On the Contributions of Different Empirical Data in Usability Testing

Maria R. Ebling †
IBM T. J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598 USA
914-784-7949
ebling@us.ibm.com

Bonnie E. John
HCI Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
412-268-7182
bej@cs.cmu.edu

ABSTRACT
Many sources of empirical data can be used to evaluate an interface (e.g., time to learn, time to perform benchmark tasks, number of errors on benchmark tasks, answers on questionnaires, comments made in verbal protocols). This paper examines the relative contributions of both quantitative and qualitative data gathered during a usability study. For each usability problem uncovered by this study, we trace each contributing piece of evidence back to its empirical source. For this usability study, the verbal protocol provided the sole source of evidence for more than one third of the most severe problems and more than two thirds of the less severe problems. Thus, although the verbal protocol provided the bulk of the evidence, other sources of data contributed disproportionately to the more critical problems. This work suggests that further research is required to determine the relative value of different forms of empirical evidence.

Keywords
Usability testing, empirical data, verbal protocol.

1. INTRODUCTION
The design cycle of a user interface consists of two main phases: generation and evaluation. This second phase, evaluation, is critical to the success of a design. Evaluation identifies areas of a design that need refinement. To be effective, evaluation cannot simply answer with a "yes" or "no" (e.g., "the interface is not usable"), but must provide detailed information about why the design does not work as anticipated or, at least, what problems users experience. In addition, evaluation must allow the problems identified to be prioritized for the purely practical reason that available time and resources will almost certainly limit the revision phase.

† This work was performed while a graduate student in the Computer Science Department at Carnegie Mellon University.

DIS '00, Brooklyn, New York.

This paper concerns the evaluation phase; more specifically, it examines the contributions of different sources of data collected during an empirical usability test. During such a test, many sources of empirical data can be used in the evaluation. These sources include quantitative data (such as learning time, number of errors, number of steps required) and qualitative data (such as questionnaires and verbal protocol [6,16,18,19]). This paper compares the relative contributions of data sources to empirical testing and specifically does not compare the relative merits of other forms of usability analysis such as cognitive walkthroughs and heuristic evaluations. Further, not all data sources apply to all situations (e.g., a think-aloud protocol is clearly incompatible with a speech interface).

Other studies have examined the relative value of *analytic* versus empirical data [2,3,10,12,13,14,21] (though some of their methods have been criticized [8]). No studies to our knowledge, however, have examined the relative value of the different types of *empirical* data. This is an important question because effort must be spent to collect and analyze any sort of data, resources are always limited, and choices as to what to collect must often be made. For example, the system will have to be instrumented to collect the users' actions and responses for quantitative performance data (up-front effort before a usability test) and the answers to free-form questionnaires will have to be read to gain the information they hold (analysis effort after the usability test). Which effort will pay off the most? This paper begins to explore this issue and highlights the need for further research in this area.

2. METHOD

2.1 Software Evaluated
We evaluated a graphical interface written to support users of the Coda File System, a distributed file system that supports high availability through the use of both disconnected and weakly connected operation [15,17,22]. *Disconnected operation* allows users to disconnect from the network, to continue to use files from the distributed file system, and to reconnect to the network transparently. Disconnections can occur either voluntarily at the user's request (e.g., when a user takes a laptop on a trip) or involuntarily as a result of a network or server failure. *Weakly connected operation* allows users to exploit low bandwidth, high latency, expensive, and/or intermittent network connections to propagate changes to and from the shared file system. Experience with the deployed system revealed that, although experienced

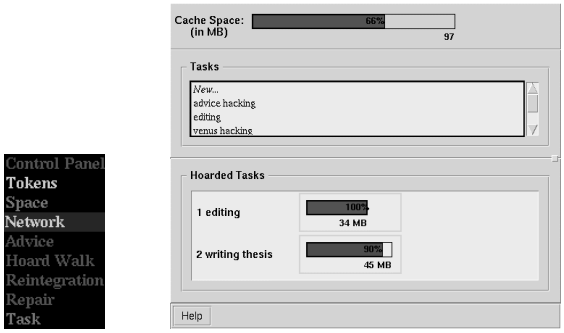
users found the functionality extremely useful, they continued to be confused by the system's behavior even after months of daily use. To address these problems and to offer users more control over network usage during periods of weak connectivity, we built a graphical user interface, called the CodaConsole [5]. It was this interface that we evaluated in this usability test.

