NAZIOARTEKO ESTATISTIKA MINTEGIA EUSKADIN 1989 SEMINARIO INTERNACIONAL DE ESTADISTICA EN EUSKADI

NEW TECHNOLOGIES IN COMPUTER ASSISTED SURVEY PROCESSING

J. G. BETHLEHEM and W. J. KELLER

EUSKAL ESTATISTIKA - ERAKUNDEA
INSTITUTO VASCO DE ESTADISTICA

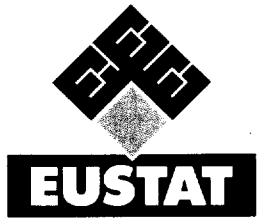
EUSTAT

NAZIOARTEKO ESTATISTIKA MINTEGIA EUSKADIN **1989** SEMINARIO INTERNACIONAL DE ESTADISTICA EN EUSKADI

NEW TECHNOLOGIES IN COMPUTER ASSISTED SURVEY PROCESSING

J. G. BETHLEHEM and W. J. KELLER

KOADERNOA 18 CUADERNO



***Lanketa** / Elaboración:*

Euskal Estatistika-Erakundea /
Instituto Vasco de Estadística

***Argitalpena** / Edición:*

Euskal Estatistika-Erakundea /
Instituto Vasco de Estadística
C/ Dato 14-16 - 01005 Vitoria-Gasteiz

***Botaldia** / Tirada:*

1.000 ejemplares
I-1989

Fotokonposaketarako tratamendu informatikoa:
Tratamiento informático de fotocomposición:

Fotocomposición IPAR, S.C.L.
Particular de Zurbarán, 2-4 - 48007 BILBAO

Inprimaketa eta Koadernaketa:
Impresión y Encuadernación:

ITXAROPENA, S.A.
Araba kalea, 45 - Zarautz (Gipuzkoa)

***Lege-gordailua** / Depósito legal:* S.S. 58/90

ISBN: 84-7542-127-10 Obra completa
ISBN: 84-7749-063-5

BIOGRAPHICAL SKETCH OF DR. JELKE G. BETHLEHEM

Jelke G. Bethlehem studied mathematical statistics at the University of Amsterdam. After obtaining his pre-doctoral degree he was employed as Research Worker at the Statistical Department of the Mathematical Centre in Amsterdam. His work concentrated on multivariate statistical analysis and development of statistical software.

In 1978 he joined the Department for Statistical Methods of the Netherlands Central Bureau of Statistics, first as Research Worker and later as Senior Statistician. His main topics were the treatment of nonresponse in sample surveys, in which he obtained his Ph. D., and disclosure control of published survey data.

Now he is chief of the Statistical Informatics Unit (a research unit within the Automation department), which concentrates on the development of standard software for processing survey data. The important fields of study are the Blaise System for computer assisted survey data collection and data processing, tabulation packages, and software for weighting sample survey data.

BIOGRAPHICAL SKETCH OF PROF. DR. IR. WOUTER J. KELLER

Wouter J. Keller studied electronics (BSc, cum laude) and applied mathematics at the Twente University of Technology (MSc, cum laude-, After his study he was employed as associate professor at the Institute for Fiscal Studies of the Erasmus University, Rotterdam and obtained his PhD (cum laude) in econometrics.

In 1979 he joined the Netherlands Central Bureau of Statistics as head of the Department of Statistical Methods and in 1987 he became the head of the Automation Department. Also since 1982 he is part-time professor of econometrics and informatics at the Free University of Amsterdam.

He works nowadays mainly on Automation of survey processing (Computer Assisted Telephone Interviewing (CATI), Computer Assisted Personal Interviewing (CAPI), and other methods for collecting, correcting and processing survey data), software development (software for statistical and econometrical analysis and time scheduling problems), microcomputers (hardware and software, in particular statistical software), management of research —and EDP— projects.

CONTENTS

INTRODUCTION

National statistical offices can improve the quality of the published statistics and the efficiency of the production process by applying recent developments in computer technology. For that purpose microcomputers are in use at the Netherlands Central Bureau of Statistics (CBS) in many steps of the statistical production process. Because of the enormous quantities of data to be processed, the microcomputer does not seem to be to most suitable instrument for this kind of work. This paper shows that the contrary is true: use of microcomputers can improve the statistical production process.

Supplying an organization with large quantities of microcomputers opens new ways towards efficient information processing, but there are also problems which have to be dealt with. The CBS faces the challenge of managing decentralized information processing on hundreds of microcomputers. With respect to software, this calls for a strong policy on standardization of automation tools. In this paper, particular attention is paid to the Blaise System for computer assisted survey processing, a tool which controls various steps in the statistical production process. Decentralized information processing is not the only problem. There also other, technical problems which have to be solved: security, back-up, archiving, software licenses and sharing of programs and data files. Local area networks, and wide area networks are proposed as a solution.

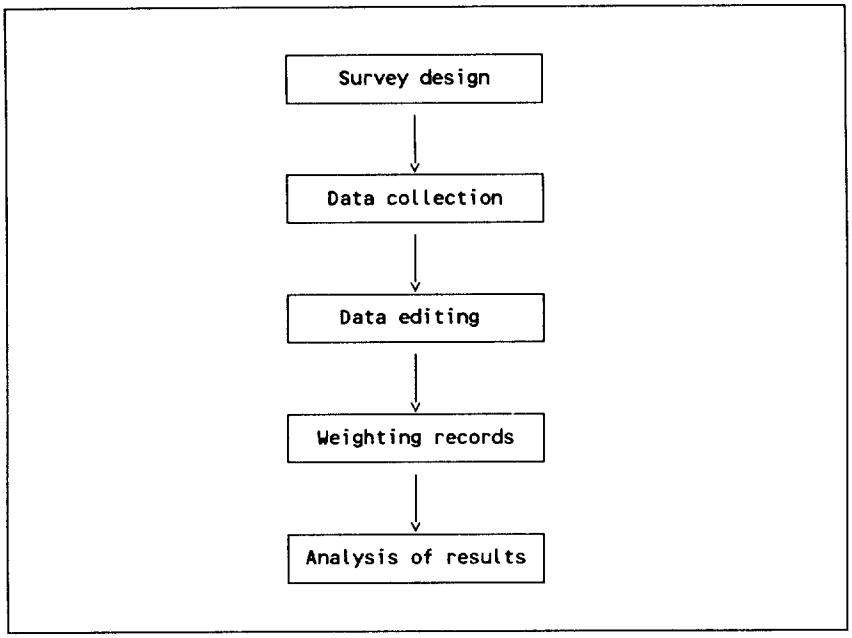
Chapter 1 is an introductory chapter which gives an overview of the various steps to be taken in the production of statistical information. Chapter 2 focuses on quality issues. What is qualtity, how is it defined, and how can computers help to improve quality? Chapter 3 concentrates on one of the most important activities with respect to quality improvement: data editing. In this chapter data editing procedures are analyzed, and the conclusion is drawn that traditional data editing can be carried out more efficiently. Chapter 4 describes the role of the computer in the statistical production process. More and more the computer is used. However, since different computer systems are used by different departments in different steps, the process is not carried out as efficiently as it should be. Decentralization and standardization are discussed as solutions to these problems. Chapter 5 discusses data collection, and particularly, computer assisted data collection. It is shown how this way of data collection improves efficiency and quality. An account is given of some Dutch experiences with computer assisted data collection. The last three chapters are devoted to the Blaise System. Chapter 6 discusses the Blaise way of data processing. It contains an overview of the system. The most important concepts of a Blaise questionnare are discussed. It is explained how a simple questionnaire is built and processed in the Blaise System. Chapter 7 shows how complexe questionnaires are developed in Blaise, and describes some advanced features of the system. The final chapter gives an overview of the software produced by the Blaise System.

CHAPTER 1. THE SURVEY PROCESS

1.1. Production of statistics

The production of statistical information is a complex process, in which data on persons, households and businesses are collected by means of surveys, and have to be transformed into accurate statistics. In this process a number of steps can be discerned, as shown in figure 1.1.

Figure 1.1. Statistical information processing



The first step is the design of the survey, in which the objectives are translated into workable definitions of important concepts. The second step is data collection, usually by means of a questionnaire. Collected data will almost always contain errors. Therefore, in the next step a data editing procedure is necessary. The data are checked for errors, and detected errors are corrected as well as possible. After editing, the data are still not ready for analysis and tabulation. The data are hardly ever representative for the population from which they originate. To correct this bias, a weighting procedure is carried out, which assigns adjustment weights to the sampled elements. Finally, a clean and representative data set is obtained, after which tabulation and analysis can be carried out.

In many cases, processing a survey will be a complex and time consuming activity, involving various departments and different computer systems. Managing and controlling the statistical production process is of vital importance with respect to timeliness and quality. In the subsequent sections we will discuss the the various aspects of the survey process in some more detail.

1.2. Survey design

Our society experiences a growing need for information. If information is required on a group of objects or individuals, a survey is an obvious instrument to collect this information. But before carrying out a sample survey, the *objectives* have to be specified carefully. In early planning stages of the survey the objectives will often be rather vague, and they have to be translated into a form that allows exact definition of what kind of information has to to be collected from whom in what way.

In the first place, the *target population* has to be defined, i.e. the collection of all elements to which the information relates. Here, we restrict ourselves surveys among persons and households. It is very important to give an exact and workable definition of a target population, because an ambigious definition will almost certainly lead to problems in sample selection and fieldwork. Next, the *variables* to be measured, have to be defined. These variables are used to make inference about the target population with respect to the objectives of the survey. The characteristics of the population are summarized in a small number of quantities, called *parameters*.

Proper selection of a sample requires a *sampling frame*. This is a administrative reproduction of the target population. Furthermore, a procedure should be defined to select elements from the sampling frame. It is important that such a selection procedure leads to a sample which properly reflects the target population. A procedure which assigns other selection probabilities to elements than intended, may lead to biased results, or results which in fact relate to a different population. On the basis of the collected data the parameters are estimated. A recipe to compute such an estimate is called and *estimator*. An estimate of a parameter does not mean much without an indication of its accuracy. The accuracy can only be computed of a proper *sampling design* is used, i.e. the sample is selected by means of stochastic mechanism, and that the selection probabilities are known.

To obtain valid information and comparable results, data collection must take place in a consistent and objective way. Wording of questions must be the same for all sampled persons. Designing a *questionnaire* is not a simple thing. The phrasing of questions requires careful attention. The questions should be clear and unambigous. If different interpretations are possible, answers will not be comparable. Concepts dealt with should be well defined in words familiar to the respondent and interviewer. Not all questions will be relevant to every respondent. Some questions relate to particular conditions, and therefore will not be meaningfull for everyone. A question about working conditions is only relevant for someone with a job, and only women can be asked about the number of children they gave birth to. The *routing structure* is an important part of the questionnaire. Routing instructions specify the conditions under which the questions should be answered. The routing should ensure that only relevant questions are answered, and that irrelevant questions are skipped.

1.3. Data collection

Statistical agencies collect most of their data by means of (sample) surveys. A questionnaire is defined, containing the questions to be submitted to the respondents. Traditionally the questionnaires are completed in face-to-face interviews: interviewers visit the respondents, ask the questions, and fill in the answers on the (paper) questionnaire. Since the interviewer can encourage and help the respondent,

the quality of the collected data tends to be good. Face-to-face interviewing is expensive. It requires a large number of interviewers, who all have to do a lot of travelling. Telephone interviewing has therefore become popular. The interviewer calls the respondents from a central unit, and no more travelling is necessary. Still, telephone interviewing is not always feasible: only people having a telephone can be contacted, and the questionnaire may not be too long or too complicated. A mail survey is still cheaper: no interviewers are necessary at all. Questionnaires are mailed to potential respondents with the request to return the completed forms. Although reminders can be send, the persuasive power of the interviewer is lacking, and therefore response rates tend to be lower in this type of survey.

During the last decade the computer has increasingly be used in the data collection stage. First this occurred in telephone interviewing. More recently, the advent of the small laptop computers made it possible for interviewers to take the computer along with them to the homes of the respondents. More about computer assisted interviewing can be found in chapter 5.

1.4. Data editing

Currently, most survey data are collected in the traditional way: on paper forms and without the use of a computer. Completed questionnaires have to undergo extensive treatment. For producing high quality statistics, it is vital to remove errors. Three types of errors are distinguished: A *range error* occurs if a given answer is outside the valid set of answers, e.g. an age of 348 years. A *consistency error* is caused by an inconsistency in the answers to a set of questions. An age of 11 years may be valid, a marital status 'married' is also not uncommon, but if both answers are given by the same person, there is definitely something wrong. The third type of error is the *route error*. This error occurs if the interviewer, or the respondent fails to follow the specified branch or skip instructions, i.e. the route through the questionnaire is incorrect: irrelevant questions are answered, and relevant questions are left unanswered.

Detected errors have to be corrected, but this can be very difficult if it has to be done afterwards, at the office. Particularly in household surveys, the respondent cannot be contacted anymore, so other ways have to be found to do something about the problem. Sometimes it is possible to determine a reasonable approximation of a correct value by means of an imputation technique, but more often an incorrect value is replaced by the special code indicating that the value is 'unknown'.

Besides data editing, another activity is sometimes carried out during this stage of the production process: *coding of open answers*. A typical example is the question which asks for the occupation of the respondent. Questions are most easy to process if a respondent selects one possibility from a list of precoded answers. However, for a question like occupation, this set of precoded answers will be long, and therefore it will be hard for the respondent to select the proper answer. This problem is avoided by letting the respondent formulate his own answer, which is copied literally on the form. To be able to analyse this type of information, the answers must be classified afterwards, which is a time consuming job, to be carried out by experienced subject-matter specialists.

1.5. Weighting

After completing the data editing process, the result will be a 'clean' data file. But this data file is not yet ready for tabulation and further analysis. The statistician likes to have a data set which is representative for the population for which he is making inference. There can be two reasons why this data set is not representative.

