The 1995 June Area Frame Project

Mark Pierzchala, National Agricultural Statistics Service USA

Blaise III (version 1.05) was used in the 1995 June Area Frame (JAF) Survey which is NASS's largest agricultural survey. This survey interviews about 50,000 farmers during the first two weeks of June, with official estimates based on these data released at the end of June. One state (Indiana) used Blaise for CAPI and for Interactive Editing while two other states (Pennsylvania and Wyoming) used the system only for Interactive Editing after data were collected on paper questionnaires. The project was successful though in CAPI two bugs were found that had to be overcome early in the collection period. In preparing for this large survey, design challenges were met so that the instrument could be used in multiple modes for 45 different versions of the survey. Performance of the instrument for CAPI and CADI was good on 486 machines. Both enumerators and editors liked the functionality and useability of the system. The Manipula utility which was used for reports and for data movement was very slow though its functionality (feature list) is greatly expanded over previous versions. Budgeting constraints will limit the implementation of CAPI for other states, however, the instrument could be used in all states in 1996 for editing. NASS views Blaise III as having great functionality and great potential but needing to mature. This survey helped Blaise III in that maturation process.

1. Design of the Instrument

The first design challenge is that the June Area Frame Survey can be viewed as a multi-level rostering situation of 4 (or 5) levels. The first level is the segment, usually a one-square mile area of land, which is the Primary Sampling Unit (PSU). Within the segment are up to 78 tracts (land within the segment operated by one operation), within each tract are allowed 35 crop fields (for Indiana CAPI, but 85 for Pennsylvania and Wyoming IE), and each field can have up to 3 crops within a crop year. Within each crop 2 utilizations are allowed but this is not really treated as a 5th level in our design. The tract is the unit of interview as the segment serves only as a unit of administration. Thus the main instrument is a tract-level instrument while a much smaller instrument is used for the segment-level data. The tract instrument will read the segment data when it needs segment-level information and the segment instrument will read the tract data when it is necessary to generate a progress report on the acreage covered by tracts within the segment. The rostering aspect of this instrument did not prove to be much of a challenge as Blaise III blocks, tables, and arrays handle this kind of situation easily.

In its current manifestation, the tract instrument contains about 8,000 overall potential questions and the same number of edits plus many computations and routing instructions. Test versions of the instrument approached 27,000 questions and a like number of edits but this was unnecessarily large. Blaise III never had any capacity limits so one problem we had in Blaise 2.5 (regarding capacity) was indeed solved. Instrument performance was good on a 486 laptop with no noticeable slowdown occurring no matter what the length of the interview. However, performance of Manipula for reports or file handling was very slow.

A related design challenge was that one operator (farmer) could operate land in 2 or 3 PSUs. It is desirable to collect all information from the farmer at one time, and without repeating common farm-level questions for each tract form that must be filled out. So parallel rosters were set up for crops fields, cattle, hogs, and land values which ask information for the tract only. Thus where one roster of 35 fields would normally suffice, there were 2 parallel rosters of 35 fields each to allow the interviewer to ask all pertinent questions as if they all belonged in one form in the first place. This worked well for interviewing but raised questions of data management and increased the size of the instrument just to handle a rare situation.

The second major design challenge was the multiple version aspect of the survey. The JAF survey has 45 versions, almost one for each state. In a previous conference I demonstrated a way to generate versions of questionnaires in a related survey. For the JAF however, a variation on this method was used. Rather than generate separate instruments we decided to use one instrument that could be driven 45 different ways based on an external file of specifications. However this involves much more than just setting flags for whether a question should be asked or not. The external file must itself be based on national-level specifications especially for crops and grain stocks. The following schematic explains:

CROPS

national-level specifications

STOCKS

national-level specifications

TOGGLES

state-level specifications

1,200 questions

TRACT INSTRUMENT

8,000 questions

TOGGLES is the central source of specification for one state. When in TOGGLES a crop (or stock) can be chosen. Meta data from the CROPS (or STOCKS) data base is then transferred into the TOGGLES data base. This prevents a lot of retyping of the same information. Information in TOGGLES includes question-by-question toggles, item codes and SAS Variable names for appropriate questions, some question wording, soft edit limits, and for the crops, a state-specific screen design for interviewers in CAPI. In all, there are about 1,200 potential ways that specifications in TOGGLES can vary from one state to the next. This method worked well; the only limitation was having enough memory to handle the external files. In fact, for Indiana where the laptop computers had only 4 Mb of memory, the external files were too large and we ended up generating a state-specific instrument where these meta data were incorporated.

