MANIPLUS, A powerful environment for managing Blaise III applications

Hans Wings and Lon Hofman, Statistics Netherlands

1. Introduction

With the introduction of survey processing software like the Blaise system more and more statistical offices, researchers and commercial marketing agencies are changing from PAPI to CADI, CAPI or CATI. After several years of experience the benefits of computer assisted interviewing are clear: better quality of data, lower costs, and more efficiency. The first generation of statistical software at Statistics Netherlands (Blaise CADI, CAPI and CATI programs, Abacus, Bascula, etc) was developed as a tool to assist in a specific stage of the survey process. With the introduction of Blaise III a next step to integrated survey processing has been made. All tools are integrated into one environment, they speak the same (meta) language and the interfaces are standardised.

Recently, a new member has been added to the Blaise III family of tools: ManiPlus. ManiPlus is a powerful tool to build applications involving Blaise. It gives survey system designers the possibility to combine tools like the Data Entry Program, the Data Viewer, Manipula, and a DOS Shell into a user-friendly package controlled by user-defined menu systems and dialog boxes.

In this paper we will present a brief overview of the components available in ManiPlus. Because the development of case management systems for CAPI is very much in the public eye, we will illustrate the features of ManiPlus using LIPS [1] (Laptop Information-system for Personal Surveys) of Statistics Netherlands as an example.

2. Case management for CAPI

A few years ago, the first steps in designing laptop case management systems were taken. Some organisations (like Statistics Netherlands with LIPS) have designed their own laptop case management systems with their own objectives or emphases. One thing these systems have in common is that they are all shells around the CAPI questionnaire programs. This is because the CAPI software is usually not capable of dealing with all the functional requirements of survey management. Although these sytems do their jobs well, they have a number of drawbacks. The survey management systems are developed with systems like database systems (Clipper, FoxPro, etc), programming languages (Pascal, C, etc) or just DOS batch files, which differ a lot from the CAPI development systems. This leads, among other things, to different user-interfaces or data representations within one survey management system instead of one face for the whole system and one data format within the whole sytem.

Other organisations have adopted existing systems. Usually these adopted systems don't satisfy the requirements of that specific organisation completely. Adjusting to these requirements is often difficult or impossible.

A disadvantage of the LIPS system at Statistics Netherlands is the dependency on the current version of Blaise. With the introduction of Blaise III the system must be rewritten or adjusted to the new version. It is obvious that a tool, like ManiPlus, which will solve the problems mentioned and avoid the disadvantages of the currently developed survey systems would be a great help.

Although every laptop case management system will have it's own requirements, many of them have a lot of functions in common. According to Nichols and Kindel [2] laptop case management systems have to be able to perform the next core functions:

1. Accept and store interviewing assignments.
2. Display a list of these assignments to the interviewer.
3. Select a case to interview.
4. Store the interviewer data and retrieve it when necessary for partial interviews.
5. Record the status or disposition of each case, such as complete, partial, appointment, or untried.
6. Initiate telecommunications or prepare diskettes for mailing.
7. Perform various system functions, such as logging in, setting and using passwords, and maintenance of core files.
8. Provide unidirectional or bi-directional e-mail communication between interviewers and supervisors.

Of course, every organisation wants to add it's own functionality to this list. So the tool needed must be able to implement the above-mentioned core functions, but it should also provide enough flexibility to add almost any required functionality.