Two pieces of the CodaConsole interface are shown in Figure 1. View (a) shows the indicator lights. The color of the indicator lights summarizes the status of the subsystem. For example, the two yellow lights, *Tokens* and *Network*, indicate that the user's tokens have expired (but no process is currently waiting for tokens) and that the system is operating weakly connected. The two red lights, *Space* and *Task*, indicate that the system is low on space and that at least one task is not available for use during disconnected operation. When the user double-clicks on the *Task* indicator light, the interface will display the Task Information screen shown in view (b). The top section of this window shows the current state of the cache space. The middle section presents a list of all tasks the user has defined. The bottom section shows which tasks the user has hoarded. For each hoarded task, the window shows its current priority (1 being most important) as well as its current availability. The availability of a task is presented in the form of a meter showing the percentage available: the color of the gauge indicates whether it is currently available (green) or not (red).

2.2 Participants

We recruited participants from among graduate students in the computer science department at Carnegie Mellon University, the same population of users who had experienced problems using the original deployment of Coda. We defined a *novice* user as a person who knew little about Coda and had no direct experience with it; we defined an *experienced* user as a person who had a Coda laptop and had operated disconnected for substantial periods of time over the course of at least a year. We observed three novice users and three experienced users.

2.3 Procedure



(a) (b)

Figure 1: This figure shows two windows of the CodaConsole interface. View (a) shows the indicator lights. View (b) shows the window that appears after the *Task* indicator light is double-clicked.

Each participant began his or her session by completing a background survey. This survey included demographic data as well as usage information regarding UNIX®-based operating systems, AFS, Coda, and the Windows® 95 Briefcase. It covered a total of nine topics and contained 32 individual questions. Most questions were category questions offering between 2 and 8 responses. Many questions included an "other" category as well as a space for them to specify that "other". A few questions asked for further details if the participant responded in the affirmative.

Participants then learned how to provide a verbal protocol. Participants listened to the experimenter giving a verbal protocol while playing the computer game Klondike. Then, the participant practiced giving a verbal protocol while playing Othello on the computer. If, after a few minutes of practice, the participant was thinking aloud and providing detailed reports about their activities and thoughts, the experimenter told the participant that this was exactly what she wanted to see. If the participant was not verbalizing these thoughts, the experimenter waited until the participant appeared to be thinking and then asked the participant what they were thinking to elicit a protocol. After they responded, the experimenter explained that that was the sort of information she was looking for. No subjects needed more than a couple of prompts. During the actual experiment, the experimenter used the standard prompt: "Please keep talking."

Each participant then learned about the interface through an on-line Tutorial that consisted of 62 individual screens and required approximately one hour. After completing the Tutorial and taking a break, participants proceeded to the Exercise component of the test. Each participant carried out 26 exercises, which we divided into three phases. During Phase I, participants were asked nine basic questions regarding the state of the system. During Phase II, participants completed five realistic tasks using the interface. During Phase III, participants diagnosed twelve realistic problems indicated by the interface.

When participants finished the Exercises portion, they then filled out a brief evaluation questionnaire. This questionnaire covered six topics and contained a total of 21 individual questions. Of these, 9 questions used a 5-choice category scale, 2 questions used a 3-choice category scale, 4 were yes-no questions, and 6 were open-ended questions. All category questions allowed the participant to add additional comments. After completing the questionnaire, participants were given the opportunity to voice further opinions and ask questions of the experimenter.

During the course of the Tutorial and Exercises, the first author observed the participants' progress on monitors in a separate room. An audio recording captured the verbal protocol; videotapes recorded the user's activities as well.

2.4 Materials

The questionnaires used in this study were designed after consulting an expert and reviewing the recommendations of survey designers [4, 7]. The version of the survey used during the test represents the result of much iteration and incorporates the suggestions of many individuals.

The Tutorial was designed specifically for the usability test, but the introductory material contained information that had been explained to many new Coda users less formally over the years.

The Tutorial formalized this instruction and added information specific to the interface under study.

An experienced Coda user designed the Exercises with realistic tasks in mind. The exercises cover issues that troubled new and experienced Coda users alike.

2.5 Measurements

For each participant, we recorded the

- time spent on each screen of the Tutorial,
- time required to complete each exercise,
- response to each exercise (e.g., written answer or change in state of interface),
- actions performed on each exercise,
- response to each item on the evaluation survey, and
- verbal protocol recorded during the Tutorial, Exercises and Debriefing.

