6th INTERNATIONAL BLAISE USERS' CONFERENCE 2000

CORK, May 2000

Blaise Generator for hi-speed data entry applications

by

Pavle Kozjek
Statistical Office of the Republic of Slovenia

1. Introduction

Modern theories of statistical survey processing consider hi-speed ("heads down") data entry mainly as a little bit old-fashioned. In the survey process it's been replaced by recent methods ant techniques, like CAI, EDI and internet data collection. However, in practice many statistical offices still have to deal with hi-speed data entry from paper forms without (or with minimum) data checking at the time of data entry. Usually it is organised on a mainframe platform, but using the systems like Blaise it can be supported on a LAN platform as well.

At the Statistical Office of the Republic of Slovenia there's still a large amount of data entered by a dedicated package for hi-speed data entry on a mainframe platform. The system is out-of-date and need to be replaced, so a new solution on a LAN Windows NT platform was developed. With the new generator (written in Blaise 4 Windows and Visual Basic), end-users can produce (on a basis of paper documentation) hi-speed data entry applications without developer's help.

New solution is based on existing methodology and documentation, and can be integrated in the existing survey process by mixing mainframe and LAN platform. At the same time it sets some basic standards and prepares ground for the redefined survey process that completely takes place on a LAN platform.

2. Reasons for development

Statistical Office of the Republic of Slovenia (SORS) is a centrally organised statistical office. Already early in 70's SORS started to explore the possibilities of building up a statistical system based on registers, and today this strategy is continued.

However, there's still a large amount of survey data collected on paper forms using traditional mailout - mailback method. One reason for relatively large batches of forms is complete coverage of population in many surveys. In some cases it is necessary, in other cases we need to introduce sampling methods. Another reason is growing number of surveys. New developments mainly make use of CAI and EDI methods and techniques, but the major part of surveys (especially from economic area) still report on a paper forms. A part of this data is captured by OCR, and the rest by hi-speed data entry. Although the share of data entered "heads down" is slowly decreasing, a reliable support for that kind of data entry should be obtained.

The existing system DCR 5000 (Data Capture & Retrieval) is a specialised hi-speed data entry software package that runs on the Unisys U Series product line (Unix operating system). The main reasons for replacing the system are: old and unreliable HW equipment, maintenance problems and poor compatibility with the new SORS information system infrastructure, based on a LAN Windows NT environment. A part of the old system was also not Y2K compliant (a rough-and-ready solution was found), so the basic functions of a new system had to be ready by the year 2000, to take over the production in case of Y2K problems.
Since Blaise supports most of SORS new developments on survey data entry, the Blaise system was a logical choice to support the new hi-speed data entry solution as well.

3. Building up the generator

Preparing replacement for the existing system was not an easy task. In general, the new system should support all the functionality of the old solution, and possibly add some improvements. But on the other hand, very small resources were available, and the system should not be too complex.
Development of generator would not be possible without some standards concerning data models. Because of data storing, these standards were taken from mainframe and (although including many restrictions) they were used as a guideline to define generator outputs. When entry is finished, final Blaise database is converted to ASCII and transferred to the mainframe. Since mainframe archiving system has limits in record length, one survey form is usually entered as a set of records of different types.
The old DCR-5000 system is a highly specialised program package and years ago it was integrated in a complete data capture-archiving-retrieval system on a mainframe. Data entrists are used to work on a special keyboards, so with the new data entry solution also the new special programmable keyboards for data entrists were obtained.
There was an important request, that users should be able to produce (on a basis of paper documentation) data entry programs without the intervention of developers. Users were not expected to write code, so generator was the solution. Data entry should be fast, and verification (double keying) must be enabled.

The development of generator began in July 1999, using Blaise 4 Windows (version 4.1) and Manipula for application development, and Visual Basic 5 for development of user interface. There was almost no documentation about the existing application, so a permanent contact with people involved in an existing production process was necessary.
Beside data models for entry and verification, generator produces many auxiliary Manipula setups to support administration and control the process. Blaise and Manipula command line parameters were widely used in development of user interface.

3. Main problems and solutions

One of the main problems was how to help and enable end-user to correctly define data model, following the existing documentation. To support this, a special Blaise data model was defined, where all the survey parameters, fields and specifications are entered. Each entry in this specification model (key is unique survey code and version) defines a new generated data model for hi-speed data entry. There are paper and on-line instructions available to the person who specifies the data model. In our case supervisor (data entry administrator) is responsible for that task.
Next question: how should look the output of the generator -Blaise data model. Generated application can not be (and don't need to be) a complex Blaise instrument. On the other hand, it has to fulfil the general needs of surveys that use hi-speed data entry. Manipula reads the parameters from the specification data model and writes the code for hi-speed entry with the following general structure:

```
DATAMODEL Survey
        FIELDS
                Field_1
                  - - -
                Field_ j
                Record_Type:  1..i
                BLOCK B[1]
                        B1Field_1
                          - -  -
                        B1Field_m
                ENDBLOCK {1}
                          - - -
                BLOCK B[i]
                        BiField_1
                         - - -
                        BiField_n
                ENDBLOCK {i}
        RULES
                Field_1
                  - - -
                Field_j
                Record_Type
                IF Record_Type = 1  THEN   B[1]   ENDIF
                  - - -
                IF Record_Type = i  THEN   B[i]    ENDIF
ENDMODEL  {Survey}
```

With this simple general structure, all data models of surveys that use hi-speed data entry are covered. There are common (form-characteristic) fields on a first level and a number of different blocks on a second level. Each block in a generated model represent a different record type. Final ASCII records, produced by Manipula setup always consists by common fields, field that define form type and a number of fields defined by that form type. Single level data model definition is also supported.

