Optimal Screen Design in Blaise

Mark Pierzchala, Westat, Inc., USA[21]

1. Abstract

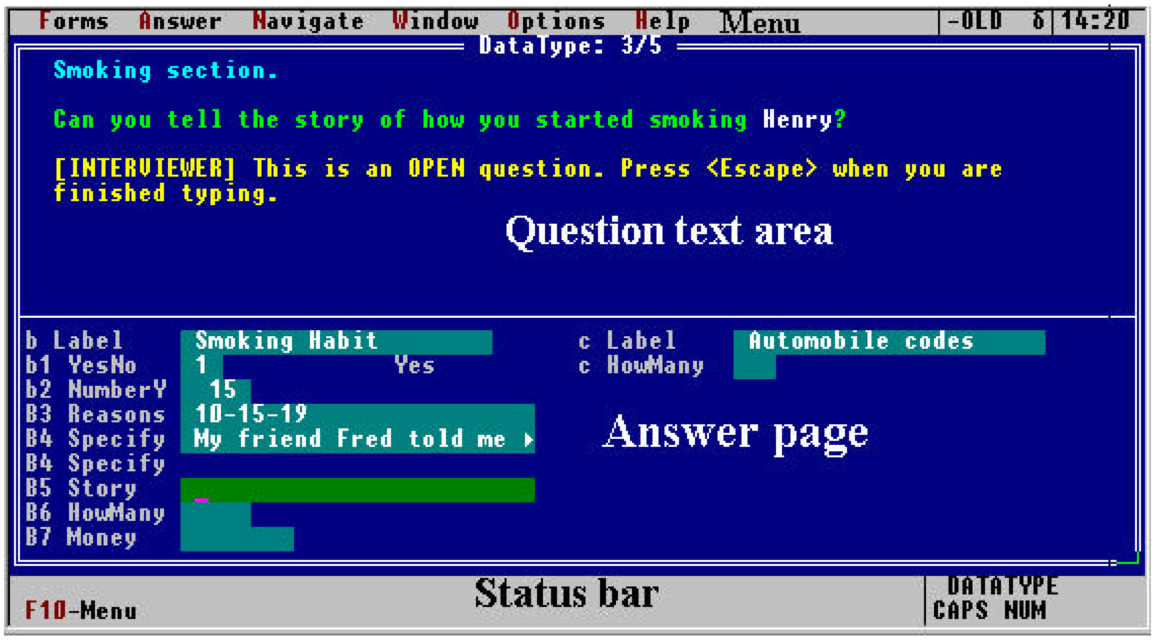
This paper discusses the enhancements that can be made to the default screen presentation of the Blaise system to make it easier for interviewers to do their jobs. Especially important are the conventions that the instrument developer can adopt to ease interviewing navigation. Suggestions that Blaise developers can employ for optimal screen presentation are offered. Examples from a datamodel called *DataType* are used to illustrate the enhancements.

Ce document discute les améliorations qu'on peut faire à la présentation de l'écran d'interviewer dans le system Blaise, especialement pour que la navigation se déroule meilleur. Quelques suggestions les programmeurs peuvent employer pour réaliser une présentation optimale de lécran sont données. Les examples d'un data model qui s'appelle *DataType* sont utilisées pour illustrer ces améliorations.

2. The split-screen interviewing presentation of Blaise

The default presentation of the Blaise system features four functional areas from top to bottom; a top menu bar, an area for question text on the top half of the screen (called an *Info Pane*), an answer page where data are entered and stored, and a status bar on the bottom as shown below.

