va satisfies a certain condition if and only if Vb satisfies another condition. One may notice that both type II and type III edits may be re-expressed as type I edits. We demonstrate this for a type II edit. The Cartesian product $f(D_A) \times g(D_B)$ is the domain of the data entity $Vc=(f(Va), g(Vb))$. The edit constricts this domain because it is equivalent to,

$$\forall c \in ((A \times B) \cup (f(D_A) \times \sim B))$$

where $\sim B$ is the complement of B with respect to $g(D_B)$. However, since our aim is to develop an edit system for practical use, the effort to bring all edits into type I form will be an unnecessary complication for the user. On the other hand, the expression of an edit in the appropriate form, from the three mentioned above, is a straightforward operation. In translating an edit into computer code the user only has to declare its type, the data elements playing the roles of Va and Vb and the sets A and B . Some examples from the application of our classification at the NSSG are reported in the following subsection.

2.1 Examples

In this subsection we present six edits taken from five surveys that the NSSG conducts regularly. They first appear in natural language, as defined by subject matter experts and subsequently they appear in table 1 translated into the language of our classification. Note that the edits are expressed by the NSSG as validity rules. When a data record fails any of the edits a warning message is issued.

1. In a flower-growing conservatory the area allotted to flower growing (field V2.3.30) should be less than or equal to 10 acres (Agriculture and animal breeding census).
2. The average weight of slaughtered animals must be between 5 and 20 kilos. It is calculated from the answers to questions 'number of slaughtered animals' (field VII5.1) and 'total weight of slaughtered animals' (field VII6.1) (Hog Survey).
3. The total number of female students that graduate at the end of the academic year (field VE2.1.F) must be less than or equal to the number of female students who enrolled at the beginning of the same year (sum of entries to fields VD1.1.20 and VD1.1.29) (Education Survey).
4. The answer to question 'number of floors' (field VE1) may be 0 only if the construction we refer to is an enclosed swimming pool (field VE.3 is NOT NULL) (Private building activity survey).
5. If the road on which the accident happened is a two-direction road (entry to field V12.30 is 2) with a safety wall between the two directions (entry to field V12.36 is 1), then the lane width, as calculated from the answers to questions 'road surface width' (field V13.42) and 'number of lanes per direction' (field V12.31), must be in the range of [2.8, 4] metres (Road Traffic Accidents Survey).
6. Questions 'Capacity of solid animal waste tanks' (field V5.16), 'Capacity of liquid animal waste tanks' (field V5.17) and 'Capacity of semi-liquid animal waste tanks' (field V5.18) must all be left unanswered if and only if the answer to question 'Do you own storage tanks for solid, liquid or semi-liquid animal waste?' (field V5.15) is 'NO' (code 2) (Agriculture and animal breeding census).

<i>Edit</i>	<i>Type</i>	<i>Va</i>	<i>f(Va)</i>	<i>A</i>	<i>Vb</i>	<i>g(Vb)</i>	<i>B</i>
1	I	V2.3.30	V2.3.30	(0,10]	-	-	-
2	I	(VII5.1, VII6.1)	VII6.1/ VII5.1	[5, 20]	-	-	-
3	I	(VE2.1.F, VD1.1.20, VD1.1.29)	VD1.1.20+VD 1.1.29- VE2.1.F	[0, ∞)	-	-	-
4	II	VE1.3	VE1.3	NOT NULL	VE1	VE1	{0}
5	II	(V13.42, V12.31)	V13.42/ (2*V12.31)	[2.8 ,4]	(V12.30, V12.36)	(V12.30 , V12.36)	{2}X{1}
6	III	(V5.16, V5.17, V5.18)	(V5.16, V5.17, V5.18)	NULL	V5.15	V5.15	{2}

Table 1: Six edits from NSSG surveys represented in our classification language.

Such a table accompanied by a table explaining the questions represented by the fields involved in each edit is the input given to the programmers that will code the edits. The programmers at NSSG found that our classification facilitated their task considerably.

All the examples presented above refer to edits that may be performed within a single record during data entry. Macro edits referring to the whole data set or to more than one data set can be handled in exactly the same way.

3. Our general approach to editing

According to our perspective, edits are not just mathematical expressions binding abstract variables. Validation logic for any given data set inherently stems from, and in fact is part of, the very nature of the actual real-world objects represented by these data. Thus, it is an inseparable part of the data model itself, rather than just a set of “externally” applied logical rules. Here, by data model we mean the abstract representation of the real-world system under survey, as a system of inter-related objects (in object oriented terminology). Therefore, the statistical variables, measured by surveys, are properties or attributes of either objects or classes of objects, while validation rules are a manifestation of the logical structure of the system of objects. Thus, for example, an edit restricting the value of a variable according to that of another variable, may be a manifestation of the fact that the two variables are properties of the same object, or of two logically related objects. This leads to the fact that, any process of identifying, expressing, formalising, storing and ultimately applying edits, can only be consistent if and only if these edits are viewed within the framework of the information model, which must therefore be known to the editing system. In this aspect, validation logic is in fact another form of information metadata and should be treated as such, i.e. semantically declared within the data model and integrated with other metadata, while actual edits to be applied can be directly generated from this metadata model. This concept is exploited in our approach in two complementary ways. First, the knowledge of the information model may be used as a guideline for a consistent identification and mapping of the edits. Second, the application of the edits, i.e. the actual

validation of the data set, can be steered to follow the usually hierarchical structure of the model, from low-level single variables, to composite ones, to objects and eventually to records and data sets. This layered iterative procedure simplifies the whole validation process.

