SOFTWARE WIZARDS FOR KEYBOARD AND MOUSE SETTINGS:

USABILITY FOR END USERS

Heidi Koester, Jennifer Mankowski, Edmund LoPresti,
Glen Ashlock, Richard Simpson
Koester Performance Research,
Ann Arbor Center for Independent Living, University of Pittsburgh

INTRODUCTION

Providing effective use of computer input devices is one key to user productivity and comfort during computer use. We have developed two software wizards that help ensure that keyboard and pointing devices are properly configured for the individual, and reconfigured as the user's needs change. A major goal was to achieve high usability by end users, regardless of their physical impairment or their prior familiarity with keyboard and mouse settings. This paper describes the usability studies we conducted and modifications made on the way to achieving that goal.

BACKGROUND

The Windows operating system has a number of built-in settings designed to accommodate a user's physical needs when using a keyboard and mouse (or other pointing device). These include Sticky Keys, keyboard repeat rate, mouse gain, double-click time, and double-click distance. Unfortunately, many users who can benefit from these settings are unaware of them [1], and setting them appropriately can be a complex and time-consuming task.

We have developed two software wizards designed to make it easier for users to adjust their own Windows keyboard and mouse settings. The Keyboard Wizard walks users through a one-sentence typing task, suggests possible changes to the keyboard settings, and lets the user decide which settings to activate. The Pointing Wizard presents a series of mouse tasks to the user, and recommends appropriate settings for the user based on the results. Both wizards will also activate their recommended settings on the computer, contingent on user approval. The initial version of both wizards is available at the KPR website (www.kpronline.com).

USABILITY GOALS

For the wizards to deliver real benefits, they must be readily usable by end users. That is, end users should be able to use the software themselves and not have to rely on a professional or other individual to help them through the process. To ensure that this goal was met, we performed a variety of informal and formal usability studies across the 2-year project. The approach and results are summarized below.

WIREFRAME PROTOTYPES

User Interface Design

A series of interviews with practitioners and end users helped us define the initial feature set for our software and establish some early usability criteria. Based on these initial ideas, we developed a wireframe prototype of the keyboard wizard, using Axure RP software. The wireframe prototype was a clickable mockup of the system, showing what each screen and transition would look like. However, it did not actually make real adjustments to the keyboard settings; it merely acted as if it did.

Wireframe Usability Methods

Across two studies, 14 individuals went through a usability protocol with a wireframe interface. 6 participants were end users, with a variety of physical impairments; 5 were AT practitioners; and 3 were caregivers or friends

of an end user. The protocol occurred over a single session, and included the following steps:

1. 14 background questions about the participant and their computer use.
2. A basic scenario that asked the participant to walk through the Keyboard Wizard interface, with no guidance from the experimenter.
3. Open-ended questions regarding specific areas of the interface for which we needed more user input.
4. Likert-type questionnaire items regarding ease-of-use and other aspects of the interface.

The sessions were video-recorded using Morae Recorder software to capture the screen and all user actions and comments. This allowed detailed review of usability issues experienced by the participants and provided a way to determine the time it took each user to complete the scenario task.

It should be noted that the wireframe interface was revised several times throughout the studies. Rather than insist on having the same prototype across all participants, we modified the interface as soon as an obvious usability issue was identified.

Wireframe Data Analysis

Specific variables that were quantitatively analyzed were: Completion of scenario task (yes/no), completion time (minutes), and usability ratings (1 to 5). However, qualitative observations of usability problems and participant comments were more important at this stage.

Wireframe Results

Questionnaire responses led to a benchmark completion time of 15 minutes or less. Actual completion times ranged from 4 to 16 minutes, averaging 8.1 minutes. All participants were able to complete the task without significant difficulty. End users rated the ease-of-use at 3.7 on a scale of 1 (low) to 5 (high).

While these results looked reasonably promising, the fact that the wireframe prototype did not actually change the user's settings was a significant limitation in the studies. The main usability issue related to understanding when and whether the wizard actually adjusted the Windows settings. Some users wondered whether they were supposed to make the recommended changes themselves. This confusion was due at least in part to poor interface design, but the limited functionality of the prototype certainly made the problem worse.

The primary revisions made were to clarify ambiguous wording, with some minor workflow adjustments. After revising the prototype through several rounds of changes, the results suggested that end users could use this interface successfully.

