

**STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE**

**COMMISSION OF THE
EUROPEAN COMMUNITIES**

CONFERENCE OF EUROPEAN STATISTICIANS

EUROSTAT

**Joint ECE/Eurostat Work Session on
Statistical Data Confidentiality**
(Thessaloniki, Greece, 8-10 March 1999)

Working Paper No. 7
English only

Topic (ii): software and computing developments

ARGUS: SOFTWARE FROM THE SDC PROJECT¹

Submitted by Statistics Netherlands²

Invited paper

I. Statistical Disclosure Control

1. Statistical offices collect information about persons, businesses, institutions, etc. through censuses and surveys. The data collected are ultimately released in a suitable form to policy makers, researchers and the general public for statistical purposes. The release of such information may have the undesirable side-effect that information on individual entities instead of on (sufficiently large) groups of individuals may be disclosed. The question then arises how the information available can be modified in such a way that the data released can be considered statistically useful and do not jeopardise the privacy of the entities concerned.

2. The aim of Statistical Disclosure Control (SDC) is to limit the risk that sensitive information of individual respondents can be disclosed from a data set. The data set can be either a microdata set or a table. A microdata set consists of a set of records containing information on individual respondents. A table contains aggregate information of individual entities.

3. In order to publish safe data, one should first have criteria to check whether a particular data set is safe according to these criteria or not. If data are not safe according to these criteria they have to be modified in such a way that the resulting data meet these criteria. These modifications, while decreasing the risk of disclosure, also imply that the information content of the data is decreased, because certain variables are coded in a less detailed fashion or values are suppressed or replaced by other values. The idea is that the modifications should be applied in such a way that the resulting information loss is

¹ This paper is an updated version of the paper "ARGUS for statistical disclosure control" by the same authors, presented at the SDP98 conference, Lisbon, March 1998.

² Prepared by Anco Hundepool and Leon Willenborg. The views expressed in this paper are those of the authors and do not necessarily reflect the policies of Statistics Netherlands.

minimised. As a rule achieving this goal is quite complicated and requires the use of specialised software tools. Such tools are μ -ARGUS for microdata and τ -ARGUS for tabular data. It should be remarked that the criteria for distinguishing safe from unsafe data is not a simple issue. It basically requires that one tries to model the behaviour of a person who, confronted with a data set, might want to identify an individual and disclose certain information of this person. The data releaser should use this disclosure scenario to develop safety criteria from this. These criteria reflect the target population of potential users of a particular data set (e.g. researchers, the general public, etc.) and the legal and organisational measures that accompany the release of these data. Dependent on this, the criteria can be more tight or less so.

4. Major research topics within the SDC area concern the development of a theory for disclosure risks from which data protection criteria are derived, how to quantify information loss in data as a result of data modifications and the development of algorithms to apply SDC techniques in a (more or less) optimal way to transform unsafe data into safe data, where "optimal" here means with as little information loss as possible, given a particular quantification of information loss. In practice one sometimes has to be pragmatic and use safety criteria that are fairly easy to apply and that can (only) be intuitively justified. Also one often has to be satisfied with modifications of the data that are not optimal (in terms of information loss) but only nearly so, because the optimisation problems that have to be solved are too complex. For more information on the backgrounds of SDC one may consult [7].

5. The current versions of μ -ARGUS and τ -ARGUS have been developed as part of an effort within an international project, viz. the SDC project, sponsored by the EU. Within the SDC project the task of Statistics Netherlands, Consorzio Padova Ricerche (CPR) and the Eindhoven University of Technology (TUE) was to cooperate in order to produce ARGUS, each specialising in particular areas. Statistics Netherlands was responsible for the interfaces and major portions of the respective kernels of both ARGUS packages. TUE and CPR were responsible for implementing software that can be used to tackle the optimisation problems that ARGUS has to solve when applying global recoding, local suppression, secondary cell suppression and rounding automatically (see [5] and [4]).