In the first place, the sample is sometimes selected with unequal probabilities, i.e. businesses are selected with probabilities proportional to their size. A clever choice of selection probabilities makes it possible to produce more accurate estimates of population parameters, but only in combination with an estimation procedure which corrects for this inequality.

In the second place, representativity may be affected by nonresponse, i.e. for some elements in the sample the required information is not obtained. If nonrespondents behave differently with respect to the population characteristics to be investigated, the results will be biased.

In order to correct for unequal selection probabilities and nonresponse, often a weighting procedure is carried out. Post-stratification is a well-known weighting method. Every record is assigned some weight, and these weights are computed in such a way, that the weighted sample distribution of characteristics like sex, age, marital status, and area reflects the known distribution of these characteristics in the population. Two major problems can make application of post-stratification difficult: empty strata and lack of adequate population information. Research has been carried out at the CBS in order to improve weighting techniques. The result was a new general method for weighting, in which weights are obtained from a linear model which relates the target variables of a survey to auxiliary variables. Post-stratification is a special case of this method. Because of the generality of the method, different weighting schemes can be applied that take advantage of the available population information as much as possible, and at the same time avoid the mentioned problems. See Bethlehem and Keller (1987) for more details. The increased power of the computer makes it possible to implement this theory. See Bethlehem (1985) for a description of the program LINWEIGHT.

1.6. Analysis

Finally, we have a clean file which is ready for analysis. The first step in the analysis phase will nearly always be calculations of frequency distributions and cross-tabulations of the basis characteristics. A table looks like a simple thing, but in daily practice there is more to it. The composition of rows and columns (often built from more variables), the quantities displayed in cells (counts, means, percentages), the way in which percentages are computed, treatment of multiple response variables, the position of totals and subtotals, disclosure control, and many other things, can make life very difficult. Good tabulation software packages can help a lot, but the usefulness of packages depends on their possibilities, the user-friendliness of the control language, and the possibility to produce camera-ready output.

The CBS uses the packages SPSS and TAU for tabulation on mainframe and minicomputers. On the microcomputer only SPSS/Tables was available. Alltough this packages can produce nice camera-ready tables, it is rather slow and cumbersome. Therefore the CBS decided to develop a new fast and user-friendly tabaulation package, named Abacus. The first release is now ready, and the initial experiences show that even large data sets can be tabulated on a microcomputer quite fast.

Many statistical agencies also carry out analyses on their data, in order to reveal underlying structures, and thus gain insight in the data. Information obtained in this way may improve a subsequent survey, and thus improve quality and reduce costs. In many cases the data are categorical, and so many well-known statistical analysis techniques for interval data can not be used. Of course, there are techniques for categorical data, e.g. loglinear analysis. Still, there is a need for other techniques which can also be applied on very large data sets. This has lead to the development and application of new techniques.

The first technique to be mentioned here is ANOTA. ANOTA (ANalysis Of TAbles) is a technique to explore possibly existing relationships between categorical variables. One of the variables is assigned the role of dependent variable, and all other variables, the explanatory variables, are considered to be predictors of the dependent variable. ANOTA resembles linear regression. The main difference is that in regression analysis the dependent variable must be a numerical variable, whereas in ANOTA the dependent as well as the explanatory variables are categorical. The estimated ANOTA coefficients have the same interpretation as regression coefficients. They measure the effect of the categories of the explanatory variables on the categories of the dependent variable. The coefficients are corrected for possible effects of other explanatory variables and therefore present 'pure' effects. In terms of ANOVA the ANOTA model is a linear model with main effects only. Due to the specific nature of the ANOTA model, no raw data matrix is required to estimate the model. It is sufficient to have all possible two-way tables. The ANOTA theory is described in Keller et al. (1985), and the ANOTA program in Bethlehem (1986).

A second theory for the analysis of categorical data is Correspondence Analysis. Through Correspondence Analysis insight can be obtained both in the degree and the type of association in two-way tables. The technique bears some resemblance to Principal Components Analysis: a contingency table is decomposed into factors. However, contrary to Principal Component Analysis, Correspondence Analysis explains the strength of association as expressed by the chi-square statistic instead of the variance. The technique assigns scale values to rows and columns in such a way that the correlation coefficient between the scaled row and column variables is maximized. The first factor consists of this set of scale values. For each following factor scale values are determined which maximize the correlation coefficient for that part of the association not accounted for by the previous factors. The scale values can be used for a graphical investigation of the association in the table. The Thermoplot compares scale values of rows and columns of a single factor, thereby offering insight in the type of association explained by the particular factor. The Biplot is a two-dimensional plot. As x-axis and y-axis thermoplots of two factors are used. This plot offers the possibility to explore the structure of the association of two factors simultaneously. The Correspondence Analysis has been implemented in the program CORAN, see Bethlehem (1988).

Both techniques (ANOTA and CORAN) use bivariate tables as input instead of the raw data matrix, and therefore analysis on a large data set can even be carried out on a microcomputer.

For more general forms of data analysis the CBS uses the SPSS package. This is a powerful package for complex statistical analysis. For simple explanatory and graphical analysis also the package STATA is used. It combines ease of use, speed and state of the art graphical techniques.

1.7. Publication

Usually the results of tabulation and analysis will be published in the form of tables and graphs. However, national statistical offices meet an increasing demand for releasing microdata files, i.e. data sets containing for each respondent the scores on a number of variables. Because of this trend and an increasing public consciousness concerning the privacy of individuals, the problems involved in releasing microdata are becoming more serious than ever before. Many statistical offices are confronted with this disclosure problems The disclosure problem relates to the possibility of identification of individuals in released statistical information (including publications on paper, tape, floppy disk, etc.), and to reveal what these individuals consider to be sensitive information.

Why is disclosure undesirable? In the first place, it is undesirable for legal reasons. In the Netherlands, for example, there is a law stating that firms should provide information to the statistical office, while the office may not publish statistical information in such a way that information about separate individuals, firms, and institutions becomes available. The privacy act, now in preparation, will probably restrict the use of private data still further. In the second place, there is an ethical reason. When collecting data from individuals, the statement is made by the CBS that the collected data will be kept confidential. In the third place, there is a very practical reason: if respondents do not trust statistical offices, they will not respond. In the Netherlands nonresponse rates in household surveys have increased over the last decade from an average of 20 to 35 percent. Hence confidence is of the utmost importance for the statistical office. The willingness of respondents to cooperate is a very important condition for the production of a statistical bureau.

To protect a microdata set against disclosure, we have to know how identification takes place in practice. Identification is closely related to the concept of uniqueness. Someone is unique in the population if he is the only one in the population with a particular set of scores on a set of identifying variables. Uniqueness in the population is vital for disclosure. Suppose some user of a data set knows that a specific person is unique in a well-defined population. Then there are two possibilities: either this person is in the sample, or he is not. If he is in the sample, he will be identified and disclosed with certainty. If he is not in the sample, no harm can be done. Knowledge of population uniqueness should not be underestimated, in particular if the data set contains variables which make it possible to detect respondents living in a small area. For example, in many small areas certain professions are unique (the doctor, the notary, the dentist). In such (sub)populations many persons are unique on a key existing of only one identifier.

From experiences with the analysis of disclosure risks of real data sets the CBS has drawn the following conclusions: In every microdata set containing 10 or more key variables, a large number of persons can be identified by matching this file with another file containing the key and names and addresses. Furthermore, response knowledge (i.e. the knowledge that a specific person is in the sample) nearly always leads to identification. Finally, analysis showed that even for only two or three identifiers, already a considerable number of persons are unique in the sample, some of them being 'rare persons', and therefore also unique in the population.

Software was developed to carry out disclosure analysis. It estimates uniqueness in the population, and removes the most obvious unique individuals. However, it will not protect the data set against disclosure by matching, and hardly against disclosure by response knowledge. Specifying more stringent criterion values will produce data sets which might to some extent be protected agains these two types of disclosure, but the subsequent loss of information set will generally be unacceptably large.

Disclosure from microdata sets is often possible, and difficult to prevent, unless the information in the data set is severely reduced. Is this the end of microdata dissemination? We think it should not be. Some types of dosclosure risks can be taken care of. And if microdata sets are released under the conditions that *the data may be used for statistical purposes only* and that *no matching procedures may be carried out at the individual level*, any effort to identify and to disclose clearly shows malicious intent. In view of the duty of a statistical office to disseminate statistical information, we think disclosure protection for this kind of malpractice could and should be taken care of by legal arrangements, and not by restrictions on the data to be released. More on the disclosure problem can be found in Bethlehem et al. (1989).

1.8. Data processing

Statistical data processing is a mixture of record-wise processing and file-based processing. In *record-wise* processing records are dealt with one at the time. Often these records are stored in a data base management systems. A record is retrieved, treated, and stored in an interactive way. A good example of record-wise processing is a data editing procedure (see section 1.4). Due to the interactive nature of the activities of record-wise processing the microcomputer is a well-suited tool for it. In *file-wise* processing a data file is processed as whole. Since this way of processing typically involves large quantities of data, it will often be carried out on a mainframe or minicomputer. Examples of file-wise processing are weighting and tabulation (see sections 1.5 and 1.6).

The power of the microcomputer is increasing rapidly. Therefore, time will not be far away anymore in which microcomputers will be used more and more for file-wise processing.

CHAPTER 2. QUALITY ASPECTS

2.1. What is quality?

Surveys are carried out in order to learn something about the population from which the sample has been drawn. Results are presented in statistics, i.e estimates of unknown population characteristics. The statistics derived from the survey will rarely correspond exactly to the unknow values of the population characteristics. The deviation of the estimate from the population value is called the *error*. Every survey operation is affected by errors, and the magnitude of these errors determines the quality of the results.

Errors can have two kinds of effect on estimates. In the first place a systematic error introduces a bias, i.e. estimates obtained by repeated application of the survey process, tend to under- or over-estimate the value of the population characteristic. In the second place, a random error introduces an extra source of variation, i.e. repeated application of the survey process will result in estimates with a larger spread around the value to be estimated.

Generally, two broad categories of errors are distinguished: sampling errors and nonsampling errors. Sampling errors are due the fact that only a sample is observed and not the whole population. Sampling errors will not occur in a census (a complete enumeration of the entire population). Nonsampling errors relate to all survey errors that stem from other sources than the use of a sampling mechanism. The surver researcher is in control of the sampling error. By making a proper sampling design and selecting a sufficiently large sample, unbiased estimates can be obtained with small standard errors. Of course, the available financial budget may impose restrictions on the size of the sample. Nonsampling errors are difficult to control. The best way to deal with nonsampling errors would be to take preventive measures at the design stage. One can think of well-considered selection of the sampling frame, careful design of the questionnaire, adequate training of interviewers, and thourough pilot surveys. However, many of the causes of nonsampling errors are inherent in the survey process and as such are virtually impossible to avoid.

There is also another aspect of quality in survey research, and that is the timeliness of the publication of the results. The production of statistical information is a complex and time-consuming process. Different departments are involved in the activities, and one department can not start working on the survey until the other department is ready with it. Specifically for large surveys it may take months, if not years, before the results are ready for publication. And the more the publication of results is delayed, the less useful it is.

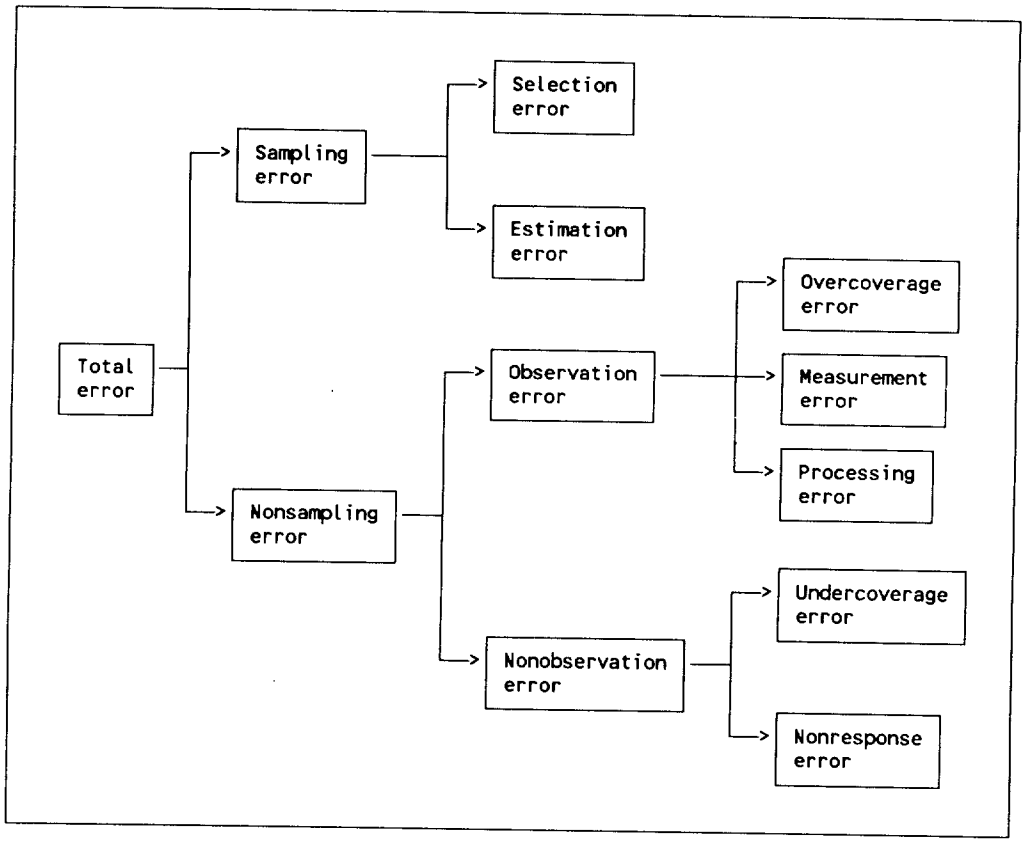
Finally, quality can be regarded from the viewpoint of costs. The survey designer should always try to minimize the costs of activities which are not aimed at quality improvement. Money can always only be spend once, and when it is spend it should help to improve quality.

