Catalogue No. 93-08

QUESTIONNAIRE DESIGN IN A PAPERLESS SOCIETY

March 1993

Ruth Dibbs, Household Surveys Division

Alison Hale, Household Surveys Division

The SLID Research Paper Series is intended to document detailed studies and important decisions for the Survey of Labour and Income Dynamics. These research papers are available in English and French, at no charge. To obtain a summary description of available documents or to obtain a copy of any, please contact Philip Giles, Manager, SLID Research Paper Series, by mail at 11-D8 Jean Talon Building, Statistics Canada, Ottawa, Ontario, CANADA K1A 0T6, by telephone (613) 951-2891, or by fax (613) 951-3253.

EXECUTIVE SUMMARY

This paper was presented on March 23, 1993 at ARC 1993, the Annual Research Conference sponsored by the U.S. Bureau of the Census.

The Survey of Labour and Income Dynamics (SLID) is a new national longitudinal survey sponsored by Statistics Canada. It was decided in the early stages of the planning and development of the survey that the mode of data collection would be decentralized computer-assisted interviewing (CAI). Many of the other CAI applications are adaptations of existing questionnaires. The challenge for the SLID Questionnaire Design Team was how to go about the process of designing a new "questionnaire" for this mode of data collection. This paper describes the learning curve associated with breaking away from the traditional method of developing questions for a "paper and pencil" questionnaire to providing specifications for questions and flows for a programmer.

TABLE OF CONTENTS

1. INTRODUCTION

The Survey of Labour and Income Dynamics (SLID) is a new longitudinal household survey being developed by Statistics Canada and is designed to follow respondents for a period of six years, collecting information on their labour market experiences, income and family circumstances. The SLID sample size is 40,000 households, split evenly between two panels with a new panel rotating into the sample every 3 years. The respondents for the first panel of 20,000 households were selected from the Canadian Labour Force Survey. The first contact with these respondents on behalf of SLID was a preliminary, or benchmark, interview conducted as a supplement to the January 1993 Labour Force Survey and as such, was a paper and pencil questionnaire. This interview was designed to collect background information on selected households at the beginning of the first reference year of SLID.

In subsequent years, respondents will be contacted in January to collect information on the jobs held in the previous year and on their job search activities between spells of employment. A second contact in May of each year will collect income and wealth data. Thus, the second contact of the first panel will be in January 1994, when the first annual labour interview will be conducted. This and all subsequent interviews will be done using decentralized computer-assisted telephone interviewing (CAI). In January and February 1993, a CAI test of this labour interview was also implemented.

A distinctive feature of SLID is that from its initial planning it had been decided to use CAI as the method of collection for all interviews after the benchmark interview. This makes it different from most surveys which use, or plan on using, computer-assisted interviewing, which was the motivation for this paper. The process of developing a CAI application for SLID was not a matter of transforming a paper questionnaire into a computer program but of specifying questions for programming. Starting from the beginning with CAI meant that we were not restricted to preserving the limitations of a paper questionnaire but on the other hand it also meant that we had to develop a new way of "designing a questionnaire" which would take advantage of CAI.

2. QUESTIONNAIRE DESIGN FOR CAI

The process began with the people involved in developing the questions for the SLID interview. We all had to learn not to think "pencil and paper" - that is not limit our ideas to only what works on a paper questionnaire. Next, we had to develop a structure for producing a questionnaire in the format of specifications to be given to a programmer.

2.1 THINKING COMPUTER-ASSISTED

One of the big advantages of CAI is that you are no longer faced with the physical limitations of a paper questionnaire. In theory you could have extremely complicated question flows and a large number of checks and edits. In reality, the risk of increasing the length and respondent burden of the interviews through too many edits, and the difficulty in programming and thoroughly testing complicated flows does limit what can be done.

