Use of Blaise and Manipula in the Annual Survey of Employment and Wages

Leo van Toor, Statistics Netherlands

1. Introduction

This paper describes the use of Blaise and Manipula in the Annual Survey of Employment and Wages of Statistics Netherlands. The survey covers approximately 75,000 establishments and 1,000,000 employees. There are two questionnaires. For each establishment there is one Employment questionnaire. This questionnaire consist of 10 questions (e.g. number of male and female employees, and for each municipality of business the number of employees). The second questionnaire is the Wage questionnaire. This questionnaire has 28 questions at the employee level (e.g. year of

birth, sex, and wage). Forms must be filled in for a sample of employees.

One of the objectives of the survey is to determine accurate statistics on the development of the number of employees (per municipality), and the development of the average wages. Therefore, it is important to carry out checks comparing figures from the current and the past year.

Since the two types of questionnaires have different return dates, two separate survey processing systems were built: an Employment and a Wage system. The global structure of both systems is approximately the same. Therefore, they are described in this paper as if they are one system.

To reduce maintenance problems as much as possible, it was decided to build the systems using standard software components, Blaise 2.5 being one of them. The developers of the system were confronted with two major problems:

1) Due the use of many external files, and extensive computations in the check section, Blaise ran into memory problems,

2) Many establishments preferred sending statistical information on tape or diskette over having to fill in paper forms.

At the same time, the Department changed its ideas about data editing. Instead of spending many resources on cleaning up each form, its was felt that editing should concentrate on only correcting errors having an impact on the quality of the published statistics. Furthermore, wherever possible, automatic editing should be implemented. In this way, human involvement in data editing activities could be reduced without it having a negative effect on data quality.

In short, the main objective for redesigning the system was faster survey processing with less people, thus improving timeliness and reducing cost.

The consequence of this approach was less focus on interactive, form oriented procedures, and more emphasis on batch-wise, file oriented procedures. Manipula was selected as the standard tool for these batch procedures. In this way, Manipula became the central part of the new survey processing system.

For batch systems, it is important to have progress and status reports. So-called 'system report files' must be produced containing information about things like execution time of modules, and number of processed records. It is important for the 'system manager' to analyse these reports before taking any further action.

The ideas about automatic editing emerged after the survey processing system was almost ready. Fortunately, the modular and flexible architecture of the system made is possible to include automatic editing modules, without getting problems with the other, already present modules.

The survey processing system had to produce a large number of tables. The size of these tables, the type calculations required, and the nature and amount of subtotals, made it impossible to use Abacus. For example, Abacus can calculate totals and averages, but not the ratio of two totals, like the average wage per hour (sum of wages divided by sum of hours). Also the table-size was beyond the limitations of Abacus. Manipula turned out to be the way out. For tables, Manipula is not as user-friendly and fast as Abacus. However the powerful Manipula language can be used to define standard setups that can be applied in many situations.

This paper describes the most important modules of the survey processing system. Not all details are mentioned, since that would ask too much knowledge of the Dutch situation. The most interesting aspects of the system are:

- Only standard-components were used: Manipula and Blaise,
- All components were tied together in MS-DOS batch files,
- The system is controlled through NOVELL-menus,
- Every module in the batch system produces a system report file,
- Manipula is used for checking the data,
- Tables are produced with Manipula using standard setup,
- An automatic editing module is implemented.

The survey processing system is described in more detail in the subsequent sections. Section 2 gives an overview of the system, from data collection to tabulation. Section 3 handles the system report files. Section 4 gives more details about the implementation of the auto-edit module. Section 5 describes the use of Manipula for tabulation. Section 6 gives some concluding remarks. More information about the details of the survey processing systems can be obtained by contacting the author.

2. Overview of the system

This section gives an overview of the various modules in the system. Where possible, the description of the modules is given in the order in which they are activated in the system.

2.1. Data collection

There are two sources of data: paper forms on the one hand, and tapes/diskettes on the other. The data on the paper forms are entered with a Blaise CADI program. The work is done by data typists with little knowledge of the subject-matter of the survey. Only simple checks are carried out by the CADI program. There are range checks, and checks that consult external files to determine the existence the establishment identification number, and the municipality code. The reasons to limit the editing in this stage of the process, are

- The data-entry process is faster;
- Focus is on errors having an impact on data quality, and checks for such errors have to be carried at a later stage,
- The ambition to use as much automatic editing as possible,
- The difference in knowledge and experience between data typists and subject-matter analysts.