3. Introducing ManiPlus

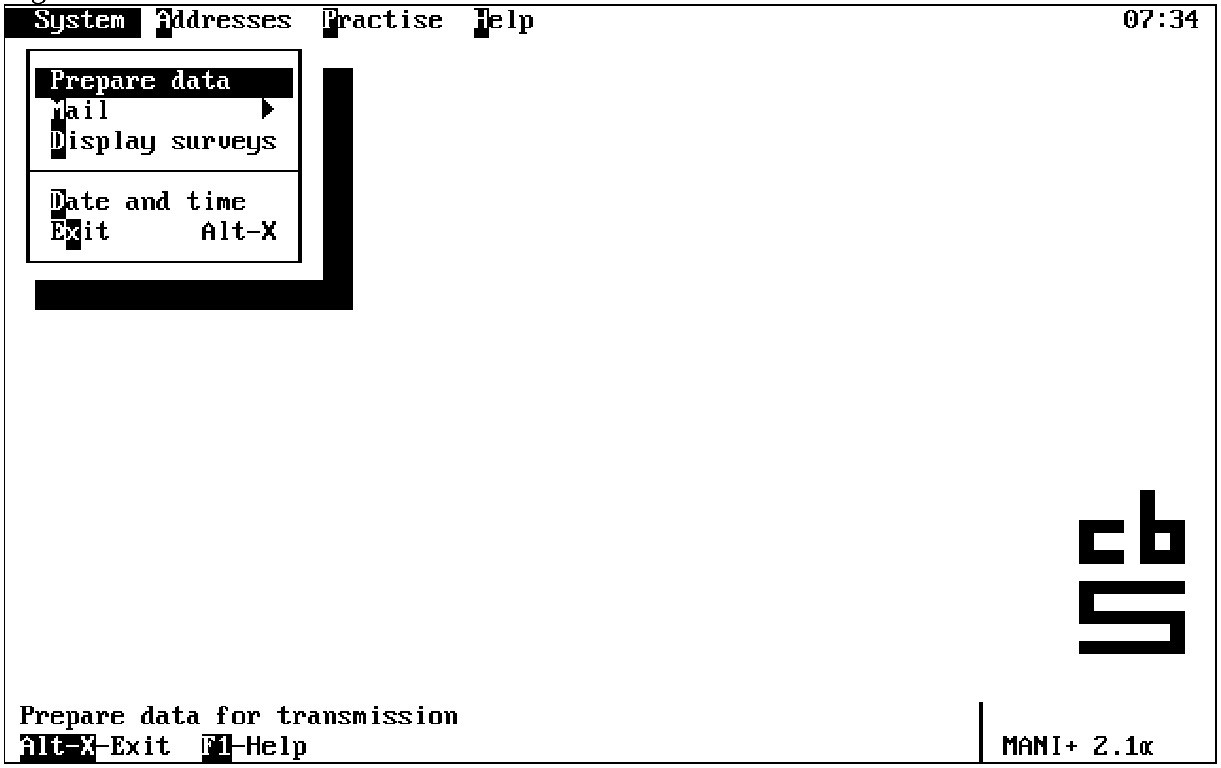
A new tool, called ManiPlus, has been added to the family of Blaise III tools. This tool is well suited for development of (survey management) systems involving Blaise. As the name suggests the basis of this tool is Manipula, so all the functionality which is already present in Manipula is also available in ManiPlus. Further tools such as the Data Entry Program, the Data Viewer, Manipula itself, a DOS Shell and a fully equipped help facility are integrated in ManiPlus. All components of ManiPlus can be controlled by user-defined menu systems and dialog boxes. Finally, file handling is not implicit like in Manipula but fully controlled by the user.

In the next sections we want to give you a feeling of what is possible with ManiPlus. This description is by no means exhaustive. We will only take a short look at some components. In each section we will show how a part of the LIPS system could be built using ManiPlus. On passing we will refer to the core functions indicated by Nichols and Kindel in order to indicate that all these functions can be implemented using ManiPlus.

