Chapter 1

The Importance of Design for Web Surveys

Design is choice
Edward Tufte (2001, p. 191)

Why a book on the Web survey design? This question really has three parts: (1) why *Web*, (2) why *survey*, and (3) why *design*? I will try to address the three parts of this question in this chapter, but first I will briefly describe the major types of Internet and Web surveys prevalent today. This will set the stage for the discussion to follow.

1.1. Internet and Web Surveys

In the relatively short time that the Internet, and particularly the World Wide Web, has reached widespread penetration in the United States and elsewhere in the world, Internet or Web surveys have rapidly emerged as a major form of data collection. It's sometimes hard to imagine that Telnet was developed in 1987, and the first graphical browser (NCSA Mosaic) was released as recently as 1992 (see www.ncsa.uiuc.edu), given the attention that Web surveys are getting in the survey profession and research literature. The adoption of online surveys has spread faster than any other similar innovation, fueled in part by the dot-com frenzy of the late 1990s but also driven by the promise of faster and cheaper data collection. For example, according to ESOMAR's global research study conducted in 2001, an estimated 20% of the U.S. survey research market could be conducted online by 2002, reaching 35% by 2004 (see www.esomar.org). More recent data reveal that

U.S. online market research spending increased from $4 million in 1996 to over $1.6 billion in 2007 (Inside Research, 2007; see http://insideresearch.com/), with online research making up some 40% of all commercial research in the United States in 2006. While the rate of growth in online research appears to have slowed in recent years, the overall trajectory has still been nothing short of remarkable.

As is often the case in such innovations, the market research sector has rapidly embraced the new method of data collection, while government and academic survey enterprises have been somewhat slower to adopt Internet survey methods, using them more as supplements to rather than replacements for other modes of data collection. But this is changing too, and Internet surveys are an increasingly important component of all sectors of the survey world.

While Internet surveys show much promise, they also suffer from a number of potential limitations. Key among these is the likely coverage error arising from the fact that not everyone in the target population of interest may have access to the Internet. Furthermore, those who do have access appear to be different from those who do not on a variety of important measures of interest (e.g., Couper, Kapteyn, Schonlau, and Winter, 2007; Robinson, Neustadtl, and Kestnbaum, 2002; Schonlau et al., 2004). The Web also presents many challenges for generating probability samples of broad populations of interest, such as all Internet users. It is unlikely we will soon (if ever) see the development of random digit dial or RDD-like procedures for sampling Internet users, nor are lists of all Internet users and their e-mail addresses likely to be developed anytime soon. Nonresponse error is also a potential cause of concern in Internet surveys, as with all other modes of survey data collection. These sources of error are not the primary focus of this book and are reviewed elsewhere (see Couper, 2000; Fricker and Schonlau, 2002; Schonlau, Fricker, and Elliott, 2002). What *is* the focus of this book is measurement error and its reduction in Internet and, more specifically, Web surveys.

1.1.1. Different Types of Internet Surveys

Thus far, I have used the terms *Internet survey*, *Web survey*, and even *online survey* interchangeably. It's time to get a little more specific and narrow the focus to what this book will cover. Internet survey refers simply to any survey in which the data are collected via the Internet. The Internet can be used for survey data collection in several different ways. A distinction can be made between those surveys that execute on a respondent's machine (client-side) and those that execute on the survey organization's Web server (server-side).

Key client-side survey approaches include e-mail surveys and downloadable executables. In each of these cases, the instrument is transmitted to sample persons via the Internet. Respondents then answer the survey questions either by using the reply function in the e-mail software, by entering responses using a word processor, or by using software installed on their computers. Once complete, the answers are transmitted back to the survey organization. E-mail surveys (see, e.g., Couper, Blair, and Triplett, 1999; Schaefer and Dillman, 1998) have fallen out of favor, largely because of technical limitations and security concerns. As I note shortly, there are other types of client-side interactivity that can be active while a respondent is completing a Web survey online, but in the previous examples, the transmission of information back and forth to the server is not happening in real time.

Server-side systems typically involve the sample person completing the survey while connected to the Internet through a browser, with the answers being transmitted to the server on a flow basis as each *submit* or *next* button is pressed. Interactive features of the automated survey instrument are generated by scripts on the Web server. A key distinction between these two approaches is whether the Internet connection is *on* while the respondent is completing the survey. Web surveys are the prime example of the second type, and are by far the dominant form of Internet survey prevalent today. The focus of this book is on *Web surveys*.

