Adapting Usability Test Methods to Improve a Data Extraction Tool *(FERRETT)*

John J. Bosley
Bureau of Labor Statistics
bosley_j@bls.gov

Cavan P. Capps
Bureau of the Census
cavan_paul_capps@ccMail.Census.GOV

ABSTRACT

This paper describes how web developers and a usability specialist collected usability data about *FERRETT*, a web-based statistical data extraction tool within extremely stringent constraints. The small development team lacked the time and other resources to implement fully the design changes suggested by extensive usability testing of the first *FERRETT* version and still perform additional evaluation to validate redesign. Instead they tested graphically redesigned screens representing design changes, using rudimentary hyperlink-based navigation functionality. They collected data from structured screen-centered dialogs with users representing major data user classes. Dialog content analysis revealed usability problems on which fine-tuning screen designs and fleshing out a complete navigation system for the redesigned site were based.

INTRODUCTION

For the past three years, the authors have collaborated in a series of laboratory and field studies of the usability of a Website, *FERRETT*, that is a joint product of their two agencies. *FERRETT* is an acronym for "*Federal Electronic Research, Review, Extraction & Tabulation Tool."* The second author leads a small programming staff who designed and now maintains *FERRETT*. The first author is a usability specialist who supports these developers as they modify and improve the site. This powerful tool (http://ferret.bls.census.gov/cgi-bin/ferret) enables users to locate, extract, download and/or perform limited on-line manipulations of statistical information drawn from any one of several survey datasets. Currently, *FERRETT* users can work with datasets from Census, Bureau of Labor Statistics, and the National Center for Health Statistics surveys.

The major functions implemented in *FERRETT* are:

- Shows a list of all federal surveys and their parts accessible through *FERRETT* and enables users to select from the list.
- Shows a list of names of all variables in the selected survey(s), and supports selection of desired data items from this list.
- If users want, shows descriptive information about data (metadata) in order to help users grasp its meaning and usefulness.
- Lets users redefine data item content on-line, for example, by restricting or grouping a continuous range of numeric values for an item.
- Lets users download selected data for client-side analysis in a variety of formats. Also lets users perform limited on-line analyses, for example, cross-tabulations.

The first version used CGI technology to implement these functions. This resulted in relatively slow performance and gave the appearance of inflexibility. It also presented the tool to the user as a series of discrete micro-tasks so that it was hard to perceive *FERRETT* as a unified tool. With the advent of Java as a programming language, the development team planned a *FERRET* interface redesign in Java. Unlike the CGI version, the *FERRETT* interface design could simultaneously support a number of data specification tasks. This avoided the time-distributed task segmentation of the first version and promised greatly improved usability.

A Usability Testing Dilemma

Replacing the CGI-based *FERRETT* interface, where graphic design was only marginally important with a Java interface capable of much more versatile use of graphic elements to support direct manipulation meant that the design process was virtually starting fresh. For this reason, the authors chose not to conduct formal usability tests of that interface, but instead subjected it to expert usability inspection, observed users in group training sessions, and scanned problems reported in voluntary comments by email. These sources of information indicated several serious usability problems with the CGI version. These included:

- Query building was tedious; using the tool forced users to work with nine separate interfaces, one after the other.

- Comprehensive instructions for end-to-end *FERRETT* use appeared on just the screen following log-on. These disappeared after that screen had been bypassed and it was not possible to retrieve them without going back to the earlier screen.
- Some of these interfaces were simple, allowing only one action. Others enabled several actions, but the user was not always alerted to that fact "above the fold," so it was possible to miss viewing controls by failing to scroll, for example.
- Navigation mixed use of on-screen custom controls with dependence on use of browser controls such as "Back" in an inconsistent manner.
- Instructions and "help" were minimal and often difficult to follow; for example, the output specifications assumed familiarity with the proprietary language of the SAS data analysis package.
- Some users were disoriented by the strictly sequential task structure. Information about actions already accomplished on earlier interfaces was not always carried over fully to subsequent ones.

The team recognized adopting a more graphics-based interface approach would provide an opportunity to solve many of these problems, and making the problems explicit provided some general framework for getting started on the redesign.

The *FERRETT* development team has internalized an appreciation of usability test findings as a key source of design guidance. However, the team is small, and while it had already committed substantial resources to writing the underlying Java code that would support "back end" data location and extraction operations, it did not have the resources to build fully functional graphical interfaces for the redesigned product for realistic usability tests. This made it necessary to adapt conventional usability test methods for use with interfaces that lacked almost all responsiveness to direct manipulation. The developers used graphics software to create bitmap interface mockups for this round of testing. Each interface image had no active elements except one (or two) hyperlinks that took the user to another interface simulating controls for a different subset of *FERRETT* functions. These screen representations incorporated graphics of controls that could eventually be "hooked" to the underlying Java software and enable users to carry out data location, extraction, manipulation and if desired, downloading.

Methodology

The "cognitive walkthrough" usability testing approach was adapted to this resource-poor environment. The minimal functionality of the interface mockups demanded a more intrusive approach, where the test administrator probed fairly extensively and participants were encouraged to describe expectations of control manipulations rather than commenting on actual interaction outcomes.

Figure 1: Bitmap of First Screen Redesign: " Select Survey and Variables"

This process of "structured speculation" is obviously prone to eliciting highly variable user responses, and another adaptation to conventional methodology was made in an effort to compensate for this weakness. This modification involved spreading the test sessions over

Here are some of the structured questions used to elicit test participant perceptions and interpretation of the new interface design:

> "This page is designed to let you select a particular survey to work with, as well as pick variables from that survey. First, study this page thoroughly and comment on any features—terminology, the written material in windows, etc.—that you find hard to understand or interpret. Pay particular attention to the text in the large window to the right and tell me how well it communicates to you how to use this page. DON'T click on anything just yet.
>
> Now, how would you find and select the most recent monthly data for the Basic CPS survey, using a feature or combination of features that you see here. What would you expect to happen when you made that selection? How would you expect the data to be presented?
>
> Any other comments?"