[21] Acknowledgements: Much of this paper is based primarily on experience gained when the author worked for the United States National Agricultural Statistics Service. Several individuals there had key roles in defining presentation standards. These individuals include Asa Manning, Roger Schou, David Knopf, Roberta Pense, Bruce Eklund, Ann Ponti, Mark Schleusener, and others. Mary Laidlaw of Westat, Inc. helped with the tables and the demonstration data model.



```
-OLD  6|14:20
DataType: 3/5
Smoking section.

Can you tell the story of how you started smoking Henry?

[INTERVIEWER] This is an OPEN question. Press <Escape> when you are
finished typing.

b Label      Smoking Habit                 c Label      Automobile codes
b1 YesNo     1                Yes          c HowMany
b2 NumberY    15
B3 Reasons   10-15-19
B4 Specify   My friend Fred told me ▸
B4 Specify
B5 Story
B6 HowMany
B7 Money

F10-Menu                                   DATATYPE
                                           CAPS  NUM
```

The two most important parts of the screen for the interviewer are the question text area and the answer page. The menu bar and the status bar are auxiliary areas of minor importance for the purpose of communicating information to the user. The split-screen approach to the display of questions and answers has long been a trademark of the Blaise system, and has been very successful and popular with interviewers. The default presentation without developer enhancements is usually well received, nevertheless, the presentation can be improved in several important ways.

The split-screen presentation of Blaise is an example of a page-based (or a forms-based) system. This is in contrast to the presentation style of some other systems which are question-based systems. A question-based system displays one question at a time to the interviewer. Experience in switching from a question-based system to Blaise teaches that the interviewers greatly prefer the Blaise page-based interface to the question-based approach. There are several reasons for this, of which a few are mentioned.

First, in the page-based system, the interviewer can see to some extent what lies ahead. A typical answer page in Blaise can display from 10 to 20 questions depending on allocation of space. This is often enough space to display a whole section, or at least major parts of one at one time. The way the routing is handled in the Blaise system, where answer cells for questions on the route are displayed and where answer cells for questions not on the route are not displayed, gives the interviewer a good understanding of the flow. Second, the interviewer can see answers to several previous responses, can verify that data have been entered correctly, and is also in a better position to catch suspicious answers that may not be caught by programmed edits. Third, the navigation in and between answer pages is very intuitive. If the interviewer is in the right column and wants to go to a question in the left column, then she or he uses the left arrow key to get there, bypassing intervening questions. As there are from 10 to 20 questions in an answer page, it is very common that a whole interview's responses, even for a long interview, are displayed in a manageable number of pages. The page organisation enables the interviewer to review great parts of the instrument by pressing the page keys.

These and other reasons make the Blaise page-based interface popular and useful for the interviewer. It is possible in Blaise to implement a question-based approach, and a few organisations have done this. However, it is the opinion of this paper that this is a retrograde step that complicates navigation (see below). Another penalty in implementing such an interface in the Blaise system is that one of the instrument files, the **.~DM* file, may be greatly increased in size than a corresponding one for a page-based format. This can use a lot of memory and can slow down the instrument performance because there is overhead in the Blaise system when changing from one answer page to another.