In addition to different types of Internet surveys, there are also a variety of ways one can access the Internet. These range from "standard" graphical-based browsers on desktop computers, to WebTV and similar devices permitting access through the television, to an ever-expanding range of mobile or portable Internet devices such as WAP, or wireless application protocol on Internet-enabled mobile phones, and handheld wireless Internet devices such as Blackberries and the like. The focus of this book is on Internet access on standard devices, using one of the regular browsers (Internet Explorer, Mozilla Firefox, Safari, and so on) and based primarily on hypertext markup language, or HTML, the common language of the Web.

Even given these restrictions, Web surveys can be designed many different ways. For example, survey questions can be designed to be presented in a separate browser window (pop-up surveys) or the main browser. They can be designed to contain a single question (e.g., "question of the day" polls) or many questions. The questions can be included in a single HTML form or distributed across many forms. They can be more or less complex. And so on. I will primarily focus on longer rather than shorter surveys (i.e., ones with multiple questions) and surveys that are the main focus of the browser (i.e., not pop-ups). This is not to say that

the design issues are not relevant for these other type of designs (they are), but simply that the context of the discussion in this chapter and throughout the book is on longer, more complex Web surveys.

Even within these limits of scope, there are variations that are likely to affect design. Two additional considerations are the target audience and whether the Web is the only mode of data collection or combined with one or more other modes. With respect to the target audience, or population of inference, Web surveys may be targeted to specialized groups or to broader targets, such as the general population. The likely experience with and knowledge of the Web is likely to vary, as is the type of equipment that respondents may possess. Similarly, Web surveys targeted at individual respondents answering on their own behalf have different design implications than those used in establishment or business surveys, where one or more respondents are answering on behalf of some larger entity. The nature of the task is likely to differ. Further, the design of a Web-only survey has different requirements than if the Web is only one part of a mixed-mode design. Ensuring comparability across the modes may place different constraints on the design. And this list can go on – there is an almost endless variety of Web surveys out there.

One particular design prescription is unlikely to be suitable for all of these. The design approach one adopts should be influenced by factors such as the target audience, the purpose of the survey, the content or topic of the questions, the importance of data quality, whether the Web is the sole medium of data collection, and so on. All these factors should be considered when choosing a particular design approach. In other words, there is no one type of Web survey, nor should there be one approach to the design of Web surveys.

1.1.2. How Web Surveys Work

Given that I have narrowed the focus to surveys consisting of several questions executed on a Web browser, we need a short aside on how the Web works – particularly with regard to HTML forms – to understand the limitation of the medium for designing and implementing surveys.

Figure 1.1 presents a rough schematic of how the Web works. The user – or respondent in our case – opens his or her Web browser by clicking on the link in the e-mail invitation or by typing in the uniform resource locator, or URL. This action sends a request via the Internet using hypertext transfer protocol (HTTP) to the Web server (i.e., the site identified by the URL) to deliver an HTML page to the respondent's browser (called the client). For most browsing activities on the

