Journal of Official Statistics
Vol. 1, No. 2, 1985, pp. 209–219.

Questionnaire Design With Computer Assisted Telephone Interviewing

Carol C. House[1]

Abstract: This paper discusses methods of questionnaire design for computer assisted telephone interviewing (CATI). Although many design principles for traditional questionnaires remain applicable, computerized questionnaires have additional requirements which must be taken into account. This paper presents general guidelines for CATI questionnaire design, describes their consequences for documentation and pretesting, and reviews the skills and time required for these tasks.

Keywords: CATI, computer assisted telephone interviewing; questionnaire design; modular programming.

0. Introduction

Computer assisted telephone interviewing (CATI) is a rapidly expanding technology which is making an important impact on the survey community (Shanks (1983)). Organizations using CATI generally agree that it precipitates major changes in many of their basic survey tasks. This paper addresses one of these tasks – questionnaire design – and presents guidelines for improved design of computerized questionnaires. Section 1 introduces major differences between computerized and traditional questionnaires, and notes how these affect the responsibilities of the questionnaire designer. Section 2 discusses the characteristics of a well-designed CATI instrument, viewed first as a survey questionnaire and then as a computer program. Section 3 shows how these principles of design contribute to documentation, testing, and client review. The final section reviews the mechanics of CATI design and the skills and time they require.

1. How CATI Questionnaires Differ

Computer assisted telephone interviewing places powerful new tools in the questionnaire designer's hands. These include: automatic branching between questions; the ability to modify the wording of each display based on prior data; and the opportunity to detect and reconcile apparent inconsistencies during the interview. They give the designer great control over the interviewing process; and

[1] Mathematical Statistician with the Statistical Reporting Service of the United States Department of Agriculture, serving as project leader for the CATI Research and Development Project from 1980 to 1984. The views expressed are those of the author and not necessarily those of USDA. The author is indebted to an anonymous reviewer and the editor for many helpful suggestions, and to colleagues at the Center for Computer Assisted Survey Methods at the University of California-Berkeley for sharing ideas on CATI questionnaire design over a period of years.

when appropriately employed can ease the interviewer's task while collecting telephone interview data which is more complex, thorough, and consistent than is possible with paper forms. But when employed without careful planning and pretesting, these same tools also can produce disastrous results by omitting applicable questions, providing incorrect feedback to respondents, and leading interviewers into situations from which they cannot easily recover. Like the little girl in the nursery rhyme, when CATI is good, it is very, very good; but when it is bad, it is horrid.[2]

The most obvious difference between CATI and traditional questionnaires is automatic branching. The next question displayed is based on previous answers or other prior information. The branching logic may become highly intricate since it is invisible to interviewers, who are simply instructed to ask each question as it appears. This invisible logic is supplied by the questionnaire designer, who must program each branch (or skip) and ensure that each has a logical path back to the main stream of the questionnaire. Once a designer gains familiarity with this tool, he or she tends increasingly to employ it for a variety of purposes that would not be considered in paper questionnaires. This often results in CATI instruments with highly complex networks of branching paths. However, the more complex these instruments become, the more difficult they are to program, test, and debug, and the more susceptible they are to undetected errors.

A second major difference between CATI and traditional questionnaire design is the option of using "fills" or inserts in displays, based on information already obtained. They may be employed to personalize questions by including the names of those asked about, to incorporate former answers in current questions, to modify interviewer instructions or the wording of answer categories, and even to display summaries of prior answers in tabular form. The interviewers are told to read the questions as they appear and frequently cannot distinguish those parts of the display which are constant from those which have been filled. The questionnaire designer has the responsibility of programming each fill and of ensuring that each works properly under all circumstances.

The third major difference is that CATI allows on-line edit and consistency checking during the interview. When an edit check fails, the interviewer is prompted to enter the response again or is taken to a probe to resolve the problem. Since these checks and probes are built into the CATI instrument, the questionnaire designer becomes responsible for preparing the survey's edit check routines and incorporating them into the flow of the interview. This generally means that questionnaire development for CATI takes more time and effort than for a paper form. It also means that the questionnaire designer must have a sufficient understanding of post-interview processing and analysis needs to perform this task appropriately.

A CATI system with its display screen adds flexibility in design in some ways and reduces it in others. The questionnaire is no longer dictated by the size and shape of a piece of paper, but it is restricted by the size of the VDT screen and its character set. Designing CATI screens is similar to designing a questionnaire on 4 x 6 index cards. One question and set of answers go on each card, and the system displays one card at a time, moving to the next appropriate card when the interviewer records an answer. The system works well for most needs, but new approaches are required both to record tabular information and to display it for interviewers in elaborately filled screens.