2.6 Data Analysis

These data were used to identify usability problems in a two-step process. This process is illustrated in Figure 2. The first step analyzes the raw data to identify evidence of usability problems. It is important to note that each source of data may require a different analysis and that not every datum becomes a piece of evidence. The second step analyzes the resulting evidence to identify actual usability problems. Once again, not all pieces of evidence will result in a usability problem. In this section, we describe each analysis and the criteria we used to identify evidence. In the following section, we will discuss the results of these analyses.

We wanted to evaluate the ease with which participants were able to comprehend the material contained in the Tutorial. Many screens of the Tutorial required users not only to read some instructional material, but also to perform some actions. Thus, the time spent on each screen was a combination of comprehending

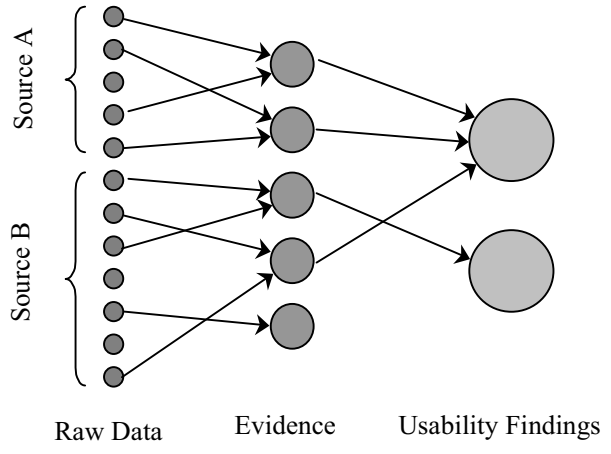


Figure 2: This diagram shows how raw data are analyzed to identify evidence and how this evidence converges into usability findings. Note that not all raw data become evidence and not all evidence becomes a usability finding.

the new material and performing some simple actions. Because all participants were experienced computer users, we assumed they were experts in using a mouse and in typing at the keyboard. To remove the time spent performing the actions (e.g., clicking and typing) required by each screen of the Tutorial, we subtracted the time a Keystroke-Level Model [1] predicted would be required to perform the various actions from the time each user spent on that screen. The time left for each screen now approximates pure comprehension time. To normalize for different length screens, we used *reading rate* as the primary measure instead of absolute time on each screen. To mitigate the effects of individual variability in reading rates, we calculated each participant's average reading rate and standard deviation over all screens of the Tutorial. We then compared the reading rate exhibited on each individual screen to this average. We identified a *difficult* screen as one for which the participant's reading rate was at least one standard deviation below their average. If only one person found a screen difficult, we did not record this datum as evidence because the slowness in reading rate might have resulted from a temporary attention lapse, a sneeze, or some other non-repeatable factor unrelated to usability. However, if two or more participants found the same screen difficult, we recorded evidence that participants found the topic covered in that screen difficult to understand.

Next, we evaluated the participants' ability to perform tasks required of Coda users. We compared each participant's response(s) to the Exercises to the correct response(s) for each exercise. We recorded an observation as evidence of a potential problem if two or more users answered an exercise incorrectly, reasoning again that if only one person made an error it may have been for reasons not related to usability.

We were also interested in the efficiency with which participants performed the tasks required of Coda users. We compared the number of actions each participant performed to respond to each exercise with *par*—the number of actions required by an experienced user (as in the game of golf). Par showed what we believed to be the minimum set of actions required to obtain all the relevant information to answer the questions in the order they were presented. Participants were said to have answered *inefficiently* if they took more than two times par actions to respond to an exercise. As with the previous measures, evidence was recorded if two or more users answered an exercise inefficiently.

We then examined the participants' responses to the evaluation questionnaire. All negative comments, as judged by the first author, were recorded as evidence. We reasoned that if something about the system was salient enough for a user to remember and comment on in this late phase of the test, it was probably important enough to be considered evidence of a usability problem.