In the remainder of this chapter we will discuss the various sources of errors, and how the use of computers may reduce them.

2.2. Sources of error

A proper classification of the different types of errors helps the survey researcher in finding and exploring phenomena which may effect the quality of survey results. Figure 2.1 contains a taxonomy, i.e. a hierarchical classification of errors.

Figuur 2.1. A taxonomy of errors in survey sampling



We already discussed the fundamental division in sampling errors and nonsampling errors. The sampling error can be divided in two components: the selection error and the estimation error. A *selection error* occurs if the real selection probabilities differ from the selection probabilities as specified in the sampling design. So, in fact the wrong probabilities will be used in estimation procedures. This can happen, for example, when some elements appear more than once in the sampling frame.

If it would be possible to repeat a sample survey, under exactly the same conditions, than the sampling mechanisme will selected different elements, with different values, and therefore, estimates will differ. The error caused by the fact that estimates are based on observations on a sample of elements which are selected by chance, is called the *estimation error*

The estimation error can be controlled by specifying a proper sampling design and sample size. In many surveys the sample size is so large that the estimation error small compared to other errors. Selection errors are hard to detect. If a sampling frame is stored in a computer, one might try to detect duplicate entries. However, in

address files it is very hard to detect that a household in fact owns two or more houses on different addresses. Generally, one has to rely on the quality of the sampling frame as it is.

We now focus on nonsampling errors. These errors can be divided in observation errors and nonobservation errors. The *observation error* is caused by incorrect collection, registration, or processing of the data. So, observation errors relate to differences between observed individual values and true individual values. Many activities are carried during the proces from observation until publication, and in all steps something may go wrong, causing the final value to differ from the true value. In contrast with observation errors, nonobservation errors are caused by the fact that observation is not possible. A *nonobservation error* occurs when either no information can be obtained from selected population elements, or when it is impossible to select one or more population elements in the sample.

For the observation error a division is possible in three components: the overcoverage error, the measurement error, and the processing error. An *overcoverage error* occurs if the sampling frame contains elements which do not belong to the population to be investigated, and such an element is included in the survey without this being detected. The second component of the observation error is the *measurement error*. Measurement errors denote a difference between the finally obtained value and the original, true value. Measurement errors can have many causes. For example, the respondent does not understand the question, or he does not want to give the true answer. The third component of the observation error is the *processing error*, caused by errors in the process of data entry, data editing, tabulation and analysis.

The nonobservation error is composed of the undercoverage error and the nonresponse error. Just like for the overcoverage error, the undercoverage error has to do with the lack of fit between target population and sampling frame. The *Undercoverage error* is caused by the phenomenon that elements in the target population are not reproduced in the sampling frame, or which can not be reached through the sampling frame. The second component of the nonobservation error is nonresponse error. *Nonresponse* is the phenomenon that from elements, which belong to target population, and which are selected in the sample, the required information is not obtained, or the required information is unusable.

2.3 Improving quality with computers

Some of the errors mentioned in the previous section can be avoided by improving the survey design. Measurement errors can be prevented by an improved questionnaire design and better training of interviewers. Frame errors (undercoverage and overcoverage) can be taken care of by using better sampling frames. Well-trained interviewers and a good management of the survey operation may help to fight nonresponse.

The computer can be particularly useful in doing something about measurement errors and processing errors. In many surveys one tries to improve the quality of the results by application of correction techniques on the collected data. But there is a better way to handle these errors. One should realize that errors are best treated by fighting them at their source. This approach agrees with the ideas of W. Edwards Deming (1986) on statistical process control in industrial production. Quality control

has always been an important issue in industrial production. Many of Deming's famous 14 points for management can also be applied to the production of statistical information. One of these points states that one should cease dependence on mass inspection. Inspection to improve quality is too late, ineffective and costly. Quality must be built in at the design stage. These statements particularly apply to data editing. By trying to detect and correct errors in questionnaire forms well after they have occurred, one fails to locate the source of the error, and consequently, it can not be eliminated. Particularly computer assisted interviewing techniques have the potential to detect errors occurring during the interview. Since both the interviewer and the respondent are there, this is also the best moment to correct these errors. This is one way of how the use of computer can improve the quality of survey results.

From the viewpoint of timeliness, use of computers also improves quality. Since computer assisted data collection integrates data collection, data entry and data editing, data is processed more efficiently, and the total production time is reduced. The improved timeliness makes the results more usefull, and in that respect, improves the quality of the data.

If quality is regarded from the viewpoint of costs, then computers are also important. Since the computer can take over some of the laborious tasks of subject-matter specialists, also the costs of carrying out a survey can be reduced. If, for example, computer assisted data collection is used, consistency checking is carried out by the computer, and afterwards no more data entry of the paper forms is necessary.

CHAPTER 3. THE DATA EDITING PROCESS

3.1. Data editing

Data editing is the process of detecting and correcting errors in the individual records, questionnaires or forms. The process is carried out with the intention to improve the quality of the results of surveys. At the Netherlands Central Bureau of Statistics much importance is attached to this aspect of survey research. A large part of human and computer resources are spend on data editing.

Currently, most survey data are collected in the traditional way: on paper forms and without the use of a computer. Completed questionnaires have to undergo extensive treatment. For producing high quality statistics, it is vital to remove errors. Three types of errors are distinguished: range checks, route checks, and consistency checks.

Range checks verifys whether an answer to a question belongs to the domain of valid answers according to the definition of the question. For example, if the answer to a question about age is defined to be in the range from 0 to 120, an answer of 348 is considered to be a range error.

Route checks verify whether the correct route through the questionnaire has been taken, i.e. whether all relevant questions have been answered, and all irrelevant questions have been skipped. For example, people looking for a job should answer questions about their previous job, and how they are looking for a new job, whereas people with a job should only answer questions about their job. Route checks are only relevant for data collected by means of paper forms. In case of computer assisted data collection no route errors can be made. See also chapter 5.

Consistency checks establish whether answers to several questions are consistent. While answers to questions can be within the valid domain, the combination of answers can lead to an inconsistency. For example, an age of 7 is possible, and also being married is possible, but if both answers are given by the same person, there is something wrong.

Particularly in the case of consistency errors, a distinction is made between hard errors and soft errors. A *hard error* indicates a real error. There is a problem which must be solved. Without correction of hard errors, the form will never be declared 'clean', i.e. error-free. A *soft error* does not necessarily indicate a real error. A soft error should be interpreted as a warning, a signal that something implausible has been encountered. If a soft error is reported, one can decide to ignore the error message without making corrections.

Detected (hard) errors have to be corrected, but this can be very difficult if it has to be done afterwards, at the office. Particularly in household surveys, the respondent cannot be contacted anymore, so other ways have to be found to do something about the problem. Sometimes it is possible to determine a reasonable approximation of a correct value by means of an imputation technique, but more often an incorrect value is replaced by the special code indicating that the value is 'unknown'.

Data editing is an important part in the survey process. Since data editing is time consuming and requires a large amount of manpower and computer resources, it is worth while to evaluate existing procedures from time to time. In 1984 the Netherlands Central Bureau of Statistics started a data editing research project. Editing procedures of four surveys were investigated. The conclusion from this project was that data editing could be carried out more efficiently. This conclusion lead to the design and development of the Blaise system, to be discussed in chapters 6, 7 and 8.

3.2. The data editing research project

In spite of a long time experience, the CBS had not much insight in the nature of the data editing activities. In particular one would like to have some indication of the profits and costs of the data editing process. To obtain insight in costs and profits, it was decided to take a close look at some existing surveys. Such an analysis should produce an inventory of the different kind of activities carried out during the data editing process, the frequency of occurrence of these activities, the time spend on them, and the number and qualifications of the people involved.

For the evaluation of existing data editing procedures four surveys were selected: two economic surveys, the Foreign Trade Survey (FTS) and the Survey on Transport of Goods by Road (STG), and two social surveys, the Labor Force Survey (LFS) and the Survey on Well-being of the population (SWB). Two surveys (one economic and one social) were very large, see table 3.1. For these surveys only an inventory was carried out. The two other surveys (one economic and one social) were of moderate size. For these surveys not only an inventory was made, but, because of their size, it was also possible to carry out a comprehensive analysis of the data sets.

Table 3.1. Evaluated Surveys

Type of survey	Size of survey	
	Small	Large
Economical	Foreign Trade (FTS)	Transport of Goods (STG)
Social	Well-being (SWB)	Labor Force (LFS)

It was hoped and expected that the evaluation would show the bottlenecks in the data editing process. Therefore, the second phase of the data editing research project was reserved for the improvement of existing procedures and the design and the implementation of new tools. Such tools should not be aimed at application in particular surveys, but should have general applicability in all surveys of the CBS.

3.3. Data editing in economic surveys

Objective of the *Foreign Trade Survey (FTS)* is to produce monthly statistics on the trade with EEC-countries and other countries. Volume and value of import and export are tabulated by commodity and country. To that end each month about 700,000 records are processed. The data are compiled from declarations coming from different sources (individual enterprises, customs and postal service) and have to be transformed into one single large file. The FTS can be characterized as a very large survey with a

small number of fields per record, which has to be processed in a short time, and with a fairly comprehensive editing process. The complexity of the editing process is due to the use of a detailed classification system distinguishing roughly 8000 different kinds of commodities.

First the forms are manually checked for completeness. In case of missing answers an attempt is made to obtain this information, if necessary by contacting the supplier of the data. A number of variables are coded, e.g. country of provenance, means of transport and place of unloading. For goods of high value the description is checked. Next the data are entered using a dedicated data entry computer. Data is entered by 40 data typists, without checking. Each data typist processes about 200 forms per hour. After entry, the data files are transferred to the mainframe computer system. In a batch-wise fashion the data are checked, and detected errors are printed on lists. These error lists are returned to the subject-matter department, where one tries to correct the errors. If necessary the supplier of the data is contacted. Corrected forms are re-entered in the data entry computer by data typists, the resulting file is transmitted to the main frame and merged with the already existing correct records. This cycle of automatic error detection and manual correction is carried out two times. In a total of 700,000 records 150,000 errors are detected, distributed over 100,000 different records. Correction takes about 3000 hours each month.

The *Survey of Transport of Goods by Road (STG)* is a small survey. It collects data concerning Dutch enterprises which have goods transport by road for hire reward as their main activity. Objective of the survey is to produce information on the structure of receipts and costs, employment figures and transport equipment figures. Once a year a sample of about 1000 enterprises is drawn and forms are send to the selected enterprises. If no form is returned, recall letters are send. If this has no effect the enterprise is visited by an employee of the CBS. The questionnaire for STG mainly focuses on the collection of financial information, energy consumption and characteristics of transport means. Not only values and amounts have to be provided by the representative of the enterprise, but he also must compute subtotals and totals. The presence of totals on the forms has two effects in the data editing process. On the hand they provide a means for checking the individual amounts, and on the other hand they are themselves an extra source of error. A large part of the editing process concentrates around the treatment of these totals.

In the first step of the data editing process the collected forms are checked for completeness and the presence of the unique identification number of the enterprise. If necessary, entries are rounded. Totals should always be positive. Entries smaller than a specified minimal amount must be set to zero. If only totals are specified on the form, this quantity must be distributed over individual entries. If inconsistencies are detected the enterprise can be contacted by telephone, but for a number of enterprises also figures from previous years are available, and can be consulted. In the first step of the editing process three types of activities can be distinguished:

1. *Real improvements.*
   This kind of activities improve the quality of the data. Among them are movement of entries to other categories, removal and modification of entries, and distribution of totals over categories.

2. *Preparation for data entry.*
These activities are necessary for efficient data entry. Among these activities are rounding off, specification of entries at the proper place and removal of all kind of irrelevant information.

3. *Superfluous activities.*
These activities neither improve the quality of the data nor simplify data entry. Among these activities are calculation of totals and balances, and indication of missing amounts by a minus-sign (-).

Table 3.2 specifies the relative share of each of these three types of activities. Apparently only a minor part of the editing activities in this phase is devoted to quality improvement.

Table 3.2. Manual activities in the Survey of Transport of Goods by Road

Type of activity	Percentage
Real improvements	23 %
Preparation for data entry	18 %
Superfluous activities	59 %
Total	100 %

In the second step the data are entered in the dedicated data entry computer by data typists. No error checking takes place in this step. After completion, the resulting file is converted to the mainframe computer. For each enterprise a number of records is generated. Each record specifies a code and a corresponding value or amount. The file consists of roughly 50,000 records, relating to 1000 enterprises. In the third step some elementary checks are carried out. With a computer program identification number and specification of the main activity of the enterprise are checked. Errors are detected in 0.6% of the records. Detected errors are corrected by means of an interactive editor. In the fourth step of the data editing process the data themselves are checked by another computer program. Totals must be positive, specified totals must be equal to the sum of the individual entries and subtotals must be smaller than totals. In 20% of the forms errors are detected and this concerns 0.7% of the records. The program produces lists of errors with diagnostic messages. On these lists corrections are made. The corrected records are re-entered in the data entry computer, converted to the mainframe computer and merged with the already present correct records. This cycle is automatic checking and manual correction is repeated two more times. After the third run the number of remaining errors is sufficiently small. All detected errors in the fourth step concern the quality of the data. Correction of these errors improves the quality of the results.

3.4. Data editing in social surveys

Up to 1987 a *Labor Force Survey* (LFS) was carried out every two years. The evaluation of data editing procedures was carried out on the 1983 survey. For each selected household there was a questionnaire with questions about the composition of the household (the type A questionnaire). Furthermore, there were different questionnaires for each member having a job (the type B questionnaire), and each non-working member (the type C questionnaire). In the 1983 survey data about

approximately 350,000 persons were collected. Some of the questions had open answers. Consequently, part of the editing process was spend on coding these open answers.