Our approach to editing requires the storage of edits along with the data in a distributed database. The way to achieve this is to modify, according to the edits, the domains that combinations of data elements would have if no edits were present. An example will make this clearer. In a market research survey, the data elements ‘age of respondent’ and ‘occupation of respondent’ form the data entity {age, occupation}. This data entity would have as domain the Cartesian product

$$(0,130) \times (\text{set of occupations}).$$

However, we know that values like (45,high school student) are not valid and this is specified by an edit like ‘IF occupation=high school student THEN age must lie in the range [13,19] years’. Our validation system will have as inputs the data model, the domains of the single variables and the edits. It will then construct the domains of data entities by taking Cartesian products. Subsequently, it will specify the sets of non-allowable values for each data entity based on the edits in which it is involved. In simple cases, e.g. categorical variables with few possible values, the user will have the option of inputting the non-allowable values. In complicated cases our aim is to let the edit system automatically calculate these non-allowable values. It is therefore necessary that it is able to determine the constraints imposed by edits on the variables involved.

The automatic determination of constraints is not always possible. A type I edit with an invertible f function is straightforward since it forces the domain of Va to become $f^{-1}(A)$. Assuming that all functions involved are invertible the edits force the variables involved to take values in the following sets:

Type I edit: $\{Va \in f^{-1}(A)\}$

Type II edit: $\{(Va, Vb) \in ((f^{-1}(A) \times g^{-1}(B)) \cup (D_A \times g^{-1}(\sim B)))\}$

Type III edit: $\{(Va, Vb) \in ((f^{-1}(A) \times g^{-1}(B)) \cup (f^{-1}(\sim A) \times g^{-1}(\sim B)))\}$

These forms define the modified combined domains of the variables and the corresponding sets of non-allowable values are determined by taking complements. When the functions are not invertible we must use **computed variables**. In our example showing how a type II edit is expressed as type I, Vc is such a computed variable. The validation system will calculate the necessary computed variables based on the edits input by the user. When the edits have been stored alongside the data, the system will only need to check for each element, entity or schema whether it takes allowable values. In other words, for the system all edits will be type I range rules.

4. Architecture of the validation system

The architecture proposed by the authors allows the representation of edits by means of algebraic operations on domains of variables instead of logical operations on values of variables. The architecture includes:

- (a) a repository holding both the information model of the survey (i.e. metadata on the multi-layered hierarchical structure of the variables) and the domain definitions of single variables;
- (b) a transient storage facility on which the data set under inspection can be stored and operated upon, while maintaining references to the metadata on the structure of the variables;
- (c) an edit definition model capable of calculating and storing in the rules repository the domains of data entities and schemas based on the edits definitions, according to the methodology presented in the previous section.
- (d) a validation engine capable of obtaining the domain of the variable under inspection from the rules repository and of checking the variable's values against this domain.

The repository, i.e. a specifically designed metadata database, is able to hold domain definitions for different kinds of single variables, as well as to store and manage the definition of the hierarchical structure of the data model, consisting of: single data elements, i.e. either simple variables of different types or computed variables and a multi-layered hierarchy of composite data elements, i.e. multidimensional variables, data records, data sets, and data schemas.

The validation engine can then extract the appropriate edits out of this metadata repository, acting in an iterative down-up manner, traversing the variables hierarchy from single variables towards the entire data set. This is done at each consecutive level, by applying the algebraic operations implied by the multidimensional variable metadata (including the declaration of edits) to the domains of the single variables to which the latter can ultimately be decomposed, in order to generate the corresponding multidimensional variable's domain.

5. Conclusions

In this paper we have presented a new classification of edits based on using Boolean arguments for expressing the relationships between the variables they involve. It creates a new formal language for expressing edits, which are traditionally declared with "IF-THEN-ELSE" statements. This enables the automation to a large extent of the procedure of programming edits. Our classification was implemented, by the National Statistical Service of Greece, in its new Integrated Information System.