Finally, we transcribed the verbal protocol recorded during the usability test. Our transcripts captured all but those utterances during which the participant read verbatim from a screen of the Tutorial or from an exercise description. Although we transcribed from the audiotapes, we referred to the videotape to clarify the participant's activities at the point of the utterance when necessary. We defined evidence as a notable utterance made by one or more users, such as expressions of surprise or frustration and indications that users were deviating from the expected path. We assigned each piece of evidence a unique identifier and

recorded it next to the appropriate utterance(s) in the transcripts. To assist in the task of identifying similar comments made by other users or at other times, we created a hierarchical listing of the comment evidence. Our hierarchy, illustrated in Table 1, was necessarily based upon our interface. The interface was organized around a set of indicator lights (Control Panel, Tokens, Space, Network, Advice, Hoard Walk, and Task) so our hierarchy contained a category for each indicator light. We added a general category to cover topics related to the interface that did not fall under any indicator. We had a final category for problems identified in the materials used in the study. If an observation applied to multiple areas of the interface or multiple categories, it was reproduced in all applicable areas. This hierarchical listing helped to focus our consideration on the subset of comment evidence within each category, rather than trying to sift through all the comment evidence at every point. We built this hierarchy part way through the analysis when the number of unique comments grew beyond about 75. During our analysis, we examined the transcripts more than once. We found that having seen a detailed comment in one pass helped us to identify a similar, but more subtle, comment on a subsequent pass.

Table 1: This table shows the hierarchy we used to sort the evidence from the four sources of empirical data.

- **Indicator Lights**
  - Notification of Events
    - General
    - Specific
- **Control Panel**
  - General
    - Commit Behavior
  - Event Configuration
  - Urgency Colors
- **Tokens**
- **Space**
- **Network**
- **Advice**
- **Hoard Walk**
- **Task**
  - General
    - Lost Data
    - Save Behavior
  - Task Information
  - Definitions
    - Task Definition
    - Data Definition
    - Program Definition
- **General**
  - Bugs
    - Interface
    - Widget Library
    - Coda
  - Cosmetic Problems
    - Global
    - Local
  - Enhancements
  - Help
  - Meters
  - Terminology
  - Widget Library
  - Windows
- **Study Materials**
  - General
  - Tutorial
    - Clarifications
    - Bugs
    - Omissions
  - Exercises
    - Clarifications
    - Bugs
    - Inconsistencies

2.7 Results

This section presents a summary of the evidence we gathered from the analyses described in the preceding section. We provide examples to give a flavor of the sort of evidence we collected from each analysis.

Our analysis of the timing data from the Tutorial revealed thirteen pieces of evidence. For example, we observed that five participants found the screen describing event configurations difficult (Tutorial Result #12, [5]).

Our analysis of the errors participants made resulted in three pieces of evidence. We noted that five users provided incorrect hoard walk advice (Exercise Result #1), that three users could not accurately describe the current space status (Exercise Result #2), and that five users failed to completely define the "fixing bugs" task (Exercise Result #3) [5].

Our analysis of the participants' efficiency in performing tasks using the interface resulted in 11 pieces of evidence. For example, we found that two users had difficulty finding the events to be configured (Exercise Result #5, [5]). For this par 3 exercise, one user required 7 actions and the other required 15.

The evaluation questionnaire provided 23 pieces of evidence. For instance, it revealed that users had difficulty finding events in the Event Configuration tab of the Control Panel (Questionnaire Result #5 [5]). One participant (N1) stated that "some event[s] were not where I expected them"; debriefing revealed that this participant was referring to the difficulty in finding events by their indicator in this window of the interface.

Our analysis of the transcripts of the verbal protocol identified more than 150 pieces of evidence. For example, we observed that many users could not immediately find the location of events in the event configuration tab of the Control Panel (Comment Evidence #17, [5]). This comment is exemplified by a segment of protocol made by one participant (N2) during the Exercise segment of the study (annotations appear in square brackets):

"So I brought up Control Panel to look at the events. [*This was the correct action to take.*]
Uhm. So, ah, 'Weakly Connected Cache Miss Advice' [*User is reading the specified event from the exercise description.*]
Is that all? I think that's Network stuff. [*User selects the Network indicator, the third element in the indicator list and suggested by the phrase "Weakly Connected".*]
No. [*User looked at the events listed for this indicator, but did not see the specified event.*]
So what else? [*User looks at the list of indicators.*]
Space? [*User selects the Space indicator, the second element in the list and suggested by the phrase "Cache Miss". User doesn't find the specified event in the events listed for the Space indicator.*]
Advice. [*User selects the Advice indicator, the fourth element of the list and suggested by the final word of the event name. User looks at the list of events.*]
Oh OK. [*User finds the specified event.*]
Makes sense."

2.8 Usability Problems

Using our analyses of the raw data, we looked for a convergence of evidence suggesting a usability problem. For example, some observations used in the previous section (Tutorial Result #12,

Exercise Result #5, Questionnaire Result #5, and Comment Evidence #17) and two other pieces of Comment Evidence point to a problem with a particular window of the interface. This problem was documented in a format similar to Dumas and Redish [4] and is shown in Figure 3.

