Lessons learned programming a large, complex CAPI instrument

James M. O'Reilly, Battelle, USA.

Presented to International Blaise Users Conference
October 1993, London, UK

Introduction

The National Survey of Family Growth is the baseline survey of female fertility in the U.S. It is conducted by the National Center for Health Statistics about every five years using a nationally representative sample of 10,000 women ages 15 to 44 . The agency elected to use computer-assisted personal interviewing for the first time in the 1994 cycle.

We will discuss a number of challenges faced in programming the CAPI instrument for the October 1993 pretest. Some were the result of implementing innovative data collection techniques, such as a custom event-history module and audio computer-assisted self-interviewing (ACASI). Others were the result of the sheer scale of the questionnaire, which grew to 27,000 lines of Blaise program code and requires two hours to administer in a stripped-down form.

Description of the Survey

The NSFG is the principal U.S. survey of female fertility. For the 1994 round a nationally-representative sample of 10,500 women ages 15 to 44 will be interviewed on a range of topics related to fertility. The major sections of the survey are:

A. Education and family background

B. Pregnancy and birth history

C. Marital and relationship history

D. Sterilizing operations

E. Contraceptive history

F. Birth control and family planning services

G. Birth expectations and desired family size

H. Infertility services

I. Background and work history information

J. Abortion (pretest ACASI experimental treatment)

K. Respondent feedback (pretest-only)

A pretest of 800 women in six primary sampling units is being conducted in October and November 1993. The pretest includes an experiment to test whether abortion reporting can be improved through alternative interviewing approaches. One-third of the women will receive an ACASI section on abortion during their in-home interview. Another third will be interviewed at a 'neutral' site outside the home. And the remainder will be interviewed in-home without ACASI.

To better understand factors affecting fertility and contraceptive use, the survey seeks detailed information about dates events occurred, relationships, types of services, contraceptive methods used, and reasons.

The RTI-Battelle NSFG Team

The worked reported here was conducted under a contract to the Research Triangle Institute from the National Center for Health Statistics. Two Battelle scientists participated as consultants to RTI. Michael Weeks of RTI is project director.

Pretest questionnaire development was lead by Dr. Judith T. Lessler of Battelle, assisted by Susan H. Kinsey of RTI. Dr. James O'Reilly of Battelle lead the CAPI development team, assisted by Chris Carson and Donna Jewell of RTI. The programming work was performed by: O'Reilly (Section A [1/3rd], B, C, D, E, G, J, Method Calendar, ACASI, NSFG_SYS driver) and Chris Carson (Sections A [2/3rds], F, H, I, K).

Selection of Blaise

Shortly after the contract for the NSFG was awarded to RTI in October 1992, an in-depth appraisal was undertaken to determine which CAPI package would best meet the goals of the project. A comparison was made of the functional capabilities of Blaise (version 2.38) and CASES (version 3.5) to manage 24 important characteristics of the NSFG instrument The two packages were found to be equally capable on a number of characteristics. However Blaise was judged superior on many of the key items. These included:

- answer entry (date types , tabular and other complex answer types, decimal answers);
- interviewer navigation (backing up, changing answers, and moving forward);
- question formatting and presentation (fills, long questions)
- computing/calculations (date math, real numbers, long string lengths);
- consistency checks,
- rosters, and
- general programming style.

Development Zeitgeist

The major reason for switching to CAPI was to improve data quality through:

- collecting more detailed demographic data on household composition, male partners, birth control methods used, and labor force history;
- extensive consistency checks on important fields;
- extensive tailoring of question texts to personalize questions; and
- other features of CAPI technology such as range checks and total control of the question routing.

In addition, for the pretest, computer-assisted interviewing technology (CAI) permitted testing techniques to improve data quality for collecting event history information, and for improved reporting of abortions through ACASI (O'Reilly, 1992).

Several general features of the project made designing and developing the CAPI system demanding. First, the survey was moving to CAPI from paper-and-pencil. This meant redesigning the structure of large sections of the instrument as well as many individual questions. The second crucial feature was that the professional staff at NCHS, RTI, and Battelle were intent on exploiting every potential advantage of CAPI to enhance the quality and efficiency of the survey.