[2] Yourdon (1975) used this to describe computer programmers.

Of more concern is the effect the segmentation of the questionnaire has on the interviewer. He or she never sees the questionnaire as a whole piece, only individual screens. This isolation has benefits – it focuses the interviewer's attention on the current question, and ensures that questions are asked in their proper sequence. But it also has disadvantages. The relevance of questions and their relationship to each other may be lost. Good design can mediate the feeling of "question isolation" or "segmentation" by providing the interviewer with enough information on each screen to adequately handle most situations. (This is discussed in the following section.)

Groves et al. (1980) and later Nicholls (1983) have observed that the greatest strength in CATI data collection – it's control of the interview – is also its greatest weakness. This control leads to many of the advantages frequently attributed to CATI. However, when the unexpected occurs, it also strips the interviewer of the ability to improvise. The burden falls on the designer to anticipate and prepare for most eventualities that might occur during the interview.

2. Characteristics of a Well-Designed CATI Instrument

A CATI instrument is both a survey questionnaire and a complex computer program. A designer must draw from existing knowledge in both disciplines to create a successful product. This section presents some of the things that help a CATI instrument work well. The discussion is divided into two parts to examine the instrument from the viewpoint of each discipline.

2.1. Viewed as a "Survey Questionnaire"

The basic characteristic of a well-designed CATI instrument is that it is an equally well-designed "survey questionnaire". To achieve this status, the designer begins with the accepted design principles of traditional questionnaires to create a survey tool that will engage the interest of the respondent, solicit his cooperation, and collect the needed information in an unbiased manner while minimizing respondent burden. Current literature contains numerous good sources covering basic questionnaire design (Payne (1951); Sudman and Bradburn (1974)). Moving beyond these basics, the CATI designer must attend to the special needs of CATI interviewing.

Good CATI design can help maintain the pace of the interview. Of the factors affecting this pace, response time for screen replacement is probably the most important. Interviewers become frustrated when they can not maintain the momentum of an interview because of delays between screens. A silent wait of even a few seconds can become awkward and embarrassing to an interviewer with an impatient respondent on the phone, and may impair respondent cooperation. The speed of screen replacement is largely beyond the control of the questionnaire designer. It depends on such factors as system software, hardware capabilities and competition by other users for available computer resources. However, the designer can moderate his own instrument's demand for these resources by limiting the number of numerical calculations and the number of times data are written and read between display screens.

A second factor affecting interview pace is more directly under the control of the designer. He or she must make each new screen immediately comprehensible so that the interviewer may proceed with the next question or instruction immediately after it appears. The display should assist the interviewer to visually differentiate instructions, verbatim questions and precoded answers. One way this is done is through the use of conventions, such as having all instructions in capital letters or enclosed in brackets. Italic lettering, used frequently on

paper forms, is generally not available on a VDT screen. In fact most screens offer only a single character set and font size. On some systems "reverse video", "highlighted" or different colored images are available and the designer can use them to visually separate functional parts of the display. Some systems offer none of these capabilities, but the designer should use whatever tools are available to develop a useful set of conventions for screen display, and use these consistently throughout the questionnaire.

The designer should avoid crowding the screen. Crowded screens cause the interviewer to waste time searching for needed information or response categories. Proper arrangement of the question, answers and instructions on the screen with "white space" between the parts is particularly important for longer questionnaires where the interviewers tend to be less familiar with individual screens. A crowded screen may be less serious in shorter questionnaires. When interviewers are throughly familiar with all screens, they sometimes prefer one crowded screen to several more carefully spaced ones.

A third factor which affects the pace is how easily the interviewer is able to handle nonstandard movement such as backing up to a previous question and changing an answer. Again much of this is out of the control of the questionnaire designer and dictated by the mechanisms that the CATI system provides for such situations. However, the designer should anticipate when the interviewer may need to review answers, such as after an edit checking sequence, and display information on the screen that will assist the interviewer in his or her task. Such information might include the names of questions most likely to be reviewed, along with simple reminders on how to move about in the questionnaire. The designer may choose to "fill" the actual answers to previous questions on the screen, so the interviewer can review these, backing up only if a change is needed.

