INTERVIEWER BEHAVIOR ON CAPI GRIDS AND LISTS

D.E.B. Potter, AHCPR, Brad Edwards, Sandra Sperry, and Richard Dulaney, Westat, Inc.
D.E.B Potter, Agency for Health Care Policy and Research
2101 E. Jefferson Suite 500, Rockville, MD 20852

KEY WORDS: Questionnaire design, CASIC, cursor movement, interviewer navigation

BACKGROUND

Now that the feasibility of computer-assisted personal interviewing (CAPI) has been established (see Baker, 1992; Couper, 1994; Nicholls, 1997), CAPI designers can direct more of their attention to usability and the most important user of computer-assisted interviewing (CAI) systems -- the interviewer. The interviewer as a source of measurement error has long been a focus of survey methods research (see for example Groves, 1989). Understanding how interviewers use CAPI can help direct CAI designers (authors, programmers) and CAI software engineers to areas where changes in design could result in "measurable gains in data quality" (Couper, 1994).

This paper builds upon previous work by the authors (Sperry, et. al., in press) that examined interviewers' use of CAPI navigation features as a means for interviewer control. In this analysis we examine how CAPI interviewers use CAI navigational features to move around on screen, specifically on grids (matrices) and lists (e.g., look-ups). To provide a context for our results, we first describe the variety of grid and list screen functionality currently available in CAI.

THE DATA

The data source for this analysis is the 1996 Medical Expenditure Panel Survey (MEPS), Nursing Home Component (NHC) (Potter, in press). The MEPS NHC is sponsored by the Agency for Health Care Policy and Research, and the National Center for Health Statistics. It is a longitudinal establishment survey of nursing homes and nursing home residents. The data were collected by Westat, Inc., using a CAPI application designed by Westat in Cheshire. The primary data source is the keystroke files generated from the first and fifth completed interviews conducted during the first round of data collection, approximately 400 hours of CAPI administration time. These data were collected by 46 non-supervisory interviewers.

The CAPI keystroke files were transactional in nature and provide a record of every keystroke the interviewer actually performed in order to collect the data. For instance, if the interviewer entered a "no" to an item, and then backed up and changed the response to a "yes", the audit trail data shows the "no," the backup, and the "yes". For additional details see Potter (in press) and Sperry, et. al., (in press).

In designing the analysis file from the keystroke records, we took four issues and constraints into account: the magnitude of the effort to create a SAS file; the schedule for the research versus the MEPS NHC production schedule; the size of the raw keystroke files (potentially containing many millions of observations); and the desire to have at least two points of measurement on the interviewers' learning curve, one at the beginning of production and another after interviewers had acquired some experience. We selected the first and fifth NHC cases completed by each interviewer because we could not wait until the entire interviewing round was complete to construct the data set; even the first five cases for each interviewer would have resulted in a data set too large to handle easily. The first case represented about 4 hours of interviewing (roughly comparable in keystroke volume and application production time to the data set examined by Couper and others, and the fifth case represented a point for each interviewer about 16 interview administration hours after starting production, in our judgement enough time to examine most learning curve issues.

Due to the non random nature of these data, no statistical tests of significance were preformed and the generalizability of the data needs to be considered when interpreting these results. However, these data were collected by a national field staff, hired by a large data collection contractor to work on a complex CAPI study for the U.S. government and are thought to be comparable to similar interviewing situations.

GRIDS IN CAI SYSTEMS

We use the term *grid* to refer to CAI screens that are displayed to interviewers as matrices, where responses to multiple questions are entered on a single screen. These have also been referred to as form-based questions, tabular displays, table format, and (in the non-survey environment) spreadsheets. Because grid screens provide a mechanism to collect multiple responses on a single screen, the CAI segmentation effect (House and Nicholls, 1988) is thought to be lower than that found with other CAI screen types. (Other techniques that pack more into the screen -- such as multiple entry fields, question overlays, and context headers

-- have also been used effectively in combatting this effect, but they do not offer the interviewer the navigational control found on the grid and thus are not discussed here.) A frequent use of grid question screens in the survey world is the enumeration of all persons in the household, with each grid row representing a person and each column corresponding to attributes of the person (age, sex). So as not to be confused with forms-based systems (such as Blaise), we will use the term grid.