To achieve this, a continuing effort was made to avoid limiting assumptions about what CAPI could or could not do. The CAPI development staff urged questionnaire developers not to assume restrictions and limitations in the CAPI system. Rather they were told to expect that CAPI could do whatever they wanted. The demographers, methodologists and survey research specialists at NCHS, RTI and Battelle were encouraged to describe whatever questions and interview structures they thought would be useful. It would be up to the CAI systems development staff to program it, or to demonstrate that it could not be done.

Development process

The process that unfolded over the 10 months leading up to the 1993 pretest entailed transforming the paper-and-pencil questionnaire into, first, a much-enlarged paper CAPI questionnaire specification. Next the CAPI-specified questionnaire version was programmed in Blaise. These two steps proceeded section by section for each of 12 major sections. Finally the separate sections were integrated into a single master questionnaire. Testing, correcting and enhancing the instruments occurred almost continuously.

We began work with substantial experience in CAPI systems development. Yet each major step in the NSFG effort seemed to be a venture into unexplored territory, with little published experience available for guidance. Fresh thinking, close teamwork, and rapid iteration of products through the specification-programming-testing-revision cycle was crucial to meeting these demands.

Among the major issues we wrestled with throughout the process were:

- What is the best way to render the various types of paper questions in CAPI?

- What is a suitable written syntax for questionnaire authors to use to specify for programmers consistency checks, range checks, variable text in questions, and question routing logic (for both valid and invalid responses).

- How does one represent the resulting CAPI questionnaire back into paper form for the public to understand what happens in an interview?

- How does one test a complex instruments that has thousands of different routes through it and, as a result of wide use of tailored files in questions, a large variety of question versions?

Each of these issues is worthy of a paper in its own right. We learned much on each. However, it's safe to say that no one on the NSFG questionnaire development or programming teams believes we have adequate answers to any of them. It's probably fair to say that we struggled with each every step of the way and never felt fully satisfied with the solutions we adopted. We hope to describe our approaches and the experience we gained in a subsequent publication.

It is these areas where innovation and development is most needed in CAI systems. It seems likely that the CAI system which make the most advances in them will become dominant in the field.

The NSFG Programming Challenge

As mentioned, the NSFG seeks to gather information on a important topic which has a variety of major components and aspects. There is nothing simple or straight-forward about it. As demographic research in this field has advanced, the need has grown for both more depth and breadth of primary data such as the NSFG supplies.

The need for a substantial volume and detail of data doesn't by itself make programming the CAPI instrument so difficult. The rub comes when that is combined with both heavy use of variable-text questions and complex question ordering so that questions are asked based on multiple conditions.

Questionnaire designers believe that the best way to help women recall the needed information is to tailor questions finely so that the language conforms as closely as possible to a woman's individual situation. The term used by the NSFG team is "conversational inteviewing". The goal is to have the questions personalized as much as needed and have the software do this, rather than ask the interviewer to fill in the correct substitute text "on the fly". The goal is to permit the interviewer to simply read the question from the screen and have it sound to the respondent quite specific and personal. Subject, object, tense, and references for questions vary based on the respondent. Many questions have four or more variant wordings.

Examples from the NSFG

Here are a few examples of demanding sets of questions in the NSFG instrument. In Section A, on education and family background, one subsection seeks to determine every living situation the women was in from birth until she lived on her own permanently. The women is asked for each situation who filled the female and male parental roles (natural

mother, step-mother, grandmother, etc.). Then she is asked whether, when, and why the situation changed (death, divorce, separation, etc.).

When non-natural parents (step- or adoptive parents or parent's girlfriends or boyfriends) are mentioned after the first time, the woman is asked if she is referring to a new step-parent (or adoptive or parent friend) or the one already mentioned. A variety of other subsidiary questions are included to deal with deaths, joint custody, and other situations. To aid recall, questions are tailored based on responses from the previous situations. Up to 12 living situations are provided for.

It was a significant effort to design questions and program Blaise so that the wide range of possible living situations that might be encountered are handled appropriately, and in a way that leads the woman to recall the situations and sequences[1]. The payoff for this effort, if respondents can adequately recall the information, will be that researchers will know for the first time detailed living situations patterns and timings along with the woman's complete fertility experience.

