BLAISE TO ITS LIMITS
Applications in a market research environment

Dirk Sikkel and Mylène van Veldhoven
Research International Nederland

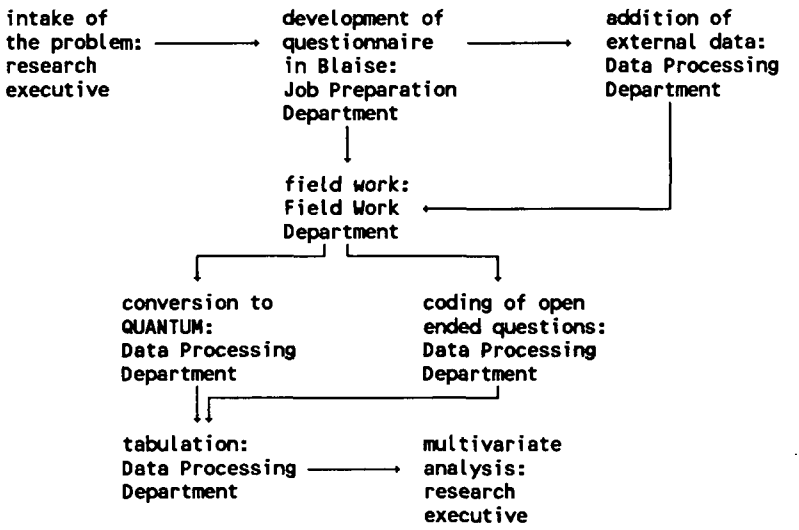
1. Introduction

During the past four years, Blaise has been used by Research International Nederland for a variety of purposes. The reason for buying Blaise was the necessity to carry out CAPI projects that were too complicated for the simple software (Ci2) that was already in use. After the first two projects, which were desperately difficult to manage, Blaise has gradually become a standard tool, embedded in the daily routine of the organization. Although not designed for a market research environment, it enabled us to solve a number of problems which at first sight seemed to be almost insurmountable; as a consequence, the problem solving ability of Blaise gave us considerable edge in some parts of the market. In this paper we will first describe the organization around the use of Blaise. Next some of the problems that have been solved by Blaise will be sketched. Finally we will comment on some of the shortcomings of the package in a market research environment.

2. Organization

The technical units in market research agencies usually are small. Only a few people are involved in the technical development of questionnaires and the subsequent data processing. At Research International, the most common procedure, based on such a small staff, is sketched in figure 1.

Figure 1. The organization around Blaise



Clients who have problems which require complex Blaise questionnaires are dealt with by a research executive with at least some knowledge of Blaise. Next the problem is taken to the Job Preparation Department, where a specialist constructs the actual Blaise questionnaire, usually in close cooperation with the research executive. Sometimes, external data are used in the interview, e.g. in a panel survey. The external data are included in the questionnaire as extra questions. The partly filled records are produced by the data processing department, based on the record descriptions which are automatically generated. After the fieldwork, the data are prepared for tabulation by QUANTUM, a tabulation package which is popular in market research. The interface between Blaise and QUANTUM is rather complex, as QUANTUM is based on 80-column records. The setup generation language of Blaise has, however, enabled us to write a satisfactory conversion program. Open ended questions first are converted to the format of a coding program that is a part of the Ci2

package. After coding these questions, the resulting data are added to the main data set. The SPSS-setup of Blaise is frequently used for preparation for multivariate analysis.

In the whole procedure, usually three people are involved: the research executive, the expert from the job preparation department and the expert from the data processing department. The intake of a job from the client is a difficult process. It is our experience that many clients usually have false expectations of both the possibilities and the impossibilities of computer assisted interviewing. When agreement has been reached with the client, it takes 2 to 4 days to build the (complex) questionnaire in Blaise. After that, the process continues more or less automatically, according to the principles of Integrated Survey Processing.

3. Trade-off processes

An important tool in market research is the trade-off. Respondents have to give preferences for product attributes. After a number of trade-off, utilities of the product attributes are computed which enable the user to construct the 'ideal' or 'optimal' product. This method, for which many specialized CAPI packages are available on the market, is called conjoint analysis. Research International has developed a variant of this technique which is often used for service quality studies. This technique is called SMART (Salient Multi-Attribute Research Technique).