Under this method of instrument production, a question (or code) in the crops or stocks part of the questionnaire can have one meaning in one state and another meaning in a second state. Thus the instrument itself does not hold all meta data. When reading data into or out of the instrument, or when flashing questions on the screen for the interviewer, meta data from the external TOGGLES must also be used. Again, this worked well in Blaise III. The external data models CROPS, STOCKS, and TOGGLES are all Blaise III data models. They are set up so that non-programmers can state the specifications without having to get into the source code. One advantage of this was seen in Pennsylvania when it turned out that specifications were wrongly stated. The state personnel (who had never before seen Blaise III) were able to go into the CROPS and TOGGLES data models and correct specifications which then allowed them to proceed without the instrument being re-prepared.

A third design challenge was the merging of CAPI and CADI considerations into one data model. Blaise III goes a very long way into making this possible and in fact has no deficiencies in this area. However, there is still considerable thought that must go into accommodating both modes. It is one thing to do post-collection editing in CADI after all data are collected in CAPI, and quite another to do post-collection editing in CADI when data are collected by CAPI in one state and by paper in another. There are considerable differences between CAPI and paper modes of collection and the way that data are handled after collection. For example, in CAPI we rarely use the empty attribute for a question. All questions on the route are to be asked and if the answer is zero then a zero must be entered. However, when data are key punched from paper, zeros are not typed, a blank then represents a zero. The upshot is that we had to program a considerable number of edits that would require an entry in CAPI but allow an empty in CADI. Also in CAPI there are a considerable number of screening questions for each section that are not item code questions. Routing instructions must then be written to accommodate CAPI where the screening questions are necessary and item code data where the screening questions are not entered. Despite these differences in modes, it was still very worthwhile to do both in one system. Well over 90% of the code was applicable to both modes and thus saved considerable resources that would otherwise have to be spent programming the same thing in two different systems. The ability to switch dynamically between modes made testing much easier than in previous versions of Blaise.

A fourth design challenge was to maintain good performance during interview. Blaise III has the ability to maintain speed no matter how long the interview and at the same time enforce all appropriate edits and routes. However, this ability works better for instruments which have structure that is common to rostering situations. We were worried that a large instrument with many blocks at the data model level would be slow in Blaise III. However, on a 486 laptop there was never any problem. We did take two preventative measures from the start based on performance research conducted in December 1994. We declared all LOCALS at the lowest level possible and we reduced the number of parameters being passed into blocks from external files by combining separate bits of information into one string (which was then picked apart as needed). The concern with performance in Blaise III stems from the way it administers parameters which then decides which blocks have to be checked. If there are a lot of parameters, then the parameter administration can take a long time. NASS instruments typically generate hundreds of parameters. This is no problem in well-structured instruments (where the structuring limits the number of parameters administered at any one time), but is a problem for instruments where there are many blocks on the data model level. We expect Blaise III version 1.1 to eliminate this problem by allowing parameters that do not have to be administered.

A fifth design challenge was to manage the survey totally by electronic means both on the laptop and on the LAN in the office. This challenge was not met. We were expecting to use the newly enhanced version of Manipula called ManiPlus which would have allowed point and shoot access to forms and the like. However, ManiPlus is a version 1.1 product which was not available at the time we went into the field. Thus laptop management was largely done both by paper and by regular Manipula generating reports from DOS BAT files. We have seen ManiPlus in alpha versions and expect this utility to cover all of our laptop needs as well as some for editing. For example, we would like to do away with the segment-level instrument next year and let ManiPlus handle its administrative functions.