6. In the remainder of this paper, two SDC packages are described that can be used to produce safe microdata (μ -ARGUS) and safe tables (τ -ARGUS). In view of the length of the paper we can only highlight the main functionality of ARGUS and not go into every detail. The main portions of both sections consist of explaining the current functionality of both packages. In the final section possible future extensions of ARGUS are described.

II. μ -ARGUS for microdata

7. First we describe the basic background philosophy of μ -ARGUS and then its main functionality. For a full description the reader is referred to the user manual corresponding to the package.

II.1 *Background to μ -ARGUS*

8. In case of a microdata disclosure of sensitive information of an individual respondent can occur after this respondent has been re-identified. That is, after it has been deduced which record corresponds to this particular individual. So, disclosure control should hamper re-identification of individual respondents.

9. Re-identification can take place when several values of so-called identifying variables, such as 'Place of residence', 'Sex' and 'Occupation', are taken into consideration. The values of these identifying variables can be assumed known to friends and acquaintances of a respondent. When several values of these identifying variables are combined a respondent may be re-identified. Consider for example the following record obtained from an unknown respondent:

'Place of residence = Urk', 'Sex = female' and 'Occupation = statistician'.

10. Urk is a small fishing-village in the Netherlands in which it is unlikely for many statisticians to live, let alone female statisticians. So, when we find a statistician in Urk, a female one moreover, in the microdata set, then she is probably the only one. When this is indeed the case, anybody who happens to know this rare female statistician in Urk is able to disclose sensitive information from her record if such information is contained in this record.

11. An important concept in the theory of re-identification is a key. A key is a combination of identifying variables. Keys can be applied to re-identify a respondent. Re-identification of a respondent can occur when this respondent is rare in the population with respect to a certain key value, i.e. a combination of values of identifying variables. Hence, rarity of respondents in the population with respect to certain key values should be avoided. When a respondent appears to be rare in the population with respect to a key value, then disclosure control measures should be taken to protect this respondent against re-identification. "Rare" means that a combination of characteristics occurs less than a certain threshold value D_k , where k is a key, implying that the threshold value depends on k . One can define the threshold value at the population level, and then use an equivalent threshold value for a sample, as usually is the case.

12. A key value that occurs less than D_k times in the population is considered unsafe, a key value that occurs at least D_k times in the population is considered safe. The unsafe combinations must be protected, while the safe ones may be published.

13. When the estimated frequency of a key value, i.e. a combination of scores, is at least equal to the threshold value D_k , then this combination is considered safe. When the estimated frequency of a key value is less than the threshold value D_k , then this combination is considered unsafe. An example of such a key is 'Place of residence', 'Sex', 'Occupation'.

14. μ -ARGUS is developed to remove a set of unsafe combinations from a microdata set, i.e. by modifying some of these key values. The current version uses two techniques for this: global recoding and local suppression. In case of global recoding several categories of a variable are collapsed into a single one. In the above example, for instance, we can recode the variable 'Occupation'. For instance, the categories 'statistician' and 'mathematician' can be combined into a single category 'statistician or mathematician'. When the number of female statisticians in Urk plus the number of female mathematicians in Urk is sufficiently high, then the combination 'Place of residence = Urk', 'Sex = female' and 'Occupation = statistician or mathematician' is considered safe for release.

15. The effect of local suppression is that one or more values in an unsafe combination are suppressed, i.e. replaced by a missing value. For instance, in the above example we can protect the unsafe combination 'Place of residence = Urk', 'Sex = female' and 'Occupation = statistician' by suppressing the value of 'Occupation' in the records in which the unsafe combination occurs. This only leads to a safe combination of scores if the number of females in Urk is sufficiently high. The resulting combination is then given by 'Place of residence = Urk', 'Sex = female' and 'Occupation = missing'.

16. Both global recoding and local suppression lead to a loss of information, because either less detailed information is provided or some information is not given at all. A balance between global recoding and local suppression has to be found in order to make the information loss due to the application of SDC measures as low as possible.