Basically, SMART works as follows. A respondent is confronted with a number of items which are aspects of service quality. A typical example is:

How long do you normally have to wait in a queue?	1. more than 10 minutes 2. 5 to 10 minutes 3. 1 to 5 minutes 4. less than 1 minute

Apart from waiting time, such items deal with friendliness of staff, competence of staff, information, complaints etc. The respondents score one or more companies with respect to these items and indicate the

importance of the items. The complicated part (with respect to Blaise) starts where the respondents have selected the eight most importance items. The respondents are presented with an imaginary company where all eight most important items are at their lowest level. So waiting time is more than 10 minutes, the staff unfriendly, incompetent, no information is given etc. The respondent is asked to raise one item one level. This is the most important improvement in this imaginary company. Next the respondent is asked to make a new improvement by raising one item one level. This process continues until all items are raised to their highest levels or no further improvements are desired by the respondents. At every step the respondent is required to make a trade off which improvement has highest priority for him. The resulting data set is analyzed with a tailor made package. Results are mainly in terms of priorities which clients put on service improvement.

Our main interest here is the technical problem of programming this procedure as a CAPI program. A simple solution is to write a dedicated CAPI program. This, however has the drawback that it is hardly possible to make flexible adjustments, according to the project at hand. So it appeared to be necessary to have a basic Blaise version that could be adapted to the needs of each individual client. This was possible, but very complex. The source code of the most complex block (where the improvements are made) is given below.

```
BLOCK RANK_LEVELS;
VAR j, k, y: integer;
    l1: array [1..40] of integer;
QUEST
   rank1 "INTERVIEWER: GIVE THE RESPONDENT THE FOLLOWING CARDS TO ORDER
          THE LEVELS
        ////$t[1]$t[2]$t[3]$t[4]$t[5]$t[6]$t[7]$t[8]$t[9]$t[10]
        $t[11]$t[12]$t[13]$t[14]$t[15]$t[16]$t[17]$t[18]$t[19]$t[20]
        $t[21]$t[22]$t[23]$t[24]$t[25]$t[26]$t[27]$t[28]$t[29]
          //On these cards the most important levels are given. With
          all these aspects, we start at the lowest level. Which aspect
          would you want to raise first. Just name the aspect number.":
   1..40;

   rank "You just raised////$t[y]//to level $l[y].//Of which next aspect
         would you raise the level? Just name the aspect number.
         ////: INTERVIEWER: ENTER ZERO IF THE RESPONDENT DOES NOT WANT
               TO RAISE ANYMORE ASPECTS
         array[2..40] of 0..40;
ROUTE
   for j:=1 to 40 do compute l1[j]:=1 enddo;
   rank1;
   compute y:=rank1; compute l[y]:=l[y]+1;
   rank[2];
   compute y:=rank[2]; compute l[y]:=l[y]+1;
   for j:=3 to 40 do
       if (rank[j-1]<>0) and (j<=n_jumps) then
          rank[j]; compute y:=rank[j]; compute l[y]:=l[y]+1;
       endif;
   enddo;
CHECK
   x[rank1]=2 "card $rank1 is not in the pile of the respondent.";
   compute l1[rank1]:=l1[rank1]+1;
   for j:=2 to 40 do
    if rank[j]<>0 then
      x[rank[j]]=2 "Card $rank[j] is not in the pile of the respondent";
      compute l1[rank[j]]:=l1[rank[j]]+1;
      l1[rank[j]]<=a[rank[j]] "The level of aspect $rank[j] is now
         $l1[rank[j]].//This is impossible as the highest level is
         $a[rank[j]].";
    endif;
   enddo;
```

A program was developed as a part of the integrated SMART-procedure, which generates a standard Blaise source code, given the texts of the items and levels (in the same way Blaise generates SPSS setups). This

procedure appeared to be workable. In 50% of the cases the simple dedicated CAPI package is used; in the more complex cases, Blaise performs in a satisfactory way.

4. Public transport

Public transport is an area where many satisfaction studies are carried out. In the Netherlands each larger city has its own public transport company, each of which has unique problems and policies. There are, however, many issues that these companies have in common. The Blaise subfile structure appeared to be very well suited for this type of problem.