To assist in the task of finding a convergence of evidence, we expanded our hierarchical listing of the comment evidence to include evidence from other origins. As before, if an observation applied to multiple areas of the interface or multiple categories, it was reproduced in all applicable areas. Once again, this hierarchical listing helped to focus our consideration on the subset of evidence within each category, rather than trying to sift through all the evidence at every point. This technique worked particularly well for finding problems associated with a single screen of the interface. To further identify problems associated with multiple screens, after identifying a particular usability problem, we considered whether it applied only to the current window or whether it applied to other windows as well.

We identified a usability problem if

- two or more sources of evidence suggest a problem,
- two or more users suggest a problem,
- one or more users crash the interface,
- one or more users identify a bug in the interface, or
- one user's evidence suggests a problem and the authors concur.

For the last bullet listed above (evidence from a single user unrelated to a system crash or obvious bug), the authors rated each piece of evidence independently (17 ratings in total), resolved disagreements by consensus, and agreed that 10 problems should be recorded. From the data collected from these six users, we identified a total of 65 separate usability problems.

2.9 Classification of Problems

We categorized each usability problem according to its *scope* and *severity* [4]. The scope of the problem indicates how widespread that problem is. A problem with *local* scope is limited to a single window of the interface. One with *global* scope applies to multiple windows.

The severity of a problem indicates how critical the problem is. Dumas and Redish [4] use a four point scale in which the first severity level represents the most severe problems and the last severity level represents the least severe problems. These levels are described as follows:

[1: Prevents Task] Prevents completion of task

[2: Significant Delay] Causes significant delay or frustration

[3: Minor Effect] Presents a minor effect on usability

[4: Suggestion] Suggests a potential enhancement

More specifically, if even a single participant failed to complete a task due to the given problem, we classified the problem as Severity Level 1 (Prevents Task). If a single user experienced significant delay or frustration because of a problem but eventually completed the task, then we classified the problem as Severity Level 2 (Significant Delay). Problems that did not cause significant delay or frustration fell into Severity Level 3 (Minor Effect). Suggestions for additional functionality or further information were classified as Severity Level 4 (Suggestions).

Our biggest classification challenge lay at the boundary between Severity Levels 1 (Prevents Task) and 2 (Significant Delay); a few problems did not clearly fall into either category. Therefore, all problems that caused the interface to crash, that revealed a serious

Problem #4: Event Configuration Tab of Control Panel layout inappropriate to task.

Evidence:
Tutorial Result IDs: 12
Exercise Result IDs: 5
Questionnaire Result IDs: 5
Comment Evidence IDs: 17, 162, 206

Scope: Local
Severity: 1
Number of Participants: 6

Explanation: In order for users to modify the configuration of a particular event, they must first know under what indicator that event is notified. For instance, to change the configuration of the *Task Unavailable* event, users must first select the *Task* indicator from the indicator listbox of the Event Configuration tab of the Control Panel. Then, they can select the *Task Unavailable* event from the event listbox. In this example, the decision to look under the *Task* indicator is relatively straightforward. However, if the event is *Weakly Connected Cache Miss Advice*, users could reasonably look under *Network*, *Space*, or *Advice*, as happened in Comment Evidence #17.

Recommendations: [The following recommendation assumes background knowledge of the CodaConsole interface.] The Event Configuration tab was redesigned during pilot testing. The original design listed all events under a single listbox. One of the early pilot users pointed out that the Event Configuration should include the indicator on which an event will be notified. The remaining users in the study used the redesigned Event Configuration tab. Unfortunately, the redesign introduced the dependency between the indicator and the event. By the end of the pilot testing, it was clear that this was a problem, but the experimenter was reluctant to make further changes so late in pilot testing. The Event Configuration tab needs to be redesigned to address the original suggestion while eliminating the need to know event/indicator pairings. The recommendation is to consider a design that contains a single list of events and a copy of the indicator lights.

Figure 3: This figure shows a sample usability problem [5]. Each problem documents the observations that identified the problem, explains the problem, and recommends an approach to addressing the problem.

omission or bug in the interface (such as one that could easily prevent a participant from completing a task), or that contributed to a participant responding incorrectly to an exercise were also classified as having prevented the task. In general, we found the distinction between these first two severity levels to be problematic. A number of serious problems appear as Severity Level 2 (Significant Delay) by this definition because users did eventually complete the task correctly; however, some of these problems appeared far more frequently than other problems that actually prevented the task because they crashed the interface. Table 2 summarizes these results.