Another example is in Section C, on the marital and relationship history. Information is collected about all husbands and co-habiting male partners. Demographic information is asked on the man's age, education, marital status at the start of the relationship, race, Hispanic ethnicity, and religion. Other questions address the co-habiting history: periods the couple lived together before marriage, the start of the current situation, and periods of living apart before the dissolution of the living situation. Next, questions are asked about first sexual intercourse: date, partner, relationship with partner, and the partner's age, race, marital status and education at the time. Finally, the woman is asked the number of intercourse partners since 1989. For each man who is not a husband, boyfriend, or first partner already addressed, question are asked about demographics and the relationship, including dates of first and last intercourse.

The challenge of this series is to pose only the appropriate questions, to do so in a manner in keeping with the sensitivity of the subject, and in a way that a woman with more than a few partners can understand. This required substantial amounts of programming to handle the question fills and to organize the intricate question routing logic. Again the benefit of this effort may be unique new information to improve understanding of various fertility issues.

Other examples of demanding sets of questions include determining contraceptive methods used month-by-month since January 1989 (discussed below) and a detailed employment history.

Shortening the interview length

The breadth of areas to be queried and the depth of detail desired produced a large questionnaire. The NSFG pushes the limits of the CAPI technology--the DOS 640k environment as well as the CAI software development system.

It also means a long interview. Preliminary testing of the full instrument in August indicated that interviews as long as three hours might not be uncommon. Some questions were then dropped. But to understand better the time demands of components of the

survey, a strategy was adopted for the pretest in which respondents get different subsets of randomly assigned sections so that the average survey length would be two hours or less. This allows all sections to be adequately pretested and provides detailed information on timing so that informed decisions could be made for the main study[2].

System development strategy

The core programming strategy to handle the scale, complexity, and tight development schedule was modularity, modularity, modularity. We made heavy use of Blaise's block structures, "include" source files, and separate Blaise instruments for major sections of the overall questionnaire. With a system that ended up with 27,000 lines of code and had significant changes, fixes, and revisions being made to it almost until it was loaded on the interviewers' laptops, it is doubtful we could have succeeded, or even survived, without this modular approach.

Table 1 gives statistics on the 10 separate Blaise instruments which combine to form the NSFG pretest system. Perhaps most indicative of the size of the system is the number of uniquely defined questions (1,826), uniquely defined blocks (203), the consistency or signal checks (3,505), the source code files (103), the questions in the data file (7,658), and the total length of the data record (58,765).

Table 1: Statistics on NSFG Pretest Blaise Instruments

	A	B	C	D	EA	EC	FGH	I	J	K	ALL
Source files	20	18	17	10	10	9	15	14	8	6	103
Source lines	5429	3944	4870	2491	3319	2591	4735	3624	1002	1681	26967
Uniquely defined questions	289	255	253	126	124	70	362	201	68	78	1826
Overall used questions	795	2537	1965	251	366	576	497	631	180	75	7873
Uniquely defined blocks	40	32	27	8	20	9	34	22	7	4	203
Subfiles	2	7	2	5	0	1	0	0	0	0	17
Defined answer categories	607	592	560	571	582	604	625	633	29	523	5326
Length of open answers	7519	15379	10718	2483	2743	1345	5748	6323	56	1417	53731
Length of error messages	6553	8044	11660	2062	13596	3583	5877	3229	565	702	55871
Length of question text	26283	27737	24125	10666	8887	7441	38813	19690	8300	7264	179206
Length of answer text	13933	14416	13501	14398	14178	14738	14923	15278	355	12796	128516
Open questions in data file	206	411	162	34	132	17	156	227	3	47	1395
Date questions in data file	165	250	586	17	63	80	28	62	0	1	1252
Enumerated q's in data file	329	1324	1033	184	100	380	216	278	144	19	4007
Set questions in data file	4	137	51	6	15	1	30	3	0	0	247
Subrange questions in data file	64	387	79	4	31	67	51	53	17	4	757
Total questions in data files	768	2509	1911	245	341	545	481	623	164	71	7658
Questions not in data files	27	28	54	6	25	31	16	8	16	4	215
Number of tables	60	132	4	0	0	28	7	62	0	0	293
Maximum data record size	7513	6062	5714	5714	2642	6626	7513	7513	2642	6626	58565
Total number of signal checks	404	645	1230	47	584	111	147	306	14	17	3505

From the outset it seemed likely that the full system would be a connected set of individual CAPI modules. The overall scale of the questionnaire was likely to exceed Blaise's stated limits. Also the NSFG design included two non-standard components--a custom event history module written in the Foxpro database language and an audio self-administered module[3].