The basic questionnaire, common to each of the cities, contained questions like "how often do you travel by bus?", "do you travel at night?", "do you have a driving license?", etc. The answers were stored in the main file. The satisfaction questions were unique to each city; the respondents were asked to score aspects of public transport on a scale between 1 and 10. Such aspects are the shelters, the driving behaviour of the bus driver, the waiting time for the tram, the cleanliness of the subway etc. The evaluations of these aspects were written to a subfile.

A drawback of Blaise is that it does not allow for rotation of questions. In this study, some rotation was required by the client. The problem was solved by rotation based on the respondent number. Four different orders of the satisfaction questions were programmed. They were picked according to the remainder of the respondent number divided by 4 (r mod 4). In the questionnaire this could be calculated as

```
compute order:= r - 4*int(r/4).
```

The ROUTE section then looked as follows

```
if order=0 then ORDER0 endif;
if order=1 then ORDER1 endif;
if order=2 then ORDER2 endif;
if order=3 then ORDER3 endif;
REORDER;
```

The blocks ORDER0, ORDER1, ORDER2, and ORDER3 contain the satisfaction questions in different orders. The block REORDER copies the data into a block of fixed order. By these procedures, using subfiles and reordering the variants to one fixed order, the ease of data processing was optimized within Blaise: each possible variable was written to a fixed field within the record.

5. An incomplete block design

A problem which resembles the order problem in the previous section arose in the context of an advertising monitor. A large bank wanted to evaluate the newspaper ads and TV-commercials of itself and its main competitors. There were six printed advertisements, say a to f and six story boards of TV-commercials, say A to F. From a pilot survey it was clear that it was impossible to have a respondent evaluate all advertisements a to f and A to F. Three from a to f and three from A to F was the maximum. Since there could be an effect of one advertisement on of another, a balanced design was required. This led to 20 combinations (6*5*4/(1*2*3)) like adeBCF or befACD. The 20 combinations were identified by the respondent number or, more in particular, by r mod 20 (the remainder of the respondent number divided by 20).

These respondent numbers were preprogrammed for each interviewer on her field diskette. As a consequence, each interviewer was assigned one combination from a to f and A to F.

6. Sensitive questions

A serious problem in survey research is the inclination of respondents to refuse to answer sensitive questions. In Blaise there is a (hidden) opportunity to enhance the privacy of the respondents when such questions have to be answered. Of course, the interviewer has the possibility to have the respondent enter the answer personally, without the interviewer watching. In principle, by paging backwards, the interviewer can read back the sensitive answer. In the following example this is prevented.

```
TYPE
   IncType = (V0T16 "0 to 1650 guilders",
              V16T22 "1650 to 2200 guilders",
              V22T32 "2200 tot 3200 guilders",
              M32 "more than 3200 guilders",
              NA "no answer");
QUEST
   IncSec "INTERVIEWER: SHOW CARD 20 'TOTAL INCOME OF HOUSEHOLD PER
      MONTH'.////Now I would like you to indicate to which class the
      total monthly income of your household belongs. If you don't want
      to say that to me you can enter it yourself. Afterwards, I cannot
      see what you have entered but it is registered with the data that
      will be processed anonymously.////INTERVIEWER: THE RESPONDENT MAY
      ENTER HIS INCOME IF HE WANTS TO":
   IncType;
   IncReal: IncType (hidden); ROUTE
   page; IncSec; IncReal; page; SIGNAL
   if IncSec<>NA then
       compute IncReal:=IncSec;
       compute IncSec:=NA;
   endif;
```

The procedure works as follows: as soon as an answer is given to the question IncSec (secret income) it is copied to the hidden question IncReal (real income). Then IncSec is reset to the answer NA, which is all the interviewer can see when the computer is returned to her. Our experience is that this procedure reduces item non response. The drawback, however, may be that lack of supervision by the interviewer may lead to non serious answers.

7. A marketing instrument

Sometimes, a Blaise questionnaire can be so convincing, that it can be used for other purposes. The problem under investigation was the question whether a new insurance product should be put on the market or not. To appreciate the benefits of the product, it is necessary to have some insight into the costs of cars. From earlier research it was known that consumers do not like to be confronted with the monthly costs of their car. This happened in paper and pencil interviews where a rough estimate was made, based on the price of the car. In general, respondents are unpleasantly surprised by the height of this estimate and usually deny it. This experience was a good reason to doubt the profitability of the test product.

