WP. 14
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Ljubljana, Slovenia, 9-11 May 2011)

Topic (iii): Macro editing methods

MacroView: a generic software package for developing macro-editing tools

Invited Paper

Prepared by Saskia Ossen, Wim Hacking, Ralph Meijers, and Peter Kruiskamp, Statistics Netherlands[1]

I. INTRODUCTION

1. Statistics Netherlands has developed a software package called MacroView that enables an efficient development of custom-tailored macro-editing tools. Generic functionalities of macro-editing tools are implemented in MacroView. These functionalities are in fact the building blocks of all macro-editing tools developed in MacroView. To actually build such a custom-tailored macro-editing tool using these building blocks, a script language can be used. Scripts specify which building blocks are combined in a macro-editing tool and how. Macro-editing tools developed in MacroView are currently being used in the redesigned Dutch Structural Business Statistics and Road Statistics. At the moment MacroView scripts are developed and tested for the redesigned Short term Statistics.

2. The first aim of this paper is to give a brief introduction of the (macro-edit) functionalities that are already supported by MacroView (see also Hacking, and Ossen (2011)). It will be shown that a macro-editing tool build in MacroView can, among others, make use of predefined:

- Aggregation techniques (like Sum, Median, Standard Deviation, and Count)
- Transformations of data
- Methods for linking datasets
- Functions for drawing grids and several types of plots
- Functions allowing for dynamic composition of filters, making it possible for a user of the macro-editing tool to dynamically select data in a grid or plot and to subsequently perform analyses for the filtered data
- Functions for efficient recalculation of plots and tables after micro-data have changed.

3. The second aim of the paper is to discuss the challenges met while developing macro-editing tools in MacroView for the Structural Business Statistics and the Short term Statistics. It was found that the main challenge regarding the Structural Business Statistics was the presentation of a huge number of variables at several different aggregation levels to the analyst in a well-organized way. For the Short term Statistics the size of the input files was found to be the main challenge as administrative records (VAT-data) are used for these statistics. At Statistics Netherlands VAT-data are available on all companies in the Netherlands, leading to large numbers of records at the micro-level.

[1] The views expressed in this paper are those of the authors and do not necessarily reflect the policies of Statistics Netherlands.

4. To reach these aims the paper is structured as follows. In section II we shortly discuss the motivation for developing MacroView. In section III an overview is provided of the functionalities currently supported by MacroView. In section IV and section V we discuss the two biggest challenges met while building macro-editing tools in MacroView, i.e. showing a large number of variables in a well-organized way, and dealing with a large number of records. In section V1 we present our conclusion and discuss future work.

II. MOTIVATION FOR DEVELOPING MACROVIEW

5. Macro-editing (or Topdown Analysis) (Granquist (1994), de Waal and Haziza (2009)) was already in use at a number of places at Statistics Netherlands before the development of MacroView started. As no generic software package was available for developing macro-editing tools, customer-tailored tools were built for several statistics in a variety of standard software packages.

6. As Statistics Netherlands is currently redesigning several statistics in order to produce them (among others) in a more efficient way (see, for example, Braaksma (2007)), macro-editing becomes more and more important. This implies that for several statistics new macro-editing tools have to be developed or existing macro-editing tools need to be adapted.

7. Although all these statistics have their own variables that need to be analyzed in a macro-editing tool, it is important to recognize that the techniques used by these statistics in macro-editing share a lot of common characteristics. For example, micro-data need to be aggregated at one or more aggregation levels, aggregated data have to be shown in grids or plots, users have to be able to navigate through aggregated data shown at different aggregation levels, aggregates have to be recalculated after the underlying micro-data are adjusted, and so on.

8. MacroView is developed to provide a common base for developing macro-editing tools for different statistical departments. A lot of functionalities useful for a macro-editing tool are implemented once and can easily be accessed via a script language. This implies that a lot of work in developing and testing only needs to be done once. This reduces the development time of such a tool considerably. In deciding which functionalities are needed, we had brainstorm and review sessions with analysts from different statistical departments. Furthermore, we performed a short literature study. All this resulted in a first production version of MacroView. This first version has been iteratively improved based on continuous feedback and new requests from users.

III. FUNCTIONALITIES AVAILABLE IN MACROVIEW

9. In this section we provide a short overview of the functionalities currently available in MacroView. This overview is certainly not exhaustive, for more details the authors can be contacted.

A. Aggregation techniques

10. In almost every macro-editing tool it will be necessary to aggregate micro-data. Several commonly used aggregation techniques are therefore predefined in MacroView and can easily be accessed via a MacroView script. Figure 1 shows an example of an excerpt of a script in which micro-data are aggregated.