The tapes or diskettes containing data are translated into ASCII subfiles with Manipula, after which these ASCII-subfiles are converted and added to the Blaise data file with the Blaise conversion tool Convert.

After all paper forms, tapes, and diskettes have been processed, the result is a raw data file. The next step is data editing.

2.2. Data checking, phase 1

The raw Blaise data file is checked by the same Blaise CADI program as mentioned in section 2.1. The data structure is the same as that of the data entry program. The CADI program is run in batch mode (with the command line parameter /C), so an integral check is performed on the data file. As a result the status parameter of each form is set either to *CLEAN* (no error), *SUSPECT* (only soft errors), or *DIRTY* (hard errors).

The data model underlying the Blaise survey specification, is an hierarchical model with three levels:

- The first, and highest level contains information at the level of the establishment,
- The second level contains information at the level of the establishment in a specific municipality,

- The third, and lowest, level is the employee level.

To be able to handle the information at the various levels, the Blaise data file has a subfile-structure corresponding to these levels.

The Blaise conversion tool Convert is used to export the clean records to ASCII files, and to delete this records from the Blaise data file. The dirty records remain in the Blaise data file. Convert is more suitable for this job than Manipula, because Convert it able to keep the subfile structure. Furthermore, it is much easier to use Convert to delete the clean records from the Blaise data file.

The disk space allocated to the deleted records is not released by Convert. As a consequence, the file size is not reduced by this operation. The Blaise tool Formman is used to clean up the Blaise data file.

The Blaise data file with the remaining dirty records is processed by the data-typists using the Blaise CADI program that was used to enter the data. Of course, no data is entered at this stage, but available data is corrected.

The clean records go to the next step in the survey process, and that is the computation of aggregated records.

The aggregation step consists of using the employee level data to compute figures at the level of the establishment in the municipality. Examples of aggregated figures are

- Number of male and female employees, and the total number of employees per municipality,
- Number of employees per age group,
- Average wage per hour, and total of wages per year.

Furthermore, these figures are linked to the corresponding figures from the previous year.

Aggregating and linking is carried out in a single Manipula run. Also, a number of checks are carried out in this run. The checks can be divided in two groups. The first group of checks deals with the data from the previous year. Examples of these so-called level 1 checks are:

- The sum of the number of male and female employees must be equal to the total number of employees,
- The sums of the numbers of employees per municipality must be equal to the total number of employees.

The second group of checks compares data from the two years. Examples are:

- The change in the number of employees must be within certain bounds,
- The ratio of the current average wage and the previous average wage must be within certain bounds.

The Manipula setup takes advantage of the possibility to create more than one output file. One output file contains the linked figures of the current and the previous year, and the other output file lists the error codes per establishment.

At a later date, also the auto-edit module was included in this setup. The auto-edit module exist of separate Manipula-setups. More details can be found in section 4.

Other Manipula setups were used to link the data records in the different ASCII sub-files with the corresponding error codes. The linked records (i.e. records with one or more error codes) are written to the suspect-data file, the other (not linked) records were written and added to the clean data file, which is a file of the format INDEX. The suspect data records are converted from ASCII to Blaise with Convert, and must be handled by a Blaise data editing program. This program differs from the Blaise program mentioned in section 2.1.

2.3. Editing current year’s data.

Current year's data is edited when the comparison with previous year show too large changes. Analysts use another (the second) Blaise CADI program for these editing activities. The previous year's data is made available through external files. The analysts work through the file with forms that have been marked as suspect by Manipula. The problem description produced by Manipula, is also available in the Blaise program. The analysts have three possibilities to solve the problems:

1) Edit the record, i.e. change a value in a current year’s field,

2) Enter an ‘OK code’, indicating that nothing is wrong. In this case, they also have to enter an explanation,

3) Enter a code indicating the previous year must be corrected.

2.4. Data checking, phase 2

The people handling this phase of the system, are almost the same as those mentioned in section 2.2. (data checking, phase 1). The differences between these two systems are:

1) Now it must be taken into account that the data from the previous year can be changed (section 2.5),

2) It may now happen that the an establishment is suspect, but the analyst has given it an ‘OK code’ and ‘error explanation’. This concerns changes in figures between years, and not consistency checks within a form,

3) The analyst may now want to edit records from the previous year.

All these aspects are taken care of in phase 2 of data checking.

In integral check is carried out on all records (with the command line parameter /C). The same Blaise program is used as in section 2.1. The reason to do this integral check is to be sure there no accidental errors are introduced during the data editing process (e.g. value range errors). To make this integral check possible, extra variables (OK_CODE, EXPLAN) are defined in the Blaise program. These variables have the attribute HIDDEN. In this way it is possible to do an integral check on other Blaise files with the same program.