3.1. Menus

Operators must be able to work with laptop interviewing system and therefore they need a menu, which guides them through the system. In the LIPS system the menu could look as follows:

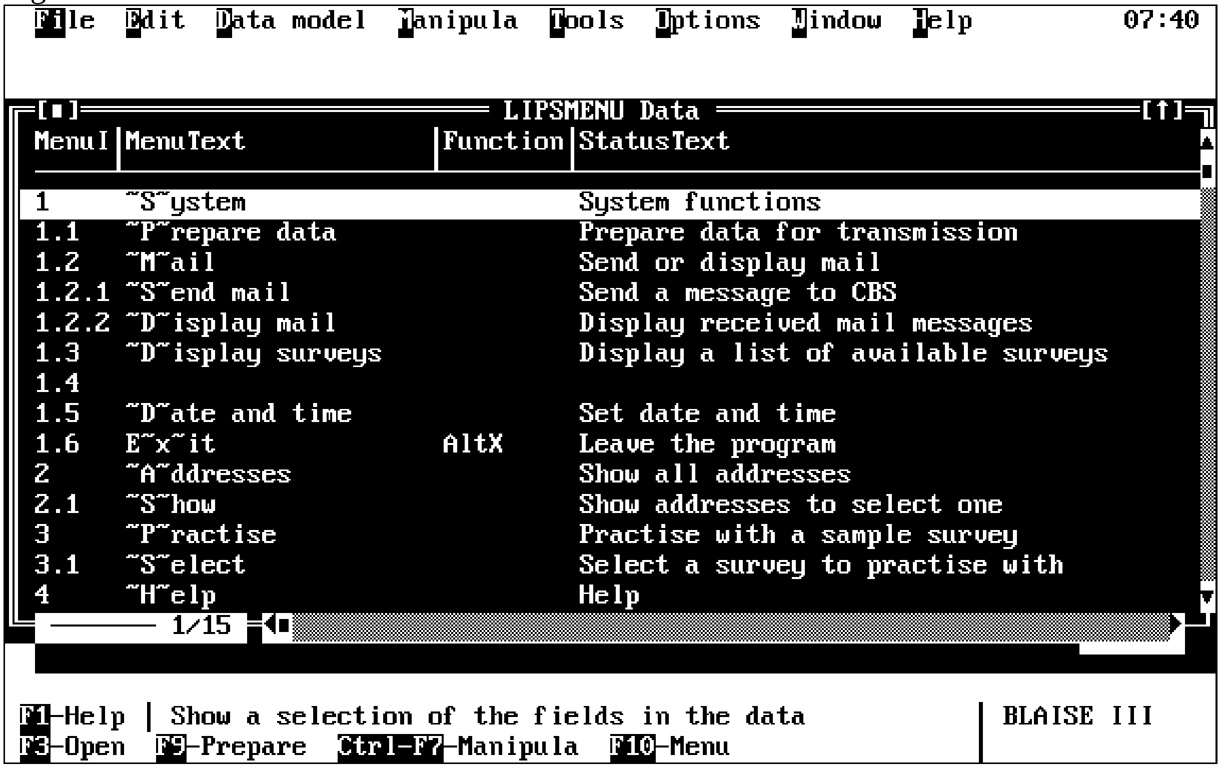
Figure F. The LIPS menu



Prepare data for transmission

In ManiPlus the menu items are defined as records in a file of a specific datamodel. The definition of that data model has to satisfy a number of conditions, which we will not explain in detail here. The following figure shows the Data Viewer of the Blaise control center with the contents of the data file which defines the menu of the LIPS system. Each line in the Data Viewer corresponds to one menu item entry in figure 1.

Figure G. The contents of the menu file



```
 File  Edit  Data model  Manipula  Tools  Options  Window  Help         07:40

[■]======================== LIPSMENU Data ===========================[↑]
 MenuI |MenuText            |Function |StatusText
 1      ~S~ystem                       System functions
 1.1    ~P~repare data                 Prepare data for transmission
 1.2    ~M~ail                         Send or display mail
 1.2.1  ~S~end mail                    Send a message to CBS
 1.2.2  ~D~isplay mail                 Display received mail messages
 1.3    ~D~isplay surveys              Display a list of available surveys
 1.4
 1.5    ~D~ate and time                Set date and time
 1.6    E~x~it              AltX       Leave the program
 2      ~A~ddresses                    Show all addresses
 2.1    ~S~how                         Show addresses to select one
 3      ~P~ractise                     Practise with a sample survey
 3.1    ~S~elect                       Select a survey to practise with
 4      ~H~elp                         Help
 ——————— 1/15

 F1-Help | Show a selection of the fields in the data            | BLAISE III
 F3-Open  F9-Prepare  Ctrl-F7-Manipula  F10-Menu                 |
```