11. An aggregation function like the one shown in Figure 1 can be reused in the script every time that it is needed to aggregate data at a different aggregation level or to apply a different filter to the micro-data serving as input to the aggregation. In illustration, the aggregate function shown in Figure 1 can be used to calculate aggregates at the SBI 2 digit aggregation level, the SBI 3 digit aggregation level, and higher digit levels (the first 4 digits of SBI are almost comparable to NACE, the fifth digit is a more detailed differentiation used in the Netherlands).

```
{*****************************************************************************}
{-- [Aggregates] Do not remove! --}
{*****************************************************************************}

AGGREGATE Calculate_Aggregate
    INPUT
        DataT : MODEL_Micro_data
    OUTPUT
        Sum_DataT : MODEL_Macro_data

    CELLS
        NumberOfCompanies := COUNT(ALL);
        AverageTurnover := MEAN(Turnover);
        SumNumberEmployees := SUM(NumberOfEmployees);
        TurnoverPerEmployee := SUM(Turnover)/ SUM(NumberOfEmployees);
ENDAGGREGATE
```

Figure 1 Example of an aggregate definition in MacroView

B. Transformations of data

12. Another commonly used functionality is the so-called transformation. In transformations micro-data or aggregated data can be linked to each other (using different types of joins). Figure 2 shows an example of a part of a MacroView script in which micro-data for the years T, T-1 (Tm1), and T-2 (Tm2) are linked. Another feature of transformations is that a broad range of mathematical calculations can be performed on the input data. This implies that new variables can be calculated, and values of existing variables can be changed. This can - to a certain extent - also be achieved by the "aggregate" functionality. The difference between aggregates and transformations is however that in transformations input and output data have the same level of aggregation, while output data in aggregates are - by definition - at a higher aggregation level than the input data used.

```
{*****************************************************************************}
{-- [Transforms] Do not remove! --}
{*****************************************************************************}
TRANSFORM TRAFO_LINK
INPUT
    Year_T : MODEL_Micro
    Year_Tm1 : MODEL_Micro
    Year_Tm2 : MODEL_Micro
OUTPUT
    Linked_data : MODEL_Linked_Data
RULES
    Turnover_t := Year_T.Turnover;
    Turnover_tm1 := Year_Tm1.Turnover;
    Turnover_tm2 := Year_Tm2.Turnover;
    Turnover_per_Employee_t := Year_T.Turnover\ Year_T.Nr_Employees_t;
    Turnover_per_Employee_tm1 := Year_Tm1.Turnover\ Year_Tm1.Nr_Employees_t;
    Turnover_per_Employee_tm2 := Year_Tm2.Turnover\ Year_Tm2.Nr_Employees_t;
ENDTRANSFORM
```

Figure 2 Example of a transformation definition in MacroView

C. Functions for drawing grids and several types of plots

13. Micro-data or aggregated data can be shown to the user in a grid or a plot. Figure 3 illustrates some examples of plots defined in the script for the Short term Statistics.

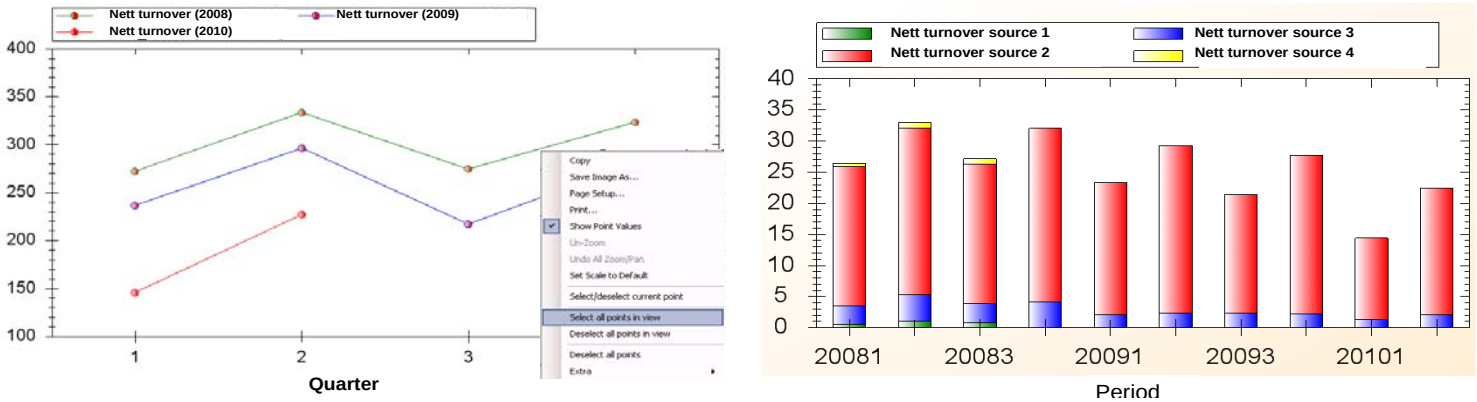

Figure 3 Examples of plots in MacroView

The plot on the left shows the nett turnover per quarter for several years. The years are drawn using separate lines, to facilitate a comparison between equivalent quarters of different years as well as an analysis of seasonal effects within the years. The plot on the right shows for several quarters the total nett turnover (height of the bar). As this turnover is obtained from a number of sources, the bars are divided in

different parts (every part is colored differently) where every part represents the part of the turnover obtained from a specific source. More types of plotting functionalities than are shown here are possible in MacroView. If a certain type of plot is not supported by MacroView, it can always be created via plug-ins, for instance, for R-packages.