3. The answer page as a question pick-list

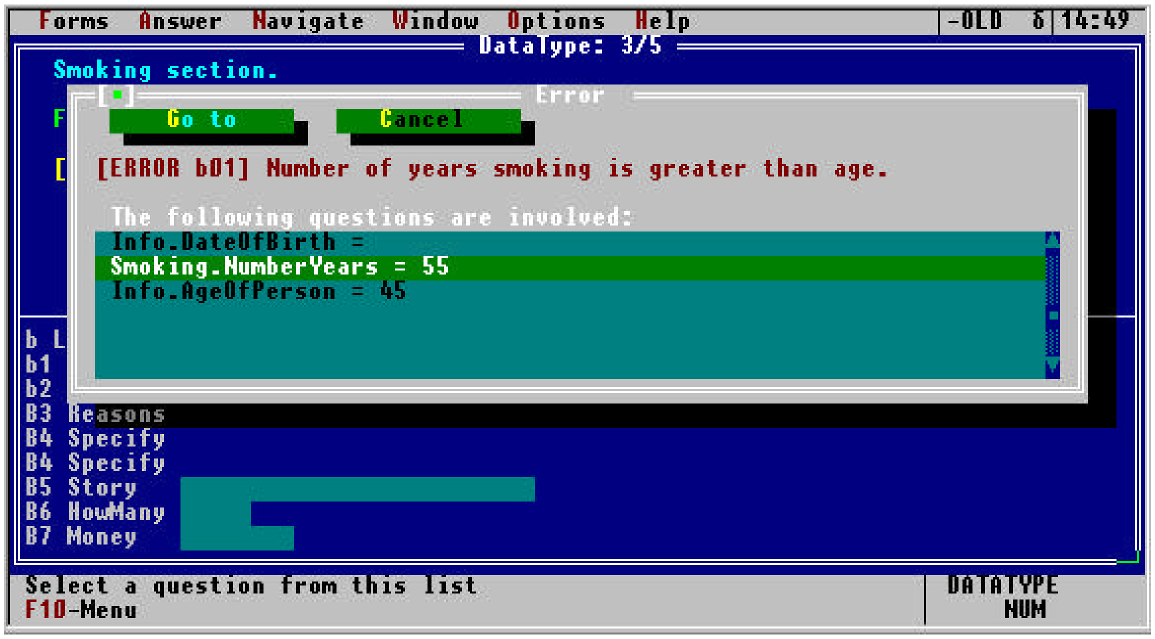
A well designed answer page may be compared in many respects to a question pick-list in other systems, only better. In some other systems, a question pick-list can be invoked by the interviewer and appears in a pop-up window. There the interviewer can pick a question to jump to and get there by invoking a jump mechanism. The usefulness of this kind of question pick-list can be limited when the instrument is very large. For example, it can be very difficult to pick out the appropriate question if the question list is long.

The answer page in Blaise gives a different answer to this navigational need. Each answer page can be viewed as part of an overall question pick-list, only in Blaise, it is well travelled territory. After the interviewer has gone through training and a few interviews (even practice ones), it is easy to back up and find questions. The developer can take a few steps here to ease backing up for the interviewer. The first is to incorporate labels in the page, the second is to display a section heading in the question text area. Both methods are discussed below.

Novice reviewers of the Blaise system often mistake the jumping capability of the Blaise system as the equivalent to the question pick-list in other systems. The jumping capability in Blaise has its place and can be very valuable, but it is of secondary usefulness compared to other methods of navigation.

4. Edit-specific pick-list

One of the nicest navigation features of the Blaise system is the edit-specific pick-list when errors are encountered in the interview. This is presented in an edit window that is automatically generated by the Blaise system. The pick-list of questions is appropriate to each edit in the instrument. In order to fix an error, the interviewer selects a question from the pick-list, presses <Enter> and jumps to the question to be fixed. There are several things the developer can do to make the edit-specific pick-list more useful. The most important is to use readable question names to identify fields in Blaise. For example, *AgeOfPerson* instead of *PersnAge*. There are other useful enhancements that can be made in the edit window that are shown below.



```
DataType: 3/5
Smoking section.
[ERROR b01] Number of years smoking is greater than age.

The following questions are involved:
Info.DateOfBirth =
Smoking.NumberYears = 55
Info.AgeOfPerson = 45

b L
b1
b2
B3 Reasons
B4 Specify
B4 Specify
B5 Story
B6 HowMany
B7 Money
Select a question from this list                    DATATYPE
                                                    NUM
```

In the edit window above, the edit message begins with a heading in square brackets which inform the interviewer that this is a hard error, the edit number is *b01*, and a succinct sentence gets the message across to the respondent and the interviewer quickly. It is also possible to include verbatim text and calculated numbers, for example rate of speed, which helps users to understand the situation better.

5. Ten methods of interviewer navigation

There are ten methods of interviewer navigation not counting the use of the mouse or pointing devices. They are listed below. The function key assignments have been changed from the Blaise defaults using the *Depmenu* files.

