Revolutionary Paradigms of Blaise

Mark Pierzchala, Blaise Services at Westat, United States
Tony Manners, Office for National Statistics, United Kingdom

Abstract

Blaise has historically featured several revolutionary paradigms that make it one of the leading computer-assisted interviewing (CAI) systems. By proper appreciation and application of these paradigms Blaise can dramatically improve the way an organisation executes surveys.

Some examples of this revolution from current users illustrate how Blaise has enabled fresh approaches to tricky challenges. These examples cover a range of issues including interviewer usability, integration of data collection with post-collection processing, development of instrumentation, and data and metadata export and manipulation. With proper appreciation and application of these paradigms it is possible to rethink the CAI process and to achieve better data quality in a cost-effective way.

The paper will speculate on how a new paradigm, that of Open Blaise Architecture, continues the pattern of Blaise contributions to improved CAI processes.

Seven Major Paradigms

We first identify 6 aspects of the system that Blaise has embodied since its first versions. While there have been enhancement and elaboration of these paradigms over the years, it is argued that the choices made in its early years have resulted in Blaise's current-day success and worldwide adoption through its adaptability and capability in handling very difficult surveys. These 6 paradigms are *intuitive usability*, *total checking and total reliability*, *structured modularity*, *development based on researcher skills*, *all metadata defined once*, and *no data without metadata*. We identify a seventh paradigm, the recently arrived *Windows openness*, and we speculate on its eventual impact.

Examples given below demonstrate how these revolutionary paradigms have enabled, and how they encourage, efficient ways of approaching all aspects of CAI and survey processing. Appreciation and use of these paradigms are cost effective and have improved data collection and processing.

In this paper, a paradigm may be deemed revolutionary if it represents an approach to a challenge that is a significant departure from previous practice. The discourse identifies how each of the paradigms is a departure from approaches taken in most other systems.

While these seven paradigms are extremely beneficial, there will always be challenges that may require senior and experienced staff to design and develop Blaise instruments along with other non-Blaise expertise and systems. This is particularly true for some very large and complex studies, studies with special requirements, or when shifting an existing survey from another CAI system into Blaise.

Intuitive usability

The default Blaise split-screen display is a readily recognizable hallmark of the system. It combines a question-display area (top part of the screen) with a page display (the bottom part). The question text that is displayed at any moment corresponds to the location of the cursor in the page.