II.2 Functionality of μ ARGUS 3.0

17. We assume that the data protector has selected a microdata set that has to be protected, and that he has chosen μ -ARGUS to produce safe data. (This implies that he uses the type of approach that μ -ARGUS supports.) The microdata are assumed to be in (the commonly used) ASCII format, and in the form of a rectangular file.

Input

18. First of all, the data protector has to explain to μ -ARGUS which variables are the identifying ones in the microdata set and where they are located in the file. If he intends to use an option of μ -ARGUS to generate particular combinations of identifying variables (the keys), he might have to supply additional information. This information actually provides levels of identifiability to the various identifying variables. The variables belonging to one particular level of identifiability also belong to the one with a lower level of identifiability. In that way nested sets of identifiers are obtained. As an example consider the case that three levels of identifiability occur in the data file. μ -ARGUS then generates all possible combinations of identifiers in which all three levels are represented.

19. The data protector should also indicate which variables are (very) sensitive, which are direct (person, household, etc.) identifiers, which variables are hierarchical, which are numerical, which is a (post-sampling) weight variable, which are household variables (that have - by definition - the same scores for all members of the same household) and, finally, which variables are regional indicators (pertaining to the same region).

20. We provide below some comments on these items:

- i. *Very sensitive variables* are sometimes taboo in microdata sets - such as the public use files at Statistics Netherlands -, particularly if there are several very sensitive categories (such as 'suicide' for 'Cause of death').
- ii. *Direct identifiers* should never be released in a microdata set, but they can be useful during the execution of μ -ARGUS, for instance to identify members of the same household. After their use during the execution of the program, they should be discarded and not be copied to the file to be released.
- iii. *Hierarchical variables* allow pruning (=chopping off digits), which is a form of global recoding. It is still possible that a hierarchical variable can be interactively recoded. This would allow one to go beyond the uniform levels of coding that pruning implies.
- iv. *Numerical variables* should not be identifiers, as the tables in which they would be used tend to be very big, and each cell would turn out to be sensitive. Furthermore it tells μ -ARGUS to which variables it can apply techniques that require numerical operations, such as microaggregation or controlled rounding. In case either of these techniques is to be applied to a variable the data protector should specify the group size in case of microaggregation, and the rounding base in case of controlled rounding.
- v. *Weight variables* sometimes pose a special problem, namely in case it is not desired that a future user of the data can use these weights to identify the (post-sampling weighting) stratum to which a respondent belongs. Via this, somewhat indirect, route he can

- disclose information related to the stratification, for instance the region where a respondent lives. This is undesirable if no regional information is allowed to be present in the file, directly or indirectly (i.e. via these weights).
- vi. For some microdata sets it might be necessary to make sure that a possible intruder is not able to reassemble the households to which individuals represented in the sample belong. This requires that checks are possible using *household variables*.
 - vii. In some cases a data protector wants to investigate if the *regional attributes* in a microdata file pertaining to the same type of region (say 'Degree of urbanization', 'Number of inhabitants' (in classes), 'Number of railway stations' of the 'Area of residence' do not, taken together, identify fairly small regions in the population). To investigate this, the data protector should also specify the name of a file containing this sort of regional information, so that μ -ARGUS has the necessary input to do the checking.

21. If the data protector wants μ -ARGUS to apply the automatic global recoding option he should provide some additional information. First of all, for each identifier he should specify a set of alternative codings. For hierarchical variables it is assumed that the allowed codings are implicitly given by the various coding levels possible. μ -ARGUS is allowed to choose one of these codings for each identifier. Also the data protector should specify a weight for each identifier, indicating the importance of this identifier to a future user: the higher the weight, the more important the variable is to such a user. Furthermore the data protector should provide a weight for each possible coding: the higher the weight the more important the coding is. The final weight associated with each coding is (proportional to) the product of the weight associated with the variable and the initial weight associated with the coding. This allows the data protector to separate the importance of the various identifiers from the importance of the codings for each variable. Of course, the choice of these weight should reflect the intended use of the data, say by the majority of the users. Furthermore the data protector should specify a weight for each identifier that quantifies the information loss due to local suppression applied to the respective identifiers.