14. For grids also several visualisation options can be chosen in the script language. The left part of Figure 4 shows an example of a grid in which for every editing cell the contribution to the total turnover of the SBI 2 digit aggregate is represented by a horizontal bar. In this way it is immediately clear which cells influence the total turnover of the SBI 2 digit the most. The example on the right of Figure 4 illustrates the feature to give cells of the grid a colored shade depending on the value. In this example, the higher the absolute value of the value in the cell, the more red the shade becomes. This option is useful when a macro-editing tool needs to show a lot of values and the user of the tool has to get a quick impression of the values needing special attention.

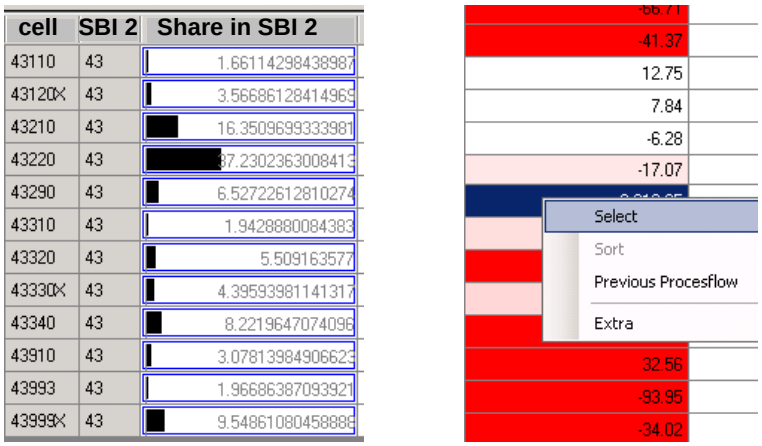

Figure 4 Examples of features available for grids

D. Dynamic filters for selecting and analyzing subsets of data

15. Figure 3 and Figure 4 also show “selection menus”. A user of a tool built in MacroView can select a subset of data with these menus. It is possible to design the script such that a selection made by a user with these menus is applied to visualisations opening subsequently.

16. An application of this functionality is illustrated in more detail in Figure 5. In this macro-editing tool at first only the screen on the left opens. After the user of the tool selects SBI ‘7222’, the table on the right appears. This table gives an overview of the size classes corresponding to the selected SBI; another possibility is to show all underlying micro-data. The exact action after selection is defined in the script according to the needs of the user.

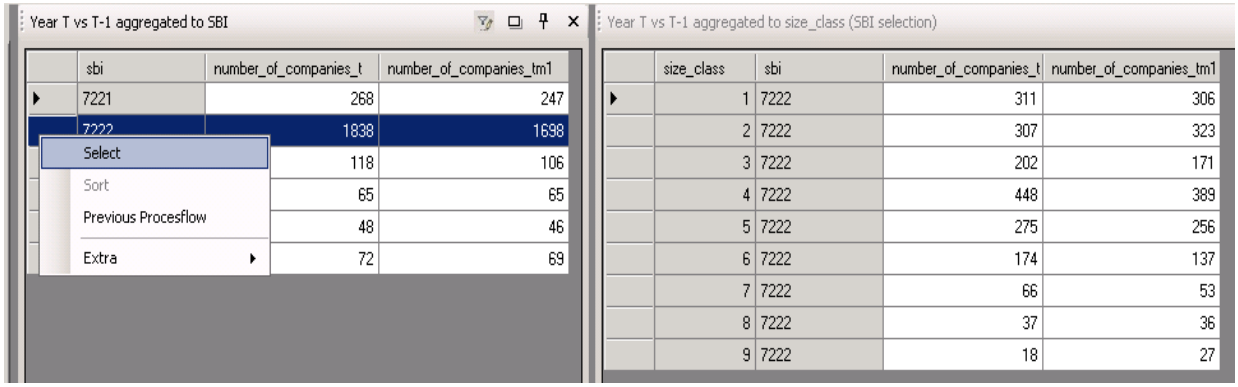

Figure 5 Example of how a selection made by a user can be applied as filter in subsequent screens

E. Efficient recalculation of all tables and plots shown after micro-data has changed

17. The functionalities discussed above are the buildings blocks of macro-editing tools developed in MacroView. How the blocks are actually used is specified in a script. Once a tool is developed and in production an analyst might (after zooming in from several aggregation levels) detect a significant error

in the micro-data. In order to be able to correct this error, a data-editor for micro-data needs to be linked to the macro-editing tool by specifying it in the script. The script can then be designed such that the editor can be opened in the macro-editing tool. Opening the editor in the tool can be done via the selection menus shown before. For example, when a grid is opened showing values at the company level (the micro-level), a row can be selected via the menu; in the script it can be specified that this action causes the editor to open with data from the selected company.

18. After a user has changed one or more values of the micro-data using such an external micro-data editor, control is given back to the macro-editing tool. In the script it can be specified that in this situation all screens that are open in the tool are updated in such a manner that they become consistent with the new value(s) for the micro-data. For the sake of efficiency, this update functionality is developed in such a way in MacroView that only screens affected by the change are updated.

19. In this chapter we have given an impression of the functionalities currently included in MacroView. Given these functionalities a new challenge arises: How to use these "building blocks" to construct a macro-editing tool that (1) shows important variables in a well-organized way to a user, (2) can efficiently be built and maintained, and (3) has a good performance. This challenge is discussed in the next chapters using two (real-life) examples.