WORKING PROTOTYPES

Keyboard Wizard Implementation

We implemented a fully working prototype of the Keyboard Wizard using the Java programming language. This addressed the limitation of the wireframe, since the working prototype actually did activate the Windows settings correctly for the user.

Usability Studies for Working Prototypes

A series of four usability studies were conducted with four versions of the working prototype. For three of these studies, the methods were very similar to those used for the wireframe studies, using the same scenario task and questionnaire. Some of the open-ended questions varied between studies, depending on particular issues that we were exploring. The fourth study was a remote beta test, in which we made the beta version available for download and asked users to complete an online survey. Key results are described below.

Results with Version 0.0

This study involved four participants: 3 end users and 1 practitioner. Results were extremely illuminating, as research notes refer to it as a "usability disaster" for one end user in particular. A major problem related to trouble with the Enter key, which activated the Next button on each screen. Holding the Enter key down too long moved through multiple screens

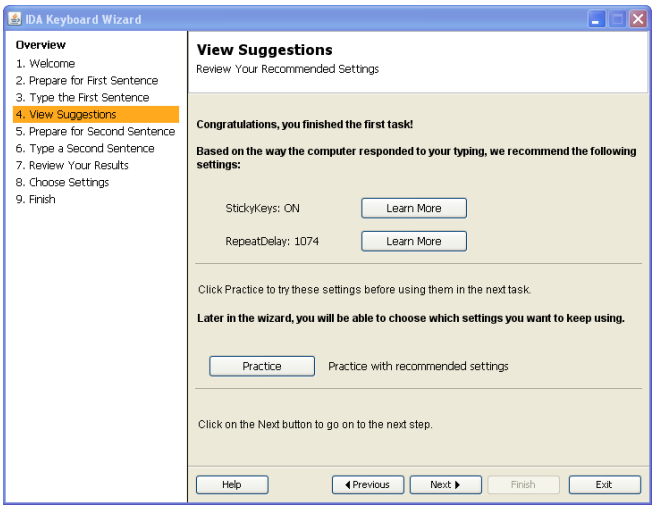

Fig 1. Screenshot from Keyboard Wizard v0.0, showing sidepanel, small font, and required second typing task.

unexpectedly, and hitting it inadvertently suddenly advanced to the next screen. This led to a large amount of confusion for this user, although she was eventually able to complete the task. The wireframe prototype did not use 'Enter' as an equivalent to clicking the Next button, so we did not have an opportunity to observe this problem in the previous studies.

While the other 2 end users did not have that same difficulty with the Enter key, there were still 11 major issues identified. Three of these were technical issues related to the decision algorithms and measurement accuracy, which again were areas that we could not evaluate with the wireframe prototype. Related to workflow, we revised the default path so that a second typing task is no longer required, while allowing users to opt-in if desired. Several ambiguous controls were identified and revised. Part of the reason these weren't identified earlier is that the wireframe prototype controls did not actually affect the underlying Windows settings. Finally, the user interface was simplified to remove the sidepanel. This was a list of wizard steps, with the active one highlighted to orient the user to their place in the process. Users tended to click on the sidepanel, expecting that it would move to that step, so the sidepanel was creating more confusion than it was preventing.

Average completion time was 11 minutes for the end users, and ease-of-use rating averaged 3.33. These compared unfavorably to values observed in the wireframe studies, as we would expect, given the usability problems observed.

Despite the numerous issues, most of the solutions were straightforward. The biggest challenge was to resolve the keyboard issues so that inadvertent Enter hits wouldn't cause problems, while still preserving 100% keyboard access for people who have difficulty using a mouse.

Results with Version 0.1 and 0.2

These two studies involved a total of 7 end users and 1 practitioner. Three end users went through the protocol for Version 0.1, including the individual who had difficulties with the Enter key in the previous study. The problems with Enter were solved, but a second user revealed issues with inadvertent spacebar presses. Since the spacebar could also act as a button click, this led to unexpected and surprising screen transitions. The algorithm issues were improved relative to Version 0.0, but there was still a need for further refinements. Finally, a response time issue emerged, as some steps in the wizard had long delays when transitioning to the next step. Overall, however, Version 0.1 represented a big improvement over Version 0.0.

Four new end users and one practitioner went through the protocol with Version 0.2. All users completed the task with ease, and the algorithms all worked correctly. Revisions at this point were refinements to screen text, additional enhancements to response time, and other relatively minor issues.