In those situations where the designer does not provide specific help in backing up, the interviewer's task is simplified if the question names are easy to remember. This is particularly important in CATI systems where the question name is used in the jumpback command. The name may directly relate to the substance of the question ("AGE" for the age of the respondent) or follow a sequential ordering within specific topics ("HOSP5" for the fifth in a sequence of questions about hospital confinement).

Because the flow of a CATI questionnaire is enforced by the computer, it tends to be rigid. A CATI instrument is flexible only to the extent that flexibility is designed into it. The designer must plan and prepare in advance for the expected and unexpected alike. Unexpected situations, difficult in any interview, are particularly troublesome with CATI because the interviewer has only the options which the instrument designer has provided.

The designer must provide the interviewer a way to record the answer to a question, no matter how unexpected that answer might be. In a question with precoded answers, the "other/no answer" category, coupled with the provision to leave a note, is very important. An instrument should facilitate unanticipated movement as well. As briefly discussed earlier, an interviewer may need to back up, review a previous answer, and return to the original place in the interview. The instrument must work as well after such a backup as it does during forward movement. Arithmetic calculations require special protection from backups, as illustrated by this common design error. The respondent is asked for the number of trips to the doctor in the last six months, and then for the number in the six months before that. The program adds the two numbers and computes the total visits for the year. An interviewer backs to review an earlier question, returns to the original place in the interview, and finds that the number of trips has been re-added, giving a final count twice

its actual value. To avoid such a problem, the sum must be re-initiated to zero and all calculations repeated each time the interviewer passes through these items.

A well-designed instrument provides adequate information on each screen. The interviewer should be supplied with the correct thing to say in all anticipated situations. The questionnaire designer must foresee additional information that may be useful and provide it as well. Common examples are: the respondent's name, the names of other household members, or the answer to a previous question. This aspect of good design can reduce the interviewer's feeling of "question isolation". Otherwise, interviewers may feel deprived of access to information they just collected – it's in the machine but not readily accessible.

Auxiliary information can be placed on the screen as part of the question, or in a special area reserved for instructions. Information used less frequently can be placed on a separate "help screen" that is called only when needed. Some CATI systems provide straightforward methods of designing "help screens," while others require extensive programming to make them work.

The full potential of CATI data collection can be achieved only with a questionnaire that exploits the power of the system. The designer can use branching and screen filling to individualize an interview so that a respondent is answering only pertinent questions.

On-line editing of data is one of CATI's most powerful capabilities. The designer should use it whenever possible, bringing inconsistencies to the attention of the respondent and asking his or her aid in resolving them. An on-line edit sequence in an instrument begins with a comparison of answers (or other data) in a "background" process. If the data are not consistent, the instrument branches to a screen that notifies the interviewer. It is important to make this screen recognizable (with a distinctive format or a "notice" at the top of the screen) because the interviewer will not be expecting to see it. The screen should supply the interviewer with a statement to solicit the respondent's help in resolving the matter without putting the respondent on the defensive about previous answers, or leading him to new ones. It may give the previous answers to questions involved in the consistency check. The designer may request interviewer notes when unexpected results appear, but the questionnaire should always give the interviewer a way to return to the main part of the questionnaire.

A CATI instrument can provide valuable feedback to the designer on the quality of the questionnaire by providing an audit trail of changes made to data during the survey process. Some CATI systems provide this automatically. With others, the questionnaire designer must take special steps to retain needed information. Such audit trails are valuable tools for the designer to locate particular questions that are giving respondents and interviewers trouble.

Interviewer notes have always provided key insights into what transpires during an interview, and CATI provides the questionnaire designer with the ability to "encourage" notes at crucial points in the questionnaire or when an unusual set of circumstances occur. For example, it may be desirable to have an explanatory note after a consistency check failure, especially if the interviewer determines that the data is correct as recorded. The designer may put a "friendly reminder" on the screen in such as situation, encouraging the interviewer to enter a command that will allow her to record notes. Alternatively, the designer may force the interviewer into recording a note, before she is allowed to return to the mainstream of the questionnaire.

2.2. *Viewed as a "Computer Program"*

This section applies the techniques of "good" programming to the problems of CATI

instrumentation. These are areas of design where CATI questionnaires are generally weak.

Perhaps the most important characteristic of a good computer program is that it works – it does what the designer intends, the way that the designer intends it to be done. In the case of CATI, the instrument must work along with the underlying CATI system to do its job efficiently and effectively. The better the designer understands the structure of that CATI system the more effective his instrumentation will be.