The ManiPlus setup which runs the LIPS system looks as follows:

```
InitProc
REPEAT
   Res:= LipsMenu.Menu('~Alt-X~-Exit ~F1~-Help')
   IF (FunctionKey <> altx) AND (Res = 0) THEN
     CASE ModuleName of
       'showaddresses'   : ShowAddresses
       'makemessage'     : MakeMessage
       'preparedata'     : PrepareData
       'surveys'         : ShowSurveys
       'datetime'        : SetDateAndTime
       'selectsurvey'    : SelectSurveyToPractise
     ENDCASE
   ENDIF
UNTIL FunctionKey= AltX
```

The instruction *LipsMenu.MENU* activates the menu in the program. After selecting a menu item the data that corresponds with the menu entry is available in the program. When the menu item *Display surveys* in figure 1 has been selected the sixth record (see figure 2) of the file *LipsMenu* becomes the current record. This record contains a field *ModuleName* to identify the procedure which is associated with the menu entry. In this case this field has the value 'surveys'. The CASE statement invokes the procedure which is associated to the selected menu item.

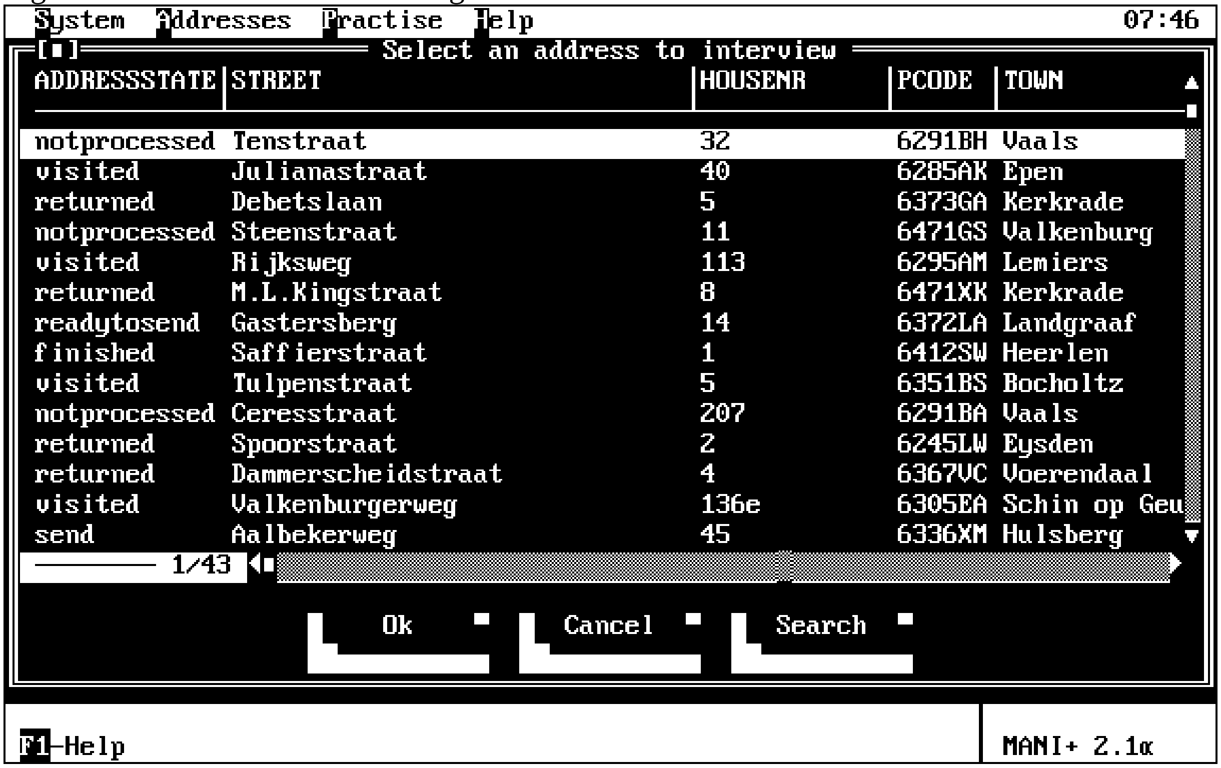
By using a REPEAT loop as the core of the ManiPlus setup, the operation of the setup is completely menu driven. A procedure (also a new feature of Manipula 2.1), which will execute the statements associated with the menu item, is assigned to every menu item.