For those reasons, as well as the fact that questionnaire development and programming had to proceed almost in parallel, section by section, to meet the schedule, we emphasized a highly modular design. Each of the 11 major sections were first developed and tested as stand-alone instruments. Any references to responses from preceding sections were modeled with a "preload" block.

The highly structured architecture of the Blaise system made it possible to organize and develop a software project of this scale. We carefully designed our system to follow that architecture and implemented it with that approach used at every level.

Impact of linking separate instruments for the NSFG

Breaking up the NSFG questionnaire into 11 separate sections was unavoidable, we believe, because neither Blaise or any other CAI package could do all the things required within the confines of a single application. One can imagine a system that might be able to do this, and possibly some vendors or developers may claim this capability. But given the realities of this project--the schedule, data collection requirements, and the budget--such claims would be hard to believe.

The chief potential impact of this approach is that the interviewer can not backup and correct a response that occurred in an earlier section and then proceed forward following the logic dictated by the current response set. This could be permitted if we allowed the interviewer to reopen completed sections for a case in edit mode. But that is not allowed in the pretest.

In general, the NSFG team believes the need to correct key fields beyond the current instrument will be rare because most key fields are subjected to consistency checks at the time of entry. To judge whether this assumption is sound, in the pretest interviewers are trained to enter a Blaise note when a changed response outside of the current instrument is mentioned by the respondent.

Structure of the NSFG CAPI System

Hardware

The PC used is a Toshiba 1900 notebook system. Its characteristics are: weight six pounds, 486/25SX processor, 2Mb RAM, 80Mb disk, VGA monochrome screen, internal modem, MS-DOS 6.0. For the ACASI section of the interview, the interviewer connects an external digital audio device (1.5 lbs with cables) to the PC. The respondent listens to the questions over headphones.

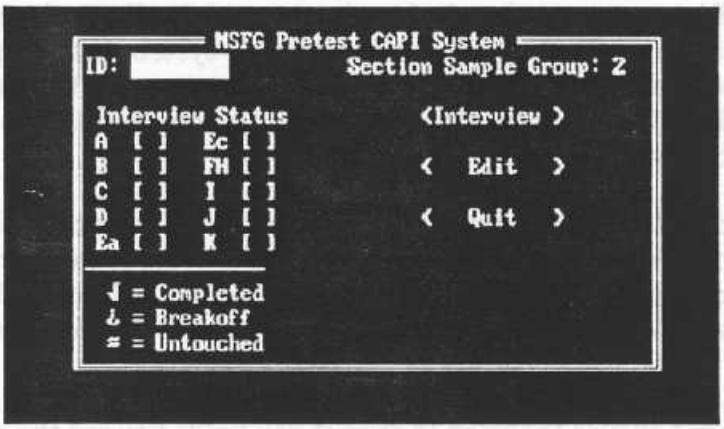
The speed of the notebook PC, while not crucial for the adequate functioning of the system, is an advantage. Transitions from section to section are quite fast, even when, during the

change, the driver program performs an overhead process, such as opening a DRF file to check for a break off. Within about one second after confirming the end of one Blaise section, the system returns to the driver, the outcome is checked, and the next Blaise section is executed. To the interviewer and the respondent, the questions flow across Blaise sections as if everything was in one piece. The only pause of more than a second or two comes at the beginning of the Method Calendar.

NSFG_SYS Driver

The top-level driver program, written in Foxpro2, controls access to the data collection modules. Called NSFG_SYS, this program takes case IDs entered by interviewers, confirms that the ID is assigned and active, and checks the status for the case of each of the 10 Blaise instruments. If the case is not completed (because it has not been started, or was started but not completed fully), NSFG_SYS starts the interview in the appropriate module--usually at the beginning, or for a break-off at the module where that occurred. The NSFG_SYS screen is shown in Figure 1.