Also, the team developing the questions for the questionnaire, or application as it should more appropriately be called, must now be familiar with a number of technical issues that they may or may not have had to deal with before. The question development team must be aware of the limitations of the software(s) being used, the skill-levels of the programmers and even the capacity of the computer which will be used by the interviewers in order to prevent massive rewriting of the application once it has been developed. It is also necessary for the persons formulating the questions to specify the edits that should be applied to the responses, as well as the possible values and skip patterns based on the answers.

2.2 QUESTION SPECIFICATIONS

A great deal of thought was put into the format of the question specifications which were to be produced by the SLID questionnaire design team for the programmers. For a number of reasons it was considered best to produce question specifications in packages, or modules, which corresponded to groups of related questions. First, this is how the design team would have to work to ensure that the flow from question to question was correct. Also, with the application broken up into modules, the programming work could be distributed among a number of programmers and the programmer assigned a module could then ensure that the data flows were correct throughout a module. This approach also helped the questionnaire design team to easily split up testing responsibilities, by assigning modules to be tested to different people.

The specifications for a module consisted of a flow chart showing the skip patterns for the questions and detailed instructions for how each question should be programmed. It must be noted that the question specifications are not only necessary for the programmers but as documentation to replace the paper questionnaire. Later an example will be used to illustrate how this was put into practice, however it is important at this point to go into more detail in how the questions were specified.

A template (See Figure 1) was developed for specifying the wording of a question and providing the information a programmer would need.

FIGURE 1

DISPLAY QUESTION FORM

1.|		**2**.Revision#		**3**.Date:	

QUESTION DESCRIPTION

4. Header_Text:	
TEXT=	
5. Special_Help:	

ANSWER DESCRIPTION

6.	QTYPE=	**7**.Format:		**8**.Prefill:	
9. Allowable Answers					**10**. Save as
11. Range	Maximum			Soft Max.	
	Minimum			Soft. Min.	

FLOW DESCRIPTION

12. Check Condition:		**13**. Go to	**14**. Notes
1. Don't Know 2. Refusal 3. ANSWER = 4. ANSWER = 5. ANSWER = 6. Default			

The format closely followed the information needed by a programmer using the CAI software developed by the Centre for Human Resource Research (CHRR) at Ohio State University, since this was the software used for much of the SLID application. With the CHRR software only one question appears on the screen at a time. Other software was used (such as FOXPRO) to program parts of the SLID questionnaire, but the same template was used. The main advantage of the template was to provide a structure to ensure that all of the people who were specifying questions for programming included the necessary detail.

Figure 1 is an example of the template used (a WordPerfect table). It is divided into 4 sections -

a) Section 1 contains control information as part of the documentation process and is not programmed.

1.|		**2**.Revision#		**3**.Date:	

b) Section 2 is for the description of the actual question.

QUESTION DESCRIPTION

4. Header_Text:	
TEXT=	
5. Special_Help:	

! *Header Text* - the information that should be displayed at the top of the screen. This is usually extra information that the interviewer may find useful when asking the question.

! *TEXT* - the question.

! *Special_Help* - used to give the name of the file containing context sensitive help information.

c) Section 3 is to specify the format of the answer to the question.

ANSWER DESCRIPTION

6.	QTYPE=	**7**.Format:		**8**.Prefill:	
9. Allowable Answers					**10**. Save as
11. Range	Maximum			Soft Max.	
	Minimum			Soft. Min.	

! *QTYPE* - the type of answer to the question. For example, a DATE or a YES/NO answer.

! *Format* - length of the answer (number of digits) and format of a date (MMYY or DDMMYY)

! *Prefill* - what information (either from a previous interview or a previous question) is to be prefilled in the answer. For example, when asking for the current address the old address may be prefilled in the answer box on the screen.

! *Allowable Answers* - list of acceptable answers to the question.

! *Save as* - the value that an answer is to be saved as. For example, a Yes answer may be saved as a 1 and a No as a 2.