When the interviewer comes onto the screen the cursor's initial placement is typically in the first cell requiring input (e.g., cell A1). In CAI systems the cursor usually follows a default path through the grid, moving the interviewer from cell to cell in a predetermined way. After typing the response in A1, the interviewer typically uses the Enter key or arrow key to initiate movement to the next cell in the default path (cell B1 for example). Where the major CAI systems vary, sometimes by quite a bit, is in their: (1) presentation of the questions on screen, (2) path of the default flow through the grid, (3) method for interrupting the default flow, (4) scrolling capability, and (5) mechanism for exiting the grid.

Question Scripting. CAI applications can vary quite a bit in their in their level of question scripting that appears on screen with the grid. Some provide exact wording in direct correspondence to each cell on the grid (e.g., "What is John's age?"). As the interviewer moves from cell to cell, the question displayed on the screen changes. Other applications are unscripted, and merely provide labels for the grid columns and rows (e.g., "Age"). Yet others offer a hybrid approach -- for example, scripting the question but not the question word fills ("What is YOUR/PERSON'S age?"). The level of scripting is determined in part by the CAI system, but also by the survey designers responding to a myriad of design constraints, the most obvious being size of the grid and the proportion of the screen occupied by the grid.

Default Path. There are two widely used default paths through the grid: (1) left to right, top to bottom (A1, B1, C1, A2, B2, C2), and (2), down the first column, then left to right, top to bottom (A1, A2, B1, C1, B2, C2). Others are possible. The default path in a particular CAI system can be "hardwired," requiring all organizations using that system to follow the same default path. However, several of the CAI systems are designed so that the default path can be changed. Some survey organizations have adopted a default flow that never varies within their organization, while other organizations vary the default specific to each design (Potter, 1996).

Interrupt the Default Flow. At one extreme are systems that provide interviewers with a great deal of flexibility in grid movement. The default path can be interrupted at any time by moving the cursor (with arrow keys) to a cell that is different from the default. With a flexible system it is much easier for interviewers to collect the information in an order that departs from the default path (e.g., following the order that the respondent is using). Should a need arise to correct information in a previously entered cell, the interviewer has only to use the arrow keys to return to a previous cell. At the other extreme are systems where the default path can only be interrupted by initiating a jumpback.

Scrolling. Some systems have the capability of scrolling off screen while still roaming within a single grid. This can result in grids that are quite large (one application featured a grid that was 60 by 90 cells). This may result in a segmentation effect within one screen, but it does provide the interviewer with enormous flexibility in data collection sequence.

Grid Exit. The method by which interviewers exit from the grid is determined by the CAI system. In question- or screen-based CAI systems, this is typically accomplished with the Page Down key, the Down Arrow key, or the Escape key. When an interviewer uses the "exit key," the next screen in the forward flow of the application is displayed. Forms-based systems also use the Page Down and Arrow keys, but although the grid leaves the screen and a new question appears, the forms metadata (some of which may be associated with the grid questions) remain on screen.

Interviewer Behavior on Grids. The Round 1 NHC application has three grid screens that require the interviewer to enter data in at least two columns. All are part of the data collected about the establishment early in the interview. Each is used to enumerate various structural relationships of the nursing home (e.g., name and type of special care units and attributes about each of the units). Thus, depending upon the structure of the nursing home the grid question may or may not appear as part of the CAPI application (Exhibit 1).

The NHC application scripts all grid questions for the interviewer, using as word fills such things as the name of nursing unit. Each of the grid columns is labeled with the attributes of the unit. The rows correspond to the various units that are being enumerated. It is possible for a grid to have a single row or several, depending upon the complexity of the establishment. The default flow through each of these grids is left to right, top down. Interviewers enter data in a cell and hit the Enter key to record the information. This simultaneously moves the cursor to the next cell in the default path (this would include moving the cursor from the end of a row to the beginning of the next row automatically). Interviewers can change direction at any time with the arrow keys. Information can be corrected in a previously entered cell by moving the cursor to the desired cell; new data can be entered by overwriting old data, or the interviewer can use the backspace key to correct the information. Interviewers exit the grid (i.e., the grid screen) with the escape key.