The procedure *InitProc* could check in a specified directory for new addresses that have to be visited and store these addresses in the appropiate files of the system (the first core function of a laptop interviewing system).

3.2. Dialogs

To implement the second and third core functions of a laptop managament system you need a dialog which helps the interviewer to select a case to interview. In LIPS a dialog with a list of available addresses is shown to the interviewer. This list contains only the necessary information of the addresses for the interviewer.

Figure H. The select address dialog



Select an address to interview

ADDRESSSTATE	STREET	HOUSENR	PCODE	TOWN
notprocessed	Tenstraat	32	6291BH	Vaals
visited	Julianastraat	40	6285AK	Epen
returned	Debetslaan	5	6373GA	Kerkrade
notprocessed	Steenstraat	11	6471GS	Valkenburg
visited	Rijksweg	113	6295AM	Lemiers
returned	M.L.Kingstraat	8	6471XK	Kerkrade
readytosend	Gastersberg	14	6372LA	Landgraaf
finished	Saffierstraat	1	6412SW	Heerlen
visited	Tulpenstraat	5	6351BS	Bocholtz
notprocessed	Ceresstraat	207	6291BA	Vaals
returned	Spoorstraat	2	6245LW	Eysden
returned	Dammerscheidstraat	4	6367VC	Voerendaal
visited	Valkenburgerweg	136e	6305EA	Schin op Geu
send	Aalbekerweg	45	6336XM	Hulsberg

The definition of this dialog in the ManiPlus setup looks as follows:

```
DIALOG SelectAddressDialog "Select an address to interview"
  LOOKUP AddrTempo
    POSITION = (1,1)
    SIZE = (77,16)
    FIELDS
      AddressState
      Street
      HouseNr
      PCode
     Town
  BUTTON ButtonResult
    CAPTION = ' ~O~k '
    VALUE = ok
  BUTTON ButtonResult
    CAPTION = ' ~C~ancel '
    VALUE = cancel
  BUTTON
    CAPTION = ' ~S~earch '
    ONPRESS = SEARCH (AddrTempo)
```

The dialog *SelectAddressDialog* is composed of different dialog elements, in this case three buttons and a lookup. The lookup is assigned to the file *AddrTempo* and only five of the fields (the information necessary for the interviewer) of each record are displayed in the lookup. When the third button is pressed the search dialog of the DataViewer is activated. With this dialog you can change the order in which the records must be displayed according to the keys which are defined in the datamodel of the file concerned. When the first button is pressed the dialog is closed and the currently selected form becomes available for further processing.

Each dialog can be composed from different dialog elements. In total, ManiPlus provides you four different dialog elements:

1. A text at a certain position,
2. A button which genererates a command when pressed,
3. A control which allows the user to enter data into a field, and
4. A lookup which displays a Data Viewer.

These four dialog elements give you the flexibility to fully design your own dialogs.