! *Range* - acceptable range of answers, mainly for numeric answers. A soft maximum range, is one which is acceptable but unusual and the interviewer would receive a message indicating this.

d) Section 4 describes the flow of the questions based on the answer given to this question.

FLOW DESCRIPTION

12. Check Condition:		**13**. Go to	**14**. Notes
1. Don't Know 2. Refusal 3. ANSWER = 4. ANSWER = 5. ANSWER = 6. Default			

! *Check condition* - list of the various answers which result in a different flow or skip pattern.

! *Go to* - which question to go to based on the corresponding Check condition.

! *Notes* - any notes that would help the programmer or to specify if any error message is to be displayed based on the Check condition.

3. AN EXAMPLE

The version of the Labour interview of SLID, which was tested early in 1993, is divided into 6 main modules and with a total of over 100 different questions (some of which are repeated depending on the number of employers in a year). The smallest module, SUPPORT, will be used to illustrate the question specification process.

The SUPPORT module contains questions on unpaid care-giving and receiving, and how it affects ones ability to work in the paid labour market. The flows are relatively simple and the module has 18 questions. Figure 2 is the flowchart for this module. Note that even though SUPPORT is the last set of questions asked of the respondents, the first question is still numbered Q1. Questions are numbered sequentially within each module and the name of the module was used as the first part of each question name. Thus the first question in the SUPPORT module would be SUPPORT.Q1.

3.1 THE SUPPORT FLOWCHART

The first thing that should be noted about the flowchart is that the complete question is not given. To be useful a flowchart should show the entire flow on one page and if the whole wording was given even a small chart would be 3 to 4 pages. Therefore, rather than the full question, an abridged version was shown. The symbol for a question on the chart indicates the type of question. A rectangle is a question for which the flow is only in one direction, that is, no matter what the answer is to a question, the next question to be asked is the same. For example, no matter what the answer is to SUPPORT.Q2 the following question will be SUPPORT.Q3.

If the question is in a diamond shaped symbol, then the flow varies depending on the answer given to the question. For example, if the answer to SUPPORT.Q1 is NO the next question asked will be SUPPORT.Q6, but if the answer is YES, the next question asked will be SUPPORT.Q2.

Some of the symbols do not have a question number by it, such as the diamond containing ~WORKED?~ after SUPPORT.Q13 in the flowchart. This is a NON-DISPLAY "question" - a skip, hidden to the interviewer, based on answers to previous questions in the current interview or based on information on the input file.