Table 2: Each problem is categorized by its scope and severity [4]. Scope is a measure of how widespread a problem is; severity is a measure of how critical a problem is considered to be.

Severity	Scope	
	Global	Local
1: Prevents Task	1	8
2: Significant Delay	4	6
3: Minor Effect	11	16
4: Suggestion	2	17

3. DISCUSSION

While analyzing the data, we realized that the verbal protocol provided the bulk of the evidence. This observation led us to ask three questions:

1. How did each source of evidence contribute to the usability findings?
2. Did problem severity influence this contribution?
3. How much investment did each source of data require?

To explore these questions, we analyzed the origins of the evidence for each of the 65 problems. Table 3 shows this analysis in tabular form indicating the number of problems of each class identified from the various sources of data. This table reveals that the transcripts of the verbal protocol contributed the bulk of the evidence. Of the 65 problems, the verbal protocol provided evidence for 61 of them. Furthermore, 40 of these problems could be identified only in the verbal protocol data, with none of the other sources contributing.

Table 3: This table shows the number of problems of each severity (defined previously) identified by each source of data and by the test as a whole. The column labeled *Number of Pieces* shows the number of pieces of evidence used in identifying usability problems found by each source of empirical data. The row labeled Total shows the total number of problems of the given severity; because some problems were identified via multiple segments of the study, the numbers in each column do not necessarily sum to the specified total. The study identified a total of 65 usability problems.

Source	Severity Level				Number of Pieces
	Prevents Task	Significant Delay	Minor Effect	Suggestion	
Tutorial	2	4	2	0	8
Exercise Accuracy	3	0	1	0	4
Exercise Efficiency	2	3	2	1	8
Questionnaire	1	3	4	6	14
Verbal Protocol	8	10	27	16	61
Total	9	10	27	19	

Our second question examined the quality of the contributions made by each source of data. Figure 4 shows the number of problems detected from evidence originating with each source of data. The Tutorial, Exercises, and Questionnaire clearly contributed disproportionately to the more critical (level 1 and 2) problems, but the transcripts almost doubled the number of level 1 problems identified by the test (5 without; 9 with) and nearly quadrupled the number of level 3 problems (7 without; 27 with). These data suggest that the verbal protocol adds valuable contributions to problems at all severity levels, but disproportionately identifies less critical usability problems.

On the other side of the cost/benefit question is the effort required to collect and analyze the data. The costs involved in performing a usability study include preparing for the usability study, performing the usability study, extracting the raw data, analyzing the raw data to identify evidence, and analyzing the evidence to identify usability problems. Some of these costs are required for any usability study and others are required only for certain types of empirical data. This effort will vary greatly as a function of the usability lab set-up and the experience of its personnel, but we offer our experience as one data point.

Preparing for the usability study included writing the on-line Tutorial, preparing the Exercises, preparing the correct answers and the expected actions, and instrumenting the interface to collect the measurements. Of these tasks, only one, the on-line Tutorial, was useful beyond the usability study. That and preparing the Exercises would have been required regardless of the type of empirical data we wanted to collect. To collect the quantitative data for the study, we had to instrument the interface and the Tutorial to collect timing data and to record user's actions. This instrumentation required a few person-days to add the necessary code to the Tcl/TK prototype and to test that the data collected were accurate. Because the data were collected transparently in the background, it did not add to the duration of the usability test itself. For each participant, we required approximately 3-4 hours to extract and post-process (comparing answers and counting actions) the 250 pieces of raw data collected. Analyzing most of the post-processed data was relatively straightforward, simply requiring us to set up a spreadsheet, and required about 10-15 hours. However, the analysis of the Tutorial timing data required substantially more

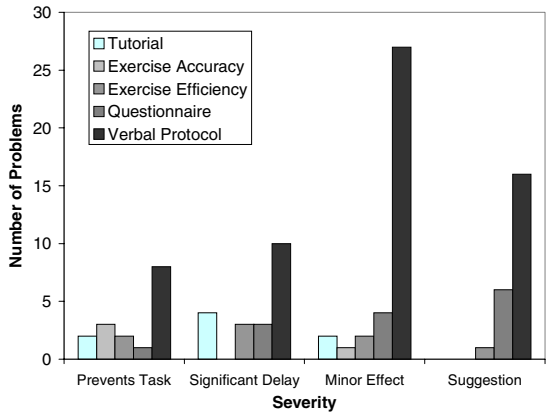


Figure 4: This bar chart shows the contributions of each source of data by the severity level (defined previously) of the problem identified.

time (perhaps 25-30 hours) because of the additional GOMS modeling and normalizing necessary to extract comprehension time.