In the first step of the editing process the forms were checked manually. For all forms belonging to the same household the identifying number was checked. This is very important, since different forms, and even parts of forms, were separated, because they got different treatments. And finally, information had to be combined again. Open answers on questions like occupation and education, were coded manually. Coding took about 10 minutes per questionnaire. Other activities took about 3 minutes per questionnaire.

When the forms were ready for data entry, they were sent to the data entry department. There forms were entered using a dedicated data entry computer. The data was processed in different flows, each flow corresponding to a different part of a questionnaire. For each flow there was tailor made editing program developed by the computer department. The routing structure in the questionnaires was very complex. Therefore, a large part of data editing was devoted to checking the routing. But also a number of consistency checks was carried out. The program performed checks and produced lists with errors. The error lists were processed by the data editing department. Corrections were carried out manually. Corrected records were entered again by the data entry department. The cycle of automatic checking and manual correction was repeated three or four times. Some indication of the efficiency of this procedure can be obtained from table 3.3. When no more errors were reported, the data file was supposed to be clean. Finally, the different flows had to be recombined again.

Table 3.3. Detected errors in the Labor Force Survey.

Type of questionnaire	Number of errors per 1000 records in run 1	Number of errors per 1000 records in run 2
B	47	13
C	54	17

The *Survey on Well-being of the Population* (SWB) is based on a relatively small sample of approximately 4000 persons. In this survey a large number of questions are asked about living conditions, e.g. housing, work, income, education, leisure time activities, health, and social relations.

First, the collected forms go through a manual phase. In this phase two types of activities can be distinguished: checking and preparation for data entry. Preparation mainly concerns copying of hand-written answers in precoded fields. During the manual phase checking leads to about 4000 corrections on the forms. An extensive analysis of these corrections showed that the mayor part (85%) consists of specifying a code 'unknown' for not answered questions. The coding 'unknown' usually means filling the particular field with nines, e.g. if income is not answered, the code 99999 is specified. An average of 13 minutes per form is spend on this type of coding. Another coding activity is the coding of the open answers on questions regarding education and occupation. This takes roughly 16 minutes per form. Including all activities during the manual phase more than 40 minutes are spend on every form.

After completion of this phase, the forms are transfered to the data entry department. Here the forms are entered on the dedicated data entry computer. Entry of all data takes about 300 hours. To avoid typing errors, data is entered twice. A record is only accepted if both entries result in the same values in the corresponding fields. After entry, the files are transfered to the mainframe computer. On the mainframe computer the cycle of automatic checking and manual correction is carried out. Checking concerns valid answers, consistency of answers and correct routing. Detected errors are listed, corrections carried out by hand, and corrected records entered anew and transfered to the mainframe. Table 3.4. gives a summary of detected and corrected errors.

Table 3.4. Detected errors in the Survey on Well-being of the Population

Type of error	Errors per 1000 records		
	Cycle 1	Cycle 2	Cycle 3
Consistency/routing	654	60	6
Routing	305	56	21

Remaining errors were dealt with by the subject matter department. No special software was used. With the statistical package SPSS relevant tables were produced. If necessary, corrections were carried out in the mainframe file.

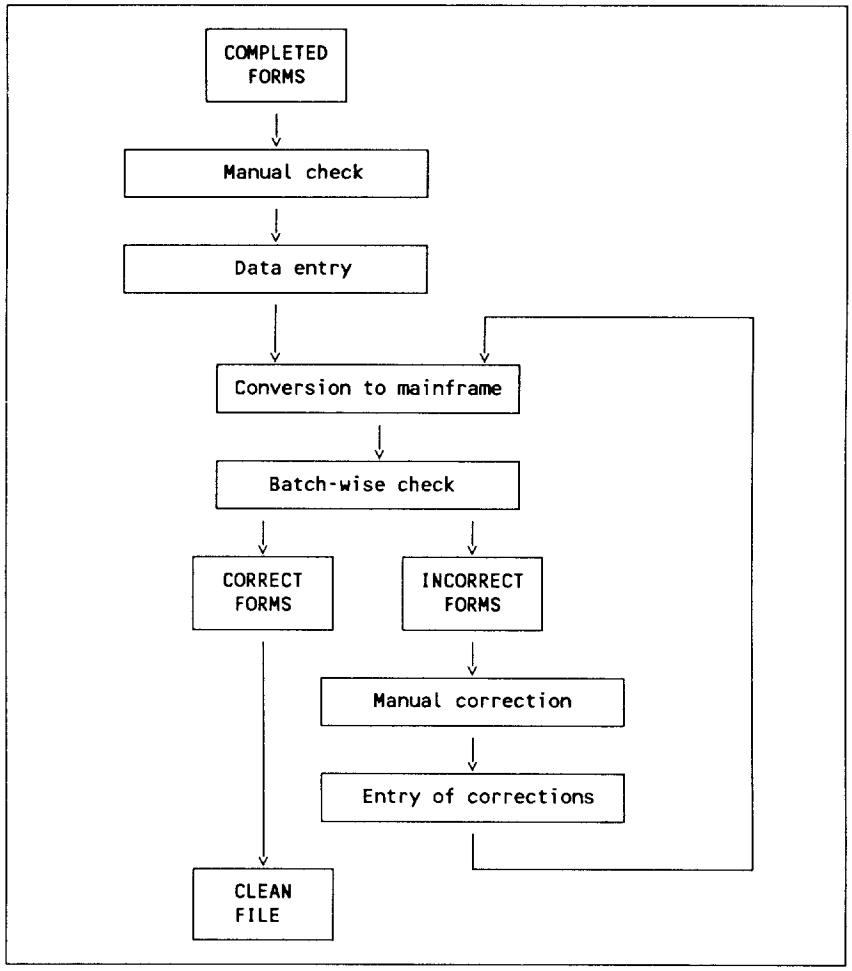
3.5. Conclusions from the project

Considerable differences were observed between the editing process in economical and social surveys. In most economical surveys the questionnaire is straightforward. The questions are answered one after another without complex routing structures. Many range checks and consistency checks are carried out. Totals of individual entries have to be calculated or checked. Often figures of firms are confronted with corresponding figures from previous years. In case of detected errors it is sometimes possible to contact the firm, either by telephone or by an employee of the CBS which visits the firm. In short, economical surveys can be characterized by simple questionnaires with a lot of checking.

Questionnaires for social surveys are often very large with complex routing structures. Error checking mainly concerns the route followed through the questionnaire and some range checks. Only a few consistency checks are carried out. To be able to correct detected errors, contact is necessary with the supplier of the information. Since this is hardly ever possible, generally correction results in setting a value to 'unknown'.

Although the data editing process differs from survey to survey, still some general characteristics can be observed which hold for nearly all surveys. The editing process is summarized in figure 3.1.

Figure 3.1. The traditional approach to data editing.



After collection of the forms, subject-matter specialists check the forms for completeness. If necessary and possible, skipped questions are answered, and obvious errors are corrected on the forms. Sometimes, the forms are manually copied to a new form to allow for the subsequent step of fast data entry. Next, the forms are transfered to the data entry department. Data typists enter the data in the computer at high speed without error checking. The computer is a dedicated system for data entry. After data entry, the files are transfered to the mainframe computer system. On the mainframe an error detection program is run. Detected errors are printed on a list. The lists with errors are sent to the subject-matter department. Specialists investigate the error messages, consult corresponding forms, and correct errors on the lists. Lists with corrections are sent to the data entry department, and data typists enter the corrections in the data entry computer. The file with corrections is transfered to the mainframe computer. Corrected records and already present correct records are merged. The cycle of batch-wise error detection and manual correction is repeated until the number of detected errors is considered to be sufficiently small.

After the final step of the editing process the result is a 'clean' data set, which can be used for tabulation and analysis. Detailed investigation of this process for the four selected surveys lead to an number of conclusions. These conclusions are summarized below.

1. *Different people from different departments are involved.*
   Many people deal with the information: respondents fill in forms, subject-matter specialists check forms and correct errors, data typists enter the data in the computer, and programmers from the computer department construct editing programs. Transfer of material from one person/department to another can be a source of error, misunderstanding and delay.

2. *Different computer systems are involved.*
   Data entry and data editing are carried out on different computer systems. Transfer of files causes delay. Incorrect specification and documentation may produce errors.

3. *Not all activities are aimed at quality improvement.*
   A lot of time is spend just on preparing forms for data entry, and not on correcting errors. Subject-matter specialists have to clean up forms to avoid problems during data entry. The most striking example is assignment of a code for 'unknown' to unanswered questions.

4. *Manual check of complex routing structures.*
   In particular for social surveys much time is spend on checking routing through the questionnaire. Vital consistency checks are not carried out.

5. *The editing process is cyclic.*
   Repeatedly the process is going through the cycle of data entry, automatic checking and manual correction, in many cases three times or more. Due to the cyclic nature this part of the process is very time consuming.

6. *Repeated specification of the data.*
   In several steps of the data editing process the structure of the data must be specified. Although essentially the same, the form of specification may be completely different for every step. The first specification is the questionnaire itself. The next specification is with respect to data entry. The automatic checking program requires another specification of the data set. For tabulation and analysis, e.g. using the statistical package SPSS, again another specification is needed. All specifications ask for a description of variables, valid answers, routing and possibly valid relations.

From the conclusions of the first phase of the research project it became clear that the process of data editing could be improved. It was decided to design a new data editing system which should not have the disadvantages of the present system. In particular, editing activities should be restricted as much as possible to one department and one computer system, preferably a microcomputer or a network of micro-computers.

Error checking and correction should be an intelligent and interactive process, to be carried out by the subject-matter specialist and his microcomputer. Instead of the traditional batch-oriented process in which the data set is processed as a whole, there

should be a record-oriented process in which records are dealt with one at the time. This reduces the cyclic nature of the editing process. After entering and editing a record by the subject-matter specialist, the record should be error-free and ready for further analysis.

Furthermore, the system should be based on a powerful language in which questionnaires can be specified, including valid answers, routing information and consistency checks. Questionnaires should be specified in a structured way. All questions pertaining the a particular subject should be contained in a separate subquestionnaire. Storage of subquestionnaires in a library system or data base may contribute to a better coordination of different questionnaires.

Using the specified questionnaire as input the system should be able to generate data editing modules automatically. In particular it must be able to produces the software for intelligent interactive data entry and data editing (CADI).

When these requirements were specified, it was realized that such a system could be much more powerful. In fact, such a system has some properties of an expert systems. The questionnaire description is the 'knowledge base', containing all knowlegde about the questionnaire and the data. The user should be able to use this knowledge to develop all kinds of data processing applications. Consequently, the system was extended to be able to take into account different modes of data processing. Indeed, using the questionnaire specification as input, the system should also generate software modules for computer assisted telephone interviewing (CATI) and computer assisted personal interviewing with laptop computers (CAPI). For traditional paper and pencil interviewing (PAPI) a printed version of the questionnaire (including routing information for the interviewer) must be possible. Furthermore, the system should be able to produce setups and system files for other data processing software, e.g. for tabulation and analysis packages

CHAPTER 4. THE ROLE OF THE COMPUTER

4.1. Historical overview

The production of statistics is requires a lot of calculations. Therefore it is not surprising that throughout history there has always been a close relationship between statistics and computers. Of course, statistics is much older than the computer. Take, for instance, the Inca of the 10th to 14th century. Each district had its own statistician, the *Quipucamayoc*. He recorded statistical information (number of men, women, llamas) on a *quipu*.

Figure 4.1. The Quipucamayoc, the Inca statistician

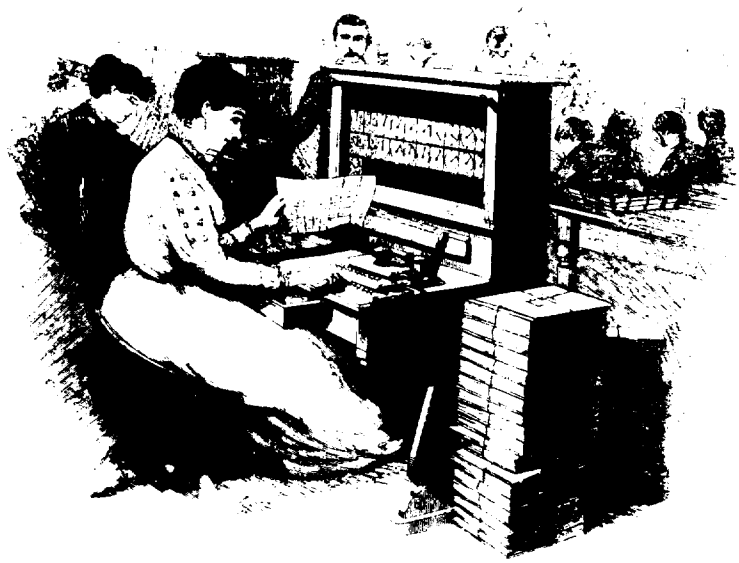
The quipu was a simple but ingenious device consisting of a main cord, and from this cord dangled smaller coloured strings which had at intervals nots tied into them. These strings recorded numbers in the decimal system. Each colour denoted a different subject. With some imagination the quipu can be seen as an early predecessor of the laptop computer.

One of the famous man in the history of statistics was Blaise Pascal (1623-1662). In a correspondence with Fermat he layed the foundation for modern probablity theory. Maybe it is not a coincidence that Pascal was also one of the inventors of a calculating machine, the *Pascaline*. He constructed this machine for his father, to help him with his tax computations. It is doubtful whether the machine was also used for statistical computations.