22. When all the necessary meta-information has been specified μ -ARGUS is able to generate a specific set of combinations (=tables) that it will check. The data protector is able to modify the set of tables generated by μ -ARGUS, if he wishes to do so. Or he can produce his own set of tables, without using the table generation option of μ -ARGUS.

23. Not only should μ -ARGUS know which combinations of variables it should check, it should also know which threshold value(s) to use. Cells with a frequency less than the corresponding threshold are considered unsafe. In μ -ARGUS different threshold values can be specified for each table.

Actions

24. If μ -ARGUS has all meta-information at its disposal, it is ready to identify the unsafe combinations. This requires a single run through the microdata. In this run the tables are calculated and the unsafe combinations are identified. The data protector can choose whether he wants to use the raw cell frequencies or smoothed versions.

25. For the data protector summary information about the unsafe combinations is prepared so that he is able to decide about the actions that should be undertaken to get rid of the unsafe combinations. He might decide to do an interactive global recoding, or to leave this to μ -ARGUS. In the latter case μ -ARGUS needs coding information that should have been specified by the data protector at the beginning of the session (namely which alternative codes there are for each identifier, and the weights to quantify information loss in

case of global recoding or local suppression). If this information is not available, the only option is that the user tells μ -ARGUS which global recodings to carry out.

26. It should be remarked that the effect on unsafe combinations by applying global recodings can be derived from the tabular information only. There is no need to go back to the microdata for this. Therefore the size of the underlying microdata set is completely irrelevant for the efficiency at which the global recodings can be carried out. This is only determined by the sizes of the tables, i.e. the combinations, that have to be checked.

27. In case of automatic global recoding there is a problem that μ -ARGUS has to solve, because of lacking information. This is related to the fact that for this option it is not necessary to eliminate all unsafe combinations by global recodings. Finding the optimum mix of global recodings and local suppressions is the goal. But μ -ARGUS lacks information when it tries to assess the information loss incurred when a set of unsafe combinations is being eliminated by local suppressions. This is due to the fact that it is not known in which records unsafe combinations occur, and therefore how the unsafe combinations overlap, i.e. have common values. There is good reason for this lack of information: at the initial run through the microdata to fill the tables to be checked, it is unknown how many unsafe combinations will turn up. This might turn out to be a very big number. Therefore an enormous amount of space could be required to store, for each unsafe combination, in which record it occurs. Apart from this, most of this information will not be used anyway at the recoding stage (which does not need this kind of information), and therefore would be an expensive waste of resources: it slows operations down and demands storage which is of no use. Therefore μ -ARGUS has to make an estimate of the information loss due to local suppressions to eliminate a set of unsafe combinations. For background information on the OR models used for automatic global recoding and local suppression refer to [5].

28. At one point this global recoding phase is over, and then possibly remaining unsafe combinations will have to be eliminated by local suppressions. This is carried out automatically by μ -ARGUS; no interactive process is possible (nor desirable!). For this μ -ARGUS needs to go to the microdata. First it needs to apply the global recodings to the microdata, and then it processes record by record, carrying out local suppressions in case of records containing an unsafe combination (after the global recodings have been applied).

29. In case the data protector had indicated that he wants to apply microaggregation or controlled rounding (by specifying the necessary meta-information) to one or more numerical variables in the file, this is the moment to carry it out. In case of controlled rounding, this can be combined with the previous step, which executes global recodings and local suppressions. In case of microaggregation, a sorting of the data (of the variable in question) needs to be performed first, so that it is possible to form the respective groups and calculate the respective group means. This has to be repeated as many times as there are variables that are to be microaggregated. In case of controlled rounding no preliminary sorting is needed but only a single pass through the data that can also be combined with a previous step in the process, that executed the global recodings and the local suppressions.

