

IMPROVING ELECTRONIC DATA COLLECTION AND DISSEMINATION THROUGH USABILITY TESTING

Elizabeth Murphy, Kent Marquis, Richard Hoffman III,
Lelyn Saner, Heather Tedesco, Chanda Harris
U. S. Census Bureau
and
Renate Roske-Hofstrand¹
Computer Technology Associates

Abstract

Electronic modes of data collection and data dissemination pose challenges to instrument designers and user-interface designers (cf. Sweet & Russell, 1996). Just as ensuring the usability of a paper instrument requires cognitive testing, ensuring the usability of an electronic instrument requires usability testing. Both cognitive testing and usability testing have their roots in cognitive psychology and its findings on human thought processes. Whether the user is providing data or retrieving data, the relative usability of different user interfaces varies with the cognitive demands placed on the user, the level of consistency with user expectations, and the relative visibility of relationships between user actions and system behaviors. Examples of usability issues in a Web-based data-collection instrument and a Web-based data-dissemination tool illustrate obstacles to achieving the goal of user-centered design. Recommendations for resolving such obstacles are based on recent usability testing.

Introduction

Designers of automated, web-based instruments and tools can learn a basic lesson from the early days of office automation. Putting a paper form on-line raises issues associated with the electronic medium, e.g., how best to support the user's ease of moving within and between fields, whether to provide edit checking, how far to depart (if at all) from exactly duplicating the paper form (e.g., Mooers, 1983). As pointed out by House and Nicholls (1988), differences between paper questionnaires and electronic data-collection forms require specification of navigational paths and acceptable alternatives. Path specification by itself, however, is not sufficient to ensure that users will be able to find their way around. The role of usability testing is to identify problems that users are likely to have in accomplishing their task goals as they proceed through an electronic data-collection form or attempt to retrieve data using an electronic data-dissemination tool. This paper discusses usability testing as a way to enhance the user's performance as a data provider and as a data retriever. Examples are provided from recent usability testing conducted at the U. S. Census Bureau.

We first present some definitions of three terms that are used throughout the paper: **user**, **user interface** and **usability**.

¹ Formerly with the U.S. Census Bureau

Definitions. For the purposes of this discussion, the **user** of a data-collection instrument may be the interviewer or, in the case of computer self-administered questionnaires, the respondent. The user of an electronic data-dissemination tool may be a government employee, a private citizen, or anyone who wants to retrieve data using the tool.

For practical purposes, the **user interface** includes the displays and controls that facilitate communication between a computer and its user(s). Users accomplish the sensory-perceptual, cognitive, and behavioral aspects of their tasks by receiving output from the computer and by giving input to the computer through the user interface. Output received from the computer provides a basis for the judgment and decision-making aspects of users' tasks.

As defined by Dumas and Redish (1993), **'usability'** means that the *people who use the product* can do so *quickly and easily* to accomplish *their own tasks*" (p. 4, italics in original, bold added). As these authors point out, "usability is an attribute of every product," not just of user interfaces to software products (Dumas & Redish, 1993, p. 4). Usability is a characteristic of the product, not of the user, although usability may vary depending on the user's prior experience with similar products.

The usability of any product is a function of its compatibility with its users' physical and/or cognitive characteristics. Take a book, for example. What makes books so usable? For one thing, pages can be turned easily; text and graphics are laid out according to the reader's cultural expectations. It is easy to learn and remember how to use books. For example, readers do not have to keep re-learning how to turn pages. (That is not to say that ease of understanding technical content is necessarily high!)

By analogy, what makes a software user interface usable or unusable to the end user? It is usable if it assists users in performing their tasks easily and quickly; if it reflects an understanding of users' goals and tasks; and if the 'look and feel'² are consistent and predictable, thus easy to learn and easy to remember. Imagine if we had to relearn how to use a book every time we went to look up something in the encyclopedia or if each chapter in a book looked or behaved differently from all the others. This may seem ridiculous to contemplate, but it is the very kind of thing that makes software user interfaces less than usable.

From Nielsen's (1993) perspective, computer-system usability is a multi-dimensional construct combining the following elements (p. 26):

- Learnability – The system should be easy to learn so that the user can rapidly start getting some work done...
- Efficiency – The system should be efficient [and effective] to use, so that once the user has learned the system, a high level of productivity is possible.