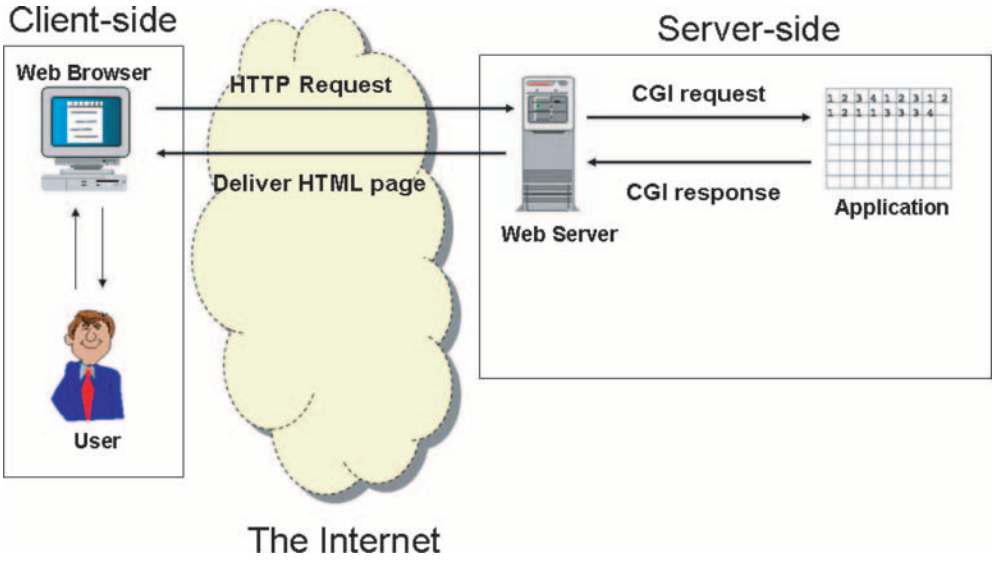

Figure 1.1 A Web Survey Schematic.

Web, this is basically the operation. Web surveys are a little more complex for two reasons: (1) data needs to be sent from the client to the server, and (2) the data need to be stored or processed in some way on the server.

For the first of these, HTML forms rather than pages are used. Forms are special types of Web pages that permit the input of information using a variety of form elements. We'll be looking at these in greater detail in Chapter 2.

To deal with the second issue, Common Gateway Interface (CGI) scripts are used. CGI is a standard for interfacing external applications with information servers, such as HTTP or Web servers. A plain HTML document that the Web server retrieves is static, which means it exists in a constant state: a text file that doesn't change. A CGI program, however, is executed in real time, so that it can output dynamic information. When the Web server receives the data from a form, it transfers this information to another application – for example, a database, a program, or a survey application – which acts on that information and sends a message back to the Web server regarding the next action to take.

A variety of different actions can occur, from simply storing the submitted information in a database, to processing complex algorithms to determine the completeness and consistency of responses, to generating new HTML forms dynamically based on the answers submitted. The CGI scripts and the survey application can

also perform survey management functions such as controlling access to the survey, verifying that the survey has not already been completed by that respondent, permitting respondents to resume where they left off, generating automatic e-mail invitations and reminders, and so on. Much of the action in a Web survey is thus on the server-side.

Another implication of this brief description is that Web surveys are not made of HTML alone. Knowing how to create a home page and how to post information on the Web is not the same as knowing how to conduct a survey. The CGI scripts and the application used to run the survey on the server are critical elements. CGI scripts can be written in several different programming languages – Perl is a common scripting language, but ColdFusion, ASP, PHP, or customized survey packages can all serve this function.

1.2. Scrolling versus Paging Designs

For surveys longer than a single screen – that is, excluding pop-up or single-question surveys – there is a continuum of design choices, ranging from scrolling designs where the entire survey is a single HTML form, to paging designs where each question is presented on a separate HTML form. Before we discuss the variety of design approaches along this continuum of scrolling and paging designs, we need to clear up some terminology.

The terms *screen*, *page*, and *form* are often used interchangeably when discussing Web survey design, but they have different meanings. A *screen* is the physical device on which the browser and its contents are displayed. In computer-assisted interviewing (CAI) surveys, where the notion of one question per screen originated, the display was constrained to be identical for all interviewers. This was certainly true of DOS-based instruments where a screen meant the same number of displayable characters for all users (no more than eighty columns by forty rows). But even in graphical user interface (GUI) environments (such as Windows or Mac OS), where scrolling and resizable windows made the notion of screen less relevant, screen size is still largely constrained to present the information in exactly the same way for all interviewers (in computer-assisted telephone interviewing [CATI] or computer-assisted personal interviewing [CAPI]) or respondents (in computer-assisted self-interviewing, or CASI). Given user control over font size and browser window size, not to mention the variety of screen resolutions (e.g., 600×800, 1024×768) available, the notion of screen has even less precise meaning for Web surveys. However, we use it to refer specifically to the information that

can (typically) be seen without the need for scrolling, given a typical or minimum set of browser settings. In other words, the term is used in terms of a design goal (visible without scrolling), even if not realized for every respondent and every browser setting. Couper, Traugott, and Lamias (2001), for example, use this notion of screen when talking about single-item versus multiple-item screens. Clark and Nyiri (2001) used the terms multiscreen and single-screen surveys to refer to paging versus scrolling designs, respectively. Dillman (2000) also uses the screen terminology.