IV. DEALING WITH A LARGE NUMBER OF VARIABLES IN A MACRO-EDITING TOOL

20. The main challenge in developing a macro-editing tool for the Structural Business Statistics was to present a huge number of variables at several different aggregation levels to the analyst in a well-organized way. A further complication was that the variables of interest differ between industries. The variable "Airportcosts" is, for example, only relevant for companies working in the air transport sector. It is useless for companies active in retail trade and even for companies active in public transport.

21. In the remainder of this section we will address these problems. We will conclude this section by explaining which measures we took to make sure that the tool could be efficiently built and maintained.

A. Showing a large number of variables at different aggregation levels in a well-organized way

22. An important challenge in building a macro-editing tool for the Structural Business Statistics was to find a way for showing a large number of variables at different aggregation levels to users of the tool in an understandable way. A first important step in achieving this was to determine for every variable to which industries or parts of industries it applied. More specific, a list was made of all "generic variables", "industry-specific variables" and "variables for a part of an industry". As we knew that users of the tool would only be focusing on one industry or even on a part of an industry, we could use this list to make sure that users will only be able to see variables that are of interest to the industry they have to analyze.

23. We also considered relations between variables. This gave the important insight that variable definitions do contain several implicit aggregation levels. In other words, there are "general variables" that can be split in more detailed variables. For example, the variable "overall expenses" covers the variables: "Purchasing costs", "Personnel expenses", and so on. The "purchasing costs" can again be split in: "Purchasing costs of raw materials and auxiliary material", "Purchasing costs of traded goods", and so on. Given this we designed flow charts like the one shown in Figure 6. In these flow charts every rectangle corresponds to a screen in the macro-editing tool under design. The arrows between the rectangles show which new screen is opened when (a specific part of) a screen is selected. Figure 6 shows, for example, that at first two screens open: one showing the overall operating expenses and (below this screen) one showing the purchasing costs, personnel expenses and so on. By clicking on the barplot referring to the purchasing costs a user can open a screen showing more detailed variables regarding these specific costs. The macro-editing idea behind this approach is that an analyst only needs to analyse very detailed variables when necessary, for example due to implausible results for more "general variables". Given such an implausible aggregated result the detailed variables need to be considered for efficiently determining the underlying cause.

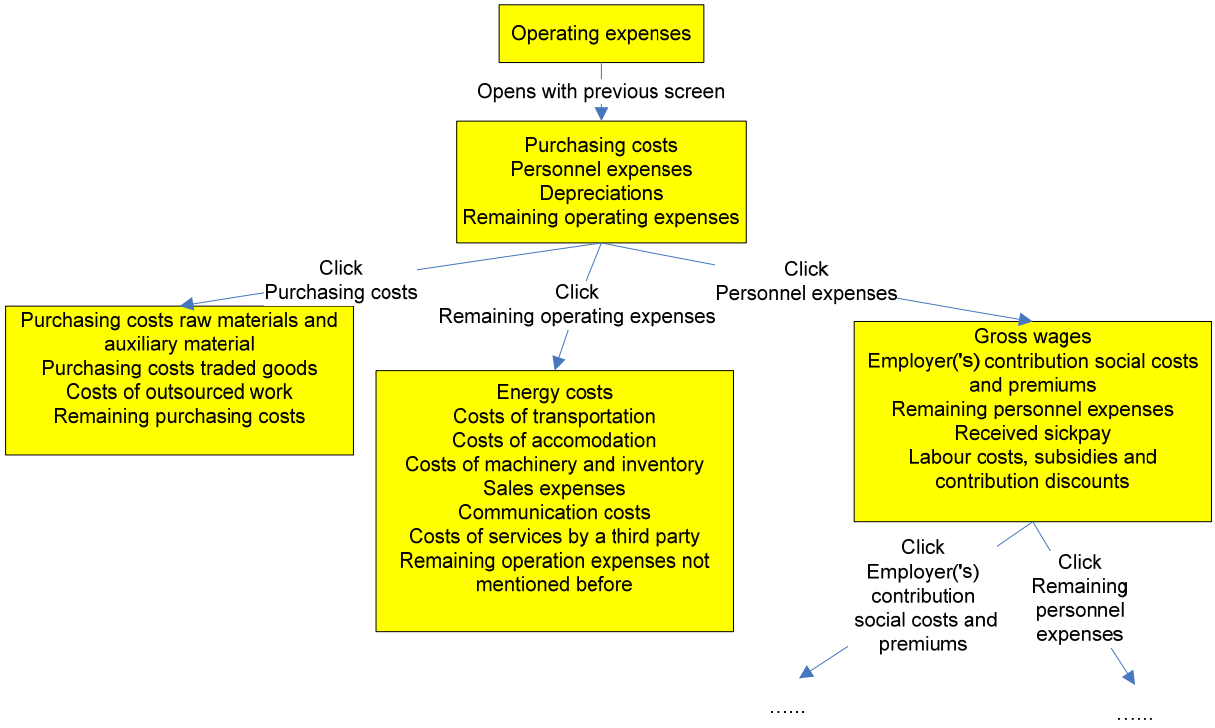

Figure 6 Example of how flow charts were made, which show how more general variables can be split in more detailed variables

Figure 7 shows an example of the actual screen that was built based on the upper two yellow rectangles shown in Figure 6.