Navigation Facility	**Function**
Recording answers in an interview	Normal forward movement, taking into account flow strictures. Blaise takes care of this automatically.
Arrow keys (Up, Down, Left, and Right)	Used for short range navigation within a page or neighbouring pages. Where a page has two or more

	columns, use the left and right arrows to skip past several questions at once (for example, moving from a question in the right column to a question in the left column with the left arrow).
<Page Up> and **<Page Down>**	Used for medium-range navigation through sections or pages of an instrument. This is a **major** way to navigate. **<Page Up>** moves the interviewer back one page. **<Page Down>** moves forward one page.
<Home>	Used to jump to the first question in the instrument.
<End>	**In interview mode,** used to jump to the next appropriate question, taking into account changes in route due to changes in answers. **In editing mode,** used to jump to the last question in the instrument.
Edit-specific pick-list	This list is automatically invoked whenever a response causes an edit to be invoked. The list is in a pop-up window along with an error message and a list of question names. If it is necessary to fix an answer to a question, the interviewer can pick any of the questions from this edit-specific pick-list.
<Ctrl Enter>, Parallel blocks	Used to break the linear progression of the interview for tasks such as making appointments, recording non-response, concurrent household interviewing, or to complete sections by different respondents. After pressing **<Ctrl Enter>,** the interviewer selects a parallel block, to make an appointment for example, from a list of parallel blocks.
<F8>, jump box	Used to jump to any field with a field tag. It is **not** a primary way of navigation to individual questions since interviewers cannot be expected to know all question jump tags. It is **very effective** for jumping to sections with section labels.
<Shift F9>, Remarks lister	Used to review all remarks made in the instrument. You can jump to the field associated with any remark.
<Ctrl F9>, Open field lister.	Used to review all open fields the instrument. You can jump to any open field.

Some of the navigational methods are used in combination. For example, it is common to use the <Home> key to get to the beginning of the instrument and then to page down (forward) to get to an early section of the instrument, and then to use the arrow keys to get to a specific question.

It is possible to use the mouse or other pointing device in Blaise. This can be a great boon to the interviewer provided that he or she is specifically trained in the use of the device. If this training is not provided, it is best to disable the mouse driver for the interviewer.

The usefulness of the methods of navigation depends on the conventions adopted in an organisation. Two areas of concern include the choice between a page-based approach and question-based approach, and in the kinds of enhancements the developers can use to ease navigation.

6. Negative effects of reverting to a question-based presentation

The following table shows how some of the navigation methods are enhanced by the page-based display of the Blaise system.

Method	Page-based effect
Arrow keys (Up, Down, Left, and Right)	Arrowing in an answer page, or in adjacent pages, enables the interviewer to see where he or she is heading, and the arrow keys move the cursor in the direction intended. In a question-based display, the left and right arrow keys are useless, the interviewers cannot see where they are heading.
<Page Up> and **<Page Down>**	The concentration of answer cells in several or many answer pages, reduces the number of screens required to display data. Thus for medium range navigation, it is possible to review many answer cells with a few key strokes. The use of a question-based interface reduces <Page Up> and <Page Down> to the equivalent of <Up Arrow> and <Down Arrow> respectively.
<Home> and **<F8>**, jump box	Since these keys are often used in combination with the page and arrow keys, their utility is also diminished when the question-based approach is used. For example, if the interviewer jumps to a section, he or she would still have to navigate one question at time to arrive at the targeted question.
Answer page as a question pick-list	The page-based presentation enables this, the question-based approach destroys this.

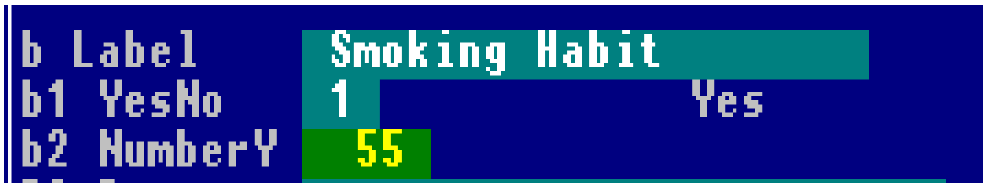
The use of arrow keys, the page keys, the edit-specific pick-list, the parallel blocks, and the jump box, can all be directly facilitated by developer-provided enhancements.

7. Answer page enhancements and standards

Given that a paged-based presentation style is adopted, there are several conventions that a developer can adopt to make the answer page more understandable to the interviewer. Page standards developed over the years are shown in the table below.