Figure 1: NSFG_SYS Screen



Foxpro2 was used because of its event-driven and other powerful programming capabilities and its Windows-like interface and screen objects. This provided the capability to build a suitable user interface for interviewers and let us program the control of all steps in the process.

NSFG_SYS calls the MS-DOS command shell to execute each Blaise instrument in order, supplying the command line parameters to start the current case in the appropriate mode. After the Blaise module is completed, NSFG_SYS regains control of the process. It checks the outcome of the last module (reading the appropriate *.DRF file and determining whether the case's record shows a complete or broken off interview). If a break-off is found, the cycle is terminated. Otherwise NSFG_SYS immediately executes the next Blaise module.

NSFG_SYS's other main function is to communicate with the PC field control system--generating events describing all interviewing actions, and handling additions and transfers of case assignments.

The CAPI Instruments

The core elements of the system are 11 stand-alone MS-DOS executable components--ten Blaise instruments (A, B, C, D, EA, Ec, FGH, I, J, K) and one custom-programmed Foxpro application to collect information on the month-by-month contraceptive method use since 1 January 1989.

These 11 components are linked together into a unified higher-level instrument through the capability of individual Blaise instruments to access the indexed data files of other Blaise instruments. This crucial capability, through the READBLAISE command, provides fast and definitive access to the current case's responses in previous sections. Because of its speed and reliability, it appears to the user that everything is happening instantaneously. Table 2 describes the data from preceding sections which are utilized in later sections.

Table 2: Data items used from preceding section

Section	Preceding section and data items
B	A:Age, DOB, marital status, living with a man, living with a boy friend, man's name
C	A-set plus B: age at menarche, currently pregnant, number of pregnancies
D	A-set plus B-set plus C: times married, ever had sex, date of first sex, # partners in last 12 months and lifetime, lived with other men, # other men lived with, date of marriage to currrent husband
EA	A-set plus B: number of pregnancies; currently pregnant; pregnancy outcome, pregnancy end date, gestation period, baby's name (each pregnancy); C: each husband's name, marriage date, end date, date began living together, date ended living together, each other man's names, beginning & ending data of living situation. D: Sterilization date, sterilization reversal date, partner sterilized
Ec	A-set plus EA: pregnancy interval information, contraceptive use information for intervals prior to 1/89. Method calendar: monthly method used 1/89 and later
F	A-set plus B:ever had an abortion C:times married, ever had sex D:sterilized, place of sterilization, date of sterilization
G	A: date of birth B: currently pregnant, number of liveborn children C: R sterile; partner sterile D: able to have child
H	A-set plus B: age at menarche, number of pregnancies, number of children born alive, date of first livebirth, date last pregnancy ended C: date of first sex; partners in last 12 months, partners in lifetime.
I	A-set plus currently attending school, highest grade complete, have a diploma, date last attended school, number of persons in household, number of children less then 13 living in household Ever lived on own B: Any liveborn children
J	EA: ever used 19 contraceptive methods Ec: sex in last three months

Two other data linking mechanisms are used. The Blaise EXTERN program is called twice by NSFG_SYS to generate indexed data files that are accessed by Blaise instruments using the READFILE command. One of these is the sample control database which indicates for the case what sections of the interview to execute and which experimental condition the case is assigned for two question order experiments included in the pretest. The second one is called at the end of the Method Calendar section and provides to subsequent Blaise sections with responses from that section.

The Blaise CONVERT program is used to translate a number of data fields on pregnancy intervals, husbands, boyfriends, sterilization, and contraceptive history from sections A through EA into an ASCII record. It is executed after the EA section completes. The ASCII

data record is then read by the Foxpro2 Method Calendar module and used to organize the monthly contraceptive method questionning.