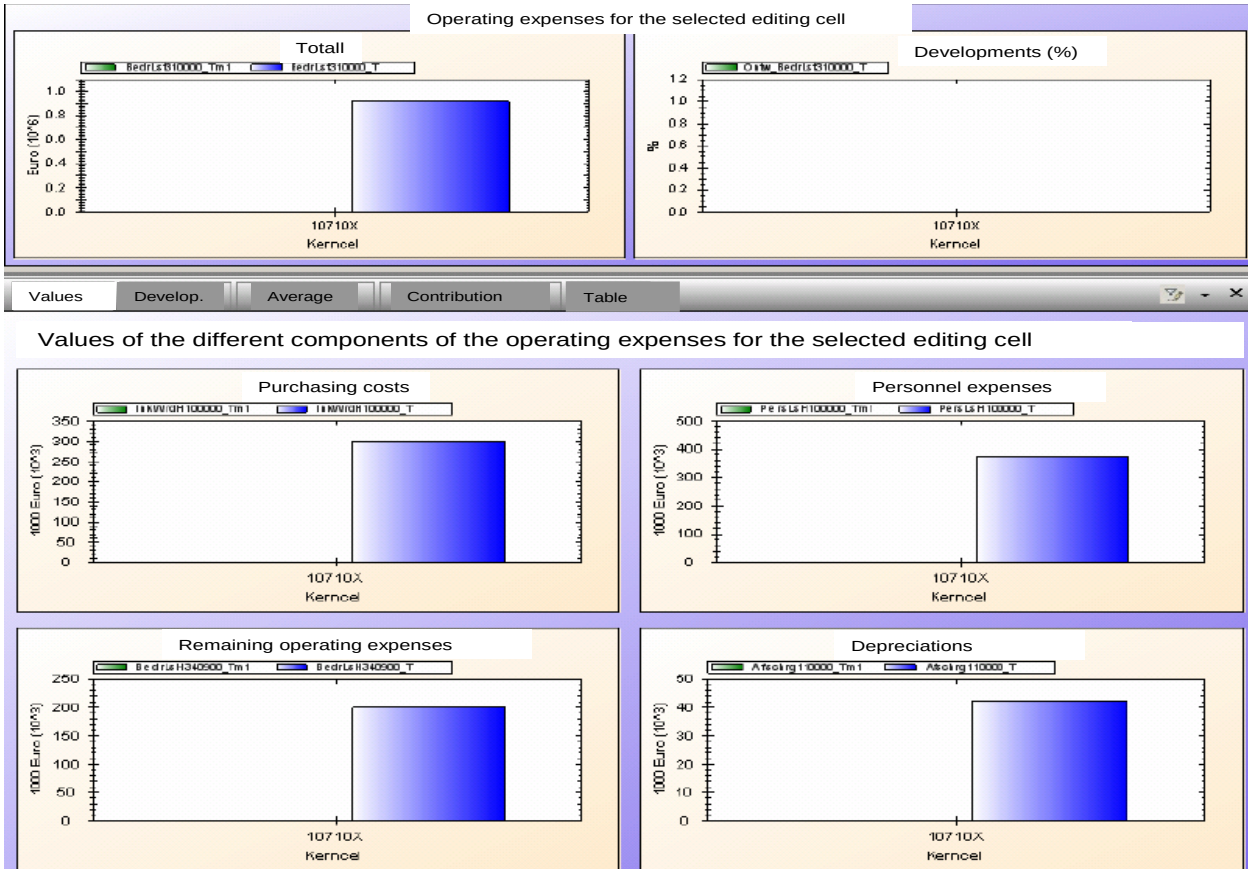

Figure 7 Example of a screen in the macro-editing tool built based on the flow chart in Figure 6

24. This example shows the “total operating expenses” at the top. At the lower part of the screen the “general variable” is split in the related more detailed variables. For these more detailed variables the following information is provided in different tabs:

- The sum over all companies belonging to the selected editing cell[2] (for all years for which data are available)
- The change of the sum compared to last year (for all years for which data are available) in percentages.
- The average value per company (for all years for which data are available)
- The contribution of the value of the “detailed variable” in the value of the “general variable” (for all years for which data are available)
- A table containing all previously mentioned values (for two consecutive years)

25. By clicking on these screens new screens can be opened that show even more detailed variables (see Figure 6). By clicking on a screen, (also) new screens can be opened showing the same variables but at a lower level of aggregation. For example, the screens on “editing cell level” can be split in “editing cell level x SBI 5 digit” and “editing cell level x size class”.