² 'Look' refers to the visual appearance of the screens, and 'feel' refers to the relative smoothness of moving around within a set of screens.

- Memorability – The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- Errors [Accuracy] —The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.
- Satisfaction – The system should be pleasant to use...

Thus, usability is largely a function of the cognitive demands associated with learning to use an instrument or tool, with remembering how to use it, with comprehending the information presented, and with recovering from error. Usability problems arise, in part, because the design of the user interface imposes cognitive demands that people are unable or unwilling to meet. For example, the design may require the user to retain more information in limited short-term memory than the user is actually capable of retaining, i.e., more than seven to nine groupings or “chunks” of information for an unstressed user (Miller, 1956). If the user is under time pressure, however, the limits of short-term memory can narrow to three chunks (Broadbent, 1975; Moray, 1986; Newell & Simon, 1972; Simon, 1976). The load on short-term memory at any moment is critical to the user’s ability to think and reason (Gardiner & Christie, 1987).

When information is lost, i.e., when it becomes unavailable for further processing, errors are likely to occur. Because people want to minimize or limit the demands on their resources of attention and memory, a design that repeatedly increases those demands beyond the user’s preferred level of mental effort will be perceived as difficult and unpleasant to use. Thus, keeping track of one’s path through three or four web pages may be easy, but keeping track of where one is in ten or twelve web pages will probably be difficult.

What You Get Should Be What You Expect. Many of the problems associated with modern computer technology have been observed in the field and documented in the Usability Laboratory’s recent usability testing of data-collection and data-dissemination applications. Usability problems typically creep into the user interface because software designers and developers frequently focus on creating and enhancing particular technological features to the detriment of actual user needs. Users need interfaces to computer systems that behave according to their expectations, that is, user interfaces that are consistent with the mental models induced in the user (e.g., Norman, 1988). A user interface that violates users’ expectations will lead them down unproductive paths through the underlying data structure, creating frustration and workarounds to minimize use of the tool.

Recent Usability Testing at the U. S. Census Bureau

At the U. S. Census Bureau, some experimental user interfaces to support data collection and data dissemination have been through at least one cycle of usability testing, with

additional testing planned for the coming months. In cooperation with their respective project teams, these tests have been designed to evaluate the software's ability to support effective and efficient user performance and user satisfaction. Usability testing has contributed to faster overall task-completion times for test users of the Census Homepage (Roske-Hofstrand, 1999). In the data-collection arena, usability testing has identified new problems associated with the switch from paper-based to computer-based instruments. In the following sections, examples are drawn from two Census Bureau projects, one in data collection and the other in data dissemination, whose names are not mentioned to protect confidential information.

Usability Issues in Web-based Data Collection. Security features and browser idiosyncrasies present major obstacles to respondents' success in Web-based surveys and censuses (cf. Sweet & Russell, 1996). To ensure security and guarantee confidentiality, respondents must enter some form of identification before receiving the survey or census. Lengthy explanations and instructions about how to do this prevent the respondent from getting into the survey or census form quickly. Internet users, however, want to move around quickly, finish with one page, and move on to another. Asking them to read long instructions carefully is counter to their normal mode of operation on the Internet. Lengthy instructions presented in sentence and paragraph form impose heavy demands on short-term memory.

Another issue is that user expertise with computers and with the Internet varies widely in a Web-based data-collection environment. Since novices and experts may try to submit their Census information over the Web, instructions not present in the paper form are provided to help respondents access an experimental, electronic short form. The instructions include an alert that respondents received if certain software was not turned on. The following alert appeared on a pale green background in the test version of this on-line instrument:

IMPORTANT! – Your Web Browser does not have 'JavaScript' enabled!

Although JavaScript is not required to fill out an <electronic data-collection form> it is required if you need to access any hyper-linked help information while you are filling out the form. If your browser has JavaScript, you should enable it now and reload this page before you continue. If your browser is older and does not have JavaScript, you should manually open a separate browser window to the following help page address before you continue: 'http://www.<34 character web address>'

None of the test users who received this message knew what it meant or what they needed to do. One test user scrolled right by this message and, when asked about it later, said she used to have a job where she was told to ignore any such messages. She had no idea what JavaScript was or what the impact would be of not enabling it.