A key concept in programming is the use of modules in a structured approach to design. A module is a self-contained subroutine which performs one specific well-defined function in the program. It is small and independent of other parts of the program. This means that all paths leading into and out of the subroutine are limited in number and carefully controlled by the designer. From a CATI point of view, a module is a self-contained group of questions connected by a general theme or function. A "front-end" module, for example, would contain the questions that deal with dialing instructions, what to do with no-answers and busy signals, and setting up appointments to reach the proper respondents at a later time.

Modules are a critical part of the design of larger, more complex instruments. By using modules the designer has broken a large task down into smaller blocks that are easier to work with. A modular instrument is easier to design, program, debug, document, modify and review with clients. Modular design also provides a mechanism for using parts of an instrument on later surveys. For example, most random-digit dialing surveys would use similar "front-end modules." Survey organizations can develop standard frontend and backend modules, or modules for randomized respondent selection for related types of surveys and interchange them. On particularly large and complex instruments, modular design simplifies the problem of having several people work together. Each can be assigned independent parts to program, test and document.

The easiest way to achieve the type of modularized design discussed above is to use what the programming world calls "top-down design." Working from the top, down, the designer "formally" identifies the major functions of the program first, and then proceeds to identify the lesser functions that extend from the major ones. Eventually, functions are broken into small, well defined parts which can be programmed as individual modules. In a CATI instrument a major function might be "obtaining an acceptable respondent," or "obtaining a health profile of the respondent." Obtaining an acceptable respondent could be broken down into the functions of "locating a household," "obtaining permission to conduct the survey," and "randomizing the respondent selection." Yourdon (1975, p. 45) says that the "key to successful top-down design is a formal and rigorous effort on the part of the designer to specify the input, the function, and the output of each module in a program or system."

Well-designed instruments are built to be modified. Most computer programs are changed during their useful lifetime, and CATI instruments are no exception. Parts or all of the instrument will be used again. Last minute bugs will be found that need to be corrected quickly before the production survey can continue. The designer should design every part with an eye toward its eventual revision. Modular design assists in this effort. Changes in one module do not affect the others, except in very carefully constructed interfaces. Keeping the design simple will also help – the object with the smallest number of moving parts is the least likely to break, and the easiest to fix when it does. Designers must make trade-offs between the advantages of complex, intricate CATI

instruments which can do sophisticated things in a survey, and the disadvantages of having instruments which are difficult to modify and debug.

Instruments should be flexibly designed so that they can handle a variety of situations with little modification. One suggestion is to use "symbolic variables" instead of "constants." For example, the designer of a food consumption survey might wish to ask a detailed set of questions if an individual eats three or more meals per week in a restaurant. Rather than code a "3" throughout the instrument to force the desired branching, the designer creates a symbolic variable called "eats," and uses it to define the logic flows. At one (and only one) specific place in the instrument, the designer gives this variable the value "3." Then if the "3" must later be changed to a "4," the change need be made in but one location.

Finally, and very importantly, a well-designed program is well documented. The importance of this, and suggestions on what this documentation should include are discussed in detail in the following section.

3. Documentation, Testing, and Client Review

3.1. Documentation

Good documentation is essential for any CATI instrument. The clients need a well documented instrument to determine whether it meets their needs. Documentation is important for testing and debugging the instrument and to assist in making changes. It can expand the useful life of a CATI instrument, by making it easier to use parts of it on future studies. Few will attempt to use part of an existing instrument unless he or she is very, very sure what it does, how it interacts with other parts of the instrument, and where it stores data.

Top-down design and modular programming make documentation a manageable job. By using the concepts outlined under the top-down design, the designer is creating a design flow of the functions of the instrument, and of the modules that fulfill those functions. Each module can then be documented separately. The programming and the documentation of an individual module is done concurrently rather than waiting until the entire instrument is completed and tested before beginning the documentation process. Then documentation can assist in the debugging and testing stages.

Thorough documentation may include the following items:

1) a flow diagram of the instrument design (as created in the top-down design process);
2) a module by module description (including a statement of the function of each module, a listing of variables, or question names, the inputs and outputs, and information on variable storage, both temporary and permanent);
3) a hard copy listing of the instrument;
4) a list of values of all symbolic variables;
5) a listing of differences between multiple versions of the instrument, whether they reside in a single program or several programs;
6) comments interspersed within the code, if permitted by the development language; and
7) other documentation normally provided for paper and pencil questionnaires.