Among the cases that we examined there were a total of 63 episodes of grid movement (Table 1). Considered a single grid episode were situations where the interviewer entered data on the grid, then moved off of the grid screen, only to return to the grid screen. These episodes were distributed roughly across the interviewers, with a few interviewers having no episodes.

The patterns of interviewer cursor movement (interviewer navigation) were quite varied across grids. Grid size, as measured by the mean number of cells in which data were entered, ranged from 11, for Grid 1, to 6 and 3, for Grids 2 and 3, respectively. The number of cells encountered (regardless of whether data were entered as the cursor passed through the cell) also varied by grid, with a mean of 23 cells for Grid 1, in comparison to 5 cells for Grid 3. What was found to be a bit more constant, but still a positive linear relationship to the size of the grid, was the ratio of cells *encountered* to the number of cells in which data were *entered*. On average, for every grid cell with data entered, the interviewers' cursor encountered approximately two cells.

High rates of interviewer navigation were also seen in the frequency of interruption to the grid default flow (an average of 1.8 interruptions, per grid). Almost eight percent of the Grid 1 keystrokes were interrupt directional keys, rather than data entry keys.

The ratio of keystrokes to the number of cells encountered, per grid, was not associated with the size of the grid (2.9 for Grid 1 and 3.8 for Grid 3). This a function of the grids' design (i.e., number of cells designed to collect text strings) rather than interviewer behavior.

We can only hypothesize as to why interviewers collected the data in the order that they did (e.g., collecting first the data in Column 1, then the data in the last row, and then completing the remaining rows, from top to bottom). A few of their movements were obvious keying errors (using the Up Arrow while already on the top row), but we are, with these data, unable to measure systematically the error.

LISTS IN CAI SYSTEMS

In CAPI applications, lists share some properties with grids. Indeed, a list can be seen as a matrix stripped of all its columns, save one. Thus, navigational movement within a list screen is limited to one dimension -- up and down -- and this limited range of movement allows us to study some navigation issues that may be masked by the more complex activity on a grid screen.

In CASIC, the concept of a "list" has been developed in ways that go well beyond any paper questionnaire capabilities. In many surveys, respondents are asked to report medical events, health conditions, sources of payment or income, employers, etc. This type of information lends itself naturally to the creation of lists during the interview. By viewing a list, the interviewer can see all previously reported instances of a certain type of information, or the universe of listings in a category. If the latter type of list is long, it is often called a directory; special search features can be supported in several of the CASIC systems to find elements in such directories. The list concept is also closely related to the concept of response categories -- a list of categories is a special form of a list, which aggregates individual elements into groups, or offers a classification scheme for responses.

Perhaps the most typical way to select from a list in current CASIC systems (DOS based) uses an entry field. A list of categories is displayed below the question, with a response code (a number) beside each category. The interviewer uses the keyboard to enter the number for the category in a field set aside for recording answer codes. In most respects, this example doesn't differ much from a categorical question in a paper questionnaire. As the list of categories grows longer, the "total question" (that is, the question text, the answer categories, and the entry area) may grow to become too large to fit on a single screen. One solution (common in CASES and Survey Craft applications) is to allow codes to overflow onto a second screen, accessed with the Down Arrow or Page Down key.

In CASIC, there is no requirement that lists be predetermined; CASIC lists can be "dynamic." A list can be designed to contain only those entities reported by the respondent in a particular interview. Each time the respondent reports a new instance of a listed item (for example, employer, prescribed medicine, or medical condition), the CAPI program can add it to the list. When the respondent reports an item that has been previously entered on the list, a link is created between the data variable and the list element. For example, if the respondent reports a visit to a medical provider who is already on the list, the interviewer chooses this provider from the list. The program then links this provider's name to the appropriate event, thus adding efficiency and accuracy to the recording of important information. The program may also allow the interviewer to call up the dynamic lists on demand. These dynamic lists have sometimes been called "rosters." However, in other CAPI systems, the term *roster* designates repeating elements in a rectangular file -- for example, all doctor visits for a person.