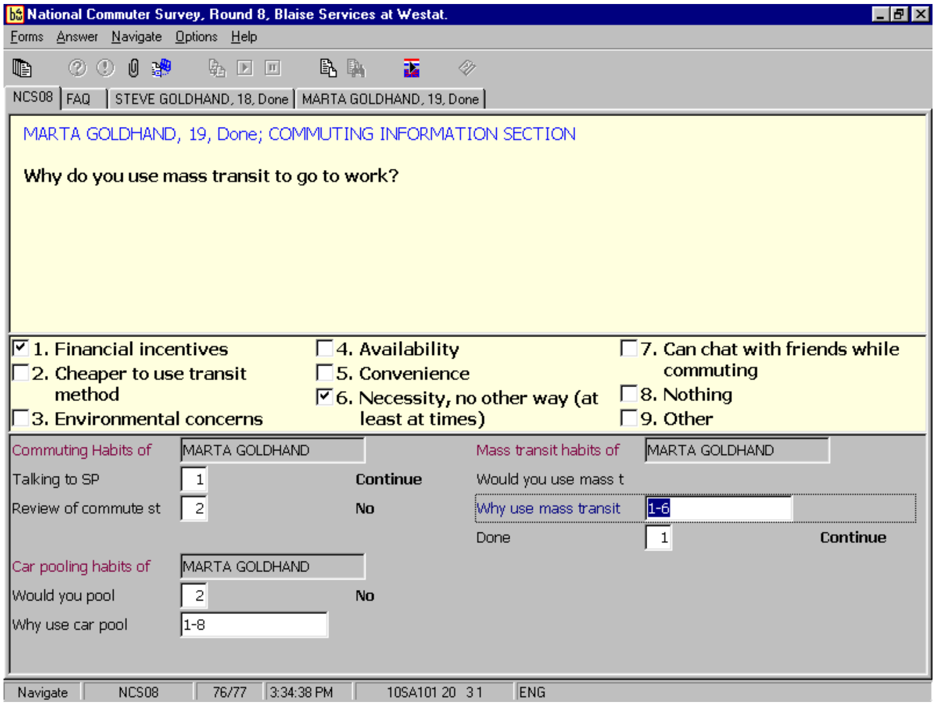

Figure 1: An illustration of the split-screen interface with a formatted Blaise page.

The page in the bottom part of the screen is the basis for the proven Blaise usability for interviewers[1]. By default it is a multi-question display that tells an interviewer about a cluster of related questions, showing which have been answered (with values), which were not on route and which are still to be answered. This information is vital to an interviewer's sense of being in control of interview and combines well with the literal control the page gives for navigation. The Blaise instrument page has many similarities to a page in a paper questionnaire and can contain labels, white space, (multi-lingual) field descriptions, texture, page numbers, and other attributes that enable the interviewer to recognize a page's topic and place in the instrument. The page, if designed well, makes a meaningful element, which enables the interviewer to build an instant mental overview of the questionnaire and where they are in it at any time. Interviewer briefing for the survey need only show the interviewers the high-level structure of the instrument in terms of topic order, and the page can indicate the current topic. The page is also a unit of navigation with the Page Up and Page Down keys, or you can navigate within a page using arrow keys or a mouse. There are many ways to navigate in a Blaise instrument but the page offers interviewers a method that is transparent for them and retains their control of navigation. It is particularly popular for backing up over short distances in the instrument (by far the most common need for backing up which is not triggered by an edit check which has its own direct movement to the variable to be changed). The higher the average data density in an instrument's pages the fewer overall pages that are needed. Twelve intuitive methods of navigation are documented in Pierzchala and Farrant (2000). As we have stressed, the Blaise page gives the interviewer the secure sense of being in control of the technical aspects of the CAI interview, something that is important for the overall quality of interviewing. It should be noted however, that there may methodological or other reasons to limit navigation: Blaise can accommodate this too.

---

[1] Interviewer choice of Blaise as the most interviewer-friendly system was a key aspect in its selection in a competitive tender for CAI software for a major British survey in the early days of CAI when such comparisons could be made with interviewers who had not previously been exposed to any of the systems.

Revolutionary aspect: The Blaise multi-question page stands in stark contrast to the default display convention of most other CAI systems that are oriented towards an item based display of one question at a time. The item-based display can result in the well-known *segmentation effect* (Groves, Berry, and Mathiowetz 1980) where interviewers have little idea of the location or context of questions as they conduct the interview. It is very difficult for them to see the relationships of questions and groups of questions to one another. In these other systems navigation and error repair is difficult even if the system architecture supports backup. The well-designed Blaise page helps eliminate the segmentation effect.

Benefits may include reduced training costs, easier ad hoc navigation and data correction during the interview, easier re-entry into the system, and lower post-collection processing effort. Training costs may be reduced because less time is spent teaching the questionnaire to the interviewer. They can see the whole questionnaire in much the same way as an interviewer can see a paper questionnaire. It is easier and more natural for interviewers to learn the instrument from the meaningful topics in pages than it is from items in item-based systems.

In Europe, where Blaise has predominated for over a decade, interviewers are expected to navigate and make data corrections. Thus instead of getting a note that someone must review post-interview, you should expect to get a corrected value instead, and if necessary, any data from a newly established route. (However, there may be methodological reasons to limit navigation back to certain areas of the instrument. In this case you may still get notes that need to be reviewed.) Clean data from the field are the default expectation in an organisation like the United Kingdom's Office for National Statistics (ONS) which expects interviewers to edit during the interview - indeed, which sees this as a key advantage of CAI. More than a decade's experience of using Blaise has shown that it can be achieved reliably.