Type of Text	Example in answer page	Enhance-ment	Purpose
Section heading	Label	Top of a column of questions in a page, or at the start of a new page.	Aids in providing context to the interviewer and in short- and medium range navigation with the page concept.
Question name	Name, DateOfBirth	To the left of the answer cell	Gives an understandable name to the question that interviewers can use in navigation with arrows or in fixing answers due to edit failure.
Question number (tag)	a, a1, b, b2	To the left of the question name	Gives an alternate name to the question that interviewers can use with the jump facility.
Start new column	Start of a new section	Top of column Done with NEWCOLUMN	This is a way to keep related questions together, and makes the page more recognisable for page-based navigation.
Start new page		Done with NEWPAGE	Usually NEWCOLUMN is better, but there are times when you want a new section to start on a new answer page.
Blank space in columns or holes in table		Done with DUMMY	Not as important as formerly for the columnar format since NEWCOLUMN eliminates the need to use multiple DUMMYs to start a section at the top of a column. However, they are still important for tables with uneven rows, especially in economic collections.

An example of the implementation of these page standards is shown in the figure below. Especially important is the use of labels in the page. These are AUXFIELDS where a brief string is computed. These labels give context and texture to the page, and are important for paging. They enable the interviewer at a glance to know where they are in the instrument. An example of a label in a page is given below.



```
b  Label      Smoking Habit
b1 YesNo      1                Yes
b2 NumberY     55
```

The text string **Smoking Habit** is computed in the RULES. In this instrument the interviewer can land on the label. This is done for two reasons; to give an introduction to the section, and to provide a jumping place for the section. The latter use of the label is made possible by the tag that is associated with the label, in this case *b*. If the interviewer wishes to move to section b, he or she presses the <F8> key, enters *b*, and presses <Enter> to jump. The code for the label is given below.

```
AUXFIELDS

  Label (b) "@/@Y[INTERVIEWER] You are now entering

             the smoking section.

             @/@/Press <Enter> to continue."

             : STRING[20], EMPTY

  .  .  .

RULES {Many lines later.}

  Label := 'Smoking Habit'

  Label

  {and more code following}
```

The label is very useful as well for editing mode where the question text is not displayed. It helps orient the data editor very quickly to the screen.

A readable question name is almost always a better identifier for a question than a question number. This is true for the developer, who writes readable code. It is also true for the interviewer who reads the readable question name in the edit-specific pick-list of questions in an edit window, or who is browsing the answer pages with the page keys. It is not necessary to limit the length of the question name to 8 characters in order to conform to limitations of downstream systems. You can either let Cameleon truncate the question name to the first 8 characters in order to generate, say a SAS data step, or you can modify a Cameleon setup in order to read a SAS Var Name from another meta data location. For example, it is possible to use the first 8 characters of the description space to state a SAS Var Name. If a question number is necessary, it is better to record it as a question tag.

8. Text enhancement standards

Text enhancement standards are used in the question part of the screen, the help text in a pop-up window, and in pop-up edit windows. These parts of the interface are used to give a lot of different kinds of information such as the question, possible answers, the name of the section, the name of the respondent, and special instructions. While the question to be posed is most important, it may take up a relatively small percentage of the amount of text that appears in these windows. The enhancements are used to distinguish between types of text. Some standard text enhancements are given in the table below.

Type of Text	**Position of Text**	**Enhance-ment**	**Purpose**
Section name	Top line of the question text area	Bright blue text on dark blue background **@B**	Orients interviewer to the location within the instrument. It often corresponds to the label in the answer page.
Question text	Below the section name in the question text area.	Bright green text on dark blue background **@G**	Indicates the information to be read aloud to the respondent.
Interviewer instructions	Below the question text.	Yellow text on dark blue background **@Y**	Indicates information to be read by the interviewer only, (not to the respondent).
Enumerated responses that are part of the question text	Below all other text	Green text on dark blue background **@G**	Indicates that for this question the response categories are to be read as part of the question text. If the enumerated responses are not to be read to the respondent they are left grey.
Fills in question text	As appropriate in question text	White text on dark blue background **@W**	Indicates text that is inserted into the question text based on a response to a previous question.