Two basic types of lists have different properties: (1) "select one" allows only one element to be selected; and (2), "code all that apply" allows the selection of multiple elements. The "code all that apply" list presents some special problems, in providing feedback to the interviewer on what has been selected and in jumping back through an item that may have been answered in different ways on several paths through the interview.

Separate attention has been devoted to lists that reside in external files and to search engines that access directory elements. (For example, see Vincent and Sanchez, 1993.) Search engines present interface consistency issues, as well as concerns about how the interviewer conceptualizes the search. The NHC application uses two external directories, a list of all U.S. hospitals and their addresses, and a similar list of nursing homes. We have exclude these from our discussion in this paper, since the interviewer leaves the CAPI application to search the files, and is then returned to the application. As such, there were no CAPI keystroke data collected about the interviewer's movement on these external lists. However, one of the lists that we *were* able to investigate was sufficiently long (600+ elements) to provide some insight into interviewer navigation on external directories, which are typically long in length.

Interviewer Behavior on Lists. The NHC application developed a version of the list that was intended to mimic the content and appearance of a document that is ubiquitous in nursing homes, the Minimum Data Set form. The MDS incorporates many lists of text items, which feature white boxes for recording check marks.

In early stages of development, the NHC choice list allowed only one selection per screen. At first it was thought the best way to indicate a selection had been made was to highlight it in reverse video. This was consistent with the backup convention for displaying previously entered data, which seemed to be accepted readily by interviewers. But having more than one selection highlighted in this way was found confusing and irritating in usability tests. Even a single choice was confusing enough, since in the default flow, when the interviewer pressed enter on a desired category, the cursor moved to the next category. Since the category the cursor rested upon was displayed in reverse video, and the selected category just above it was also in reverse video, it appeared as though the interviewer had made two selections. To make matters worse, since most of these choice lists really had to allow for multiple choices, we forced the interviewer to go through a loop for each selection. This proved extremely cumbersome in usability testing; it appeared as though an earlier choice had been erased.

The solution to these problems was found by combining the visual properties of the paper MDS (as it looked after it was filled in by nursing home staff) and the dynamic properties of a code-all-that-apply question in CAPI. The final NHC application supports abstracting from the MDS, by presenting a question that maps to the MDS, displaying the text items associated with the question, and allowing the interviewer to place a checkmark next to each item that is checked on the MDS. When the interviewer first comes to an item, the cursor is on the first item (indicated by highlighting the text for the entire item in reverse video). The interviewer moves the cursor down (and up) the list using the arrow keys, page up, page down, home, and end. When the cursor is highlighting the desired item, the interviewer presses the enter key. A checkmark is then displayed next to the selected item. The checkmark can be toggled on and off with the enter key. Most of the choice lists allow more than one item to be selected; each selection is indicated by a checkmark on the screen (Exhibit 2).

It should be noted that many items in the NHC application are categories or simple lists that do not feature this "choice list" design, but use the more traditional display of codes and categories, with an entry field to indicate the selection. Concerns might be raised about the lack of consistency in the user interface between these "entry field" items and the choice lists. We have not observed, however, no instances of interviewers having difficulty switching between these two approaches. Interviewers are aided by the visual distinction between the two approaches -- the choice list always appears inside a box, with the cursor highlighting an entire category in reverse video, whereas the entry field approach with its codes looks more like many traditional paper questionnaires.

In CAI, the entry field approach often has the advantage of fewer keystrokes, since there is no cursor movement. But the choice list concept is the more "natural design," to borrow a term from D.A. Norman (1988), since the cursor and the checkmark provide the interviewer with visual cues that are directly related to the choices made or to be made, while the code and entry field approach requires a translation. This suggests that the choice list technique might be associated with fewer errors. It also hints at some of the possibilities of object-oriented or graphical user interfaces (GUI) for CAPI applications.

For longer lists -- those with more than 10 or 15 elements -- an additional search feature, beyond the cursor keys, was developed within the Cheshire system to speed interviewer movement within the list. On these lists, the interviewer can type the first character of a text field, and the cursor moves to the first list element that starts with that character. The interviewer can specify the text more narrowly, by typing additional characters. We refer to this type of list as "search enabled." (Blaise has a similar capability.)