During re-entry into the system, it is possible for the interviewer to review data previously entered to get an idea of what the interview is all about. Lower post-collection costs come about in three ways. One is fewer errors and notes to be reviewed post collection. Second is that the page-based overview is also available and beneficial to the data editor. Third is that Blaise has powerful edit and note-review facilities (see below).

Examples of U.S. instruments that take full advantage of the page-based usability include the Consumer Expenditure Survey (U.S. Bureau of Labor Statistics/U.S. Bureau Of the Census, Stanford, Altvater, and Ziesing, 2001), the June Area Frame Survey (U.S. National Agricultural Statistics Service, Pierzchala, 1996), and the Spanish Bladder Cancer Survey (U.S. National Cancer Institute/Westat, Frey 2000). All ONS surveys use the page concept: recent examples are the Expenditure and Food Survey (Office for National Statistics, Gatenby 2001), and Survey of Psychiatric Morbidity among Adults 2000 (Office for National Statistics, Singleton 2001).

Total checking and total reliability

Every time a new or changed data value is entered, Blaise re-executes the rules of every block affected by the new value. That is, it re-evaluates flow, re-computes assignments, and re-assesses the edits within all affected blocks. If appropriate, the routing of the instrument is changed or an edit is invoked immediately. This continual and global re-checking guarantees the integrity of data relationships in the questionnaire no matter where in the instrument the user makes changes. Total checking and total reliability is manifest whether in data collection mode or in post-collection editing mode.

Revolutionary aspect: The continual re-checking of appropriate blocks is in contrast to limited or no rechecking seen in other systems. When the re-checking is started, Blaise makes no assumption about the correct route of the interview. Rather it determines anew the correct route every time there is a new value. The states of the instrument are constantly re-assessed and are re-established every time a new data value is entered or the form is brought into memory. A variant on this is passive checking in edit mode where the data editor can delay the checking until a function key is pressed, allowing several values to be changed before the next re-assessment of the form.

The constant re-checking is one of the architectural enablers of the 12 methods of navigation. The interviewer can navigate and make corrections with confidence that the constant global re-checking will ensure the correctness of the route and edits. This paradigm also allows one instrument to handle both interviewing and editing mode, even where editing is on data from paper questionnaires. An example of this is from the Quarterly Agricultural Survey (NASS) where every quarter half the cases are collected in Blaise CATI and half are done in the field on paper. The same Blaise instrument handles both the CATI and the data editing no matter where the data originated (Schou, 1995). Thus post-processing is a natural capability of the Blaise instrument, eliminating the need to rewrite all data relationships in another system to handle post-collection review.

Other examples of multi-mode use include the UK's CAPI/CATI rotating panel Labour Force Survey (ONS/Manners 1992, Elliot 2000), and Early Childhood Longitudinal Survey-Kindergarten cohort (U.S. National Center for Educational Statistics/Westat, Dulaney and Allan, 2001).

Structured modularity

The Blaise development language has several modular constructs including types (pre-defined pre-codes), procedures, blocks, and tables. The block is the most important and most common of these modular constructs. All of these modular constructs can be used or reused in different parts of the same instrument or in different instruments. Blocks and other modular constructs model natural questionnaire structures, such as a section of a questionnaire, a table, or a row within a table.

An instrument may be considered to be a collection of blocks and other modules and this modularity offers maximum order and a maximum degree of flexibility. It is possible to parameterize a block and use it in a variety of different circumstances. Specification and development can be block-based, independent of the context of the whole instrument. Instrument development can proceed on two levels, that of constructing blocks (and other modules) and that of tying them together.