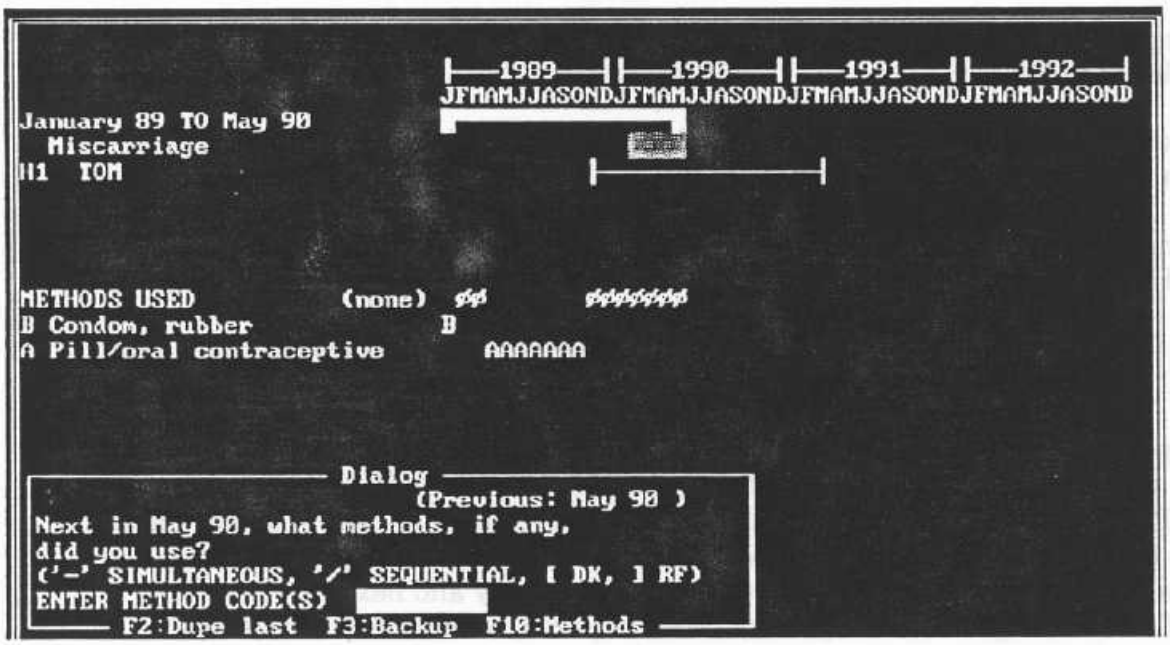
Method History Application

Perhaps the most important purpose of the NSFG is gathering information about contraceptive practices used by women. Demographers desire month-by-month contraceptive use for the last four or five years. Used in conjunction with information on pregnancy intervals and gestation periods, this data provides critical information about a variety of fertility issues.

However, recalling information of this detail and precision over such a long time span is obviously problematic for many women, especially those who have changed methods a number of times. Survey methodologist believe the best way to enhance a women's recall of events of this type over a long time period is through use of a calendar that places the events in the context of other milestones and memorable occurrences in her life.

We designed and implemented such a calendar using Foxpro2. Figure 2 shows the method history screen. It is used with the respondent and the interviewer sitting next to each other and the interviewer reading the questions from the dialog box and entering the responses. The respondent is able to see the screen revised based on her responses. A wide range of information from the previous Blaise modules related to male relationships, pregnancies, gestations periods, and so forth are passed into the method calendar automatically.

Figure 2: Method Calendar Screen



In the example, the questioning focuses on the period from January 1989 until the woman's miscarriage in May 1990. Years and months form the columns on the right side of the screen. The pregnancy interval (January 89 to May 90) is depicted graphically on the next line. The following row shows with a bar of hatched characters the months the woman was

pregnant (February to May 1990). The next few row shows the male relationship(s) for the period--her first husband Tom, from November 1990 (when they began living together) to March 1991 (when they separated)[4].

The bottom half of the screen displays the contraceptives used by month. In the dialog box at the bottom the question text for each month is displayed for the interviewer to read. As the women responds concerning the contraceptive she used for the month the interviewer keys the appropriate letter and that use is indicated on the screen above.

The method calendar proceeds pregnancy interval by pregnancy interval. Other functions included in the calendar are a help screen to show the 19 methods and codes, keys to duplicate the previous month's response and to back up and correct the last response, and input routines to permit entry of multiple method use in a month (either sequential or simultaneous).