Let's consider the cognitive demands of the above alert: To understand this message, the user needs to know about technical aspects of Web browsers. Users need to know the meaning of having a feature enabled or disabled. They need to understand that they will have a difficult time accessing any help they might need once they get into the data-collection form if they do not enable the missing feature and reload the page. They have to know how to do the enabling and the reloading. They need to know whether their browser has the feature or whether they need to open a separate browser window to support access to help. They need to know how to manually open a separate browser window. Each of these high-level cognitive demands implies a constellation of interrelated knowledge structures developed over time by the software designer but completely lacking in the average user. Interpretation of unfamiliar information, such as that contained in the alert, places heavy demands on short-term memory (Gardiner & Christie, 1987).

To someone unfamiliar with the technical aspects of the Internet, the above alert is an extremely complex message that poses many obstacles to interpretation and comprehension. It is akin to the instructions for programming older VCRs that were so notoriously difficult for the average (non-technical) person to follow. Respondents simply want to get to the data-collection form and provide their information. However, the required actions must be performed before accessing the form because enabling and reloading cannot be done inside the form. Thus, those who ignore the message at this point will not be able to access help and will not understand why help is inaccessible once they do get into the form. A goal of usability testing is to identify otherwise hidden contingencies such as this so that they can be clarified and made apparent to the user.

The following revised version of the JavaScript alert will be tested in a second round with at least 100 users:

IMPORTANT! Your access to help is disabled or not available.

- If you think you will need to access any hyper-linked help information while you are filling out the form, you need to enable a feature called "JavaScript." If your browser has JavaScript, you should enable it now and reload this page before you continue. (JavaScript is usually enabled/disabled from within the 'options' or 'preferences' section of your browser's 'pull-down' menus.)
- After you have enabled JavaScript, 'Click Here' to go back to the <project Home Page> and start fresh. (This message should disappear.)
- If you cannot, or do not wish to enable JavaScript, you should still be able to obtain some of the help information provided with the <on-line form>, although access will be less automatic.

Although the revised version is admittedly longer than the original, it begins by explaining why the user should be concerned. Instead of using the unfamiliar technical term (JavaScript), it begins by saying in plain language that the user cannot access help information: "Your access to help is disabled or not

available.” This version of the alert introduces the user to JavaScript and demystifies what the user will need to do to activate this software. The bulleted format helps to break the text into a step-by-step process, reducing some of the demand on the non-technical user’s ability to hold unfamiliar information in mind. The amount of programming jargon is reduced, and more explanation is given for unfamiliar terms. Instead of being presented on a green background, which implies that everything is normal, the revised alert should appear on a yellow background. Within Western cultures, yellow carries with it implications that things may not be quite normal and that caution is warranted.

A navigational issue that surfaced during usability testing was that some users carried over their habit of using the tab key to move both within and between items. This presented a problem because the cursor was sometimes captured by hyperlinks that intervened between one item and another. In extreme cases, when the user tabbed between items, the cursor disappeared along with unfilled data-entry fields. It turned out that the developer’s model of navigation was “by scrolling only,” but this was in direct conflict with what many users wanted to do. Since it was not possible to change the data structure to allow tabbing, this issue was resolved by adding an instruction to users that they should navigate by scrolling. Ideally, however, users should be permitted to navigate by whichever means they prefer, tabbing or scrolling or a combination of both, within the limits of available methods.

The effectiveness of these changes will be evaluated in the second phase of usability testing on this Web-based data-collection instrument.

Usability Issues in Web-based Data Dissemination. If users of a data-dissemination Web site need to specify a geographic area of the United States as the basis for answering a question, the user interface should give them a way of selecting a single state or multiple states. User-interface developers know that check boxes support multiple selections and radio buttons support single selections. In this case, however, it did not occur to them that users would try to make multiple selections using radio buttons. So they used a pre-programmed widget to open up a list of all the states, with a radio button preceding each state name. This widget allows only selection of a single state.