3.2. Testing And Debugging

The concept of top-down modular design is a great aid to the designer in testing and debugging an instrument. Lower level, individual modules can be programmed, tested and debugged separately. The designer can be sure that each individual part is working correctly before combining it with another. The modules are fairly easy to debug because

they are reasonably small and have a clearly defined function. If a module is documented as it is completed, it is clear how the module should function, what it needs as inputs, and what the outputs should look like.

There are a number of steps in testing and debugging a given module. First, the designer will visually verify the instrument logic and wording as he is writing it. Second, syntax errors are eliminated so the instrument will execute. Most CATI systems have some automated way of checking for syntax errors. Third, the designer must manually check the logical ends to all branching paths. He or she must examine all "fills" and edit checking sequences to ensure that they are working properly and that the edit parameters are correct. Fourth, the designer should evaluate any external or input data (such as a name and telephone file) to make sure that the data are of high quality, and that their layout in the file is properly defined in the instrument. Fifth, the designer should test the instrument's reaction to unusual movement by an interviewer – backups, skips, change answers, etc., to make sure that the instrument continues to function correctly in all of these situations. The final step is to review the instrument's performance as a high quality questionnaire, particularly in respect to some of the points discussed earlier in this paper.

Once lower-level modules are functioning, the tester can begin putting them together to debug the interfaces. Once the entire instrument is working to the designer's satisfaction, it should be pretested. Never use a CATI instrument in production surveys without an adequate pretest.

3.3. *Client Review*

The top-down modular design of the CATI instrument is the framework for client review, as it is for documentation and testing. The review begins with the overall design of the instrument and its various functional modules. The client and designer can agree on the overall design of the instrument before any programming begins. The documentation discussed earlier will assist the client in evaluating the instrument and determining if it fulfills his needs. Since the documentation is developed as the modules are programmed, the client review can be ongoing as well.

Modules, used to divide programming and debugging into manageable pieces, can now do the same for client review. The flowchart of the overall design will help the client keep the functions in perspective as he looks at each module intensively. Some clients may wish to examine the written code or copy of the questionnaire – it isn't as formidable for a single module as it would be for an entire instrument. In most cases, he will want to execute the module so he can see what it looks like on the screen and follow the various branching paths to their conclusions.

4. Basic Requirements: Development Languages, Skills and Development Time

Three topics, related to the mechanics of programming a CATI questionnaire, are discussed below – the programming language used to develop the questionnaire, the skills needed by questionnaire designers, and the time required for this development. The discussions are not technical. Rather, they are aimed at the reader without substantial CATI experience. The programming languages are discussed first because their choice affects the other two topics.

4.1. *Development languages*

Two types of languages are used in CATI systems for programming questionnaires – special purpose languages developed specifically for this job, and general purpose languages such as FORTRAN or BASIC.

Specialized questionnaire development languages reduce the need for programming skills to the point that individuals holding the responsiblity for the design of paper and pencil questionnaires can typically maintain these responsibilities using CATI. Most CATI systems include a specialized development language. The system may require that designer submit a "batch" questionnaire program or it may interactively "prompt" the designer for information. An interactive system specifically asks the designer for all necessary information, e.g., "Do you wish to add an additional question?" The designer responds on the keyboard. A batch system requires that the designer create a prototype of a question using a set of predefined conventions. These indicate to the system when a question begins and ends, what the question numbers, answer codes, and commands are. To do this, the designer creates a text file using a word processor, or text editor, and types a set of questions basically the way they will look on the screen during the interview.

Each approach has merit. Interactive design makes it easier to begin. A designer of traditional questionnaires with virtually no previous programming skills can begin writing CATI questionnaires as soon as he or she is taught to logon to the system and to use a keyboard. In contrast, the batch system requires that the designer first master several key elements: text editing, predefined conventions, and the set of commands that create the questionnaire from the text file. The relative advantage of the interactive system disappears as the CATI designer gains experience, and the designer eventually may feel that the interactive queries actually slow her entry of the questionnaire. The batch approach to instrumentation more closely resembles traditional computer programming. This may be advantageous for more complex designs.

Sometimes a general programming language such as FORTRAN or BASIC is used to design each CATI questionnaire. A skilled programmer writes a program that generally combines two separate functions of a CATI system – the design of a specific questionnaire and execution of that questionnaire during an interview. This approach can work well for short, simple questionnaires and specialized applications. However, it permanently places a skilled programmer between the questionnaire designer and his questionnaire. This may change in the future. Recent advances in Fourth Generation programming languages are bringing sophisticated programming capabilities to less skilled users. These developments may make the use of general purpose languages more feasible.