2. Blaise III CAPI in the Field

The CAPI part of the test met most expectations once two bugs were overcome. As interviewing software Blaise III performed well. The 35 interviewers in Indiana were able to handle a variety of difficult situations including non-response, handling two tracts at once for one operator, error correction, and the large field table roster. The majority of the interviews were collected outdoors, sometimes in adverse conditions. In addition to the computers the interviewers had large aerial photos where they had to draw off the farmer's land and fields, so logistics were difficult. Interviewers persevered despite these challenges and the fact that farmers were often impatient as they were behind in their planting. We were relatively pleased with this aspect of the project.

However the bugs initially caused a lot of consternation for both interviewers and office staff. The first bug was in Manipula which would sometimes blow up in a runtime 216 error when copying forms with comments. The fix was to read data from Blaise III format to ASCII, transmit them, then read them back into Blaise III. This worked but reduced the overall efficacy of CAPI because subsidiary information was lost. Comments, DontKnows and Refusals, and error suppressions did not survive the Blaise to ASCII to Blaise dance (comments were read out to a separate ASCII file).

The second bug concerned a part of the grain stocks section. The data would be collected as expected and saved when the interview was closed. However, if an interviewer brought the data up for review this part of the instrument wouldn't be on the route. Thus when the form was saved a second time that part of the grain stocks information would be lost (temporarily for the most part, we had backups). We don't know the extent of the problem here, but most interviewers did not normally review forms, they just sent them. An ad-hoc patch was put in the instrument to prevent this from happening.

Both problems were discovered late on Thursday, June 1. Solutions were conceived and executed on Friday, June 2. On the positive side, NASS discovered it could deal with adversity, even modify procedures, during the operational survey where the data collection period lasts only 15 days. However, NASS does need to think more in advance about how to handle such emergencies and how to test to avoid them.

NASS had a variety of problems associated with hardware. The subnotebook computers came from 3 different purchases; as a result we had Zeos, NEC, and AMS laptops (laptop models don't stay in production very long). The Zeos computers were 2 years old and by the second week of the survey 6 of 15 had broken down. It is possible that 4 pound subnotebooks are not as reliable as 6 pound subnotebooks. The one-year-old NEC and new AMS computers held up better but the NEC battery life was often disappointing. In fact some interviewers had to switch to paper when all 3 of their batteries were dead. Charging batteries was difficult because none of the interviewers had external battery chargers. The computer could be used to collect data or it could be used to charge batteries, but it could not be used for both at the same time during a personal interview.

The NEC and AMS computers had 4 Mb of RAM; the Zeos had 8 Mb. The smaller amount of memory was barely sufficient for the instrument. In fact it was difficult to recall some forms for review after they had been closed. They could be recalled, but only after Blaise III had been 'exercised' by bringing up 2 or 3 other forms first. NASS will have to buy more RAM and battery chargers.

3. Interactive Editing in Blaise III

Interactive editing proceeded well from a Blaise perspective. Several minor updates in the instrument were needed but that was a NASS problem. Data were read directly from an 80 column item-code format into the Blaise III data set and out again with Manipula. Cameleon was used to set up a considerable part of the Manipula programs that did this. The new editing interface was very well received by the users. On occasion Blaise would have to read information into the form from a double external file read. The first would obtain an ID from the list frame, the second would obtain stratum codes associated with the ID. The first file was about 2000 lines deep, the second about 90,000 lines deep. This double read of external data was executed almost instantaneously by Blaise. Pentium machines were used to handle the batch activities; otherwise 486 machines were used by most editors. All 45 of our offices have equipment and staff in place to handle the interactive edit; thus we expect to see a move to make this fully operational. The instrument has already been written to handle all states and all we need are state-level specifications which are entered into the TOGGLES data model. Also needed is a lot of testing by operational personnel.

Overall the experiment was very successful. This article is being written on June 26 while the survey ended June 16. Thus several weeks of review is still necessary to completely evaluate our use of Blaise III in the June Area Frame.