30. Other data modification techniques that can be applied are rounding and top and bottom coding.

Output

31. After the actions on the microdata set have been completed, μ -ARGUS produces several output files. First of all there is the modified microdata set, produced from the input microdata set. This modified microdata set is safe according to the confidentiality rules that

have been used by the data protector (through τ -ARGUS), and can therefore be released. The second file should contain the updated metadata information. This file is different from the input metadata file only if variables have been globally recoded, or new variables have been produced (e.g. as a result of microaggregation or rounding). In case of automatic global recoding and microaggregation the names of new categories or variables have been specified in the metadata. In case of interactive global recoding the user should provide new label names. The third file is a report file, containing summary information on the modifications that have been applied to the original file by τ -ARGUS. The idea is that this file should be kept as a document by the data protector.

III. τ -ARGUS for tables

32. In the present section τ -ARGUS is discussed, first by providing some background information to motivate the various design choices, and then by describing its main functionality. Contrary to the situation of microdata, the disclosure problems for tables are more transparent than those for microdata. This is because tables are generally less rich in structure than microdata.

33. For tables concepts like "disclosure risk" and "information loss" are more clearcut and less controversial than in case of microdata - or so they seem. The result is that the hard part of protecting tables is essentially a matter of solving well-defined optimisation problems, rather than puzzling over modelling issues as in case of microdata.

III.1 Background to τ -ARGUS

34. τ -ARGUS is intended for producing safe tables. The current version of τ -ARGUS can only handle a single table (together with its marginals) of dimensions less than 5. τ -ARGUS can handle two kinds of tables, namely magnitude tables and frequency count tables. The difference is important for several reasons: first when employing a disclosure risk model to a (set of) table(s) in the way of defining sensitive cells, and second when protecting a table. In the latter case it makes a difference if the cell values can only take integer values or not.

35. The safety of a table is determined by the existence of sensitive cells and whether the cell values in these cells can be considered sufficiently protected. τ -ARGUS identifies sensitive cells in magnitude tables by employing a "dominance rule" (see e.g. Section 6.2 in [7]). This rule states that a cell of a table is unsafe for publication if a few, n say, major contributors to a cell are responsible, when adding their contributions, for at least a certain percentage p of the total of that cell. A common choice is $n=3$ and $p=70\%$, but τ -ARGUS allows users to specify other parameter settings. Applying a dominance rule to a frequency count table implies a thresholding rule: nonempty cells with a frequency less than the threshold are considered unsafe, whereas those with a frequency above the threshold are considered safe (this is comparable to a thresholding rule for microdata). In some cases this approach makes sense, but in other cases it does not (see [7], Section 6.3). When it does not, considerations motivated by group disclosure are taken into account, which go beyond those concerning individual disclosure considerations that are usually being applied.

36. It should be noted that for τ -ARGUS to identify the sensitive cells in a magnitude table, it needs, for a dominance rule with parameters n and p , apart from the cell totals, the sums of the top n contributors in each cell. If a user is permitted to lump rows or columns in the table together (in order to protect sensitive cells), then it is useful for each table cell to store the individual top n contributions of that cell instead of their sum. This allows τ -ARGUS to calculate the top n contributions for each cell that has been created by lumping

two or more cells together. It simply requires that the top n contributions for all cells are merged and the top n contributions for this new cell is calculated.

37. Once τ -ARGUS has identified all sensitive cells in a table, it helps a user to protect them through the execution of certain SDC techniques, such as cell deletion, cell suppression, table redesign or rounding. Some of these operations have to be carried out interactively, using inputs provided by the data protector, while others can be done automatically by τ -ARGUS itself.

38. A complication that typically exists in case of protecting tables - and what makes the exercise difficult - is the presence of additional constraints in the data, such as additivity constraints in case marginal tables are present or non-negativity constraints of cell values. Due to the presence of these constraints cell suppression is usually not quite what it suggests: an interval of feasible values for a suppressed cell (in a pattern of such cells) can be calculated, rather than that the suppressed value is completely unknown. For rounding, the constraints that apply to the original table are imposed on the rounded table as well. Besides, the rounded table (and its rounded marginals) should be close to the original table (and its marginals) as well, assuming a suitable metric to measure distances.