The ‘clean’ records are converted to ASCII and deleted from the Blaise file. The ‘dirty’ records remain in the Blaise file. This is all done with Convert. Normally there are no records left in the Blaise-file after this phase.

Manipula is used to select the records with the code ‘CORRECTION PREVIOUS YEAR’ from the data file of the previous year. Then Convert is used to convert these records to a Blaise file. The treatment of these records is described in sections 2.5 and 2.6.

The Manipula setup of section 2.2 is used to carry out calculations, and to link the figures. The only differences are that previous year’s records may have been edited (see section 2.6), and that the remaining suspected data must be handled again with the data editing program (for the current year). Switches in the Manipula setups take care of the different use of these setups.

2.5. Editing the data of the previous year

Comparing figures of the current with those of the previous year is only useful if it can be assumed that the figures from the previous year are correct. So, where possible, checks are carried out on previous year's data, and where errors are detected, they are corrected. The selected and converted data from the previous year are edited by analysts with a Blaise-program (the third). Note that the previous year's data is not confronted with the data of yet an earlier year.

2.6. Data checking, phase 3

After the data from the previous year has been edited, an integral check is carried out on the forms to make sure the status of each form is updated. The program Convert is used to convert ‘clean’ records into ASCII, and to delete these records from the Blaise file. The ‘dirty’ records remain in the Blaise file. This remaining Blaise-file is cleaned up with Formman, and then handled again with the data editing program (of section 2.5). The ‘clean’ ASCII records are translated into the a Manipula INDEX file. Then the data checking phase 2 module is used for calculating previous year’s figures.

The phase 2 data checking module uses the corrected previous year records instead of the original records. The advantage of this method is that by making a copy record and editing this copy record, it is afterwards possible to see which records, and how many records from the previous year were corrected. It is also possible to undo corrections, if necessary.

The process mentioned in the sections 2.1 to 2.6 is repeated as many times as necessary to obtain a sufficient amount of response to tabulate.

The activities in the sections 2.2, 2.4 and 2.6 are managed by the ‘SYSTEM MANAGER’. He activates systems, and judges their progress. He has a sufficient amount of subject-matter knowledge to be able to make informed decisions.

2.7. Weighting

After the clean data file is ready, weights have to be computed in order to make the sample representative for the population. A file with population frequencies of a number of auxiliary variables is available for this purpose. These calculations were made with Manipula.

Also some data manipulations are carried out during this phase. For example, ISIC codes are attached to establishments, and birth dates are transformed into ages.

A new data file is created, the so-called the standard file. This file forms the basis for computing population estimates.

2.8. Macro checking

The estimates based on the standard file are still not ready for publication. First, preliminary estimates are computed. These estimates are used to carry out some checks at the macro level.

Manipula is used to compute estimates of population statistics for the current and the previous year. Examples of these statistics are the number of employees and average wage per hour in each ‘cell’, where a cell is a combination sex, age and economic activity.

The computed statistics of the current and previous year are brought together in one file, after which Manipula computes changes. Extreme changes are marked as being suspect. In this case, all relevant data about the cell are written to a print file.

Every extreme change has his own ‘suspect code’. The cell data and the suspect codes are written to a second output file. These suspect codes (on the cell level) corresponds with one or more error codes, which are assigned on the level of the establishment. The link between these errors and the suspect codes is recorded in an ‘error relation file’.

The ‘error relation file’ is used to select all records with an error code corresponding to a suspected cell in the ‘clean’ data file. With Manipula, the selected records are deleted from the ‘clean’ data file.

The data editing and data checking processes that follows, has already been described in sections 2.2 to 2.6.

3. System report files

All modules in the batch systems generate system report files. These files contain important information about the progress of the process. The information should help in solving problems that may occur. Each system report file contains information about the time the module was started, the execution time, and the number of processed records.

In the following example, the system report file has the standard name DAY.PRN, and it is controlled by an MS-DOS batch-file. The general structure is displayed in figure 3.1.