Foxpro2 was used to develop this application because of its strong character-based, windows tools, project development system which permits rapid development of a complex application, event-driven programming capabilities, and other tools.

Blaise complements the Foxpro2 system well and provides important tools that permits the smooth integration of the method calendar application in the flow of the series of Blaise instruments. The steps in this process are:

- The Blaise EA module reads responses from sections A to D on the woman's age, pregnancy status, pregnancies, male relationships, and sterilizations and saves that information along with responses to questions on contraceptive method use before 1989 in the EA Blaise data record.

- On completion of the EA section, the Blaise CONVERT program write out an ASCII data file of the EA section record.

- The Method Calendar then begins, reads the needed fields from the ASCII data file using a data dictionary and proceeds to collect its data.

- On completion, the method calendar writes out an ASCII data record of information needed by the next Blaise section, EC.

- The Blaise EXTERN program is run to build an indexed Turbo Toolbox data file that EC will read.

Another important Blaise feature used in development was the SETUP language. We developed a custom setup which generated a comma-delimited data file describing the EA ASCII file. This was read into Foxpro2 to become the data dictionary used to access the EA data for the method calendar. This was used often during development as the EA data record was altered and allowed development to proceed quickly and flexibly. Without it, some sort of hard-coding solution might have been required, which would have made changing and enhancing the method calendar much more difficult.

Orchestrating the ACASI Instrument

Another complex element in the NSFG is integrating the ACASI section into the flow of the interview[5]. The ACASI section is placed after the last regular substantive section ('I') and before the interview evaluation section ('K'). The problem faced is that the audio system's driver, which runs in the background as a TSR while the Blaise ACASI section operates, uses 140k of RAM. This driver provides the interface between the Blaise section and the digital audio system which plays the screen text at the same time as the questions are put on the screen by Blaise.

The early, heavyweight sections, especially 'A' and 'B', need about 600k to run. So there is no way they could function with the audio TSR active. A mechanism had to be designed to revamp the DOS environment to bring in the audio driver just before starting the ACASI section and then remove it. This is done by the NSFG_SYS program. It not only controls whether the ACASI section is presented to the interviewer for the current case. It also performs the needed steps to orchestrate the ACASI section.

The PC's AUTOEXEC.BAT contains the following lines at the bottom

```
REM NSFG ACASI hooks. Written 08/27/93 13:18:43
IF EXIST \BEGACASI.BAT CALL \BEGACASI
IF EXIST \BEGACASI.BAT CALL \DELBEG
IF EXIST \ENDACASI.BAT CALL \ENDACASI
IF EXIST \ENDACASI.BAT CALL \DELEND
REM NSFG ACASI hooks. Written 08/27/93 13:18:43
```

NSFG_SYS writes the four DOS batch command files mentioned in the above lines. On the next reboot of the system, if they exist, they are called from the system's AUTOEXEC file. The first file runs the audio driver and starts the ACASI Blaise instrument for the current case. The second batch file, which runs after the ACASI section is completed, deletes the first batch file and performs a 'cold' reboot of the system. When that happens ENDACASI.BAT is called, which restarts the NSFG_SYS program with appropriate command line parameters to move the interview directly into the remaining section 'K' for the case. The last batch file, DELEND.BAT is executed after the entire interview is completed and NSFG_SYS is executed. It deletes ENDACASI.BAT.

This cascading sequence is triggered by NSFG_SYS when an ACASI interview is required. It writes the four batch files and performs a 'cold' reboot to set things in motion.

Other important Blaise functions for the NSFG

We have already mentioned how critical to the NSFG are the Blaise system's modular architecture and its capabilities to pass information between Blaise instruments and to and from other data records. Other more specific Blaise features were also quite important in developing the instrument.

One is the ability in Version 2.4 to create user-defined functions in Turbo Pascal which can then be used as extension of the Blaise language was very helpful. The custom function we used most was a short one to translate a cardinal to an ordinal number. One passes an integer value such as 1, 3, or 5 and the function returns the string 'first', 'third', or 'fifth'.

While this was very simple to create, it was used many times in constructing the fills for the 'conversational' style questions used throughout NSFG.

As noted, the NSFG asks often about the dates various events occurred. We make much use of the JULIAN() function and date calculations to construct fills and consistency checks.