III.2 Functionality of τ -ARGUS 2.0

39. We assume that the data protector has chosen τ -ARGUS to produce safe tables. Contrary to the use of μ -ARGUS the current version of τ -ARGUS requires the use of an externally called LP-solver package (Xpress), to carry out local suppressions and controlled roundings.³

Input

40. All the tables are built by τ -ARGUS from scratch from a microdata file. For magnitude tables it is necessary that, along with the table to be protected, information is made available to apply the dominance rule (i.e. top k contributions per cell, where k is a parameter used in the dominance rule to be specified by the data protector). Although for frequency count tables it is strictly not necessary to start tabulating in τ -ARGUS; this could be done elsewhere. Since every contribution to a cell total is the same, there is no need to calculate this from the base file. On the other hand it does not harm either.

41. The data protector can request τ -ARGUS to protect several tables, calculated from the same base file. These tables are then protected one by one, and independently of each other. In order to describe to τ -ARGUS which tables to calculate from which microdata file, and how to identify the sensitive cells in them, the data protector should provide the necessary input information. The information concerning sensitive cells is provided through the specification of the dominance rule to be used or a threshold value, depending on the type of table at hand. In case several tables have to be protected it is possible to use a separate sensitivity rule for each table. By default the system takes the cell value as the suppression weight (i.e. high values get a high weight, and will therefore be less likely suppressed). But it is possible to use other suppression weights.

Actions

³ The LP-solver Xpress that the τ -ARGUS 2.0 requires should be independently purchased by the data protector with Dash Associates in the UK.

42. After the data protector has made the necessary preparations τ -ARGUS can proceed to produce the tables from the microdata set, which is the base file. If a user has specified multiple tables, they will be dealt with one by one, and mutually independent. We describe the situation for one table.

43. In case a magnitude table is to be protected, τ -ARGUS will not only produce this table but also additional tables containing for each cell the top k contributions to each cell (or less if there are no more contributions in a cell). On the basis of this information and the parameters for the dominance rule it can calculate the unsafe cells. Summary information on these cells will be shown to the data protector. In case of a frequency table, τ -ARGUS can also calculate the sensitive cells, using a threshold rule or a dominance rule as specified by the user. Irrespective of the kind of table, i.e. a magnitude table or a frequency count table, the sensitive cells in the table (and its marginals) are known.

44. The user should know decide whether to apply a table redesign first or to apply another technique. If he decides - on the basis of the relatively high number of unsafe cells - to apply a table redesign, then he should decide which spanning variables to take for that and how to recode these spanning variables. This recoding has to be done interactively, where the user can either come up with a suggestion to combine certain rows, columns, etc. or to pick a coding from a list of possibilities, provided he has prepared such a list when he started the session. He might want to repeat this recoding procedure, in which he also has the option to undo earlier made recodings. At some point the data protector might decide that there have been enough recodings. If there are no sensitive cells left he is done, and he has produced a safe table. If not, he has to decide which other techniques to use to the current table. He can choose from two techniques: cell suppression and rounding.

45. If he chooses to apply rounding he has to specify a rounding base. If he decides to apply cell suppression he should specify a set of weights for τ -ARGUS, indicating for each nonsensitive cell how much loss it means if this cell would be suppressed. Both the rounding and the secondary suppression are carried out automatically by τ -ARGUS. In case of controlled rounding the result is a rounded table with rounded marginals that is also additive. After having finished these procedures, the data protector has the possibility to undo the results obtained so far and go back to the table redesign phase. From thereon he can do the process all-over, starting with a clean slate. At one point this process terminates and the data protector should tell τ -ARGUS that he is satisfied with the result. If the data protector did not use input files containing the recoding information, but did the table redesigning interactively by producing recodings "on the spot", he has to specify certain new meta-information for the protected table, i.c. labels of variables and categories used. τ -ARGUS then produces the necessary output files.