The debriefing questionnaire took approximately 15-20 hours to prepare. Each participant required 10-20 minutes to respond to the questions in our survey. To ensure that we could read each participant's handwriting and understand the answers to the questions, we reviewed the questionnaire before allowing the participant to leave and asked for clarification if necessary. This added another 5-10 minutes to each participant's test. Analyzing the responses was simply a matter of placing them in the hierarchy, discussed previously.

The verbal protocol took no up-front instrumentation beyond scheduling the use of our User Studies Lab and configuring its recording equipment. However, it required that we train participants to provide a good verbal report, adding 10-15 minutes per participant to the duration of each test. Transcribing the verbal protocol required 6-8 hours per participant, about a 3- or 4-to-1 ratio of transcription-time to tape-time. Examining these transcripts in detail to identify the Comment Evidence required a substantial amount of time, perhaps another 2-4 hours per hour of verbal protocol.

Once all the raw data had been analyzed, we had to organize the resulting evidence and identify usability problems, an effort whose duration increases with the quantity of evidence, not the type of empirical data. The basic form of the organizing hierarchy was built from the first author's intuition of the organization of the evidence, with modifications to the hierarchy occurring out of necessity as the process of organization proceeded. Sorting the evidence into the hierarchy took only slightly more than pure reading time. Analyzing that evidence from the hierarchical framework was also not difficult, but did require a moderate amount of time to document each problem and consider alternative approaches to addressing the problem. Again, this effort increases with the number of problems, but not the type of data.

Given our experience of the costs and benefits of the empirical data we used in this study, we believe that the common wisdom that says "thinking aloud may be the single most valuable usability engineering method" [20, p. 195] is borne out by our data. However, because the more objective performance data and the easier-to-analyze questionnaire data contributed disproportionately to the more severe problems, our data suggest that these are also important sources worth the effort of collecting and analyzing.

An additional advantage to collecting multiple forms of empirical data beyond the accounting presented in this paper is the confidence that arises from seeing a convergence of evidence. When the reading rate indicated that an instructional screen was difficult, the user performed the task it taught well above par, and then commented on how difficult it was both concurrently in the verbal protocol and afterwards on the questionnaire, it becomes indisputable that there is indeed a usability problem.

4. LIMITATIONS and FUTURE WORK

A limitation of this analysis is that we do not know how many problems we would have identified simply by observing the user without transcribing the verbal protocol and analyzing it. Although we did not maintain complete real-time observations as each participant proceeded through the study, our informal notes suggest that we would have missed critical comments.

This study also does not address whether the identification of these problems, no matter which evidence was used, actually leads to an improvement in the design of an interface. Future research should extend all the way through re-design and re-testing (e.g., [11]) to deployment.

The most serious limitation of this result is that it arises from just a single usability study. This usability study could be an anomaly. Therefore, further research is required to confirm that this result holds across different experiments, experimenters, and test conditions by observing a variety of usability studies performed by different parties. Further, future studies should collect the detailed data needed to determine the exact cost and benefit of transcribing the verbal protocol. Any future work should also avoid the evaluator effect [9] by including several analysts for each verbal protocol.

5. CONCLUSIONS

If these further studies show that our results hold across experimenters and usability tests, it suggests that usability analysts trying to improve the design of a system should routinely collect multiple types of empirical data in any usability test. If, however, resources are limited and the primary goal is to identify the most severe problems, our results suggest that collecting performance and questionnaire data should be sufficient. On the other hand, because the verbal protocol had such overwhelming coverage of the problems, it suggests that researchers assessing the predictive power of an analytic technique (e.g., [2,3,10,12,14,21]) might be able to justify collecting only verbal protocol data as evidence for the usability problems predicted by an analysis.

6. ACKNOWLEDGMENTS

We thank Satya, Jane Siegel, Coda developers and users, the participants in our usability test, and Guerney Hunt for their contributions to and support of this work. We thank the reviewers for their helpful comments and insights.