3.3. Data Entry

If an interviewer has selected an address to interview, you need to start a data entry program (DEP). The DEP of Blaise III has been integrated into ManiPlus, so it is not necessary to start it outside the management program in a DOS shell. There are two ways to start the DEP in ManiPlus:

1. If you know the data model and the name of the survey beforehand (that is to say while designing the ManiPlus setup) the DEP can be started as a method of a file with *F.EDIT(S)*, where F is a file identifier of a Blaise III update file. With the text *S* you can pass the same command-line options like for the stand-alone DEP, except for the meta-file name and the option /F (because they are already defined in the UPDATEFILE section of *F*). Two extra options are useful:

/X	exit the DEP after treatment of one form
/G	start the DEP in the get form mode

The form that has been manipulated in the DEP will be available if the DEP is exited. The DEP can be executed on any Blaise III update file defined in the ManiPlus setup.

2. However, in LIPS, which supports multiple survey interviewing, the data models and the data files are unknown at design time. What survey will be used depends on the case an interviewer selects to interview, so the data model and data file to be used are determined at run time. In this case you start the DEP with *EDIT(S)*, where *S* must contain the name and location of the meta file and optionally the name and location of the data file.

The part in the ManiPlus setup which takes care of selecting an address and starting the DEP for this address looks as follows:

```
CreateAddrTempo(ActiveSurvey)
SelectAddressDialog
IF ButtonResult=ok THEN
  DetermineActiveSurvey
  EditResult:= EDIT(DataDirectory+ActiveSurvey+
                  ' /F'+DataDirectory+ActiveSurvey+
                  ' /K'+AddrTempo.Spil_ID+
                  ' /X')
  UpdateAccountOfAddress
ENDIF
```

The procedure *CreateAddrTempo* creates a temporary file (a new file type in Manipula 2.1) to store all available addresses in memory. With the already discussed dialog *SelectAddressDialog* the interviewer selects a case to interview whereupon the DEP is started for the selected address only. If the interview is (partially) completed the DEP takes care of storing the data of the interview. The procedure *UpdateAccountOfAddress* takes care of accounting for the interview. This is comparable with the fifth core function of a laptop interviewing system.

3.4. Executing Manipula setups

A lot of survey systems use DOS batch files to execute successive Manipula setups. The execution of a Manipula setup (that is to say executing ManiDoit with the corresponding prepared MSF file) is also integrated in ManiPlus. To implement, for example, the preparation of data which has to be sent to the central office (the sixth core function of a laptop survey system) you need a Manipula setup which extracts the data from the Blaise data file. Because LIPS supports multiple survey interviewing, the data file from which the data has to be extracted is unknown beforehand. A possible solution to this problem is to provide the system with a Manipula setup for each survey, which will take care of extracting the completed interviews from the corresponding data file. In LIPS the interviewer invokes this setup by selecting the submenu item *Prepare* in the menu *System*. This menu item executes the procedure *PrepareData* in the setup:

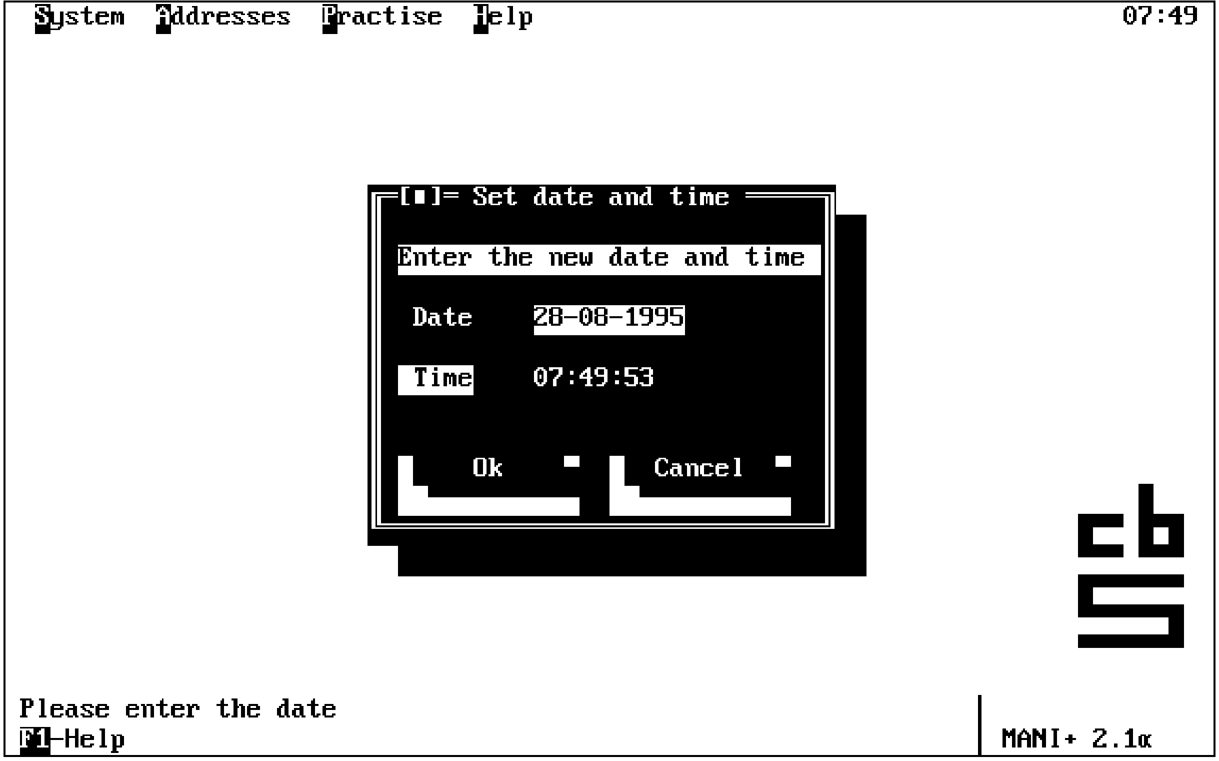
```
PROCEDURE PrepareData
  SelectSurveyDialog
  CASE ButtonResult OF
    ok:
      IF CONFIRM('All completed interviews of the survey '+IntSurv.SurveyName+
                             ' will be prepared for export and deleted in the data file. Are you sure ?')
THEN
         Res:= CALL(SetupDirectory+IntSurvSurvey > NUL')
      ENDIF
    allsurveys:
      IF CONFIRM('The completed interviews of all surveys will be prepared for export '+
                             'and deleted in the data files. Are you sure ?') THEN
        IntSurv.Reset
        FOR I:= 1 TO  IntSurv.RECORDCOUNT DO
          IntSurv.READNEXT
          Res:= CALL(SetupDirectory+IntSurv.BlaiseName+' > NUL')
        ENDDO
      ENDIF
  ENDCASE
ENDPROCEDURE
```

With the CONFIRM function a message box with a YES- and NO-button comes up. The result of the CONFIRM function is *true* if the YES-button is pressed. If the user confirms the selected menu item the Manipula setup is executed with the instruction CALL. The parameter of the instruction contains the name of the setup.

3.5. Running DOS commands

To show how a system function (the seventh function of the eight core functions) may be implemented in ManiPlus we use the date and time setting function in LIPS as an example. On the laptop the interviewer can change the date and time of the system. Entering the new date and time is done by a dialog:

Figure I. The set date and time dialog



Enter the new date and time

Date 28-08-1995

Time 07:49:53

Please enter the date

In the ManiPlus setup this dialog is defined with:

```
DIALOG DateAndTimeDialog "Set date and time"
  SIZE = (31,12)
  TEXT ('Enter the new date and time',2,2)
  CONTROL Date
    LABEL = ('Date',2,4)
    POSITION = (10,4)
    STATUS = 'Please enter the date'
  CONTROL Time
   LABEL = ('Time',2,6)
   POSITION = (10,6)
   STATUS = 'Please enter the time'
  BUTTON ButtonResult
   CAPTION = '   ~O~k   '
   POSITION = (2,9)
   VALUE = ok
  BUTTON ButtonResult
   CAPTION = ' ~C~ancel '
   POSITION = (16,9)
   VALUE = cancel
```