Edit label and edit number	In the edit window	[ERROR 101] [WARNING 10] Red text on grey background **@R**	Distinguishes between hard and soft edits. The edit number is used to communicate problems back to the developer.
Edit banner	After the edit label and number in the edit window	Red text on grey background **@R**	Gives a short summary of the problem to the interviewer and respondent. Often, this is all that is necessary. Example: "Total does not equal sum of parts."
Edit message	After the edit banner in the edit window	Green text on grey background **@H**	Verbatim text to be read to the respondent if necessary to fix the problem. Example: "The total you just gave does not equal the sum of the items you just gave. Let us review and fix this."
Fills in edit message	As appropriate in the edit message text	White text on grey background **@X**	Indicates text that is inserted into the edit message based on a response to a previous question, or on a calculation done by the system.

The use of a particular text enhancement indicator such as *@G* does not commit the organisation to any particular colour scheme. The indicator *@G* may mean green for one organisation and yellow for another as Blaise allows this definition to be changed dynamically.

9. Function key assignments

Blaise allows you to map many interviewer and editor tasks to function keys F1 through F10 (F11 and F12 are not recognised by Blaise). The strategy employed is to map the most common tasks to one-stroke function keys. Common function key assignments may be:

Function	Hot key	Description
Help, question-by-question	**F1**	**Display help text.** Developers can create question-by-question help for an instrument.
Calculator	**F2**	**Display the calculator.** To copy a calculation into the answer cell use the following sequence. <Ctrl Ins>, <Esc> (gets rid of the calculator), <Shift Ins>.
Next language	**F3**	**Switch to the next language** in the set of languages available in the instrument.
Previous language	**F4**	**Switch back to the previous language** in the set of languages available in the instrument.
Don't Know	**F5**	**Record a** *Don't Know* **response** for the current question.
Refusal	**F6**	**Record a** *Refusal* **response.**
Save and Continue	**F7**	Save data to the hard drive and continue with the interview.
Jump	**F8**	**Jump to a specified question or section tag.** If the interviewer knows the question or section tag, he or she can jump to it if it is on the route. This is most usefully used to jump to a section tag that is attached to a label in the answer page.
Make, inspect, or modify a remark	**F9**	**Open remark window at any question.** Type a new remark, modify an existing remark, or inspect the remark in the window. Press **<Esc>** when the remark is complete.
Menu	**F10**	**Move cursor to the menu bar.** Developers can disable the menu bar for interviewers.
Navigate through remarks	**Shift F9**	**Review remarks.** Blaise displays the remark windows, starting with the first remark. Right arrow displays further remarks, one at a time. Jump to the remark by pressing <Enter>.
Navigate through open questions	**Ctrl F9**	**Review responses for open questions.** Similar to navigating remarks above.

10. Minor variations on the page-based presentation

The page-based presentation that Blaise gives works very well for all question types except one. The exception is for enumerated types of many responses. In this situation, the space in the question text area can be too limited to pose both the question and to list all the answer possibilities. This limitation is due to the way that the Blaise screen uses space. In a standard 25 line DOS window, fully six lines are used by the Blaise system itself, far more than in Blaise 2.5. This leaves only 19 lines to the developer. To make matters worse, only about 76 characters are allowed per line of text instead of 80 that was available in previous versions. This space limitation for the question text area has led several organisations to experiment with the ability in Blaise to modify the default presentation. The most common screen modification is to lower the dividing line between the question text area and the answer page. This is done with the *Modelib* configuration files. The exact mechanism for doing so is beyond the scope of this article. However, it is appropriate to make a few remarks here.

The modification of the *ModeLib* file is a challenging process. On one hand, it gives an organisation tremendous flexibility in screen design. (To see that this is true, read Chapter 6 of the Developer's Guide.) On the other hand, the parameters in the *Modelib.txt* file are not well documented. It can take a great deal of experimentation to arrive at the correct combination of values of parameters. It is best for an organisation to define several standard styles and to limit developers to those possibilities.