A major breakthrough in statistical computation took place around 1880. In those days Herman Hollerith was employed at the U.S. Census Bureau. He observed the soul-killing, monotoneous work of hundreds of clerks who were involved in processing the censuses, and he decided to do something about it. He was inspired by the work of the Frenchman Joseph-Marie Jacquard, who had developed an automatic weaving loom in 1801. Punched wooden cards controlled the weaving of the cloth so that any desired pattern could be obtained automatically. Hollerith used the same principle in

the construction of his famous punched card machine, which was successfully used in the U.S. census of 1880 and 1890. The answers to census questions were recorded in punch cards, which were fed into a tabulation machine having 40 dials, so that the answers to 40 questions could be processed at the same time. With these Hollerith machines the results of the census were published already six weeks afters the completion of the fieldwork. After some years Hollerith left the U.S Census Bureau and started his own business for the construction of his machines, which ultimately became IBM.

Figure 4.2. The Hollerith machine



The first programmable computing machine was constructed by Konrad Zuse in Germany in 1936. A similar machine, the Mark I, was completed in the U.S. in 1944. The memory of both machines consisted of electro-magnetic relays, and therefore they were not very fast. Speed increased dramatically after the relays were replaced by electron tubes. The UNIVAC was one of the first general purpose computers. It was used by the U.S. Census Bureau for the 1950 census.

The development of new technology continued with an increasing speed. First the error prone tubes were replaced by transistors, and then transistors by chips. In the sixties and seventies the large mainframe computers emerge, to which many terminals could be connected. The statistician got the possibility to operate the computers himself. Statistical packages like SPSS, BMDP and P-STAT helped him to carry out his data analysis.

Finally, in the eighties, we see the rapid advent of the microcomputer. Due to its relatively low price and its userfriendliness, the microcomputer appears at almost any desk. The growing market for microcomputers generates an explosive supply of statistical software. The availability of hundreds of packages makes it very difficult to select the proper one. The decision to purchase a specific package can be based on various criteria. Keller (1986) investigated the usefulness of 15 statistical packages.

The concluding table can be found in appendix 1. Keller and Crompvoets (1989) also investigated several database packages for use on microcomputers. The results of that research are summarized in appendix 2.

In the beginning the computer was only used for sorting, counting and tabulation, but in the sixties and seventies, with the emergence of the mainframes and the statistical packages, statisticians were able to carry out extensive statistical analyses. Also the computer was increasingly being used for data editing. More recent is the use of computers for data collection, i.e. computer assisted telephone interviewing (CATI), and computer assisted personal interviewing with laptop computers (CAPI). And, of course, statistical publications are composed with word processers.

4.2. The use of microcomputers

The CBS attempts to produce more statistics with less people and within shorter space of time, while maintaining the quality of published statistics. It is obvious that an increased use of automation tools helps in realizing these goals. The idea that automation should be exclusively be carried out by computer specialists is out of date. More and more the subject matter statistician is computer-aware and computer-minded, and therefore asks more urgently for suitable software and hardware. Simple and straightforward electronic data processing is now carried out by the subject matter departments themselves, leaving design and maintenance of complex information systems to be carried out by the computer specialists of the automation department. All interactive work (including data entry and data editing) is carried on microcomputers, while most batch-oriented work (weighting and tabulation) is concentrated on mainframe and minicomputers.

The CBS makes an increasing use of microcomputers in many steps of the statistical production process. On the one hand, this opens new ways towards efficient information processing, but on the other, this creates new problems that have to be dealt with. If every department purchases its own machines and software, data, software and training are not exchangeable between departments. Additionally, backup and archiving are responsabilities better served by central EDP departments and often neglected by the end user responsible for his microcomputer. Also, distribution of new releases of software packages, including their documentation, is often cumbersome in large organisations with a lot of standalone microcomputers. Finally, the communication between microcomputers is only possible by exchanging floppies, with all the dangers of loss of confidential information considering how easy it is to take e.g. 1.44 MByte floppies outside the office. In other words, a large number of standalone microcomputers leads to chaos, and some coordination and standardization is asked for.

To provide this coordination, the best route to go in our opinion is PC-LAN's (Local Area Networks), with floppy-less workstations and centralized backup. Additionally, to enable the exchange of data, programs and experience, a strong standardization is called for. If everyone uses the same spreadsheet, for example, one training course suffices for the whole organisation and exchangebility of people, programs and data is garantied by the choice of one spreedsheet for all departments. Distribution of new software releases on a LAN is also much easier, since with one command on can upload the new version to all fileservers.

So, while the use of the instruments is brought closer to the subject-matter specialists at the departments (decentralisation), to standardize and coordinate the environment in which these end users operate asks for strong centralization.

With respect to software, standardization means that only the following software is available for CBS-users:
- Programming languages: Pascal (micro), and Cobol (mainframe)
- Databases: Paradox (micro), and Oracle (mainframe)
- Data collection, entry and editing: Blaise (micro)
- Tabulation: SPSS (micro/mainframe), Abacus (micro), Tau (mainframe)
- Analysis: SPSS (micro/mainframe), Stata (micro)
- Graphics: Freelance (micro)
- Planning: TimeLine (micro)
- Spreadsheet: 1-2-3 (micro)
- Word processing: PC-Write (micro)

4.3. The automation infra-structure

The CBS solution to the problems with stand-alone microcomputers is networking. Every deparment has its own local area network (LAN), running onder Novell software, and ARC-net hardware. Ten to fourty microcomputers are connected to a file server with an average storage capacity of 200 Mb. Security is guaranteed by means of password protection in a login-procedure, by encryption, and by a write-lock for diskettes. In the future, diskettes will even become superfluous, and diskette stations can be removed completely. Since all software and data files are stored on file servers, sharing is now very simple.

The CBS has approximately 60 LAN's. For each LAN, a full-time LAN administrator and a deputy administrator are appointed. These two people are vital for installation and maintenance of software, and for user support. All LAN's are connected by a 'backbone' ARC-net (which will be replaced by Ethernet in the near future). Also the large CDC Cyber mainframe and the CDC 930 minicomputers are connected to this backbone, using Ethernet and bridges. The two sites of the CBS (Voorburg and Heerlen, which are 300 km apart) are connected by a 2 Mbps link.

Archiving and backing-up the LAN's is carried out in a centralized way by the automation department (total backup nearly 10 Gigabyte!). It is clear that version control and updating software can be realized very simple in such an environment. All software licenses are based on concurrent usage, which is controled by home-made software.

Allocation of microcomputers to departments is carried out by the board of directors, each half year, in batches of several hundreds of microcomputers. Number of software licenses (in terms of concurrent 'counts') and documentation sets, amount of disk space, and number of printers are determined by simple rules based on the number of microcomputers allocated to the department. In order to cope with the problem of educating large numbers of new PC-users, the CBS runs an average of 50 one-day courses per month (occupying three fully equipped lecture rooms every working day). More details about the automation infra-structure can be found in Keller and Metz (1988).

CHAPTER 5. DATA COLLECTION

5.1. Traditional data collection

The questionnaire must be completed by all selected persons, and this can be accomplished in a number of ways. Traditionally, there are three modes of data collection: mail interviewing, telephone interviewing and face-to-face interviewing.

The first mode is the *mail interviewing*. Questionnaires are sent by mail to the respondents with the request to complete the form and to send it back to the statistical office. Since no interviewers are necessary, this is a cheap mode of data collection. An additional advantage is the absence of the intrusive effect which may be experienced if the interviewer is present. However, the mail questionnaire has also drawbacks. The interviewer can not control and guide the process of answering questions. He can not help if the respondent gets mixed up. He can not explain concepts, and can not make use of display cards. Furthermore, wording of questions and layout of the questionnaire is even more important. Moreover, the lack of control by the interviewer may cause a high nonresponse rate.

In many surveys among persons or households *face-to-face interviewing* will be prefered. Interviewers visit the selected addresses and attempt to get the questions answered in a personal conversation. A point for consideration is sending a letter to selected addresses anouncing the visit of the interviewer, explaining the purpose of the interview, and convincing the respondent of the confidentiallity of his answers. An unannounced visit may find the respondent not at home or reluctant to open the door. An announcement letter may even contain the telephone number or address of the interviewer. On the one hand, it gives the respondent the possibility to make an appointment for a more convenient date. On the other hand, there can also be an opposite effect: the respondent can call the interviewer to cancel the interview. Generally, face-to-face interviews produce high quality results, but this mode of data collections has the drawback of high costs. Interviewers must be trained and payed. A large part of the time of the interviewers is spent on travelling.

Another mode of data collection is *telephone interviewing*. The sampling frame is a telephone directory, but also random digit dialing is a way to select persons or addresses. Less interviewers are required for data collection. Furthermore, no time is spent on travelling. So, telephone interviewing is cheaper than face-to-face interviewing. But a conversation by telephone has restrictions. The interview may not be too long, and questions may not be to complex. A clear restriction is that only persons having a telephone can be interviewed. In some cases the population of telephone owners will not cover the target population. Particularly in lower socio-economic classes, and e.g. among students, the coverage rate is substantially lower.

In practice, the selected mode of data collection will always be a compromise between quality and costs. Whatever mode of data collection is used, the result will always be a pile of completed forms. The big problem with these forms is that they will never be completely error-free. So a subsequent data editing process is necessary. However, detected errors are very hard to correct, since the respondent is not available anymore. Therefore, it will pay off to design a way of data collection that allows for error detection and correction during the interview. Computer assisted data collection is an attempt in that direction. It will be discussed in the next section.

5.2. Computer assisted data collection

In computer assisted interviewing the paper questionnaire is replaced by a computer program containing the questions to be asked. The computer takes control of the interviewing process. It performs two important activities:

1. *Route control.*
   The computer program determines which question to be asked next, and displayes that question on the screen. Such a decision may depend on the answers to previous questions. Hence it relieves the interviewer of the task of taking care of the correct route through the questionnaire. As a result, it is not posible anymore to make route errors.

2. *Error checking.*
   The computer program checks the answers to the questions which are entered. Range checks are carried immediately after entry, and consistency checks after entry of all relevant answers. If an error is detected, the program gives a warning, and one or more of the answers concerned can be modified. The program will not proceed to the next question until all detected errors have been corrected.

Application of computer assisted data collection has three major advantages. In the first place it simplifies the work of interviewer (no more route control), in the second place it improves the quality of the collected data, and in the third place data is entered in the computer during the interview resulting in a clean record, so no more subsequent data entry and data editing is necessary.

In the last decade an increasing use of computers is made in data collection. An already established technique is Computer Assisted Telephone Interviewing (CATI), see e.g. Nichols and Groves (1988[a], 1986[b]). This is a form of telephone interviewing in which the computer selects the proper question to be answered. This question is displayed on the computer screen, and thus can be asked by the interviewer. The answer is typed in and the computer checks it for range errors and consistency errors. If an error is detected, the computer warns the interviewer that something is wrong, and corrections can be made.

More recently is the technique of Computer Assisted Personal Interviewing (CAPI). It is a form of face-to-face interviewing in which interviewers use a small laptop computer to ask the questions and to record the answers, instead of the traditional paper form. Dutch experiences with CAPI are discussed in the next section.

5.3. Dutch experiences with CAPI

At the end of 1983, small computers came on the market that could easily be carried around in, say, a shopping bag. They weighted less than 2 kilograms, and had a keyboard and a very readable screen. The CBS purchased three machines of the type NEC PC-8201A, with 32 Kb Rom and three banks of 32 Kb Ram. They had a screen of 8 lines with 40 characters. The first experiment was planned in May 1984, in the context of a Price Survey. In this survey prices of commidities are observed in order to calculate the monthly consumer index numbers. The interviewers visited shops with the computer, recorded prices, reported changes in commodities, located and recorded new shops in case a specified shop closed down, and answered special

questions in extra questionnaires. Although there were some problems with the equipment, the experiment was a success for various reasons. In the first place, it became possible to formulate the requirements a laptop computer should sataisfy. In the second place, the experiment showed that respondents (shop keepers) had no objections against computer assisted interviewing. And in the third place, despite experienced inconveniences with the equipment, the interviewers developed a positive attitude towards this way of working. And training required only a couple of hours.

After the succes of the Price Survey experiment, it was decided to carry out another experiment, to test the reactions of the general public when confronted with a laptop computer. Another three machines were purchased, and this time a regular questionnaire was programmed with questions borrowed from the Consumer Sentiments Survey. A total of 173 interviews were conducted with the laptop computers and 167 interviews were carried out in the traditional way with paper and pencil. A comparison of the results showed that the use of laptop computers did not increase the nonresponse. There were no indications of feared psychological ('Big Brother') effects. The interviewers quickly learned to handle the equipment after only one hour of training. The reacted positively and realistically to the new developments.

Figure 5.1. Computer assisted interviewing with a laptop computer

Due to the success of these experiments, the Netherlands Central Bureau of Statistics started in 1987 with full scale use of CAPI in a regular survey: the continueous Labour Force Survey. A new type of laptop computer was selected: the EPSON PX-4, running under de operating system CP/M. The choice was based on criteria like size, weight, performance, capacity, readability of the screen, convenience of the keyboard. About 300 machines were purchased and distributed under the interviewers. Each month the interviewers, equiped with laptops, visit 10,000 addresses. After a day of interviewing, they return home and connect their computer to the power supply to recharge the batteries. The laptop computer is also connected to a telephone and modem. At night the collected data is automatically transmitted to the statistical office. The next morning the interviewer finds a recharged machine with a clean workspace, ready for new interviews. More details about these CAPI surveys can be

found in Van Bastelaer et al. (1987[a], 1987[b]) and Bemelmans-Spork and Sikkel (1985[a], 1985[b]).

In the CAPI surveys discussed above the interviewing programs were dedicated, taylor-made programs. However, for large scale application of CAPI in many surveys, we need a standard tool for the development of CAPI programs. The Blaise System to be discussed in the subsequent chapters, is such a tool.

CHAPTER 6. THE BLAISE SYSTEM

6.1. What is Blaise?

Blaise is the name of a system for efficient collection and processing of survey data. The basis of the Blaise System is the Blaise language, which is used to create a formal specification of the structure and contents of the questionnaire. This specification acts as a knowledge base out of which the system extracts information necessary for automatic generation of various computer programs for data collection or data processing. In this way the Blaise System controls and coordinates a large part of the survey process: design of the questionnaire, data collection, data editing and data analysis.