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA), Air Force Materiel Command, USAF under agreement numbers F19628-96-C-0061 and F19628-93-C-0193, NSF Award #IRI-9457628, the Xerox Corporation, and the Intel Corporation. Some of the equipment used in the process of conducting this research was acquired through the National Science Foundation Equipment Grant #9022511. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Intel, Xerox, DARPA, NSF, or the U.S. Government.

7. REFERENCES

[1] Card, S. K., T. P. Moran, and A. Newell. "The Keystroke-Level Model for User Performance Time with Interactive Systems." *Communications of the ACM* 23, 396-410 (1980).

[2] Cuomo, D. L. and C. D. Bowen. "Understanding Usability Issues Address by Three User-system Interface Evaluation Techniques." *Interacting with Computers*, 6(1), 86-108 (1994).

[3] Desurvire, H. W. "Faster! Cheaper!! Are usability inspection methods as efficient as empirical testing?" In Jakob Nielsen and Robert L. Mack, eds*., Usability Inspection Methods*. John Wiley, NY (1994).

[4] Dumas, J. R. and J. C. Redish. *A Practical Guide to Usability Testing.* Ablex Publishing Corporation, Norwood NJ (1993).

[5] Ebling, M. R. "Translucent Cache Management for Mobile Computing." Ph.D. Dissertation. Carnegie Mellon University (1998).

[6] Ericsson, K. A. and H. Simon. "Verbal Reports as Data." *Psychological Review* 87(3), 215-51 (May 1980).

[7] Fink, A. and J. Kosecoff. *How to Conduct Surveys: A Step-by-Step Guide.* Sage Publications, Newbury Park, CA (1985).

[8] Gray, W. D. and M. C. Salzman. "Damaged Merchandise? A Review of Experiments That Compare Usability Evaluation Methods." *Human-Computer Interaction* 13(3), 203-261 (1998).

[9] Jacobsen, N. E., M. Hertzum, B. E. John. "The Evaluator Effect in Usability Studies: Problem Detection and Severity Judgements." *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting.* HFES, 1336-1340 (1998).

[10] Jeffries, R., J. R. Miller, C. Wharton, and K. M. Uyeda. "User interface evaluation in the real world: A comparison of four techniques", *Proceedings of CHI*, (1991).

[11] John, B. E. and S. J. Marks. "Tracking the Effectiveness of Usability Evaluation Methods." *Behaviour & Information Technology* 16(4/5), 188-202 (1997).

[12] John, B. E., and M. M. Mashyna. "Evaluating a Multimedia Authoring Tool with Cognitive Walkthrough and Think-Aloud User Studies." *Journal of the American Society of Information Science*. 48(9), (1997).

[13] Jorgensen, A. H. "Using the thinking-aloud method in system development." In Salvendy, G., and Smith, M. J. (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems.* Elsevier Science Publishers, Amsterdam, 743-750 (1989).

[14] Karat, C.-M., R. Campbell, and T. Fiegel. "Comparison of empirical testing and walkthrough methods in user interface evaluation." In *Proceedings of CHI*, 397-404 (May 1992).

[15] Kistler, J. and M. Satyanarayanan. "Disconnected Operation in the Coda File System." *ACM Transactions on Computers Systems.* 10(1), 3-25 (February 1992).

[16] Lewis, C. "Using the 'thinking-aloud' method in cognitive interface design." *Research Report RC9265,* IBM T. J. Watson Research Center, Yorktown Heights, NY (1982).

[17] Mummert, L. B., M. R. Ebling, M. Satyanarayanan. "Exploiting Weak Connectivity for Mobile File Access." In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles,* 143-155 (December 1995).

[18] Newell, A. and H. A. Simon. *Human Problem Solving.* Prentice-Hall, Englewood Cliffs, NJ (1972).

[19] Nielsen, J. "Evaluating the thinking aloud technique for use by computer scientists." In Hartson, H. R., and Hix, D. (Eds.), *Advances in Human-Computer Interaction*, Vol. 3, Ablex, Norwood, NJ, 69-82 (1992).

[20] Nielsen, J. *Usability Engineering*. AP Professional, NY (1993).

[21] Nielsen, J. and V. L. Phillips. "Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared." In *Proceedings of INTERCHI'93,* 214-221 (1993).

[22] Satyanarayanan, M., J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere. "Coda: A Highly Available File System for a Distributed Workstation Environment." *IEEE Transactions on Computers.* 39(4), 447-59 (April 1990).