In this example two controls are used to enter data. The controls provide input lines to enter the new values. An auxiliary field is assigned to each control to recieve the values of the input lines. The text dialog element is used to display an additional text in the dialog. The *DateAndTimeDialog* is invoked by the following procedure just by mentioning the name of the dialog:

```
PROCEDURE SetDateAndTime
  DateAndTimeDialog
  IF ButtonResult=ok THEN
    RunResult:= RUN('DATE '+DATETOSTR(Date)+' > NUL')
    RunResult:= RUN('TIME '+TIMETOSTR(Time)+' > NUL')
  ENDIF
ENDPROCEDURE
```

To execute the system commands the RUN function is used. With the RUN function you get full access to DOS. This function releases as much memory as possible before executing the command. The result is zero if the command was executed succesfully, otherwise it is non-zero. The parameter of the RUN function is the DOS-text of the command to be executed.

When the *OK-button* in the dialog *SetDateAndTimeDialog* has been pressed, the value of the auxiliary field *ButtonResult* is *ok* and the two DOS commands are executed. In this case a redirect to the NUL device is used, so the DOS commands are executed ″in the background″.

The last of the eight core functions of a laptop case management system can also be implemented using the RUN function. Sending an e-mail to the central office, for instance, is easily implemented with a RUN function that invokes a program which provides the appropriate functionality.

3.6. The Help facility

Nowadays every software system provides a help facility to guide the user through the menu and the interface of the system. Providing help information for an interactive system like a laptop survey management system is, of course, very important. With ManiPlus you get a hypertext help facility which links help texts to menu items or dialog elements. It is also possible to invoke a help text within the MANIPULATE section. The help text is stored in a help file.

4. Final remarks

With the introduction of ManiPlus the next step to integrated survey processing has been made. ManiPlus provides all the necessary components to develop survey management systems. Obviously, ManiPlus can be a great help in developing survey management systems. It is now up to you to explore the possibilities of this new tool. ManiPlus has a lot more to offer than we have dealt with in this paper. Not only survey management systems can be built with ManiPlus. The multi-survey shells at NASS [3] (National Agricultural Statistics Service), for instance, are a good example of a system which can be rebuilt using ManiPlus.

While rebuilding the LIPS system with ManiPlus we noted a tremendous decrease in development time. What took one computer programmer almost half a year was rebuilt with ManiPlus in several weeks. Mind you! You still need some programming experience to build survey systems with ManiPlus.

References

[1] L.P.M.B. Hofman, W.J. Keller*: Management of Computer Assisted Interviews in the Netherlands*. In: **Journal of Official Statistics**, Vol. 9 No.4, Statistics Sweden, 1993, pp. 765-782.

[2] W.L. Nicholls II, K.K Kindel: *Case managment and communications for Computer Assisted Personal Interviewing*. In: **Journal of Official Statistics**, Vol.9 No.3, Statistics Sweden, 1993, pp. 623-639.

[3] R. Schou, M. Pierzchala: *Standard multi-survey shells in NASS*. in: **Essays on Blaise 1993**, First International Blaise Users Conference (1993), pp. 133-142.

[i] WENZOWSKI, M.: *ACTR: A Generalized Automated Coding System*. **Survey Methodology**, 14,2, 299-307.

[ii] TOURIGNY, J. and MOLONEY J.: *The 1991 Canadian Census of Population Experience with Automated Coding*. Statistical Data Editing, Volume 2. United Nations Statistical Commission, Economic Commission for Europe, Conference of European Statisticians. United Nations, New York.

[iii] WENZOWSKI, M. et al. (1995). *ACTR v3 User Guide*. Internal Document, Research and General Systems, System Development Division, Statistics Canada, Ottawa, Canada.

[iii] **Author's Address:**
Research and General Systems
System Development Division
R. H. Coats Building, Section 13-A
Ottawa, ON K1A 0T6 CANADA

Tel.: 613.951.1347
Fax.: 613.951.0607
email: wenzows@statcan.ca