Most of the fields on a form level are usually SORS standard, so their specifications can be pre-defined and imputed. A question was how to enable and support correct entry for the field "record type". This is an important key field for generation of applications, but treated in a very different way in a different surveys. Usually it need to be separated in two fields, to enable correct generation of data entry application. Specifications (type, length etc) for the data fields in blocks are entered into the specification model directly from the survey documentation. There are a few checks included, obtaining correct positions of fields.

Another hard task for supervisor is how to define key fields of generated data model. Due to speed of entry only the secondary keys are defined. Usually the key is composed of sequence or ID number of form, and type of record. In the generator key definition is now supported and facilitated by presenting all possibilities in a closed question.
All the hardest problems for supervisor when specifying data model (like specifying key fields and fields for record type) were additionally explained to the supervisor.

Another problem was verification. Reference files are not a good solution, since response should be immediate. The idea was to put both fields (for entry and verification) in the same data model. There are only entry fields on the screen during the entry session, and only verification fields during verification, which is based on time sequence of entered records. But this approach disables common data base, so in the first step partial data bases (covering one batch of paper forms) are created and verified, and in the second step they are merged in a common Blaise data base of a survey. Verification process added a lot of administration, but seems to be inevitable if the entry is really hi-speed.

Final step is data transfer to the mainframe archiving system, so the rest of the process on mainframe can remain unchanged. Storing ASCII data on a mainframe is certainly not an optimal solution: it requires additional administration, and a part of information (metadata) is not used. We hope it is temporary (until completely supported on LAN), but until that time it should be automated as much as possible. Job control script for mainframe archiving system is generated on LAN and transferred together with data, so data transfer can be controlled by the end-user as well. This part of a system has only a basic functionality and is still under development.

Hi-speed data entry needs standard screen layout and standard commands. Both was discussed with end-users and prepared in Blaise, using Modelib Editor and DEP Menu Manager. Combining these tools and programmable keyboard, the work for data entrists didn't change very much. With a few exceptions all the main commands have the same keys as before, and all commands are executable without mouse.
Last version of generator uses Blaise 4.3 (build 4.3.2.436) and Visual Basic 5.

4. Preparing data entry application step by step

A new data entry application is defined and generated through VB "development" user interface that integrates all the necessary interactive and batch processes. The interface is used by supervisor -data entry administrator who specifies and generates data entry model. Another, different user interface is prepared for "production" data entrists.

The user responsible for defining data model (supervisor) has to execute the following steps:

- Step 1: Defining a new data model for hi-speed data entry. Supervisor enters all the necessary specifications into Blaise data model for generation.

- Step 2: Generating all applications. Before compiling, generated datamodels for data entry and verification are presented on the screen (Blaise editor), so additional improvements can be made if necessary.

- Step 3: Testing in development environment. Complete process (entry, verification, creating final data set, file transfer to mainframe) should be tested before it is implemented in production.

- Step 4: Transfer of application files to the production environment (different folder on the same server).

Before the production work starts, the application is shortly tested once again in production.

5. First experience

First application in production was one of monthly traffic surveys with only about 1000 records. No serious problems were noticed and there are already some new data models defined. New system was in general well accepted between end users. There are still some problems to be solved with verification, which is more different compared to the old solution. It seems like we succeed to bring it close to the end-user, but on the other hand this made the system more complex and more complicated to maintain. We hope that the "balance" between users and developers tasks is still OK. To leave the process control completely to the users, the administration still has to be improved.

6. Conclusions

In future, hi-speed data entry at SORS should be gradually replaced by recent data entry techniques, which can be better integrated in a modern information systems. But while migrating to a LAN platform and redesigning processes, an efficient solution to support traditional hi-speed data entry is necessary to obtain the statistical production.
What did we get with the new application ? There are some answers:

- functionally seems to be a successful replacement for the old one
- it is flexible and well integrated in the SORS information system (and the future strategy)

- Blaise application development is expanded (not just CAI) - in-house standardisation
- generator can be used to support process, running completely on a LAN platform, so it is an important step in the migration to the new environment
- it could serve as starting point for non-EDP people (or people who learn Blaise) when developing data models
- by adding checks, comments etc. generated data model can be used as conventional CAI application

There are also some negative points: generator became relatively complicated to maintain (due to verification module, special user's needs etc); we still don't know exactly how far we need to go with administration and automation of the process. But in general we believe that our solution is a good choice in the existing situation, and that it will contribute to the development and modernisation of the survey processing system at SORS.