Figure 3.1. General structure of the system report file

```
ECHO START SYSTEM: xx  > DAY.PRN
SYSTIME               >> DAY.PRN
MAP n:=.....          >> DAY.PRN
MAP INS s1:=.....     >> DAY.PRN

     ECHO information >> DAY.PRN

     MANIPULA ..... D=DAY.PRN

     CONVERT  PF=n:x.cvt
     COPY DAY.PRN + n:x.DAY
     DEL n:x.DAY

     FORMMAN  PF=n:y.frm
     COPY DAY.PRN + y.DAY
     DEL y.DAY

     NPRINT   ..... >> DAY.PRN

     COPY x + y z   >> DAY.PRN

MAP DEL n:          >> DAY.PRN
MAP DEL s1:         >> DAY.PRN
SYSTIME             >> DAY.PRN
ECHO END SYSTEM: xx >> DAY.PRN
NPRINT DAY.PRN
```

The user has no complete control over the names of the files produced by Convert and Formman. Therefore, MS-DOS batch commands are required to add information generated by these tools to the system report file. Unfortunately, the integral check with CADI does not produce any process information at all. If necessary, such information can be generated with Manipula.

4. An example of auto-editing.

Analysts correct many small errors without contacting the establishments concerned. If errors can be corrected in such a relative simple way, it is worth while considering doing the corrections automatically. Moreover, many of these corrections do not seem to have a substantial impact on the accuracy of the produced statistics.

To be certain that automatic editing does not make things worse, the results have to checked. The general rule is: after auto-editing, there must be less errors than before. If this is not the case, the auto-editing must be cancelled. Such a rule can be tested with existing modules in the survey processing system. The module(s) determining the error codes is carried out twice, one before, and once after auto-editing. By comparing the results a decision can be taken whether to accept or reject the auto-edit.

Here is an example of an auto-edit. If the difference between the total number of employees and the sum of male and female employees is less than 10, an auto-edit is carried out. The Manipula specification of this edit is shown in figure 4.1.

Figure 4.1. Example of an auto-edit in Manipula

```
IF (errorcode = '01') AND ABS(Male + Female - Employees) < 10 THEN
   Male:= ROUND(Employees * (Male / (Male + Female)));
   Female:= Employees - Male;
ENDIF
```

5. A standard method for tabulation with Manipula.

The survey results consist of a large amount of tables. These tables are made by Manipula. To avoid having to develop and maintain a large number of Manipula setups, the decision was taken to standardise the setups as much as possible. For each type of table, a standard setup was developed and stored in a separate file. This approach to tabulation has a number of advantages:

- Less development efforts,
- Less maintenance efforts,
- Easier administration of the tables,
- Easier to work with by the people of the Department.

The following elements of the table specifications are stored in a standard way:

- Record descriptions,
- Calculations,
- Coding files,
- Subtotal files,
- Text files.

One of the most interesting elements is the way to make complex sub-totals. Suppose, the input file contains the variable age (with a range from 16 to 64 year, in single year age classes), and the output must contain sub-totals for the following new age classes: 16-64, 16-18, 19-20, 16-20, 21-22, 23-24, 21-24, 25-29, 30-60, 61-62, 63-64, 60-64, 23-64, 21-64. Then, two actions must be undertaken:

1) Create a file with all the codes for the classes in the input file (16 to 64). Let each code be followed by a code representing the (aggregated) class in the output file. Here is an example of a small part of such a file:

```
16 010300
17 010300
18 010300
  .
  .
60 0811121300
61 0911121300
62 0911121300
63 1011121300
64 1011121300
```

2) Make a Manipula setup with two input files, the first input file is the data file, and the second input file concerns the input and output class codes. The manipulate section of the

Manipulate setup uses a duplicate instruction. After duplication, the records must be sorted and summed in a sort section. Here is an example of such a Manipula setup:

```
INPUTFILE "DATA"
           Age               1  STRING[2]
           No_of_Employees   3  INTEGER[7]

        INPUTFILE "DUPL" LIST
           Age               1  STRING[2]
           Age_sub           4  ARRAY[1..5] OF STRING[2]

        OUTPUTFILE
           Age               1  STRING[2]
           No_of_Employees   3  INTEGER[7]
        VAR
           loop : INTEGER;
        MANIPULATE
           FOR loop:=1 TO 5 DO
              IF Age_sub[loop]<>'  '
                 THEN Age:=Age_sub[loop]
                      DUPLICATE;
              ENDIF;
           ENDDOE;
        SORT
           Age;
           No_of_Employees, SUM;
```

6. Conclusion

Five years ago, the Annual Survey of Employment and Wages covered approximately 60,000 establishments with 350,000 employees. It took 27 statistical employees to process the data. After the redesign of the survey in 1994, the number of establishments increased to 75,000 with 1,000,000 employees. Due to the more efficient survey process, the substantially larger amount of data is now taken care of by only 22 people.

The redesign required a different way of working of the people involved. That took some time of getting used to. However, the change turned out to be a success. More data is processed with less people in less time, without reducing the quality of the produced statistics.