Using Blaise 4.7 for data entry, checking, error reporting and transforming for further processing

Krstanova Orhideja, SSO R..Macedonija, R.Macedonija

I like to tell you about our experience in using BLAISE survey processing system. We use BLAISE for data entry and control about a year. During that period all new data entry programs were made in BLAISE. Some of the data entry applications for the ongoing surveys were rebuild in BLAISE too. Finding it very quick and convenient especially for large databases our intention is to use it for major part of the surveys we provide.

The things are set like this: The application for certain survey in BLAISE is settled on server in special folder for that survey. Operations performed are: data entry with online control, batch control and reporting errors using manipula, corrections and transforming data using cameleon, in formats suitable for further processing. Regarding that our database is DB2 for all surveys, and that so far we don't use direct communication between DB2 and BLAISE, the data are always transformed in text format so that it would be appropriate for further manipulation or maintain. Sometimes data is transformed in SAS format according to the needs. All these operations are performed thru interface built in Visual Basic 6 and are partly automatic. But we have vision and work on it to make environment, which will provide possibilities for whole processing data to be more automated.

1. Introduction

The first introduction with BLAISE in SSO of Macedonia was about nine years ago. The very first BLASE data entry application was built for Labor Force Survey and with the professional assistance given by the Danish-Finish Consortium.

For many years, this was the unique data entry application in Blaise software. Last few years IT-sector in SSO of Macedonia is looking for a way for more efficient and up to date work in fields like data entry, editing, maintaining, metadata, documentation and so forth. In that context, BLAISE was found appropriate because it allows quick and consistent work especially with large databases. Nearly all the new data entry applications in the last year were built in BLASE.

2. Data entry application

Things are organized like this: there is a folder on the server named with the same name as the survey, the application was built for. That folder contains VB-6 application that serves like interface through which the person that will perform the data entry can reach the application.

The applications are made in blaise data files were code contains standard blocks and tables that corresponds with the questionnaire design.

In most data entry applications data controlling is provided on line. Depends on the construction of the questionnaire for the certain survey the control is performed within the table rules or in datamodel rules or in both. For most of the surveys there is a data base with some basic information like address, business unique number, classification, and so on. Every observed unit has a unique identification number for easier identification. Through this unique number all data for the unit can be linked.

During data entry to save time and disable errors in the basic information this data are taken from mentioned data bases. Next part of the code is presenting this operation:

```
.
RULES
  ISTRAZ :=('SUM22')
  GOD :=('2007')
  RBR {unique number}
  SOP
  OPST
  NKD

  IF Adresar_SUM22.search (RBR) then
    Adresar_SUM22.READ
    SOP:=Adresar_SUM22.SOP
    OPST:=Adresar_SUM22.OPST
    NKD:=Adresar_SUM22.NKD
  ELSE
    ERROR INVOLVING(RBR)  "It is not in the address book"
  ENDIF
.
```

The on line control performs logical and mathematical controls, usually using If…then structure. Part of code as an example is given below.

```
.
P41313

If (P43=2 or P43=3) and (P4131=0 and P4132=0 and P4133=0 and P4134=0 and P4135=0 and P4136=0
      and P4137=0 and P4138=0 and P4139=0 and P41310=0 and P41311=0 and P41312=0 and P41313=0) then
      error involving (P4121)"G25: Ko P43=2 ili P43=3 togas mora da ima odgovor vo koja zemja i kolkav e ostvareniot odmor "

ENDIF
.

Or
.
T1[5]
T1[6]
T1[7]
T1[8]

If (T1[5].k1 <> T1[6].k1 + T1[7].k1 + T1[8].k1) or (T1[5].k2 <> T1[6].k2 + T1[7].k2 + T1[8].k2) or (T1[5].k3 <> T1[6].k3 + T1[7].k3 + T1[8].k3)  then
     error "GR 2: vkupnoto e razlicno od zbirot @r ^ T1[5].k1 , @r @r ^ T1[5].k2 , @r @r ^ T1[5].k3 vo polinjata "
ENDIF
.
```

Some times the data have to be controlled with the data from the previous period. If it is possible for these cases, on line control is performed too. As it is shown below during executing data entry for SUM22 survey, certain data are controlled with the SUM22_1 which are data from the previous period.

```
.
Tabela1
Tabela2
Tabela3

If Sum22_1.search (RBR) then
        Sum22_1.READ
      FOR i:=1 TO 30 do
          If Tabela1.T1[i].k2 <> Tabela1.T1[i].k1 - Tabela1.T1[i].k3 +
Sum22_1.Tabela1.T1[i].k2 then
              error "GRESKA : T1[i].k2 e razlicno, vo sporedba so prethodniot mesec "
          ENDIF
      ENDDO
ENDIF
.
```

Persons that perform data entry are well skilled and familiar with the applications and surveys for qualitative data entry.