The design tool found in *DEMOS\DESIGN* under the Blaise system directory offers limited help. This tool, a combination of a Maniplus menu system and several Blaise data models, is intended to help the developer define different screen and behaviour configurations. However, some parts of the design tool are not well executed. There is undefined and inconsistent jargon that is used in the menuing system and the question text of the data models. The excellent help facilities of the Blaise system and of Maniplus are not used. Three of the data models for specification, *Grids*, *Field Panes*, and *Toggles* are somewhat helpful, in that they help you to understand the *Modelib.txt* ASCII file. The *Toggles* data model does a very good job of explaining the behaviour possibilities of the *Modelib* file.

Since the design tool is of limited value, and since often what is needed is a lot of iteration, it is better to use a text editor on the *Modelib.txt* file and modify the parameters directly. The trouble with this method is that the *Modelib.txt* file is difficult to understand on first (second, third, . . ., nth) glance. However, if you persevere you can get the results you want. One approach is to bring the *Modelib.txt* ASCII file into the control centre and edit it there. Also present in the control centre is a data model. You can make changes to the *Modelib.txt* file then use the menu options *Tools, Own program* to execute the command that transforms the *Modelib.txt* file to its binary equivalent. Then re-prepare the data model, invoke it, and see what changes have been wrought. By doing everything in the control centre, it is possible to get 10 to 20 iterations per hour. This is not the thing you want your developers spending a great deal of time on. Thus the organisation would benefit if it assigns this thankless task to an individual, define several options, and let developers choose from these.

To lower the dividing line between the question text area and the answer page, you must adjust two aspects of the default presentation. First increase the area of the so-called *InfoPane*. Then decrease the area of the *Grid*. If you do the former and not the latter, the answer page extends below the bottom line. The instrument will scroll vertically as it needs to in order to enable responses to be entered. However, when some answer cells are not visible to the interviewers when they navigate by paging, they can become very confused. The only situation where it might be appropriate to extend the answer page below the bottom line is in a repetitive rostering situation not done in a table. For example, say a block has ten questions in it and this block is repeated many times but for different topics. If NEWCOLUMN is used to start each instance of the block, and a label is used to define the topic in each block, then it probably does not matter if some answer cells are below the bottom line. The interviewer will quickly learn the pattern and know where to find things. In this situation, this is similar to the horizontal scrolling found in the table.

11. Changes in the default presentation

If there was one thing that should be changed in the default presentation, it is the elimination of some of the cosmetic display lines used in the Blaise screens. None of the framing white line needs to be there, either top or bottom, right or left. Too much space is taken by this framing line which has no functional value (it makes the screen look nicer until you have to overcome space limitations). The bottom two lines which make up the status bar are also not used. On occasion, some information is displayed there, but few if any users ever notice that information. It would be most helpful if the status bar were to disappear. Given this extra space, many organisations would not have to experiment with the *Modelib* file.

12. Summary

The default Blaise presentation is superb for almost every situation. Where it falls short for the display of enumerated responses, Blaise provides the possibility to alter the presentation style, once different styles are developed. The developer can take several simple steps to enhance the Blaise style to make the job of the interviewer and data editor easier.

---

[i] Bushnell (1995), Computer Assisted Occupation Coding, Essays on Blaise 1995, Third International Blaise Users' Conference, Statistics Finland

[ii] Dodd, T. (1985). An Assessment of the Efficiency of the Coding of Occupation and Industry by Interviewers. OPCS New Methodology Series, 14.

[iii] White, P. (1983). The Continuous Manpower Survey - Feasibility Study. OPCS Survey Methodology Bulletin, 15.

[iv] Martin, J., Bushnell, D. and Campanelli, P. (1996). A comparison of Interviewer and Office Coding of Occupations. Submitted to the Journal of Official Statistics

[v] Office of Population Censuses and Surveys (1990). Standard Occupational Classification, Volumes 1 and 2. London: HMSO

[vi] Elias, P., Halstead, K. and Prandy, K. Computer Assisted Standard Occupational Coding (1993). HMSO