We chose to look at nine lists. These represent the majority of lists used in the application and provided a wide variation in list length, window size, and search features. The number of encounters examined, per list, was about the same for six of the lists (n=363, per list), or a maximum of 8 encounters, per interviewer, per list. For the remaining lists, the number of encounters examined ranged from 73 to 3,700, per list. Four of the lists were short (mean of 7 elements), with no search enabled feature; four were mid-size (mean of 25), with search enabled; and one was long (600+), with search enabled (Table 2). The window size ranged from only 4 to 9 display lines. The ratio of window size to list size varied from 1.00 to .01. All except the long list were "code all that apply"; the mean number of elements selected, per list, ranged from 1.0 to 3.9. The prescribed medicine list is the extreme outlier (n=3,707), with 604 elements and a window size of 9. In many respects, it functions like an external directory; indeed, the source data is an external file of drug names, their forms and strengths (RedBook File, 1995).

As one would expect, the interviewers used far fewer keystrokes on the short lists -- a mean of 4.6 (unweighted) per encounter, compared to 26.1 on the mid-size lists. On average, for these two list types, there was a positive linear relationship between list length and number of keystrokes.

One might expect more keystrokes on lists with lower window size ratios, just to view all options or to reach options that are not visible in the window. But in fact, after adjusting the rate of mean keystroke use for the number of elements selected on the list, this was not the case. (Henceforth, unless reported otherwise, all rates of keystroke use are reported as mean rates, per element selected, per list.) Among the short lists, the mean number of keystrokes was largest (4.0) for the list with the second largest window size ratio (.71). Among the two lists that had a similar window ratio (.67), the rate of keystroke use was considerably less, and almost half for one of lists. For each short list element selected, there were, on average, 1.06 uses of the Enter key. Very little of that was associated with Enter key errors (or error correction), which averaged .06 keystrokes. The Down Arrow was the most frequently used navigation key (1.5 uses), a rate more than six times that of the Up Arrow, the second most frequently used cursor key.

Encounters on the mid-size lists generated much more cursor activity; on average, 18.1 keystrokes per element selected, in contrast to 2.8 keystrokes on the short lists. As expected, the Down Arrow was used the most frequent (10.6 uses). But, in contrast to the short lists, the Up Arrow was used considerable more often; for every Up Arrow use there were 3 Down Arrow uses, in comparison to 1:7, on the short lists, suggesting repeatedly scrolling up and down the lists, as you would an old Rolodex.

What was surprising, was the high use of the Enter key on the mid-size lists (3.3 uses, per *single element* selected). A substantial portion of this activity (2.3 uses) was associated with selecting elements in error and then correcting those selections. Even more surprising, the two lists with the highest Enter key error rates -- 3.97 and 5.02 keystrokes, in contrast to the other lists which averaged .26 or below -- were not the longest lists, nor the lists with the smallest window size ratio, but rather on the lists that were encountered the least (73 and 193 encounters, in comparison to 360+ for the other lists).

These rarely encountered screens presented lists of medical conditions that were *not* printed on the standardized forms used in nursing homes. Thus, the lists were essentially an on-line coding activity. Some of the interviewers who found conditions reported with minor language variations may have had difficulty, searching up and down the lists, sometimes making errors; unsure of their selections.

In comparison to the mid-size lists, the very long list had a *lower level* of navigational keystroke activity (8 compared to 18). We believe that this suggests that the search feature was used more frequently on the long lists. However, the rate of Page Down key use (approximately once for every two elements selected) suggests that the search function was not used universally.

This list was encountered more often (3,707) than all eight other lists combined, for two reasons. First, it is the only list we examined that is not "code all that apply" (and therefore interviewers encountered it each time they discovered a "new" medicine recorded in the patient's chart). And second, it reflects the extremely high use of prescribed medicines in the nursing home population -- 20 or 30 drugs are not uncommon for many residents. It also stands out from the other lists in other respects. It is very long (600+), more than 10 times longer than the next longest list. We also see more frequent use of the Page Up and End keys on the prescribed medicine list than for all other lists, reflecting the large size of the list.