FIGURE 2

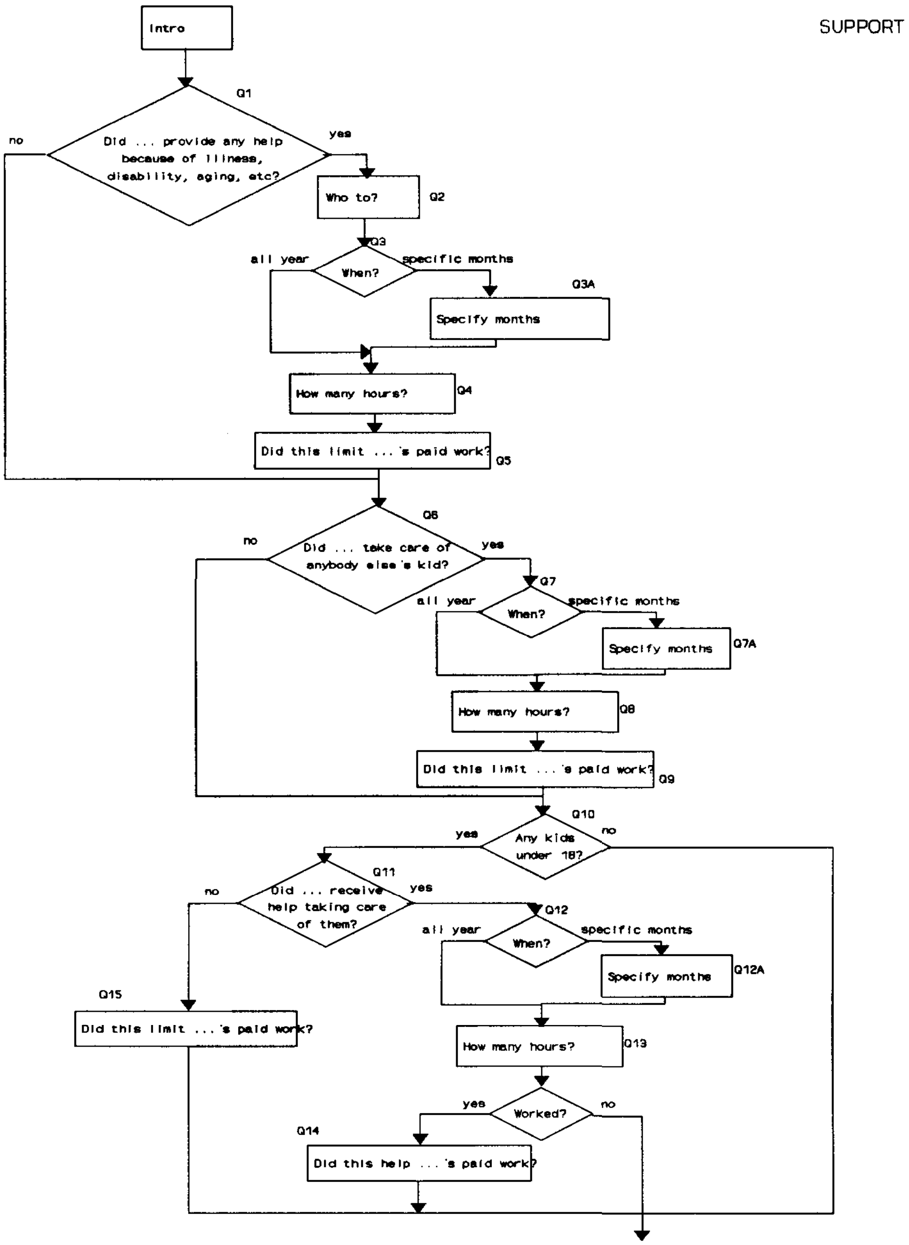
SUPPORT
Intro
Q1
Did ... provide any help because of illness, disability, aging, etc?
no
yes
Who to?
Q2
Q3
When?
all year
specific months
Q3A
Specify months
How many hours?
Q4
Did this limit ...'s paid work?
Q5
Q6
Did ... take care of anybody else's kid?
no
yes
Q7
When?
all year
specific months
Specify months
Q7A
How many hours?
Q8
Did this limit ...'s paid work?
Q9
Q10
Any kids under 18?
yes
no
Q11
Did ... receive help taking care of them?
no
yes
Q12
When?
all year
specific months
Specify months
Q12A
Q15
Did this limit ...'s paid work?
How many hours?
Q13
Worked?
yes
no
Q14
Did this help ...'s paid work?

FIGURE 3

DISPLAY QUESTION FORM

1.|	QNAME=SUPPORT-Q1	**2**.Revision#	2	**3**.Date:	November 26

QUESTION DESCRIPTION

4. Header_Text:	%FNAME% %LNAME%
TEXT=IN %SURVYEAR%, DID %FNAME% %LNAME% PROVIDE ANY UNPAID HELP TO ANYONE WHO HAD TROUBLE TAKING CARE OF HIM/HERSELF BECAUSE OF AGE, A DISABILITY, A PHYSICAL OR MENTAL ILLNESS, OR FOR ANY OTHER REASON?	
5. Special_Help:	

ANSWER DESCRIPTION

6.	QTYPE=yes/no	**7**.Format:	integer	**8**.Prefill:	
9. Allowable Answers					**10**. Save as
Yes No					1 2
11. Range	Maximum			Soft Max.	
	Minimum			Soft. Min.	