The term *page* is used in two different ways in the survey design world. First, it is a holdover from the days of paper-based surveys and is used analogously to mean what can be seen on the screen at one time (in the same way that a single page of paper is visible at once). In the world of HTML, the word *page* is synonymous with document, and can be the size of a single screen or many screens. Most Web sites, for example, have a home page and many other pages or HTML documents that one can navigate to with the aid of hyperlinks.

In HTML, pages and *forms* are not exactly the same thing. A form is a specific type of page. While HTML pages are typically static, in that they are designed for the display of text and images, forms are interactive. Forms permit the submission of information *to* the Web server, as opposed to simply retrieving information *from* the server. Two features of HTML forms distinguish them from typical Web pages: (1) the fields or tools that permit the respondent to enter information or select responses and (2) the processing script that takes the information submitted and converts it into a usable format, that is, data. HTML forms are thus the principal tool for online surveys, which require a two-way flow of information between the respondent's browser and the Web server. E-commerce also makes widespread use of HTML forms – for most sites in which one has to submit some information (a name, address, credit card number, flight date and time, etc.) forms are most likely to be used. One way to distinguish a form from a page is that the former has one or more action buttons (*submit*, *reset*, *next*, etc.). Despite the difference between form and page, the terms are often used interchangeably. Thus, Lozar Manfreda and Vehovar (2002) use "page" to contrast one-page designs with multiple-page designs.

To summarize, I will attempt to limit the use of the term *screen* to what is visible within the browser window at any particular point, while using *page* and *form* to refer to different types of documents in HTML. While they have different meanings in HTML, I will use the terms synonymously.

In HTML terms, multiple-page designs are designed with a single question – or at most a small number of related questions – per form. More, specifically, part

of the reason for such a design is to obviate the need for scrolling, and designers typically constrain the dimensions of the form such that it is visible in its entirety on the screen, under most common browser settings. In practice, depending on the length of a particular question and the number of response options, such survey designs may well require scrolling for some of the items.

Thus, a single-form or single-page survey is at the opposite end of the continuum to a single-question-per-form survey. The former has one submit button for the entire survey; the latter has one for every question in the survey. I will refer to single-form surveys as scrolling surveys and multiple-form surveys as paging surveys, to denote the primary actions respondents take to navigate through the instrument.

Web surveys can thus have any number of questions per form, ranging from only one to all of the survey items. Similarly, a Web survey can comprise any number of forms, from a single one for the entire survey to a separate form for each item. These define endpoints of a continuum, and there are many reasonable design alternatives along this continuum.

1.2.1. Scrolling Survey Design

In the prototypical version of this approach, the entire questionnaire is contained in a single HTML form, with one or more action buttons (e.g., submit, cancel) at the very end. This approach is most akin to a paper-based self-administered survey, where skips and routing are not automated, respondents can browse back and forth through the instrument, answering questions in any order, changing answers at will, and answering all or some of the questions before submitting the response. Dillman is an advocate of this approach, noting that "in general, I prefer questionnaires that scroll from beginning to end, a method that most closely resembles the general experience of using the Web" (Dillman, 2000, p. 395).

From a technical standpoint, the entire questionnaire is loaded into the respondent's browser (client) at the outset, and there is no further interaction with the survey organization's machine (server) until an explicit action is taken – *send*, *submit*, or whatever the button may be called. Clicking on a radio button or check box or entering text in a field communicates no information to the database running the survey. This information remains in the respondent's browser until explicitly transmitted to the server and then to the database for further processing.

If a respondent answers all questions but closes the browser before submitting the responses, or presses the *cancel* or *clear* button in the survey (if offered), no