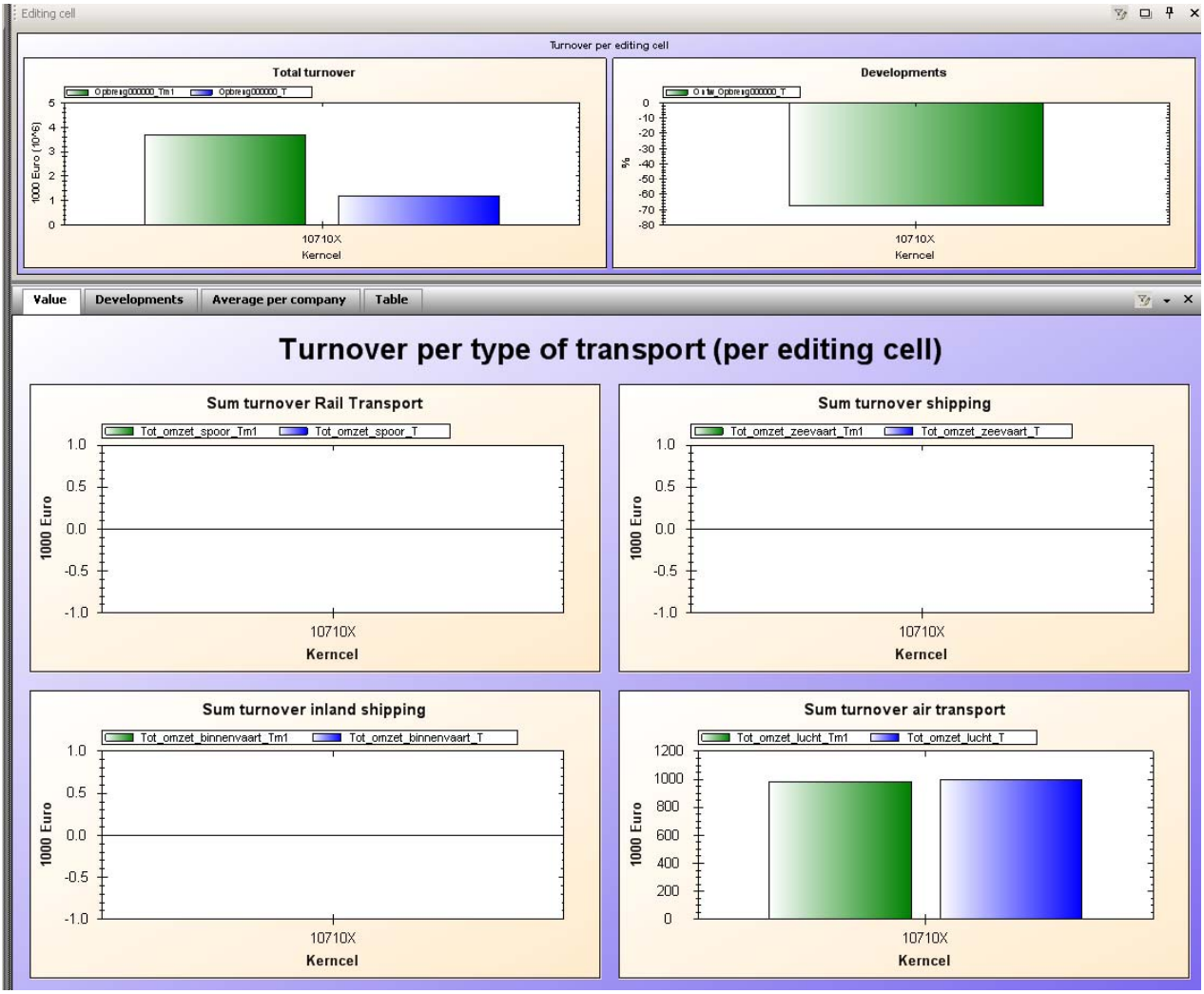

Figure 8 Example of a screen that can be used to open screens showing variables only relevant for a part of an industry

26. As this approach implicitly results in the creation of a relatively large number of screens, we tried to limit the number of variables shown. An important step in doing so was to make sure that users are not bothered by variables not of interest to the (part of an) industry they are analyzing. A complication

[2] An editing cell contains a group of companies being analyzed together in the editing process.

regarding the variables that are only relevant for a part of an industry was that it was decided that only one tool per industry would be built. To be able to only show the “screens for a part of an industry” when relevant, we designed the industry-specific tools in such a manner that these screens open only when an analyst clicks on a plot showing a dedicated variable relevant only for that specific part of the industry, for example the total turnover of the airport industry. After a click on a plot in which this variable is shown all screens for air transport open in the transport industry tool. An example of this plot that can be used to open screens relating to air transport is shown in Figure 8.

27. To all screens showing aggregates we added the possibility to add comments. Sometimes results that are implausible at first glance can turn out to be correct after a thorough analysis. In this case the analyst can add a comment stating, for example, that the strange value turned out to be correct and why it turned out to be correct. The comments are also shown to other analysts working at the same data.