FLOW DESCRIPTION

12. Check Condition:		**13**. Go to	**14**. Notes
1. Don't Know			
2. Refusal			
3. ANSWER =	Yes	SUPPORT-Q2	
4. ANSWER =	No	SUPPORT-Q6	
5. ANSWER =			
6. Default		SUPPORT-Q6	

3.2 QUESTION TEMPLATE

Figure 3 is the completed question template form for the first support question, SUPPORT.Q1. The words surrounded by percentage signs are variables that are to be used in the question. For example, %FNAME% %LNAME%, is the respondent's first and last name.

From the template, we can see that the question name is SUPPORT-Q1, the respondent's name is to be displayed at the top of the screen as a header and it is a YES/NO type of question. In the flow description, we see that a YES answer will result in SUPPORT-Q2 being asked next and a NO answer will result in SUPPORT-Q6 being asked. There are no special flow instructions to an answer of DON'T KNOW or REFUSAL, so if either of these answers is given the next question asked would be the Default, SUPPORT-Q6.

4. HOW DID THE PROCESS WORK?

As can be expected, it took awhile for the specification process to go smoothly. First, the persons providing the specifications had to become used to providing the information in the standard format. In addition, the question template went through a few modifications to get to its current form. The main change to the template was to set it up so that some of the information on the form could be imported into the CHRR software without being re-typed by the programmer, particularly the question itself. This functionality is provided by the CHRR software and was achieved by using many of same key words in the template (such as TEXT=) that the CHRR software uses. Not having to re-type the questions not only cut down on the possibility of typographical errors but also sped up the programming of the questions. This was a definite plus when specifications for some questions were still being prepared less than two months before the test of the Labour interview had to go into the field.

Originally, we had thought that the flowcharts and the question specifications would be all the documentation needed by the people testing the SLID application. Although the modular flowcharts were very effective at the beginning, when integrated testing of the complete application began, it was necessary to add higher level flowcharts showing the linking of the modules and the routing and flow of the components. It soon became apparent that this was entirely too much paper to wade through so a package was prepared of the flowcharts and the question wordings. Finally, the document covering all of the questions being asked in the January 1993 test of the labour interview was more than 140 pages, but it did work well.

As a result of the in-house testing a number of areas requiring changes were identified. The problem was how to keep the written specifications up to date so that they reflected what was in the CAI program. We had great intentions of being diligent about updating the specifications at the same time as the program. However as it got closer and closer to the time for the field test, this was not possible. As soon as the test was in the field however, the detailed specifications were verified against the program and updated as necessary. Not too surprisingly, this exercise uncovered some minor errors in the program which will be corrected before the next test later this year. <u>It is very important</u> that these specifications are not allowed to become out of date since they will be used as a reference to find out what is going on behind the scenes of the program. They also will be used when the application is rewritten in a few years, something which is inevitable as CAI continues to develop and better tools are available.

5. CONCLUSIONS AND PLANS FOR THE FUTURE

It must be obvious by now that there is nothing paperless about the process described here. It is only the computer assisted collection of the data that does not require the use of paper. It should be noted that using CAI does not get rid of the need for something that resembles a paper questionnaire showing all of the questions, possible answers and skip patterns. Many users will want this level of detail and even the interviewers conducting the survey would like this as a reference. SLID is now preparing such a document on what was programmed for the January, 1993 test.

The next SLID application to be programmed is the second contact in May of the respondents in the January test to collect income and wealth data. The templates will not be used for this interview since it is not so much a series of questions but a list of dollar values being collected. As such, the application is being programmed as a list on the screen which the interviewer will scroll through to get to the items for which the respondent has values. But as was mentioned earlier, the more complex labour interview will continue to be specified using the process described here. It has proved to be an efficient method to standardize the work of a team in producing the details needed for programming, while at the same time providing the documentation required for any future development.