The Blaise System runs on microcomputers (or networks of microcomputers) with the operating system MS-DOS. It is designed for use in a subject-matter department. It is not necessary to be a computer expert to use the Blaise System. It is specifically designed to enable subject-matter experts to input their knowledge in the system, and to take care of all subsequent processing steps.

In the Blaise philosophy the first step in carrying out a survey is to design a questionnaire in the Blaise language (questions, possible answers, routing, and relationships between answers). This specification is stored in the computer. Each computer program generated by the system makes use of this information, and in this way the Blaise System enforces consistency in all steps of data processing.

Blaise derives its name from the famous French theologian and mathematician Blaise Pascal (1623-1662). Pascal is not only famous for his contributions to science, but also for the fact that his name was given to the well-known programming language. The Blaise language has its roots, for a large part, in this programming language.

The Blaise System can produce two kinds of programs. In the first place, a CADI-program can be generated. CADI stands for Computer Assisted Data Input. The CADI-program is an interactive system for data input and data editing. The subject-matter specialist works through a pile of forms with a microcomputer, processing them one by one. He enters answers to questions in the proper fields and after completion of the form, he activates the check option, to test routing and consistency (range errors are checked during data input). Detected errors are reported and explained on the screen. Errors can be corrected, by consulting the form or calling the supplier of the information. After elimination of all errors, a clean record is written to file.

The CADI-program concentrates on processing data which have already been collected on paper forms. However, the BLAISE System can also be used for data collection. The system is able to produce CAPI and CATI software. In the case of CAPI (Computer Assisted Personal Interviewing) the interviewing program is loaded into a laptop computer. The interviewer takes this computer to the homes of the respondents. There the program takes control of the interview. In the case of CATI (Computer Assisted Telephone Interviewing) interviewers call the respondents from a central unit, and carry out the interview by telephone, with the help of the computer on their desk. Again the interview program takes control.

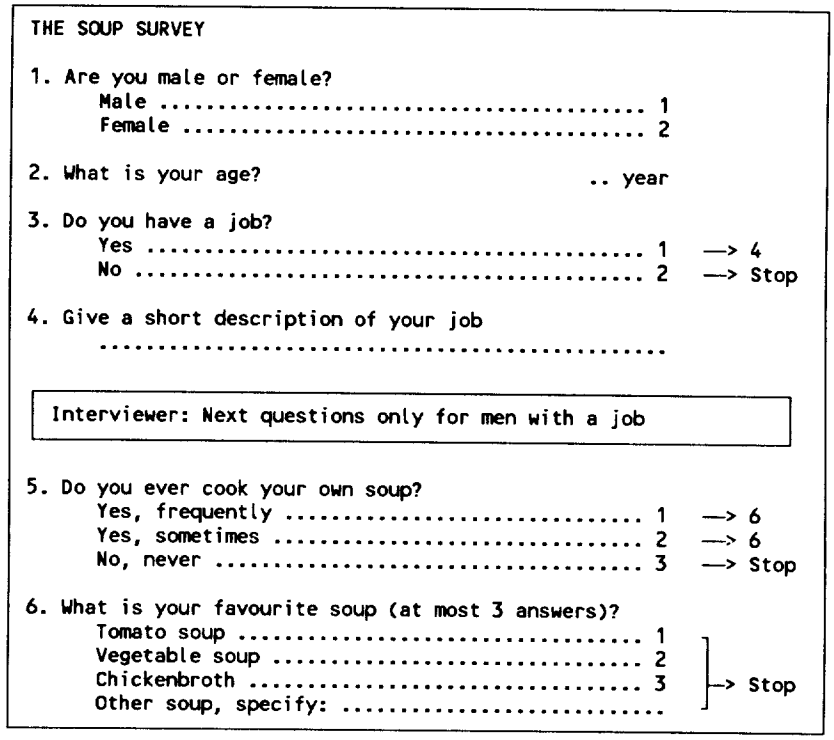
The Blaise System is not only an efficient tool in data collection and data editing, it can also be helpful in data analysis, although it does not claim to be a system for data analysis. There are already enough statistical packages on the market. However, Blaise is able to create an interface to a number of packages. Presently, setups can be produced for the statistical packages SPSS and Stata, and for the database package Paradox. The system also has an interface to the home-made tabulation package Abacus.

Generation of a Blaise program proceeds in a number of steps, First, a text editor is used to enter the Blaise specification of the questionnaire. Next, the systems checks the questionnaire for syntax errors. If an error is detected, the system returns to the text editor and positions the cursor on the approximate spot of the error. After correction, the specification is checked again. If no errors are detected, the specification is transformed into Pascal source code, that can be compiled. If the compilation is completed succesfully, an executable program is obtained.

6.2. A simple questionnaire

This section shows how a simple questionnaire can be specified in the Blaise language. Making such a specification shows some resemblance with preparing a recipe, say for soup. This approach will be followed in the explanation. But first, we start with a small, traditional questionnaire, see figure 6.1.

Figure 6.1. A simpe questionnaire



The questionnaire consists of questions, descriptions of possible answers, jump instructions to other questions, and instructions for the interviewer. Figure 6.2 contains a Blaise specification of this questionnaire.

Figuur 6.2. The Blaise specification of figure 6.1.

```
QUESTIONNAIRE Cooking;

QUEST
   SeqNum  "Sequence number of the interview?": 1..10000 (KEY);
   Sex     "Are you male of female?": (Male, Female);
   Age     "What is your age?": 0..120;
   Job     "Do you have a job?": (Yes, No);
   Descrip "Give a short description of your job": STRING[40];
   CookSoup "Do you ever cook your own soup?":
            (Yes      "Yes, frequently",
             Sometime "Yes, sometimes",
             No       "No, never");
   Soup     "What is your favourite soup?":
             SET [3] OF
            (Tomato   "Tomato soup",
             Vegetabl "Vegetable soup",
             Chicken  "Chickenbroth",
             Other    "Other soup");
   OthSoup  "Specify which other soup": STRING[20]

ROUTE
   SeqNum; Sex; Age; Job;
   IF Job = Yes THEN
      Descrip;
      IF Sex = Male THEN
         CookSoup;
         IF CookSoup IN [Yes, SomeTime] THEN
            Soup;
            IF Soup = Other THEN OthSoup ENDIF
         ENDIF
      ENDIF
   ENDIF

CHECK
   IF (Age < 12) THEN CookSoup = No ENDIF

ENDQUEST.
```

In the specification some words, like QUESTIONNAIRE and ENDIF, are printed in upper case. These words have a special meaning in the Blaise language. Their use is reserved for special situations, and therefore they are called reserved words. To emphasize this special meaning they are printed in capitals. However, reserved words may also be typed in lower case. The first line of the specification in figure 6.2 (QUESTIONNAIRE Cooking) is the identification of the questionnaire, and the end of the specification is indicated by the reserved word ENDQUEST.

In preparing a recipe three steps can be distinguished: laying ready the ingredients, mixing the ingredients, and tasting the result.

Step 1: Preparing the ingredients

For the preparation of a recipe, first all ingredients must be layed ready. Likewise, in a questionnaire first all questions must be specified. This takes place in the *quest paragraph*. Every question specification follows a simple scheme. First, the question name is specified, followed by the question text. This is the text of the question as presented to the respondent or printed on the paper form. The question text must be placed between quotes, and followed by a colon. The last part is the answer definition, describing the possible answers to that question. The question paragraph starts with the reserved word QUEST. Question definitions are separated by a semicolon.

The difference with traditional questionnaires is that the *question numbers* in figure 6.1 are replaced by *question names* in figure 6.2. So we do not talk about question 2, but about question Age, and refer to question Soup instead of question 6. It is important to identify questions by names instead of numbers. It improves the readability of the questionnaire, and problems are avoided in case questions have to be added or deleted.

The quest paragraph of figure 6.2 contains the specification of eight questions. There is one question (SeqNum) which does not appear in figure 6.1. Moreover, this question has a special attribute KEY. Forms processed by Blaise must have a unique identification key. Such a unique key is established by assigning one ore more questions the attribute KEY. In this case there is one key question SeqNum. Consequently, different forms may not have equal sequence numbers.

The Blaise language offers different types of questions. In the first place, it is possible to define an *open question*. On such a question any text is accepted as an answer, provided the lenght of the text does not exceed the specified maximum length. An example is the question Descrip. On a *numerical question* a number is expected as an answer. This number must be in the specified range. The question Age is an example. For a *precoded question* an answer must be selected out of a specified list of possibilities. The question CookSoup in figure 6.2 is an example of such a precoded question. Each possible answer is defined by a short *answer name* (e.g. Yes) and, optionally, a longer *answer text* (e.g. "Yes, frequently"). The answer name is used internally, and in the CADI program, to identify the answer; the answer text is presented to the respondent. The possible answers are identified by names and not by numbers. Just like in the case of questions names, the use of answer names improves readability and maintanability of questionnaire specifications. Sometimes the respondent must be allowed to select more than one answer from a list. For this case Blaise offers the *set question*. Such a question is created by adding the reserved words SET OF to a precoded question. Optionally, the maximum number of answers to be selected may be specified between square brackets. The question Soup is a set question. Other question types supported by Blaise are *date questions* (accepting a date in various formats) and *array questions* (in fact a series of questions, all with the same question text and possible answers).

There are two pre-defined answers which can always be given in response to a question of any type: DONTKNOW and REFUSAL. These answer possibilities do not have to be defined. They can be assigned with the special function keys.

Part 2: Mixing the ingredients

In the preparation phase all ingredients are mixed in the proper order. The analogue for the Blaise specification is the *route paragraph*, announced by ROUTE. It describes which questions are to be asked under which conditions and in which order. There is a simple rule: writing down the name of a question means asking it. In the example the questions SeqNum, Sex, Age and Job are submitted to every respondent. Only respondents with a job are asked to give a description of their job. And only males with a job are asked whether they ever cook their own soup. Only people who cook their own soup are asked to specify their favourite soup.

The example does not present the most general form of the conditional routing. It is also possible to specify questions which are asked if the condition is not satisfied:

```
IF condition
THEN
     ask questions
ELSE
     ask other questions
ENDIF
```

The route paragraph forms the heart of the Blaise specification. It describes the structure of the questionnaire in a compact and readable way. The Blaise way of route specification is much more powerful than the traditional way which uses jump-instructions. That is illustrated in figure 6.1 where special instructions for the interviewer are required to keep the respondent on the correct route.

Part 3: Tasting the result

When a recipe is ready, the result should be tried out to check whether it is tasteful. Likewise, checks should be carried on questionnaire data to see whether there are errors. Traditional paper questionnaires contain questions and routing, but no checks. Blaise offers the possibility of including consistency checks in the specification of the questionnaire.

Range checks establish whether an answer to a question falls within the domain of valid questions according to the definition of the question. Range checks are generated automatically from the question definition, and carried out instantaneously after the entry of the answer. Route checks are only relevant if the data is collected on paper forms. *Route checks* verify whether the correct route through the questionnaire has been taken, i.e. whether all relevant questions have been answered, and all irrelevant questions have been skipped. Route checks are derived from the route paragraph and, therefore, do not have to be specified explicitly. *Consistency checks* establish whether answers to several questions are consistent. While answers to questions can be within the valid domain, the combination of answers can lead to an inconsistency. Consistency checks are specified in the check paragraph. The structure of the check paragraph shows much resemblance with that of the route paragraph, with the exception that route instructions are replaced by conditions that have to be verified.

The check paragraph in the example should be interpreted as: *Respondents under 12 years do not cook their own soup.* The check describes a condition which should be satisfied. If the condition is not satisfied, the CADI, CAPI or CATI program will produce an error message. This error message will contain the text of the check as specified in the check paragraph. If this is a complex mathematical expression, it will not be very helpful in explaining what is wrong. Therefore, it is also possible to assign texts to checks. In that case the texts will be used in the error message instead of the mathematical expressions. The check paragraph example could also be formulated as

```
CHECK
   IF (Age < 12) "the respondent is under 12 years" THEN
       CookSoup = No "he/she does not cook his/her own soup"
   ENDIF
```

Then, the displayed error message will look like *IF the respondent is under 12 years THEN he/she does not cook his/her own soup*

If a check in the check paragraph discovers an error, it is called a hard error. A *hard error* relates to a real error. There is a problem which must be solved. Without correction of hard errors, the form will never be declared 'clean', i.e. error-free. Blaise also knows the concept of soft errors. A *soft error* does not necessarily indicate a real error. A soft error should be interpreted as a warning, a signal that something implausible has been encountered. If a soft error is reported, the user can decide to suppress this error message without making corrections. Checks for soft errors must be specified in the *signal paragraph.* The structure of that paragraph is identical to that of the check paragraph. The only difference is that the signal paragraph starts with SIGNAL instead of CHECK.

CHAPTER 7. COMPLEX QUESTIONNAIRES

7.1. Subquestionnaires

Chapter 6 presented some elements of the Blaise specification: the quest, route, check, and signal paragraph. These elements can be used to specify simple questionnaires. However, in practice questionnaires are often larger and much more complicated. Blaise has a number of facilities for efficient treatment of such questionnaires. Some of these facilities will be discussed in this chapter.

One of the more useful concepts is that of the *subquestionnaire*. Large questionnaires often deal with several, distinct subjects. Blaise promotes a modular approach to development of questionnaires. The basic idea is to construct a small questionnaire for each subject. Such a subquestionnaire has all the elements of a simple questionnaire, i.e. questions, routing, checks, etc. Once all subquestionnaires are completed and tested, they are brought together in one large questionnaire.

An example illustrates this approach. Figure 7.1 presents, in a schematic way, a questionnaire containing eight questions about three subjects: questions 1, 2 and 3 relate to demographic characteristics, questions 4, 5 and 6 deal with work, and questions 7 and 8 are about cooking.