3. Checking and error reporting

Although the main parts of errors are prevented by on line control during data entry, sometimes there is need the entered data to be checked more times in different ways. These operations are provided through manipula file type. If the errors that can not be solved by the people that perform the data entry are expected, code for data entry (in BLAISE data model) is a little bit different. If this type of error is expected, previously presented code will look like this:

```
.
T1[5]
T1[6]
T1[7]
T1[8]

If (T1[5].k1 <> T1[6].k1 + T1[7].k1 + T1[8].k1) or (T1[5].k2 <> T1[6].k2 + T1[7].k2 +
T1[8].k2) or (T1[5].k3 <> T1[6].k3 + T1[7].k3 + T1[8].k3) then
     error "GR 2: vkupnoto e razlicno od zbirot @r ^ T1[5].k1 , @r @r ^ T1[5].k2 , @r @r ^
T1[5].k3 vo polinjata "
       greska[2]:=2
     endif
.
```

The difference is in *greska[2]:=2* (error[2]:=2) where the errors are defined, then in rules

```
.
RULES
  for i:=1 to 7 do
    greska[i]:=0
  enddo
.
```

All this is used in manipula data file for making a report on errors. The report contains the unit number (expressed through unique identification number) in which the error is found and the number of the error. With the unit number and the number of the error, the questionnaire can be easily found and correct the data where it is necessary.

Code for the report on errors is presented below.

```
SETTINGS
  DESCRIPTION = 'Error report'

USES
  DATAMODEL outf
    fields
      oneline:string[100]
  ENDMODEL
  SUM22 'SUM22'

INPUTFILE Infile1: SUM22('SUM22', BLAISE)
OUTPUTFILE OutFile1: outf ('errors.txt', ASCII)
settings
makenewfile=no  {errors are wrote always in the same text file}
MANIPULATE
IF infile1.recordnumber = 1 then

oneline:='$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
  OUTFILE1.WRITE
  oneline:=' ' OUTFILE1.WRITE
  oneline:='Date:   '+ datetostr(sysdate)+'   Time:   '+timetostr(systime)
  OUTFILE1.write
  oneline:=' ' OUTFILE1.WRITE
  oneline:='List of errors for key (RBR):' OUTFILE1.WRITE
  oneline:=' ' OUTFILE1.WRITE
  oneline:='RBR:     G1 G2 G3 G4 G5 G6 G7  :   ' OUTFILE1.WRITE
  oneline:=' ' OUTFILE1.WRITE
  oneline:='------------------------------------------------------------------------------------------
-----------------------------------------------------------------------'
  OUTFILE1.WRITE
ENDIF
{rbr is unique identification number}
IF greska[1]+greska[2]+greska[3]+greska[4]+greska[5] > 0  then
  oneline:=rbr + '       '+ str(greska[1]) +'  '+ str(greska[2]) +'  '+
  str(greska[3]) +'  ' +str(greska[4]) +'  '+ str(greska[5]) +'  '+
  str(greska[6]) +'  '+ str(greska[7])
  OUTFILE1.WRITE
ENDIF
```

If there is a need for other types of checking a appropriate manipula file is built.

4. Transforming for further processing

As further processing are made in SAS or ACCESS software, data from the BLAISE data files need to be transformed.

If further processing is performed in SAS, Blaise data is transformed through CAMELEON so that the formats can be used.

If further processing is performed in ACCESS or have to be input in DB2 database, blaise data is transformed in text file through manipula, sometimes made with cameleon and sometimes separate manipula file is made.

5. Future activities

For now the data entry is performed through an interface built in Visual Basic 6 and it is partly automatically. Data editing, processing and transforming is performed more or less manually. An environment which will provide possibilities for whole processing data to be more automated will be established when some decisions regarding data entry, editing, maintaining, metadata, documentation and so on are made.

At this stage we experience only basic BLAISE software, but we expect and work on to establish surrounding that will allowed us to use the advanced possibilities in favor to more efficient and up to date work.

Contact: orhideja.krstanova@stat.gov.mk