To investigate the problem, a short but complicated Blaise questionnaire was constructed. The questionnaire started with the price of the car. Next, the costs of the car were broken down to depreciation (yearly), fuel (monthly), tax (yearly), insurance (yearly) and maintenance (monthly). After entering the answers, a summary screen appeared in which the total monthly costs were calculated, visibly broken down to its components. Moreover it was indicated whether the costs were plausible, given the price of the car. When the costs were implausible, the interviewer went over the components for a second time. The results were clear for the respondent and caused no irritation. This clarity of the costs made it easy to introduce the new product and show its benefits. As a result, the product was accepted by the majority of the respondents.

The result of the project was not only a clear positive advice to launch the new product, unexpectedly the questionnaire also showed how to sell it. Presently a dedicated program is being developed which has to be used by the salespeople of the client on laptop computers. It is based on the Blaise questionnaire. From information of the current car of the consumer and usage and insurance data, the benefits of the new product are convincingly calculated and presented.

8. The scenario approach

The most advanced project, or rather method, based on Blaise is the scenario approach. This method makes use of the programming flexibility of Blaise by breaking down a complex consumer decision into a number of detailed separate steps. Here we give an example of mortgages.

The problem starts with a paradox. When respondents are asked in a superficial way why they have taken a particular mortgage they usually answer that the interest rate was the main argument to go to a particular bank or other financial institute (for convenience we will only talk of banks). This, however, cannot be true, for the vast majority of respondents compare only two or three banks with respect to anything. So necessarily, all other banks are excluded for other reasons. In order to investigate this problem, the following sets were distinguished.

A = all banks

B = all banks known by the respondent

C = all banks not refused *a priori* for general reasons

D = all banks not refused *a priori* because of their mortgage products

E = all banks where the respondent has asked for information

F = the bank where the mortgage was effected

G = the banks where the respondent was a client

The sets A through F are ordered, i.e. A⊃B⊃C⊃D⊃E⊃F. Of all banks in the sets B\C, C, C\D, D, D\E, E, E\G and G\E questions were asked why they were in that particular set. With each of these sets, different questions are associated, e.g. for the set B\C:

```
NegBank "INTERVIEWER: SHOW CARD 5.////Can you tell me why you don't
        want to have nothing to do with $MortBank[j] a priori?":
array [1..20] of set[5] of
(Number "they treat you like a number",
 NotKnow "I don't know this bank enough",
 UnFriend "they treat you unfriendly",
 Rich "it is a bank for rich people",
 OldFash "old-fashioned institute",
 NegImpr "I don't know exactly, but I have a negative impression of
         this bank",
 Squeeze "they try to squeeze as much money as possible out of you",
 Business "a bank for business people",
 UnPers "you are treated unpersonally",
 DisTrust "I don't trust this bank enough",
 TooSmall "the bank is too small",
 TooFar "the bank is too far away",
 NotExp "insufficient expertise",
 InConsid "they are too inconsiderate",
 Expens "they are rather expensive",
 IntCli "they are unaware of the interests of their clients",
 Black "black money, ties with South-Africa, etc.",
 Stories "I heard all kinds of negative stories about this bank",
 Oth "other");
```

In order to keep track of the sets, and the associated questions to be asked, indicator arrays were defined, like

I_E {bank indicators in set E}: array[1..20] of integer;

if I_E[k]=1 then bank k is in set E; if I_E[k]=0, then it is not in set E. The final step (from E to F) was analyzed by asking exactly the issues on which the banks in E were compared and the pros and cons of each of the banks. The final outcome of this process was a completely new picture of what is important for marketing mortgages in which interest rate has its own, rather modest, place.

9. Problems and prospects

So far, the experiences with Blaise which are described in this paper are success stories; lack of space prevents us to describe more of them. There are, however, also problems with the package, sometimes due to the basic structure, sometimes due to the neglect of typical market research requirements. The main problems are: lack of speed (especially in questionnaires like in section 8), no easy possibility for rotation of both questions and answers, no possibility of masking answers in set-questions, no easy way of addressing set-response categories by for-loops, no checks for sample quota, no easy interface (including data sharing) with external programs. Although Blaise allows for larger questionnaires than most of its competitors, it is unclear when Blaise reaches its limits.