It is interesting to compare keystroke frequency between the first and fifth interviews on these lists. On the four short lists, mean keystrokes were slightly lower in the fifth interview than in the first. But on four of the five mid/long lists, mean keystrokes *increased*, even to the point of doubling for two questions. But for the prescribed medicine list, the mean keystroke measure dropped about 20 percent. We believe this indicates the interviewers' increased level of familiarity with the list, and building proficiency with the search enabled feature on an extremely long list.

One issue that has received a fair amount of attention in both the HCI and the CASIC literature is consistency. All else being equal, it is considered better to present a consistent interface to the user. The user builds expectations about the system based on first encounters; a requirement to accomplish something one way in one encounter and a different way in the next can frustrate the user and lead to entry errors or other problems (Shneiderman 1991; Norman 1988; Couper 1994; Saris 1991). Does the inconsistency in the treatment of lists in NHC (with regard to the search enabled feature and the window size) present problems for interviewers? We did not see evidence of problems in the feedback from interviewers, but the pattern of entries on the four mid-size lists suggests that at least some interviewers applied the same cursor movement techniques that they had used on previous short lists, and this led to highly inefficient keystroking.

It should also be noted that the list screens displayed no instruction to indicate whether the search function was enabled. It may be that the interviewers were conditioned by their experience with screen-specific help to expect a message on the screens for which the search function was available. The lack of such a message meant these screens lacked the explicitness that is one of Couper's (1994) ten design goals for the CAPI interface.

CONCLUSION

This case study explored interviewers' navigational activities on CAPI grid and list screens. The screens examined provided interviewers' considerable flexibility with which to direct (navigate) their cursor on screen. Using data collected from production keystroke files, it was possible to quantify navigational activities that occurred within the context of the field interviewing environment.

While the numbers upon which the grid analysis are based are small, they do suggest several things; most importantly, that interviewers exhibited substantial navigation on grid screens, and that the level of navigation is in a positive liner relationship to the grid size. For every cell in which data were entered, interviewers encountered about two cells. For every grid screen, the default flow was interrupted, on average, 1.8 times. We have no video tapes of the interviews, and thus can offer no empirical evidence as to *why* the interviewers exhibited the behaviors they did (e.g., collecting first all the data in the first grid column, and then collecting the data in the last row). We do offer a hypothesis: that, at least some of the time, the navigation was to accommodate the interviewing situation -- collecting the data in the order that the respondent wanted to provide it. The rate of default flow interruption suggests that substituting a more linear questionnaire design, for the more natural grid design (when more appropriate), may make the interviewers' job more difficult.

These descriptive analysis of interviewer navigation on lists suggests a positive linear relationship between the level of navigation (mean keystroke use) and list length, when comparing short to mid size lists. However, after adjusting for the number of elements selected per list, and excluding the two lists with high error rates, the levels of navigation were quite similar for the two mid-size lists of similar design, but of different lengths (13 to 45 items). This suggests the possibility that there is a threshold to list length, and as it is reached, the rate of navigation might become fairly constant once interviewers are familiar with the list. The frequency with which interviewers encountered a list was found to be negatively associated with both the rate of navigation and the Enter key error rates, pointing to the importance of interviewer training in a CAPI environment. The lower navigation rate on the long list, in comparison to the mid-size lists suggested that the interviewers used the search feature on the long 600 element list, but not necessarily on the mid-size lists, which were considerable shorter (mean length of 25). This provides some evidence that perhaps CAPI screens need to inform interviewers when a search function is enabled; simply training the interviewers appeared insufficient in this instance. An alternative, and we believe the preferred approach, would be to make the function a standard CAPI feature.

The authors wish to thank Bill Nicholls for his insightful comments on an earlier version of this paper. Readers who wish to obtain a more comprehensive version of this paper, including tables, exhibits, references and footnotes, should contact the authors.

The views expressed in this paper are those of the authors and no official endorsement by the Department of Health and Human Services and the Agency for Health Care Policy and Research is intended or should be inferred.