Consistency checks are used throughout the NSFG. The total number of checks implemented is 3,505. These checks are essential to producing the higher data quality sought with CAPI. Many of these checks are of date ranges. But many other types of checks are used as well. The structured Blaise CHECK/SIGNAL consistency check system made programming and using these checks quite effective.

Problems in Blaise

Most of the difficulties we encountered in programming the NSFG in Blaise were the result of making use of more exotic features of the system which are not documented extensively. Or, it was because we did not read the Blaise documentation carefully. Once we learned how Blaise wanted a step done, things worked properly. Throughout the work, we received very responsive and authoritative support via voice, FAX, and CompuServ from the Blaise team at the Netherlands CBS, particularly Maarten Schuerhoff and Peter Stegehuis.

Perhaps the most significant problem that seems to be Blaise's fault, rather than ours, is in the timing of READBLAISE execution and imputations in a CHECK paragraph. We found often that there was a synchronization problem, so that the accessing of data from an earlier instrument via READBLAISE was not available to the current instrument instantly at the start-up.

We solved the problem by putting in a dummy pause question at the top of each instrument. This required the interviewer to touch a key to continue. With that pause implemented, the synchronization takes place reliably in new interview situations. It also is reliable in reentering a completed or broken-off case when the user moved down the form question-by-question, or page by page using the PgDn key. But, particularly in sections which accessed a number of data clusters from different sections, reopening a section and touching the [END] key to go to the first unanswered questions often yielded unreliable results. Through training, we have prepared the interviewers to function so that these problems will not occur.

We also encountered situations with large complex instruments where a Blaise instrument would 'lock up'--simply freeze so that a reboot of the PC was required. This seemed clearly related to available memory in DOS. Sections would run flawlessly on the Toshiba notebooks with 600k of available memory, but might lock on development systems running on our Novell network with 550K. Also one section, 'B', seemed prone to lock-up even on the Toshiba. So we had to reduce the data structure--from 16 to 12 pregnancies and from 4 to 2 possible children per pregnancy. These adjustments are unlikely to impact on any sample woman.

Given how strenuously we are driving Blaise[6], it is probably somewhat tactless to criticize it for failing to do everything we demand instantly and flawlessly. But we mention these anyway.

Conclusion

With the NSFG pretest beginning field work just as this paper is completed, it is premature to claim too much about the success of this development effort. Still the testing to date makes all on the NSFG team hopeful and, at least some of us, optimistic. Overall, we believe that the Blaise system has performed very well in this demanding application and, further, that we could not have done it with any other system.

References

O'Reilly, J. 1992. "Audio Computer-Assisted Self-Interviewing: New Technology for Data Collection on Sensitive Issues and Special Populations." 1992. With M. Hubbard, J. Lessler, and P. Biemer. A paper presented at the Annual Meeting of the American Statistical Association. Boston, Massachusetts. August 1992.

O'Reilly, J. 1993. "Evaluation of the suitabillity of CASES and Blaise for development of the NSFG-5 CAPI instrument". Paper presented to 1993 Field Technologies Conference. Chicago, May 1993.

FOOTNOTES

---

[1] Yet we admit that this section is still not conversational enough, and will require further refinement for the main study.

[2] This is an unexpected example of the value of CAPI. Within days of learning of the problem of the length of the interview and the development of the section sampling strategy, it was implemented.

[3] The ACASI instrument is a written in Blaise, but it implements audio accompaniment of the screen through drivers and programs running simultaneous to Blaise. This requires fundamentally restructuring the MS-DOS running environment for just that section

[4] Other information shown on the screen, but not used in this examples, is periods of non-intercourse and periods when sterile. This is depicted on the top row above the years and month column labels.

[5] As mentioned, one of the principle goals of the pretest is to determine whether reporting of abortions is increased through the use of a self-administered audio instrument (ACASI), or through conducting the entire NSFG interview outside the home at a neutral site. The sample of 800 is randomly divided into three treatment groups--in-home interview plus ACASI, neutral site, and regular in-home CAPI interview.

[6] In some sections we have four or more READBLAISE or READFILE statements, and some are executed up to 12 times to access all the needed clusters.