Figure 7.1. A questionnaire about three subjects

```
1. What is your sex?
2. What is your age?
3. What is your marital status?

4. Do you have a job?
5. Give a short description of your job
6. What is your yearly income?

7. Do you ever cook your own soup?
8. What is your favourite soup?
```

Applying a top-down approach to development of this questionnaire, we start with the route-paragraph of the overall questionnaire. It could look like this:

```
ROUTE
   General; Work; Cooking
```

The structure of the questionnaire is clear. There are three parts. With one command (General) a number of general questions are asked. The next command (Work) takes care of the questions about work, and the final command (Cooking) deals with the cooking questions. The questionnaire consists of three subquestionnaires, which are dealt with one after the other. In Blaise such subquestionnaires are called *blocks*. Blocks can be interpreted in two ways. In the first place, a block is a sub-questionnaire, i.e. a small questionnaire with all the properties of a questionnaire. The only differences are the reserved words BLOCK and ENDBLOCK at the beginning and end, instead of QUESTIONNAIRE and ENDQUEST. In the second place, a block is a special *type definition*. A question of this type can be defined in a quest-paragraph. If such a 'super question' is encountered in the route-paragraph, the whole

corresponding subquestionnaire is executed, instead of one question. Figure 7.2 contains an example of the Blaise specification of figure 7.1.

Figure 7.2. Blaise specification of figure 7.1.

```
QUESTIONNAIRE Demo;

BLOCK GenPart;
   QUEST
      SeqNum  "Sequence number of the interview?": 1..10000 (KEY);
      Sex     "Are you male or female?": (Male, Female);
      Age     "What is your age?": 0..120;
      MarStat "What is your marital status?": (Married, Unmarried)
   ROUTE
      SeqNum; Sex; Age; MarStat
ENDBLOCK;

BLOCK WorkPart;
   QUEST
      Job      "Do you have a job?": (Yes, No);
      Descrip  "Give a short description of your job?": STRING[40];
      Income   "What is your yearly income?": 0..1000000
   ROUTE
      Job; IF Job = Yes THEN Descrip ENDIF; Income
ENDBLOCK;

BLOCK CookPart;
   QUEST
       CookSoup "Do you ever cook your own soup?":
                (Yes       "Yes, frequently",
                 Sometime  "Yes, sometimes",
                 No        "No, never");
       Soup     "What is your favourite soup?": SET OF
                (Tomato    "Tomato soup",
                 Vegetabl  "Vegetable soup",
                 Chicken   "Chickenbroth",
                 Other     "Other soup")
   ROUTE
      CookSoup; IF CookSoup = Yes THEN Soup ENDIF
ENDBLOCK;

QUEST
   General: GenPart; Work: WorkPart; Cooking: CookPart;

ROUTE
   General; Work; Cooking

ENDQUEST.
```

To be able to store the data, the Blaise System reserves space for every question in the questionnaire. There are situations in which this approach consumes more space than really is necessary. Suppose we have a questionnaire containing a block with 12 questions. Furthermore, suppose that this block of questions is to be asked of each member of the household. So we introduce an array question, in which each element is a question of a block-type. The upperbound of this array must be such that every household can be dealt with. Suppose we take an upperbound of 20, implying we expect households to have no more than 20 members. This implies that the system reserves space for 12 x 20 = 240 questions. That consumes a lot of memory space, only a small part of which will be used in most cases.

It is possible to define a relational storage structure, by assigning the special attribute SUBFILE to a block. This causes all data corresponding to that block to be stored in a separate file, and only for those instances in which one or more of the questions in the block are really answered; empty occurences of the block will not be stored. The next section contains an example of such a relational file structure.

7.2. Tables

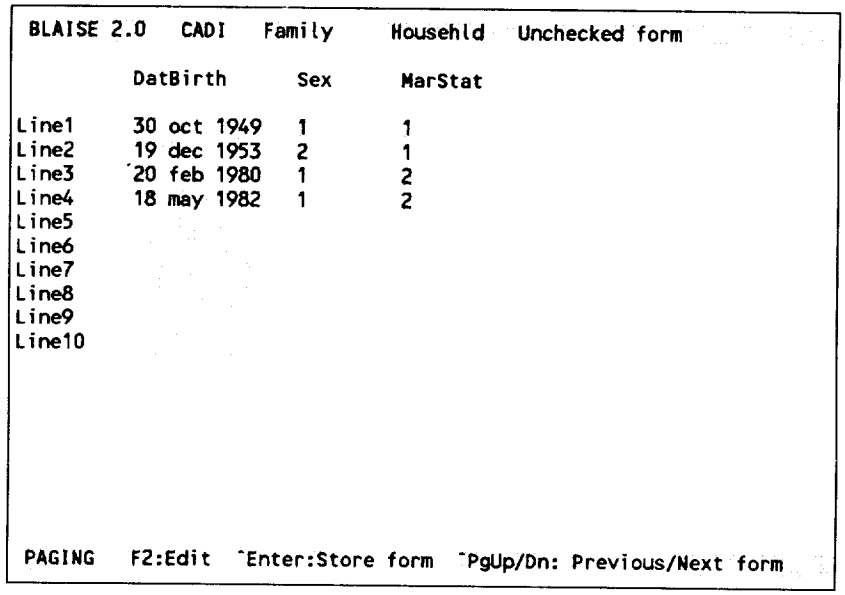
Many paper questionnaires contain questions grouped into tables. An example is a household table. The rows of the table represent the members of the household, and the columns denote questions to be submitted to the members.

Figure 7.3. Blaise specification of a table

```
QUESTIONNARE Family;

QUEST
   SeqNum "Sequence number of interview": 1..10000 (KEY);
   HHSize "What is the size of the household?": 1..10

TABLE HHTable;
   VAR
      Person: INTEGER
   BLOCK HHLine (SUBFILE);
      QUEST
         DatBirth "Date of birth?": DATETYPE;
         Sex      "Sex": (Male, Female);
         MarStat  "Marital status": (Married, Unmarried)
      ROUTE
         DatBirth; Sex; MarStat
   ENDBLOCK;
   QUEST
      Line: ARRAY [1..10] OF HHLine
   ROUTE
      FOR Person:= 1 TO 10 DO
          IF Person <= HHSize THEN Line[Person] ENDIF
      ENDDO
ENDTABLE;

QUEST
   Househld: HHTable

ROUTE
   SeqNum; HHSize; Househld

ENDQUEST.
```

Tabels can also be defined in a Blaise specification. The definition is equal to that of a block, with the exception that BLOCK and ENDBLOCK are replaced by TABLE and ENDTABLE. The rows of the table are block-type questions defined inside or outside the table. Each row-block must contain the same number of questions. Figure 7.3 contains a Blaise specification of a household table, recording date of birth, sex and marital status of at most 10 persons per household. The rows correspond to block-questions Line[1] upto Line[10]. Each row is of the type HHline, and contains three questions: DatBirth, Sex and MarStat. So the table has three columns. Since the block HHLine is assigned the attribute SUBFILE, all data relating to this block is written to a separate subfile, but empty rows are not stored. Figure 7.4 shows how this table is displayed on the screen by the CADI-program.

Figure 7.4. A table on the screen



7.3. Other feautures

Blaise contains many other features which enable the development of complex questionnaires, and simplify the use of such questionnaires. Some of these features are mentioned below.

Use of external information

Sometimes checks must be carried out that make use of information in other files. An example is a business survey in which financial data is collected. Particularly large businesses will be surveyed again and again, and this makes it possible to compare the figures of the present year with the figures of past years. Strange deviations may indicate that something is wrong with the data.

Blaise has a facility to extract information from other files and to use it in checks. Such a file may either be an index-sequential file that is constructed using one of the programs in the utility programs of the system, or a Blaise data file created by another Blaise program. To be able to use an external file, Blaise must know how the file is composed, and this is specified in an *external paragraph*. Once an external file has been defined, it can be used in two ways. In the first approach it is used as a *lookup table*. It can be used to check whether a certain entry really exists in the file. In the second approach a record is read from the file, and the information can be used in various ways.

Interactive coding

An important part of processing surveys is the coding of answers to open questions. Examples of coding are the classification of purchased goods, social status, occupation, or industrial activity. Traditionally, coding is carried out manually by subject-matter experts. Therefore, it is both time consuming and expensive.

In many cases codes to be assigned correspond to a *hierarchical classification*. The code consists of a number of digits, and the first (left most) digit of the code defines a global classification, and each following digit represents a refinement of the category indicated by the previous digits. An example of a hierarchical classification is the ISCO, the International Standard Classification of Occupation, of the ILO (the International Labour Organization). The first digit produces a classification in 10 major groups. Each major group is split into a number of minor groups. The process of refinement continues until a final classification into 5 digits is reached. Each digit in the code of a hierarchical classification has a specific meaning. For example, a computer operator has code number 34220, which is composed of major group 3 (Clerical and Related Workers), minor group 34 (Computing Machine Operators), and finally 34220 (Electronic Computer Operator).

A computer assisted approach to coding is implemented in the Blaise System. The coding module can be used in two different ways, denoted by stepwise coding and dictionary coding. *Stepwise coding* starts by entering the first digit of the code, by selecting the proper category from a menu. After entering a digit, a subsequent menu is presented containing a refinement of the previously selected category. So, the description becomes more and more detailed until the final digit is obtained. In the case of *dictionary coding* a verbal description is entered, and the computer tries to locate it in an alphabetically ordered list. The list is displayed, starting at the point as close as possible to the entered description. The list can be made so that almost any description, including permutations, is present. Stepwise coding can be combined with dictionary coding in a very simple way. Start with stepwise coding until a point is reached where it is not clear any more which category to select. Then change to dictionary coding. The coding module will display an alphabetically ordered list, only containing a subset of descriptions of which the first group of digits is identical to those already selected.

To implement computer assisted coding, a special type of question must be defined in the Blaise specification: a *code question*. This is accomplished by using the reserved word CODE in the answer definition and specifying a number of parameters and files.

Computations

In the route, check, and signal paragraph computations may be carried out. In these computations questions and variables may be used. Before variables are used, they must first be defined in a *variables paragraph*. Results of computations can be assigned to variables. The value of the variable may be used to determine the correct route, or to carry out checks. In the check and signal paragraph results of computations may also be assigned to questions, thus making *imputation* possible: if the answer to a question is not specified or incorrect, a substitute can be computed. A large number of standard functions is available for use in computations. Examples are ROUND, ABS, EXP, and SUM.

Variable texts

Sometimes the need is felt to adapt texts of questions or possible answers to the situation in which they are used. Blaise allows the possibility of including values of variables or answers to questions in question texts, answer texts, or error messages. This is achieved by specifying the name of the variable or question in the text, preceeded by a dollar-sign. In the example the question Transprt is included in both the question text and the answer texts of the question Reason:

```
QUEST
   Transprt
      "How do you usually go to your work?":
      (train, bus, tram, metro, car, motcycle, bicycle);
   Reason
      "Why do you go by $Transprt and not by car?":
      (NoJam    "No problems with traffic jams",
       Comfort  "By $Transprt is much more comfortable",
       Environ  "Use of $Transprt is better for the environment",
       Health   "Going by $Transprt is better for your health",
       NoCar    "Does not have a car")

ROUTE
   Transprt;
   IF Transprt <> Car THEN
      Reason
   ENDIF
```

If the respondent answers that he usually goes by bicycle, the CAPI program will present him with the following question Reason:

```
Why do you go by bicycle and not by car?
1: No problems with traffic jams
2: By bicycle is much more comfortable
3: Use of bicycle is better for the environment
4: Going by bicycle is better for your health
5: Does not have a car
```

Use of variable answer texts is only meaningful in CAPI/CATI programs; In CADI programs answer texts are never shown on the screen.

CHAPTER 8. BLAISE PROGRAMS

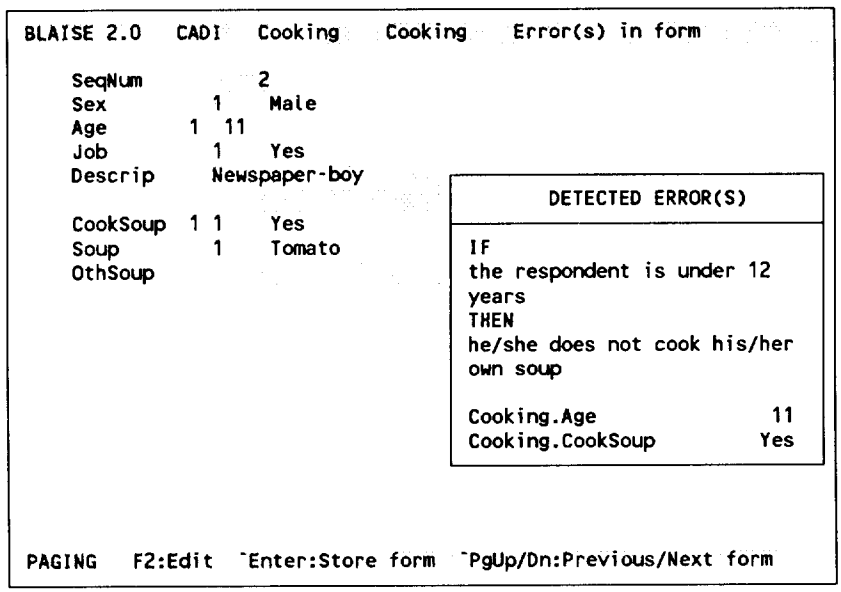
8.1. The CADI program

The Blaise CADI program offers the user an interactive system for data input and data editing. The screen layout of the CADI program is the result of an attempt to reconstruct a paper form on a computer screen. However, redundant information is removed: a question is indicated by its name, rather than by the full question text. The full information is always available via help windows.

The route statements in the route paragraph are interpreted as checks. Route errors are treated in the same way as consistency errors. The CADI program does not force the user to answer certain questions or skip other questions. The user may answer any question and has complete freedom to page through a questionnaire. This was done to encourage a user to copy the form exactly, when entering data.