But, say the user wants to select the District of Columbia, Maryland, and Virginia. Not understanding that the radio button convention is a single selection, the user opens the tree widget and clicks on the radio button for the District of Columbia. The radio button gets a little black dot in the center showing that selection. Now the user scrolls down to Maryland. The District of Columbia is no longer visible. The user clicks on the radio button for Maryland. It gets a little black dot showing that selection. Now the user scrolls down to Virginia. Maryland is no longer visible. The user clicks on the radio button for Virginia. It gets a little black dot showing that selection. The user thinks she has made all of her selections, but what has actually happened?

The District of Columbia was de-selected when the user clicked on Maryland, and Maryland was de-selected when the user clicked on Virginia. As a result, the readout of

selections shows only Virginia. The user says, “I can’t get the geography. I know I selected all of them,” and must switch to a different strategy (i.e., search). There is no message from the system to explain what has happened. Everything depends on the user knowing the conventional behavior of check boxes and radio buttons. In usability terms, this is a case of hiding system behavior from the user. Whenever something happens that is outside of the user’s view, e.g., due to scrolling, it should be reported to the user.

Although the behavior of checkboxes and radio buttons may be explained in an accompanying tutorial, usability testing has shown that every user who dutifully goes through the tutorial is not likely to remember the distinction when she encounters radio buttons for the first time in the actual user interface. This may be the case for several reasons. Perhaps the amount of information in the tutorial exceeds the user’s ability to internalize new material. More clearly, the information presented by tutorial is not immediately accessible when the user is working on trying to find the answer to a question. Any tutorial information that has made it from short-term memory into long-term memory has to be retrieved, and retrieval from memory has its own set of problems and obstacles (e.g., Baddeley, 1990).

A user-interface design that depends on the user’s retrieval of information from memory assumes that the information has been encoded in memory to begin with. It is quite common, however, for users to skip over instructions or descriptions, especially if they are in a full sentence/paragraph format. If the instructions or other information have not been perceived, they cannot have been read and understood, much less incorporated into a schema or mental model in long-term memory. If the user-interface design cannot succeed unless the user reads and understands all the verbiage presented on the screen, it is being set up for failure.

Many instances of non-perception occurred in our usability testing of a web-based data-dissemination tool. For example, instructions placed above a search field informed test users that they could legally enter only certain terms. However, it is quite clear from watching the videotapes of test users that many of them typed illegal terms in the search field without noticing or reading the instructions. They were surprised when their searches produced “no result”. Similarly, many test users overlooked the displayed instructions for how to build a complex query. Later, when the instructions were pointed out to them, these users said they had not seen or read the instructions.

Many users expect to be able to complete their tasks without reading instructions. They have a loosely defined, pre-existing mental model based on previous experience; and, if the user interface does not behave according to that model, they are stymied. One test participant, for example, was familiar with the look and feel of a CD-ROM lookup function. She described herself as “frustrated” by the look and feel of the tool being tested. There are no easy answers to these kinds of usability problems, but expecting users to be familiar with a particular sub-set of user-interface elements is expecting too much.

The same data-dissemination tool provides a step-by-step guide to users in the left frame of the visual display. However, the videotapes show that many users never looked at the left frame. They focused on the displayed content in the middle of their field of view. In many Web applications, a table of contents or similar, non-critical information is given in the left frame. Thus, there is no prior basis for users to have developed an expectation for finding task guidance in the left frame.

Another kind of usability problem does have an easy answer. This problem is the opposite of the user's not seeing something that is displayed: Nothing is displayed when something should be displayed. For example, in the data-dissemination tool being tested, users are able to call up maps at various levels of detail, from nationwide to census-tract level. One test user went to extreme lengths to confirm that she had, indeed, found the map that she wanted. Why? The map was not labeled with the name of the county that she had selected! There was some lower-level labeling but no county name. We have recommended labeling all maps with names corresponding to the names that users select from the list of geographic areas. Basic information of this kind should never be kept a secret from data users.

In Web-based data-dissemination environments, user expertise with computers and with Census data varies widely. In any case, the user's goal is to find the information she is looking for as quickly and easily as possible. If the information is hard to find using a particular data-dissemination tool, the user will become frustrated and will be less likely to use that tool again. Based on the usability team's analysis of navigation problems in testing of web-based applications, users are able to navigate appropriately to second and third level hyperlinked pages, but usability tests indicate that they encounter problems beyond the third level. Typically, they lose track of where they are and where they have been.