The person conducting the test session would then click on a hyperlink that would bring up a screen on which the actual result of selecting a particular survey would produce in the fully operational version. This screen replaced the instructions in the large grayed window on the right with a listing of all the variables included in the Basic CPS survey, in a columnar format according to the headings shown—"In Survey," "Variable Labels," etc. The participant was asked to comment on this result; specifically, was it consistent with expectations and if not, how it differed from those expectations, or what was expected instead of what appeared.

After similar structured dialog about each new screen's content, the test conductor would navigate to the next screen (or show the participant where a hyperlink was placed and direct the participant to move forward) and the process would be repeated.

Nine volunteers, all but two of whom were BLS employees thoroughly familiar with much of the data accessible through *FERRETT* (the remaining two being graduate students in a local university economics department), took part in individual test sessions using the mockups of the redesigned interface. The results were examined for relatively clear indications of usability problems, particularly serious ones. The developers used these "findings" to make changes to the interface mockups, which were then tested in a second round of individual sessions to see if usability improved. Four BLS professionals, one policy researcher from a local non-profit institute, a journalist and a sociology graduate student participated in this round. These sessions were videotaped for review.

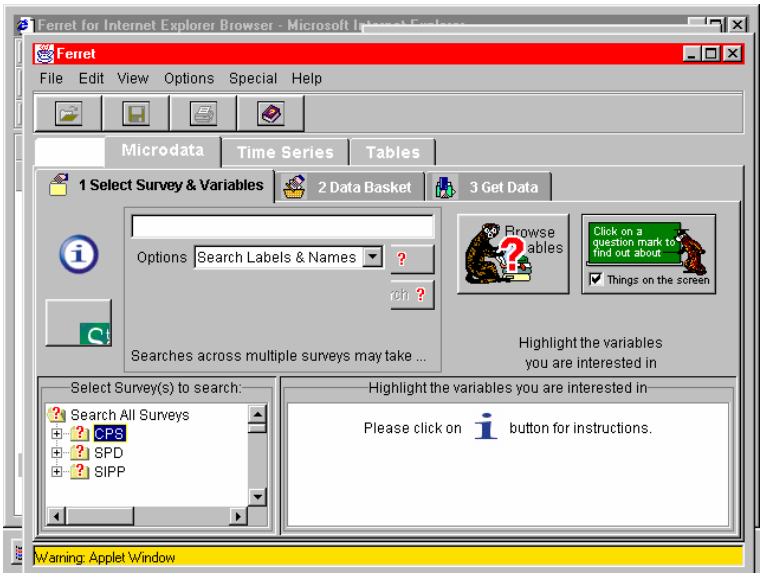
Results

This gradual "bootstrapping" approach placed minimal demands on the over-stretched development resource pool, and yet it tentatively validated usability gains through slow but steady interface design improvements. The resulting screen design that was "frozen" in order to link its features to the underlying software and make it fully functional contained many unresolved usability problems, to be sure. However, the cumulative findings of several iterations of small, structured usability tests and accompanying incremental design improvements helped avoid "show-stopping" usability problems in the first fully-functional redesigned interface.

The redesigned screen in Figure 2 shows a number of usability enhancements to meet deficiencies the testing uncovered. This functional interface is implemented as a Java applet. The improvement include:

- Implementation of all of *FERRETT"*s functions on just three multi-purpose screens.
- All navigation enabled by interface controls so there is no need to use browser controls at all.
- Tabs guide but do not constrain the most common task performance sequence.
- Context-appropriate guidance is continuously available in pop-up windows.
- Interface provides more capability to change selections or even start over without the need to navigate back to a fixed starting point.

Figure 2: First Screen of Current Beta Version of *FERRETT*



Usability testing and further enhancement is continuing. Meantime, the Java version of *FERRETT* is web-accessible as a beta version at http://ferret.bls.census.gov/java/msie.htm for Internet Explorer.

Summary

This paper has been about scaling back conventional usability test methods to accommodate time and resource constraints. How well did this work in practice, and what was learned? Here are some important summary observations.

- Give priority to implementing the function(s) that represent the core of the major task(s) the interface must support. In this case, navigation from one screen to the next was the area where improvement was needed most. Therefore this was the focus of the limited functionality built into the screen mock-ups.
- Work through relatively simple tasks. Don't try to have users "imagine" how they would tackle complex tasks. Such tasks may impose cognitive burdens that are unrealistic in that the final interface will provide greater support. It's better to do many repetitions of a variety of simple tasks than to spend time trying to "talk the user through" a complicated procedure.
- Explore thoroughly the participants' understandings of both verbal and graphical cues on the screen. This exploration should focus on expected results of an action—what do they expect the system to do in response to, say, a button click?
- Where there are misunderstandings, probe to determine whether it would be preferable to change the text, such as a button label, the kind of widget used—radio button vs. check-box, or whether some supplementary information ("help") appears to be needed.
- Where expectations don't match intended function, that is, where errors are likely, probe to explore user concepts of how to recover from the error. If they don't have a clue, this can highlight badly needed new design features.

By following simple rules such as these, a surprising amount can be learned about prototypical designs. This knowledge can enable developers to design a full-function interface with greater confidence that it will work reasonably well.