Output

46. The output that τ -ARGUS produces consists of two files. First of all there is the protected table. The user can choose between three formats to output this table: ASCII or WK1. The latter format is accepted by many spreadsheet packages that can be used to produce a nice print of the table. It should be noted that, deliberately, τ -ARGUS has no facilities itself to layout the table. The reason is that this is an activity that does not belong to the field of data protection proper, and for which there is software available already. Like in case of μ -ARGUS, a logfile is produced containing information about the changes that have been made to the original table.

47. The third file contains the meta-information concerning the new table, i.c. a description of the variables and codes used.

IV. Possible future extensions of ARGUS

48. The current versions of μ -ARGUS and τ -ARGUS have been developed as part of an effort within an international project, viz. the SDC project. Within the SDC project the task of Statistics Netherlands, Consorzio Padova Ricerche (CPR) and the Eindhoven University of Technology (TUE) was to cooperate in order to produce ARGUS, each specialising in particular areas. Statistics Netherlands was responsible for the interfaces and major portions of the respective kernels of both ARGUS packages. TUE and CPR were responsible for implementing software that can be used to tackle the optimisation problems that ARGUS has to solve when applying global recoding, local suppression, secondary cell suppression and rounding automatically (see [5] and [4]).

49. The task of the SDC project was broader than developing ARGUS. Methodological studies in the SDC area were an important task as well. These methodological studies focussed on disclosure risk models for both microdata and tables, the disclosure scenarios and empirical disclosure risk assessment for microdata (see [3]), hierarchical and linked tables (see [6]) and, finally, flexible tabulation and geography in connection with GIS (see [2]). Most of the outcomes of these researches can be used as input for future versions of μ -ARGUS or τ -ARGUS. But also developments outside the SDC project may have their impact on the future shape of ARGUS. For instance a recently developed technique like PRAM (see [1]) is likely to find its way into μ -ARGUS.

50. The current versions of both ARGUS packages have been designed to run under Windows 95. This is mainly an issue concerning the interfaces of both packages. When developing the ARGUS packages a deliberate goal was to implement the respective kernels independently from the respective interfaces, and to write the kernels in such a way that mainly standard C++ commands were used, so that porting the packages to another platform (say Unix or Linux) would merely require re-implementing the interfaces. Future versions of both ARGUS packages will be adapted to work with new releases of Windows.

51. In the meantime, a shift of thinking has taken place and an entirely different approach is being considered. This approach is based on a kind of client-server model and allows ARGUS to be used through remote access. This remote access approach would have the advantage that a powerful server can be used to do the heavy computational work, such as searching through large search spaces or solving linear or integer programs. Giving statistical offices access to the functionality of ARGUS, running on a remote server at say CBS or Eurostat through the Internet, to protect their data yields new opportunities and challenges. These need to be further researched.

References

- [1] P.-P. DE WOLF, J.M. GOUWELEEUW, P. KOOIMAN and L.C.R.J. WILLENBORG (1998). Reflections on PRAM. *Proceedings SDP98, Lisbon*.
- [2] O. DUKE-WILLIAMS and PH. REES (1998). Look-up Table Techniques for Producing New Census Data for Small Areas. *Proceedings SDP98, Lisbon*.
- [3] M.J. ELLIOT, C.J. SKINNER and A. DALE (1998). Special Uniques, Random Uniques and Sticky Populations: Some Counterintuitive Effects of Geographical Detail on Disclosure Risk. *Proceedings SDP98, Lisbon*.
- [4] M. FISCHETTI and J.J. SALAZAR (1998). Modeling and Solving the Cell Suppression Problem for Linearly-Constrained Tabular Data. *Proceedings SDP98, Lisbon*.
- [5] C.A.J. HURKENS and S.R. TIOURINE (1998). On Solving Huge Set-cover Models for the Microdata Protection Problem. *Proceedings SDP98, Lisbon*.

- [6] F.M. MALVESTUTO and M. MOSCARINI (1998). An Audit Expert for Large Statistical Databases. *Proceedings SDP98, Lisbon*.
- [7] L. WILLENBORG and T. DE WAAL (1996). *Statistical Disclosure Control in Practice*, Springer-Verlag, New York.