Revolutionary aspect: Specification and development can be structure-based as opposed to an item-based approach. You can build a library of types, procedures, blocks, and tables, and combine these with project-specific modules. The ability to re-use blocks and other modules encourages and enables the creation of organisation standards and standard survey management modules. It is even possible to define instrument architecture for a program of surveys and drop survey specific blocks into that pre-defined organization (Manners, 1998), (Pierzchala and Manners, 1996). An additional advantage of pre-developing blocks and modules and using them across surveys is in the way this enables data to be compared across surveys (Manners ET al, 2001). This modular aspect of instrument specification and development is used to great effect in NASS where every quarter 44 instrument versions of the Quarterly Agricultural Survey are generated (Pierzchala, 1992), and where one June Area Frame instrument is driven 44 different ways by an on-line specification (Pierzchala, 1995 and 1996).

Methods of handling variations on a theme have been developed in NASS (Pierzchala, 1992 and 1996) and in the U.S. Bureau of Labor Statistic's Consumer Expenditure Survey (Stanford, Altvater, and Ziesing, 2001). There, many so-called 'laundry-list' tables share common table-level specification and source code files. The row-level specifications and source code differ but are driven to a great extent by an on-line specification database, while much of the rest of the row-level sources follow agreed-upon model code.

ONS also makes extensive use of standard high-level block definitions such as table-level constructs, and also of model code. This allows the researcher to focus on the subject matter and block-level details without having to worry about how these are all tied together at the instrument level (Manners, 1998). For example, the basic structure of the Expenditure and Food Survey (the interview element of which has a mean length of 90 minutes) is a set of Blaise tables in which each member of the household has a constant position (fixed by the first respondent's original choice of order in listing them). Household members can actually be interviewed within this structure in any order (and in differing orders in differing blocks, though this would be an unusual choice) and on any number of different occasions, to suit availability. Such flexibility is not unusual, but the important point here is that it is achieved with minimal effort. Blaise handles all the overhead of keeping track of who is being interviewed when - no additional programming is required. The effectiveness of this structure has been proved on the UK's Family Expenditure Survey (the forerunner to the Expenditure and Food Survey) since 1994.

Development based on researcher skills

Since it is possible in Blaise to divorce block-level development from higher-level instrument integration, it is also possible to have researchers program the blocks themselves. If you can do SPSS re-coding tasks you can do basic Blaise (Manners, 1998). This includes field definition, flow, edits, and computations. The existence of standard organisation-level blocks and programming and display standards relieve the researcher from having to worry about anything except for the subject matter at hand. This has been put to use at ONS (Manners, 1998) and NASS where subject-matter specialists do the subject matter related instrumentation directly.

Revolutionary aspect: This style of instrument development stands in contrast to computer programmers working from specifications. If properly implemented, and in the right circumstances, this can save time and effort and reduce communication problems. ONS, in particular, argues that it improves quality because the researchers are able to design directly in the instrument medium, as they used to do when the medium was paper. Among the important lessons that researchers learn by this method is how differently they should think about designing CAI instruments from the ways they thought about paper instruments. This method of instrument development is best implemented within a program of related surveys, for example agricultural production surveys in NASS (Schou 1995). Specialty applications, difficult and new requirements, and instrument-level block integration are best saved for researchers who are highly experienced in Blaise, or for true computer programmers.

All metadata defined once

It is possible, methodologically desirable, and most efficient when as much metadata as possible is defined in the Blaise instrument. This includes metadata used for data collection, post-collection review and data editing, exporting of data and metadata, and integration of Blaise instruments into an overall infrastructure. Blaise has several features that make it possible to handle metadata depending on mode of processing. These include changes in display, changes in instrument behaviour, and the ability to include mode in IF conditions within the instrument. Additionally Blaise allows you to define much descriptive metadata including metadata about export that might fundamentally recast the structure of the data into an analytical form quite different from the structure of the data needed for data collection (Pierzchala and Farrant, 2000). The additional Blaise capability of metadata manipulation allows you to generate customized data export routines and downstream data definitions from the metadata in the instrument.