28. A ‘strange’ value can of course also turn out to be wrong after considering, for example, the underlying micro-data. To be able to view the micro-data we made it possible to open grids showing these data once the user of the tool reached the “editing cell level x SBI 5 digit” or “editing cell level x size class” aggregation level of a screen. For example, when a user selects a bar referring to size class 8 in a barplot in a screen at the “editing cell x size class” level, a grid opens showing data on all companies belonging to the selected editing cell and size class 8. The reasoning behind showing only a selection of micro-data is that in this way a user is not bothered by companies not belonging to the editing cell or size class he or she is analyzing at that moment. To every grid showing data at the micro-level we added a link to the external micro-editor. We also designed the script underlying the tool such that the “update functionality” of MacroView would be called after the micro-editor reported a change in the micro-data. As a result of this a user of the resulting macro-editing tool can open the micro-editor by selecting a company in the grid and edit the data of this specific company. Once editing is finished, all aggregates are recalculated and all open screens are updated in such a manner that they become consistent with the adjusted micro-data.

B. Efficiency in building and maintaining the industry-specific tools

29. Since different industry sectors needed a tool showing different variables, special attention had to be given to efficiency. This efficiency relates both to the effort needed for building the scripts underlying the tools as well as the effort needed for maintaining the scripts once the resulting tools would be in production. For example, in building the tools, it would be efficient when scripts specifying screens showing variables relevant for more than one industry were developed only once.

30. In practice first a MacroView script was written for a “generic macro-editing tool for the Structural Business Statistics”. This generic tool contains all screens that are relevant for (almost) all industries. This “generic tool” functions as a solid base for the industry-specific tools. To implement the industry-specific tools we used the “include” functionality of MacroView. This allows developing scripts for industry-specific parts of a tool in separate files and to refer to them in the “generic tool”. By referring to these files using the “include” functionality, MacroView treats the parts of the scripts specified in separate files as part of the main script of the generic tool. Hence the scripts of all industry-specific tools consist of the “generic script” plus some industry-specific includes.

V. DEALING WITH A LARGE NUMBER OF RECORDS IN A MACRO-EDITING TOOL

31. In developing a macro-editing tool for the Short Term Statistics the main challenge was the large number of records included in the input tables. The number of records is large because the tool needs to be able to show micro-data for almost all companies in the Netherlands. For a lot of these companies the micro-data are derived from VAT information. The micro-data for a company in a specific analysis period can even be composed of several ‘VAT-records’. As it also has to be possible to show these underlying “VAT-records” in the macro-editing tool, all VAT information needs to serve as input to the tool.

32. The analysis in a macro-editing tool starts by showing aggregated data. Micro-data are only shown when necessary. A user of the tool performing an analysis wants the aggregates to be shown

quickly. Thus, when a user selects a specific editing cell, he or she expects that aggregated data are shown for the selected editing cell immediately. When he or she subsequently selects another editing cell, aggregates have to be shown for that editing cell quickly. In order to be able to fulfil these requirements given the large amount of input data, we considered two possibilities:

- All required aggregates are defined in the script underlying the tool using the standard functionalities of MacroView (see Figure 1 for an example of an aggregate definition in MacroView). All micro-data serve as input to these aggregates. This has the advantage that it is possible to define the MacroView scripts such that MacroView can recalculate aggregates and update screens when micro-data are changed using an external micro-editor. The disadvantage of this approach is that calculating aggregates based on millions of records requires time, implying that a user of the resulting tool has to wait until all desired aggregates are calculated. As tools developed in MacroView always perform these calculations every time a script is run, the same calculations are performed every time the user runs the script. Even when in most cases no change in the underlying micro-data has occurred. This is not a problem in the tool for the Structural Business Statistics discussed before as these calculations are performed so quickly that the user not even notices it. But given the large number of micro-records serving as input for the tool for the Short term Statistics, we considered a more efficient option.
- This more efficient option was that several aggregates are calculated outside the main tool and serve as input to the tool. Thus the script only specifies in which database tables the aggregates can be found and that these tables have to be read before MacroView functionalities can be applied to them. Using this approach a script can be designed resulting in a tool showing aggregates and giving the user the option to make selections. Also the micro-data can serve as input to the tool. The script can be designed in such a way that the user can open a grid with a selection of these micro-data when he or she observes implausible aggregates. Given that these grids only show selections (like in the tool for the Structural Business Statistics) dynamic filters can be applied to the micro-data such that these grids open without delay after a click of the user. In fact, the script can be designed such that the only difference for a user not changing micro-data is that plots and grids showing aggregates open more quickly. The disadvantage of this approach is of course that MacroView can not recalculate the aggregates after a change in the micro-data. The obvious reason for this is that the formulas needed for calculating these aggregates are not incorporated in the MacroView script. As a result, the input files containing the aggregates need to be updated outside MacroView after which MacroView gets a trigger to read the input tables again and to subsequently update its screens.

33. The latter option was chosen. Figure 9 shows a screenshot of the tool that is currently being developed. In the left part of the screen a “tree” is shown, containing references to all screens currently available in the macro-editing tool for the Short term Statistics. By clicking on a reference the user can open the corresponding screen. This means that a user can switch continuously between all available screens. The way in which the tree is organized also stresses the “macro-editing” concept, i.e. the more detailed the information the more the reference is nested down the tree.