4.2. Skills for CATI Questionnaire Designers

Questionnaires should be developed by individuals skilled in the principles of good design and with an understanding of the interviewing process, regardless of whether those questionnaires are for traditional or computerized implementation. CATI in no way lessens the need for normal expertise in this area – it requires all those skills *and* some new ones. These new skills involve programming and a fuller knowledge of the total data collection effort.

CATI will require that all designers develop some programming skills and an understanding of the characteristics that make a well-designed program. The extent of these requirements is heavily dependent on the choice of questionnaire development language and the complexity of specific data collection efforts. At a minimum, the designer, like the CATI interviewers, must learn to work with a computer and overcome basic machine fear. The more complex the questionnaire, however, the greater the need to treat the questionnaire as a real computer program, and follow the guidelines discussed in this paper.

CATI questionnaires, in all but the simplest cases, incorporate most aspects of the total survey design including the edit and summary phases. Previously, the questionnaire designer might take only a minor interest in the post-interview phases of a study. Sometimes the latter phases are not even planned until the data collection is complete. With CATI, the questionnaire designer must take an active role in these other areas, and make them a part of the original design.

4.3. Development time

CATI questionnaires take more time to develop than do traditional questionnaires, primarily because a CATI questionnaire integrates more of the total survey design. Much of this is merely a shift of efforts from the end of the survey to the questionnaire design phase. Does CATI take more time *overall* to develop? This is difficult to generalize – so much is dependent on the particular surveys being conducted. However, one can say that CATI clearly has very definite advantages for organizations with recurring surveys and with surveys requiring rapid summarization and publication.

The Statistical Reporting Service (SRS) of the U.S. Department of Agriculture conducts surveys with both characteristics. These surveys have several repeated data collection periods throughout the year. The interviews are relatively short – most take less than ten minutes. Those currently adapted to CATI contain twenty to thirty substantive questions, collecting factual data about farming operations. In addition, there are many edit checking sequences, at least one for every substantive question. The results of these surveys are published within days of the completion of the interviewing. SRS has found that it takes approximately 80 to 120 staff hours to program one of these questionnaires, using a trained CATI questionnaire designer, a paper and pencil version as the base, a listing of needed edit checks, and the frontend and backend modules from similar CATI questionnaires. This represents development and debugging time, but does not include pretesting the questionnaire. For someone without CATI design experience, the same task might take four of five times longer. SRS uses the Berkeley CATI system which has a batch mode, specialized questionnaire development language called Q. (See Shanks, et al. (1980); House and Morton (1983); Morton and House (1983)).

5. References

5.1. References Cited in the Text

Groves, R., Berry, M., and Mathiowetz, N. (1980): Some Impacts of Computer Assisted Telephone Interviewing on Survey Methods. In the Proceedings of the Section on Survey Research Methods. American Statistical Association, Washington, D.C.

House, C. and Morton, B. (1983): Measuring CATI Effects on Numerical Data. In the Proceedings of the Section on Survey Research Methods. American Statistical Association, Washington, D.C.

Morton, B. and House, C. (1983): Training Interviewers for Computer Assisted Telephone Interviewing. In the Proceedings of the Section on Survey Research Methods. American Statistical Association, Washington, D.C.

Nicholls, W. (1983): Development of CATI at the U.S. Census Bureau. In the Proceedings of the Section on Survey Research Methods. American Statistical Association, Washington, D.C.

Payne, S. (1951): The Art of Asking Questions. Princeton University Press, Princeton, N. J.

Shanks, M. (1983): The Current Status of Computer Assisted Telephone Interviewing: Recent Progress and Future Prospects.

Sociological Methods & Research, Vol. 12, No. 2, pp. 119–142.

Shanks, M., Lavender, G., and Nicholls, W. (1980): Continuity and Change in Computer-Assisted Surveys: The Development of Berkeley SRC CATI. In the Proceedings of the Section on Survey Research Methods. American Statistical Association, Washington, D. C.

Sudman, S. and Bradburn, N. (1974): Response Effects in Surveys. Aldine, Chicago.

Yourdon, E. (1975): Techniques of Program Structure and Design. Prentice-Hall, Inc., Englewood Cliffs, N. J.

5.2. Reference Not Cited in the Text

Frey, James H. (1984): Survey Research by Telephone. Sage Publications, Beverly Hills.

Received October 1984
Revised June 1985