Apart from facilitating the traditional implementation of editing, our classification can form part of our envisaged approach to editing. This approach, as we have briefly shown in section 3, relies on viewing the variables measured in a survey as representations of real world objects which have relationships imposed by their nature. These relationships naturally lead to edits. These edits, expressed in the language of our classification may be

inverted, where possible, and lead to modifications of variable domains. A computerized editing system may then check whether data records take allowable values or not. If inversion is not possible, the edits imply computed variables which can be stored alongside the original ones and can be treated exactly like them.

The formulation of our approach to editing is still at an initial stage and leaves many subjects to be researched in the future. First of all, we need to examine how it will be possible to check for logical inconsistencies between edits and to isolate the minimum set of fields requiring imputation, at a smaller computational cost than the Fellegi-Holt approach. We must also investigate ways of applying selective editing as part of our system. This requires finding a way of scoring records according to the impact they have on the estimates obtained from the data set. We have not so far tried to calculate such scores. Macro editing does not pose problems from a theoretical point of view but may be difficult in practice to incorporate in our system. All these issues are targeted for further research as part of the INSPECTOR /IST project.

References

- [1] Chen, B., Winkler, W. E. and Hemmig, R. J. (2000). Using the DISCRETE edit system for ACS surveys. *Technical report*, U.S. Bureau of the Census.
- [2] Engstrom, P. (1999). Process data in the Swedish standard program for editing – GRETA. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [3] Engstrom, P. and Granquist, L. (1999). Improving quality by modern editing. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [4] Fellegi, I. P. and Holt, D. (1976). A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, 71, 353, 17-35.
- [5] Freund, R. J. and Hartley, H. O. (1967). A procedure for automatic data editing. *Journal of the American Statistical Association*, 62, 341-352.
- [6] Granquist, L. (1997). The new view on editing. *International Statistical Review*, 65, 381-387.
- [7] Granquist, L. and Kovar, J. G. (1997). Editing of Survey Data: How much is enough? In *Survey Measurement and Process Quality*, L. Lyberg, P. Biemer, M. Collins, E. de Leeuw, C. Dippo, N. Schwarz and D. Trewin (eds.), New York: Wiley, 415-435.
- [8] Luzi, O. and Pallara, A. (1999). Combining macroediting and selective editing to detect influential observations in cross-sectional survey data. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [9] Naus, J. I., Johnson, T. G. and Montalvo, R. (1972). A probabilistic model for identifying errors in data editing. *Journal of the American Statistical Association*, 67, 340, 943-950.
- [10] Nordbotten, s. (1963). Automatic editing of individual statistical observations. *Conference of European Statisticians, Statistical Standards and Practice – No 2*, United Nations, New York.
- [11] Nordbotten, s. (1965). The efficiency of automatic detection and correction of errors in individual observations as compared with other means for improving the quality of statistics. *Bulletin of the International Statistical Institute*, Proceedings of the 35th session, Belgrade, 417-441.
- [12] Nordbotten, S. (1999). Strategies for improving statistical quality. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [13] Petrakos G., Farmakis G.(2000) A Declarative Approach to Data Validation of Statistical Data Sets, based on Metadata. *Statistica Applicata*, Vol.12.N.3, 2000.
- [14] Pritzker, L., Ogus, J. and Hansen, M. H. (1965). Computer editing methods – some applications and results. *Bulletin of the International Statistical Institute*, Proceedings of the 35th session, Belgrade, 442-465.
- [15] Revilla, P. and Rey, P. (1999). Selective editing methods based on time series modeling. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [16] Revilla, P. and Rey, P. (2000). Analysis and quality control from ARIMA modeling. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Cardiff, UK, 18-20 October 2000.
- [17] Statistics Canada (1998). Functional description of the Generalized Edit and Imputation System. Technical report.

- [18] Todaro, T. A. (1998). Evaluation of the AGGIES automated edit and imputation system. *Technical report*, National Agricultural Statistics Service, U.S. Department of Agriculture.
- [19] U.S. Bureau of the Census (1996). StEPS: Concepts and Overview. Technical report.
- [20] Vickers, P. (1999). An overview of edit and imputation in the 2001 UK census. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [21] Waal, T. de (1996). CherryPi: a computer program for automatic edit and imputation. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Voorburg, the Netherlands, 4-7 November 1996.
- [22] Waal, T. de (2000a). SLICE: generalized software for statistical data editing and imputation. *Technical report*, Statistics Netherlands.
- [23] Waal, T. de (2000b). New developments in automatic edit and imputation at Statistics Netherlands. *Report presented at the UN/ECE Work Session on Statistical Data Editing*, Cardiff, UK, 18-20 October 2000.
- [24] Winkler, W. E. (1999). State of statistical data editing and current research problems. Report presented at the *UN/ECE Work Session on Statistical Data Editing*, Rome, Italy, 2-4 June 1999.
- [25] Winkler, W. E. and Draper, L. R. (1997). The SPEER edit system. In *Statistical data editing*, Vol. 2, UN Statistical Commission and UN/ECE, Geneva, Switzerland, 51-55.