VI. CONCLUSIONS AND FUTURE WORK

34. In this paper a short summary of the functionalities already available in MacroView has been given. These functionalities can be seen as the generic building blocks for all macro-editing tools developed in MacroView. Every tool consists of a script specifying, for example, how these separate building blocks are combined, and which data need to be used. In this way the generic building blocks can be used to build custom-tailored macro-editing tools. Also two practical examples were given of macro-editing tools build in MacroView. In discussing these examples we considered the challenge of presenting a large number of variables to a user in a well organized way, as well as the challenge of handling a large number of records at the micro-level.

35. Future work will consist of using MacroView for building macro-editing tools for other statistics than the ones mentioned in this paper. Next to that we will be analyzing how the existing tools are

currently used by analysts. The insights obtained from these analyses can be used to improve existing tools further and to develop future tools.

36. In the remainder of this year we will also finish the (English) documentation describing all functionalities of MacroView in detail and giving examples of scripts in which these functionalities are combined into a macro-editing tool. Also a MacroView course will be given such that more and more people become able to build and maintain macro-editing tools in MacroView.

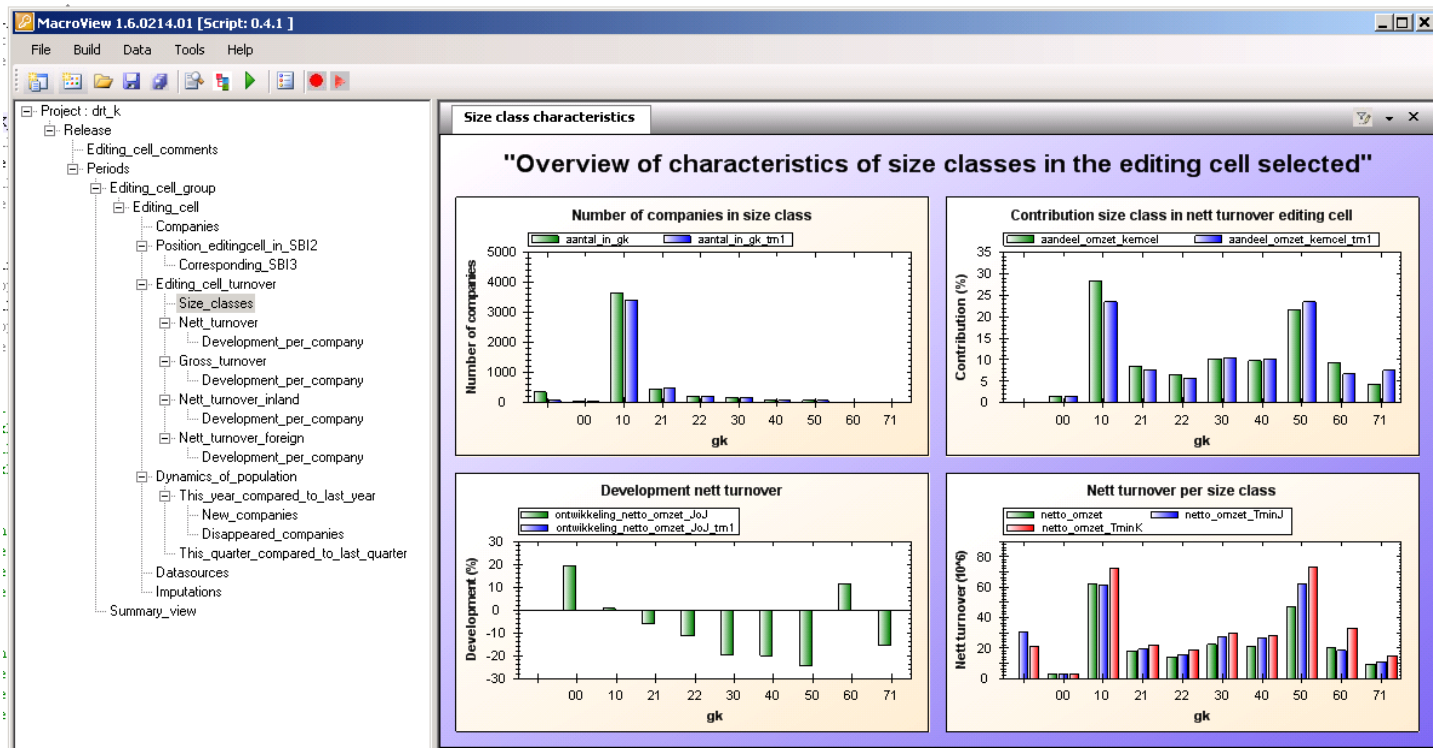

Figure 9 Screenshot of the tool that is currently developed for the Short term Statistics

VII. REFERENCES

Braaksma, B. (2007), *Redesign of the chain of economic statistics in the Netherlands*, Seminar on Registers in Statistics - methodology and quality, Helsinki, Finland

Granquist, L. (1994), *Macro editing: A Review of some Methods for Rationalizing the Editing of Survey Data. Statistical Data Editing*, Volume No. 1: Methods and Techniques

Hacking, W.J.G, and Ossen, S.J.L (2011), *Applying Macro Editing in MacroView*, NTTS Conference

De Waal, T. and Haziza, D. (2009), Statistical editing. Handbook of Statistics, Volume 29, *Sample Surveys: Theory, Methods and Inference*, Editors: C.R. Rao and D. Pfeffermann