While entering data, the user is not bothered by route or consistency checks. This allows for fast data entry and, again, encourages a user to copy his form exactly. Checks are performed when the user presses the check key, or at the end of the form. Thereafter, the screen changes slightly. The top right corner reports the presence of errors. The names of questions involved in errors are followed by error counts. A user may jump to such a question (keys are available to jump to questions involved in errors). An explanation of the problem is given if a function key is pressed. Such an error message consists of a description of the error, and a list of the questions involved. Figure 8.1. contains an example of a detected consistency error. Error counts are given for the questions Age and CookSoup, indicating that those questions are involved in an error.

Figure 8.1. A consistency error in a CADI program



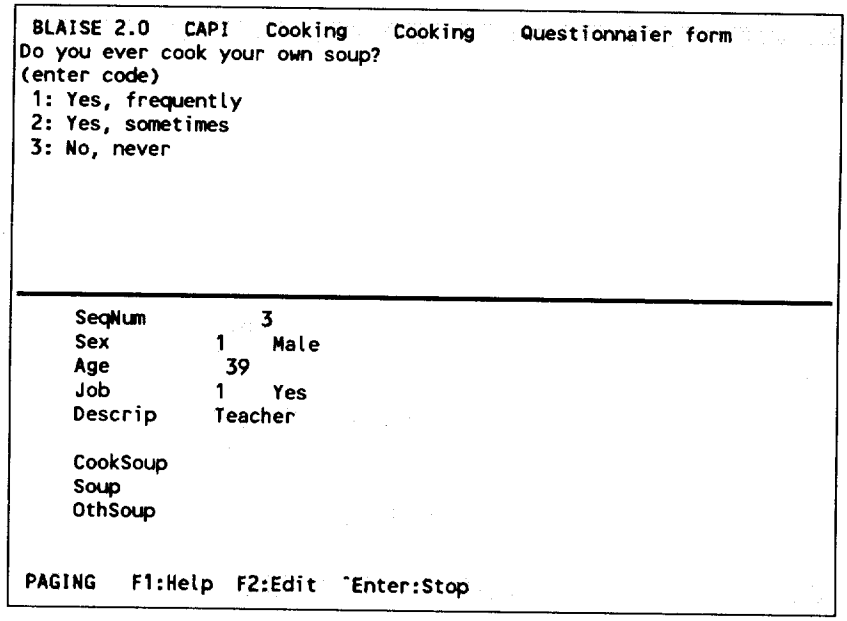
As mentioned before, Blaise distinguishes *hard* and *soft* errors. Hard errors must be corrected in order to obtain a clean form. Soft errors just give a warning that something might be wrong. For soft errors separate counts will appear on the screen, just in front of the hard error counts. Report of a soft error can be suppressed.

8.2. The CAPI/CATI program

The Blaise System generates programs for CAPI or CATI. For Blaise, CAPI and CATI programs are identical, so the system has no separate options for CAPI and CATI. The only difference between CAPI and CATI is the hardware used: a laptop computer for CAPI, and a desktop computer for CATI. Of course, there is a difference between CAPI and CATI with respect to the management of interviews: call management in CATI and address management in CAPI. The Blaise System takes care of both types of management by introducing a special block (called APPOINTMENT). Questions in this block can record all kinds of management information, like date and time of appointment, addresses, telephone numbers, speciale instructions, etc. Questions can be stored in a special appointment file. This file can be consulted by the interviewer, and interviews can be activated from this file. After succesful completion of an interview, its entry in the appoinment file can be deleted.

CAPI/CATI programs usually provide a user with just one question at a time. As soon as this question is answered, it disappears from the screen, and the next question is displayed. This gives the interviewer only a very limited feed-back on the entered answers. It is difficult for the interviewer to keep track of where he exactly is in the interview. The CAPI/CATI programs generated by the Blaise System apply a split screen approach. The upper part of the screen shows the question and possible answers; the lower part of the screen is a window on the current page of the questionnaire. Thus, the user can view a few answers a the time and relate them together. See figure 8.2 for an example of a CAPI screen.

Figuur 8.2. A CAPI screen



The CAPI/CATI program applies *dynamic routing*, i.e. the software takes control of the routing. After an answer has been entered, the program determines which question is to be asked next, and automatically jumps to that question. This implies that no questions can be skipped that must be answered, nor that questions can be answered that must be skipped. Thus, dynamic routing implies that an interviewer cannot violate the routing structure of the questionnaire. In this respect the CAPI/CATI program differs form the static routing as supported by the CADI program. The difference arises because of the completely different use of CADI and CAPI/CATI. In CADI, a paper form is entered in the computer, after which the errors are corrected; in CAPI/CATI, we want to prevent anything from going wrong at all.

Just as in a CADI program, the user can move backward through the questionnaire, either one question at the time, one or more screens, or to the start of the interview. Moving backward through the questionnaire is restricted by the routing structure of the questionnaire. A user can only move backward to questions that were previously answered. An interviewer can always change a previous questions. This implies that the routing can be changed anytime during the interview

The CAPI/CATI program applies dynamic error checking. As soon as a consistency error is encountered, an error message is displayed on the screen, together with a list of all questions involved. Via a menu, the user may select the question to jump to in order to correct the error. Hard errors always have to be corrected; soft errors may be suppressed.

It is always possible to interrupt an interview. If the respondent wants to postpone the interview, the APPOINMENT block is activated, and a date and time for the continuation of the interview can be recorded. To deal with the case of nonresponse, a special NONRESPONSE block may be included in the questionnaire. This block should contain the questions asked in case of refusal to continue. The NONRESPONSE block and the APPOINTMENT block are not part of the normal routing. They are only activated in the case of an interrupted interview. Therefore, no questions of the type NONRESPONSE or APPOINTMENT have to be defined.

8.3. ASCII conversion

Data entered with a Blaise program are stored in a special format, which makes it is impossible to use the Blaise files straight away in other programs and packages. Therefore, the Blaise System offers various tools to convert the data files to ASCII files. The ASCII files to be produced can be in *fixed format* (the values of a variable are in fixed positions in the record), or *free format* (the values are not in fixed positions, but separated by a special symbol). It is possible to convert all forms, or only clean forms (no errors), dirty forms (hard errors) or suspect forms (soft errors).

Generated ASCII files should be properly documented. The utility menu can produce a dictionary describing the variables (names, labels, positions) in the ASCII file.

It is not always possible to enter data with a Blaise program. For example, data may be sent to the statistical office in machine readable form (e.g. diskette or tape), or the amount of data is very large and simple, and therefore entered in the traditional way, at high speed and virtually without error checking. In both examples the result is a raw, unchecked (dirty) data file. Such files can be imported in the Blaise System, if a Blaise specification is made which fits the structure of the data file. With this specification, one of the utilities performs the conversion. If a CADI program is generated too, all converted records may be checked, and assigned the status clean, dirty or suspect in one run. After that, a subject-matter specialist processes the incorrect forms in the usual interactive way.

8.4. Interfaces

Blaise data files can be made suitable for use by other packages or programs. In the first place, the Blaise data files can be converted to ASCII files, as described in the previous section. In the second place, a setup file can be created. Such a setup file contains a description of the ASCII files in a format which can be read and understood by the particular package.

In this way a setup can be created for the statistical package SPSS/PC. By simply running the setup from SPSS, an SPSS system file is created. In an analogous way an interface to the statistical package Stata is obtained. Two files (a dictionary file and a do-file) are generated. After running the do-file from Stata, a simple save-command is sufficient to produce a Stata system file. The Blaise System also has a link to the home-made tabulation package Abacus. Since this package directly reads Blaise data files, no conversions is necessary. Therefore summary tables can be produced very efficiently.

Sometimes the data is not used for statistical analyses. In case simple reports and summaries are needed, e.g. for management purposes, it should be possible to import the data into a database package. Therefore, the Blaise System offers an interface to the database package Paradox. A so-called script file can be produced, which can be run from inside Paradox, with a database as result.

If the Blaise data is stored in a relational structure, there are a number of subfiles. With the exception of Paradox, most packages will not be able to handle this. In that case, there are two possibilities: setups are generated for each subfile, or the relational structure is removed by generating one large flat ASCII file with a corresponding setup.

LITERATURE

Van Bastelaer, A.M.L., F.A.M. Kerssemakers and D. Sikkel (1987[a]): A test of the Continuous Labour Force Survey with hand-held computers: Interviewer behaviour and data quality. In: CBS select 4, Automation in Survey Processing, Netherlands Central Bureau of Statistics, Voorburg.

Van Bastelaer, A.M.L., L.M.P.B. Hofman and J.K. Jonker (1987[b]): Computer Assisted Coding of Occupation. In: CBS select 4, Automation in Survey Processing, Netherlands Central Bureau of Statistics, Voorburg.

Bemelmans-Spork, E.J. and D. Sikkel (1985[a]): Observation of prices with hand-held computers. Statistical Journal of the United Nations Economic Commission for Europe, vol. 3, no.2.

Bemelmans-Spork, E.J. and D. Sikkel (1985[b]): Data Collection with hand-held computers. Proceedings of the 45th session, International Statistical Institute, Book III, topic 18.3.

Bethlehem, J.G. (1985): LINWEIGHT User Manual. CBS report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G. (1986): ANOTA 3.0 User Manual. CBS report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G. (1988): The Program CORAN 2.0 for Correspondance Analysis. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G., A.J. Hundepool, M.H. Schuerhoff and L.F.M. Vermeulen (1989[a]): Blaise 2.0 / An Introduction. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G., A.J. Hundepool, M.H. Schuerhoff and L.F.M. Vermeulen (1989[b]): Blaise 2.0 / System Description. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G., A.J. Hundepool, M.H. Schuerhoff and L.F.M. Vermeulen (1989[c]): Blaise 2.0 / Language Reference Manual. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G., A.J. Hundepool, M.H. Schuerhoff and L.F.M. Vermeulen (1989[d]): Blaise 2.0 / CAPI/CATI Operator Guide. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G., A.J. Hundepool, M.H. Schuerhoff and L.F.M. Vermeulen (1989[e]): Blaise 2.0 / CADI Operator Guide. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Bethlehem, J.G. and W.J. Keller (1987): Linear Weighting of Sample Survey Data. Journal of Official Statistics 3, pp. 141-154.

Bethlehem, J.G., W.J. Keller and J. Pannekoek (1989): On Disclosure Control of Microdata. To be published in JASA.

Deming, W.E. (1986): Out of the Crisis. Cambridge University Press, Cambridge, Mass.

Keller, W.J. (1986): Statistical Software for microcomputers. COMPSTAT 1986, Proceedings in Computational Statistics, pp. 332-337.

Keller, W.J. and J.B. Crompvoets (1989): A Survey of Database Management Systems for PC and PC-Networks. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Keller, W.J. and K.J. Metz (1988): On the impact of new data processing techniques at the Netherlands Central Bureau of Statistics. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Keller, W.J., A. Verbeek and J.G. Bethlehem (1985): ANOTA: Analysis of Tables. CBS-report, Netherlands Central Bureau of Statistics, Voorburg.

Nicholls II, W.L, and R.M. Groves (1986[a]): The status of computer assisted telephone interviewing: Part I- Introduction and impact and cost and timeliness of survey data. Journal of Official Statistics 2, pp 93-115.

Nicholls II, W.L, and R.M. Groves (1986[b]): The status of computer assisted telephone interviewing: Part II- Data quality issues. Journal of Official Statistics 2, pp 117-134.

APPENDIX 1. STATISTICAL PACKAGES FOR MICROCOMPUTERS

Package	Use of re-sources	Ease of use	Data manage-ment	Presen-tation	Descr. statis-tics & testing	Multiv. analysis	Time series analysis	Speed	Precision	Value for money
ABC	+	°	°	-	-	-	-	°	-	°
BMDPC	-	-	°	°	°/+	°/+	-/°	°	°	°
GAUSS	+	-	°	°	-	-	°	+	+	°
ISP	+	°	°	°	°	-	-	+	-	°
MICROTSP	+	+	°	°	-	-	+	+	°	+
NCSS	+	+	°	°	+	°	-	°	°	+
PSTAT8	-	°	+	+	+	°	-	-	+	°
RATS	°	-	°	°	°	-	+	+	°	°
SAS/PC	-	°	+	+	°/+	-/+	-	-	+	°
SORITEC	°	°	°	°	°	-	+	-	+	°
SPSS/PC+	-	+	+	°/+	+	-/+	-	+	+	+
SST	+	+	+	°	°	-	°	+	°	+
STATA	+	+	+	°/+	+	-	-	+	+	+
STATGRAPHICS	°	°	°	+	+	+	°	-	+	+
SYSTAT	°	°	+	°	+	+	°	°	+	+

Legend: - = bad / fair
° = fair / good
+ = good / excellent

APPENDIX 2. DATABASE PACKAGES FOR MICROCOMPUTERS

Package	Use of re-sources	Rela-tional facil-ities	Design, Vali-dation	Forms	Query	Report	Appli-cation, 4G Lan-guage	Concur-rency, Security, Recovery	Ease of learn-ing	Ease of use	Per-form-ance
Paradox	+	+	°	+	+	+	+	+	+	+	+
DataEase	+	°	+	°	°	+	-	°	+	+	°
Dbase III+	+	-	-	-	-	°	°	°	°	°	°
Rbase V	°	°	+	+	°	+	°	°	°	°	°
Oracle	-	+	°	+	+	+	°	°	-	°	+
Informix	°	+	°	+	+	+	+	+	°	+	°
XDB II	+	+	°	+	+	+	°	+	+	+	+
Ingres	-	+	°	+	+	+	+	-	°	+	+
Magic	+	°	+	+	-	+	+	°	-	+	
Focus	°	°	°	°	-	+	°	°	-	°	

Legend: - = bad / fair
° = fair / good
+ = good / excellent