Across these two studies, average completion time was 5.5 minutes for end users. Ease-of-use rating averaged 4.83 (on a scale of 1 to 5).

Beta Test with Version 0.9

Given the high usability observed with Version 0.2, we moved to the next step of creating a beta version for wider distribution. While we were still interested in usability at this point, a main goal of the beta test was to identify any glitches in program installation and execution across a wide range of computers and computing environments.

17 individuals participated, after responding to notices to the QIAT and RESNA listserv communities. 14 of these were practitioners, and 3 were end users with physical impairments. Participants were asked to walk through specific tasks with the wizard and complete a questionnaire. Unlike the previous studies, however, they were not observed or recorded while using the software. No significant functional or usability problems were reported. Ease-of-use rating averaged 4.4.

POINTING WIZARD

The Pointing Wizard recommends Windows settings related to the mouse and other pointing devices. We did not implement a Pointing Wizard prototype until we were confident in the usability of the wizard interface. Once the Keyboard Wizard beta test was complete, we developed the Pointing Wizard to use a very similar user interface. Six end users participated in a usability study for Pointing Wizard, with average completion time of 6 minutes and ease-of-use rating of 4.5.

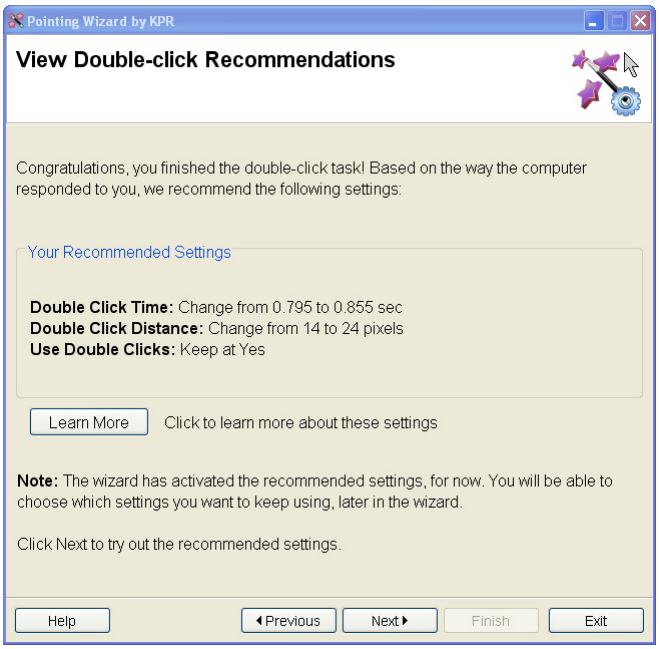

Fig 2. Screenshot from Pointing Wizard v1.0, with clearer organization, larger font.

DISCUSSION

By the completion of these usability studies, both Pointing Wizard and Keyboard Wizard were ready for general release, and we were confident in their high level of usability for end users as well as practitioners. Each type of study performed was useful in different ways. The wireframe prototypes provided an efficient way to explore overall look-and-feel and workflow issues. Due to their limited functionality, however, extensive testing was still required with fully functional prototypes. This is particularly true for this application in which it was impossible to make the wireframe fully mimic all aspects of the real system. Remote beta testing was a valuable final step to ensuring proper functioning across various computing environments with a larger number of users.

Developing an interface which was usable for a variety of users required several design iterations. Our previous experience involved developing software for AT practitioners, who represent a different user group from end users who may not have an in-depth knowledge of the computer access domain and may have significant physical impairments. Most of the lessons learned are applicable in many design scenarios. These include:

1. A wireframe prototype can tell you some things, but real-world functionality is crucial, especially in this application;
2. Don't expect users to thoroughly read every word on every screen;
3. Explaining unfamiliar concepts is a challenge. Provide the basic information in the interface, with more details available in a Help system;
4. Error prevention is crucial. Make the interface as bullet-proof as possible, while preserving user navigational control via keyboard or mouse.

ACKNOWLEDGEMENTS

This work was funded by a Phase II SBIR Award, National Institutes of Health. We also thank the many participants for their time, effort, and insights.

REFERENCES

[1] Stevenson, B., McQuivey, J., "Examining Awareness, Use, and Future Potential." Commissioned by Microsoft Corporation and Conducted by Forrester Research, Inc. www.microsoft.com/enable/research/default.aspx, 2003.