This finding on loss of orientation is consistent with long-standing, often replicated research finding that people do better with broad but shallow menu structures.³ Both our finding and the menu-depth finding are also consistent with the known limits of short-term, working memory, as discussed previously. A person simply cannot keep track of his or her train of thought while keeping in mind a successful path through the user interface, if more than three levels are involved. Since the human tendency is to limit or reduce the load on short-term, it stands to reason that users will get lost in cyberspace after the third level of drilling down looking for information.

Further usability testing is planned on a revised version of this Web-based data-dissemination tool. An objective of this testing is to determine whether the re-designed, complex query capability behaves according to experienced query builders' mental models of how it should behave.

³ For a detailed discussion of depth versus breadth, complete with mathematical models, see Norman, 1991.

Conclusions

Moving from paper-based to electronic media poses serious issues, both for survey methodologists and for user-interface designers. Just as cognitive testing has shown that survey items can be improved by removing obstacles to comprehension, usability testing shows that user-interface design can be improved by paying attention to human limitations and expectations. Cognitive demand can be effectively reduced by reducing demands on short-term memory and by designing user interfaces according to explicit models of system behavior that users can depend on. When the cognitive demand associated with the user interface is more reasonable, users are able to focus on their tasks of providing data in response to Census questions or finding just the right Census data to answer their own questions. These are the goals of user-centered data collection and dissemination.

References

- Baddeley, A. (1990). *Human memory*. Boston: Allyn and Bacon.
- Broadbent, D. E. (1975). The magical number seven after fifteen years. In A. Kennedy and A. Wilkes (Eds.), *Studies in long-term memory* (pp. 3-18). New York: Wiley.
- Dumas, J. S., & Redish, J. C. (1993). *A practical guide to usability testing*. Norwood, NJ: Ablex.
- Gardiner, M. M., & Christie, B. (1987). *Applying cognitive psychology to user-interface design*. New York: Wiley.
- House, C. C., & Nicholls, W. L. II. (1988). Questionnaire design for CATI: Design objectives and methods. In R. Groves et al. (Eds.), *Telephone survey methodology* (pp. 421-436). New York: Wiley.
- Miller, G. A. (1956). The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97.
- Mooers, C. D. (1983). Changes that users demanded in the human interface to the Hermes Message System. In A. Janda (Ed.), *CHI'83 conference proceedings: Human factors in computing systems* (pp. 88-92). New York: Association for Computing Machinery (ACM).
- Moray, N. (1986). Modelling cognitive activities: Human limitations in relation to computer aids. In E. Hollnagel, G. Mancini, and D. D. Woods (Eds.), *Intelligent decision support in process environments* (NATO Advanced Study Institute F21, pp. 273-291). Berlin: Springer-Verlag.
- Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Nielsen, J. (1993). *Usability engineering*. New York: Academic Press.
- Nielsen, J. (1997). Usability testing. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (pp. 1543-1568). New York: Wiley.
- Norman, D. A. (1988). *The design of everyday things*. New York: Doubleday.
- Norman, K. L. (1991). *The psychology of menu selection: Designing cognitive control of the human/computer interface*. Norwood, NJ: Ablex.

- Roske-Hofstrand, R. (1999). *How usability testing helped Census Homepage re-design*. Bureau of the Census Marketing Spotlight Program Presentation.
- Shneiderman, B. (1998). *Designing the user interface*. Reading, MA: Addison-Wesley.
- Simon, H. (1976). The information-storage system called human memory. In M. R. Rosenzweig and E. L. Bennett (Eds.), *Neural mechanisms of learning and memory* (pp. 79-96). Cambridge, MA: MIT Press.
- Sweet, E., & Russell, C. (1996). A discussion of data collection via the Internet. *American Statistical Association 1996 Proceedings of the Section on Survey Research Methods* (pp. 774-779).

Acknowledgements

Thanks to the reviewers who gave so generously of their time, effort, and wisdom:
Fred Conrad, Eileen O'Brien, and Mark Wallace.