Thus it is possible to accomplish multiple modes of data collection (e.g., CATI, CAPI) *and* data editing with the same system, *and* export data including metadata in any way you need them, all from one metadata specification.

Revolutionary aspect: The ability to accomplish data collection and data review and editing in the same system eliminates the need to reprogram the same metadata in a different system for post-collection processing for the purpose of cleaning data (however, there may be other reasons to include some of the same metadata in a downstream system, but that can usually be exported too). NASS uses the same instruments for both data collection and data editing in the Quarterly Agricultural Survey with the additional complication that about half of the cases for data editing are collected on paper questionnaires by field interviewers while the other half are collected in CATI. This is true despite the fact that there are fundamental differences between CAI and paper data collection in NASS (Pierzchala, 1996). The statement of both data collection metadata and data editing metadata in one system also promotes a more systematic consideration of what the differences between the two modes should mean. If data collection and data editing are separated into two systems, it is often the case that unintended differences in handling of the data between the two processes often creeps in and may go unnoticed. The Early Childhood Longitudinal Survey-Kindergarten cohort (NCES/Westat, Dulaney and Allan, 2001) and the UK's Family Expenditure Survey (Manners, 1993) offers another example of the utility of using one system for both tasks, despite some fundamental differences.

No data without metadata

When you define the blocks and fields of a Blaise datamodel, you define the data layout at the same time. This means that the data and metadata of an instrument are very tightly bound, and in fact cannot be separated in development. This relieves the developer from having to worry too much about the details of database organisation. Consequently the developer can concentrate on the subject matter of the study. The developer or instrument designer do have to be cognizant of data organization issues and export considerations (Pierzchala and Farrant, 2000). The tight binding between data and metadata also allows robust iterative development of an instrument. As long as old metadata and data files are both are archived, it is very easy to update the database of an instrument to reflect a new data definition.

Revolutionary aspects: Instrument development and database layout are taken care of in the same development step. You don't have to worry about synchronising two disparate steps of the process. Thus instrument development can proceed more rapidly. Additionally, all necessary information about data description and other metadata are in one place. A particularly dramatic example of the benefit of this paradigm is in the NASS Quarterly Agricultural Survey where 44 thematically related but considerably different instruments are generated in batch, every quarter. The compilation of each instrument results automatically in the creation of a unique database definition. The population of the database survey management data and other data is also done automatically in batch based on generated links between each database and the agency data infrastructure.

Windows openness

Windows openness is a new Blaise capability that works through COM and Active X components. This allows you to integrate Blaise with other systems. The Blaise API (Application Programmer's Interface) allows access to data and metadata from another system such as Microsoft Visual Basic. You can also build external file lookups using a standard relational database system as the external file. And you can call the Blaise data entry system as a DLL from another system.

Revolutionary aspects: The Blaise openness gives you the ability to add functionality within the context of an established CAI system, making use of the years of experience and testing which are built into it. Blaise will never be able to give you everything you need for all surveys. Windows openness allows you to develop special capabilities that Blaise does not yet have or may never have.

We see the first production applications of this capability in survey management tools. For example, a survey management shell could be written in Visual Basic with survey management data held in an Access database, but with direct access from the survey management shell to the Blaise data and metadata. A second kind of application will be in the more flexible extraction of data and metadata. Instead of exporting data en-masse, it will be possible to select fields and set up a custom export from a VB selection application, complete with qualifying metadata. Another application could be the development of special data entry programs, for example for special data collection needs not supported by traditional Blaise DEP.

But it is now also possible for third party developers to create value-added tools that work on the periphery of the core Blaise system. These might be commercial endeavors, the development of specialised or proprietary capabilities, or more informal trading of specially developed tools.