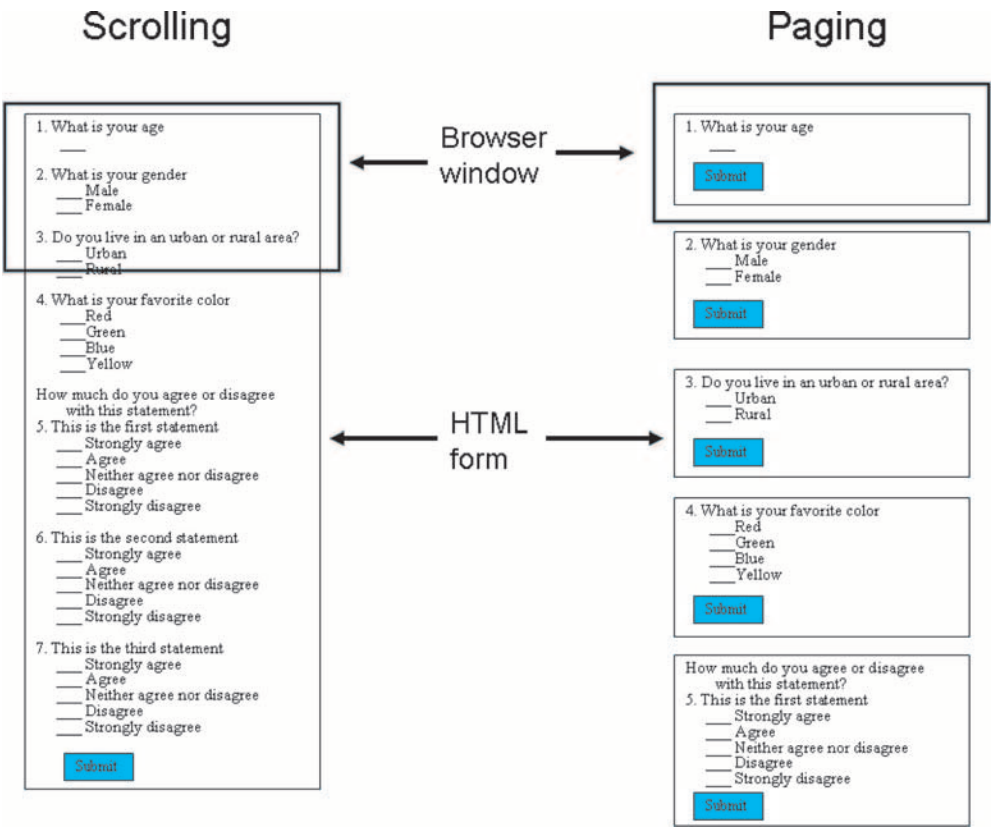

Figure 1.2 Scrolling versus Paging Designs.

information is transmitted to the server – in the same way as if a mail survey respondent were to complete the survey then fail to mail the questionnaire back to the survey organization.

There are several potential advantages of the scrolling approach to Web survey design. These include the following:

1. The design is similar to that of a paper survey, so it may be best for multimode studies in which one would want to minimize the possibility of differential measurement errors arising through mode differences.
2. Respondents can readily determine the length of the instrument and review forthcoming questions simply by scrolling to the end or noting the position and size of the scroll bar.

3. Similarly, respondents are free to move back and forth through the instrument and answer the questions in any order they prefer. This means they can back up and change answers to previous questions and can choose to omit questions if they so desire.
4. This approach is relatively easy to program. Further, given that standard or generic HTML is used throughout, this approach may be less susceptible to technical difficulties or browser compatibility problems.
5. Given that the instruments are simpler and no complex code is involved, the downloading of a static instrument may be marginally faster than a comparable interactive survey.
6. Given the above, there is less interaction with the server, reducing the likelihood of failures or errors during transmission.

However, there are also several potential disadvantages of using this design approach. Some of these are as follows:

1. The survey must be completed in a single sitting. One cannot do part of the survey, close the browser or disconnect from the Internet, and return to complete the rest later, unless additional programming has been done to accommodate this possibility.
2. All responses may be lost if the respondent fails to press the submit button after completing the survey.
3. Respondents see all questions, even those not yet answered and those that may not be relevant. This may lead respondents to choose the shortest path through the instrument as a way of satisficing (Krosnick, 1991). In other words, the respondent may be thinking, "If I answer 'yes' to this question, I have to answer the next 20 questions, but if I answer 'no' I get to skip ahead."
4. Skips and routing are not automated. As we know from research on paper surveys (Redline and Dillman, 2002), respondents are susceptible to both errors of omission and errors of commission when they have to make skip or flow decisions, and these are likely to occur in Web surveys too.
5. A variety of interactive design features are unavailable (unless embedded client-side scripts are used). This means that range checks, consistency checks, calculations, running totals, and other edits are not possible. Of course, these can be programmed to occur after submission of the form to the server, returning the results and accompanying messages to the respondent for follow-up. But it is usually best to provide this kind of feedback at