References

Dulaney, R. and Allan B. (2001). A Blaise Editing System at Westat. *Proceedings of the 7th International Blaise Users Conference 2001,* Washington, D.C.: Westat.

Elliot, D. (2001). Changes to the design of the Labour Force Survey. *Survey Methodology Bulletin no.47 July 2000.* Note: from no.48 (see reference to Gatenby below) the bulletin changed its name slightly with the addition of "Social".

Frey, R. (2000). Developing a Blaise Instrument for the Spanish Bladder Cancer Survey. *Proceedings of the 6th International Blaise Users Conference 2000*, Kinsale, Ireland. Central Statistical Office, Cork.

Gatenby, R. (2001) Overview of the development of the Expenditure and Food Survey (EFS). *Social Survey Methodology Bulletin, no.48, January 2001*, ONS, London

Groves, R., Berry, M., and Mathiowetz, N. (1980). Some Impacts of Computer Assisted Telephone Interviewing on Survey Methods. Pp. 519-524 in *Proceedings of the Section on Survey Research Methods*. Alexandria, VA: American Statistical Association.

Manners, T., Murgatroyd, L., and Flatley, J. (2001). Harmonised Concepts and Questions for Government Social Surveys, ONS, revised ed., *www.statistics.gov/harmony/default.asp*.

Manners, T. (1998). Using Blaise in a Survey Organisation Where the Researchers Write the Blaise Datamodels. *Essays on Blaise (1998),* 5th International Blaise Users Meeting. Lillehammer: Statistics Norway.

Manners, T., Cheesbrough, S., and Diamond, A. (1993.) Integrated field and office editing in Blaise. *Essays on Blaise 1993*, 2nd International Blaise Users Meeting, OPCS (now ONS), London 1993.

Manners, T. (1992). New developments in computer assisted survey methodology for the British Labour Force Survey and other OPCS surveys. *Proceedings, Annual Research Conference,* U.S. Bureau of the Census, 1992.

Pierzchala, M. and Farrant, G. (2000). Helping non-Blaise Programmers to Specify a Blaise Instrument. *Proceedings of the 6th International Blaise Users Conference 2000*, Kinsale, Ireland. Central Statistical Office, Cork.

Pierzchala, M., and Manners, T. (1998). Strategies for Producing CAI instruments for a Program of Related Surveys. In *Computer-assisted Survey Information Collection Methods.* New York, NY: John Wiley and Sons. Presented at the InterCASIC Conference (1996), San Antonio, TX.

Pierzchala, M. (1997). Optimal Screen Design in Blaise. *Essays on Blaise 1997,* 4th International Blaise Users Meeting. Paris: INSEE.

Pierzchala, M. (1996). CAI and Interactive Editing in One System for a Survey in a Multimode Environment. *Proceedings of the Data Editing Workshop and Exposition*. Statistical Policy Working Paper 25, Federal Committee on Statistical Methodology, Statistical Policy Office, Office of Information and Regulatory Affairs, Office of Management and Budget, Washington, DC.

Pierzchala, M. (1995). The 1995 June Area Frame Experience. *Essays on Blaise 1995*, 3rd International Blaise Users Conference. Helsinki: Statistics Finland.

Pierzchala, M. (1992). Generating Multiple Versions of Questionnaires. *Essays on Blaise 1992*, 1st International Blaise Users Meeting. Netherlands: Central Bureau of Statistics.

Schou, R. (1995). Developing a Multi-Mode Survey Processing System. *Essays on Blaise 1995,* 3rd International Blaise Users Meeting. Helsinki: Statistics Finland.

Singleton, N. (2001) Psychiatric Morbidity among Adults 2000 - First Release at *www.statistics.gov.uk/pdfdir/psymorb0701.pdf.*

Stanford, V., Altvater, D., and Ziesing, C. (2001). Programming Techniques for Complex Surveys in Blaise. *Proceedings of the 7th International Blaise Users Conference 2001,* Washington, D.C.: Westat.