CS 66.460
Introduction to Graphical Human–Machine Interfaces
Spring 1997
IN-CLASS EXERCISE #3
Wednesday, February 26 and Monday, March 3, 1997
Heuristic Evaluation of a Paper Mock-Up

The Goal: To gain experience identifying usability problems of a user interface. In particular by performing a **heuristic evaluation** of a paper mock-up of a user interface.

The Interface: Figure 1 shows a design for a system to provide weather information to travelers. TRAVEL*weather* (a non-existing system) can provide information about the weather at 3AM, 9AM, 3PM, and 9PM for the current day as well as the two next days, using reported readings for past weather and forecasts to predict future weather. The interface is designed for use with on a graphical personal computer with a mouse, and will appear in a separate window on the screen.

Figure 1: Screen design for a hypothetical system to provide weather information and forecast to travelers.

The user operates the interface by typing the desired time into the box in the upper right part of the screen. If the user types a date other than today or the next two days, or if the user types a time other than the four times for which information is available, the system will show an alert dialog box with the following error message: `''Weather Data Not Available.''` The only button in the error message box is an "OK" button. Clicking the OK button will make the dialog box go away and will reset the date and time specification to the previous value.

The user changes the map display by editing the boxes for zoom magnification and for the center of the map. The system ensures that only integers can be typed in the map magnification box by simply beeping every time the user presses a non-number key in that box. If the user types anything in the map center box other than a valid set of coordinates (an integer from 0 to 90 followed by the letter N or S followed by an integer from 0 to 179 followed by the letter W or E), the system will show an alert dialog box with the following error message `''Unknown Map Coordinates.''` The only button in the error message box is an "OK" button. Clicking the OK button will make the dialog box go away and will reset the coordinates to their previous value.

With respect to all three input boxes, the user's changes take effect as soon as the user clicks the mouse outside a box after having editing it.

The Goal: The goal of this exercise is to determine as many usability problems with the TRAVEL*weather* interface as you can by performing a heuristic evaluation of the interface.

What To Do: Divide up into small groups of 4–5 people. Feel free to rearrange the seating in the room so as to facilitate interaction among members of your team.

Perform a heuristic evaluation of the interface. Remember to evaluate the entire interface, including both the figure and the text describing what happens as a result of various user actions. The result of the heuristic evaluation should be a list of the usability problems in the interface with reference to some established usability principle violated by that aspect of the user interface. It is not sufficient just to say that you do not like something; explain *why* you do not like it with reference the heuristic (see the attached list of heuristics or you may make reference to other usability results).

Try to be as specific as possible and list each usability problem separately. For example, if there are three things wrong with a certain dialog element, list all three with reference to the various usability principles that explain why that particular aspect of the interface is usability problem. There are two main reasons to note each problem separately: First, there is a risk of repeating some problematic aspect of the dialog element, even if it were to be completely replaced with a new design, unless one is aware of all its problems. Second, it may not be possible to fix all usability problems in an interface element or to replace it with a new design, but it could still be possible to fix *some* of the problems.

As an example, here is one usability problem:

> The name of the system is displayed much too prominently. By making the name smaller, room could be provided for alternative dialog elements, or the screen could be made less busy. (Simple and Natural Dialog)

Once you have identified the usability problems, you should give it a **severity rating** using the rating scale shown below.

The Deliverables: Your group will meet for part of class on both Wednesday, February 25, and Monday, March 3. After meeting with your group on March 3, be prepared to discuss some of the usability problems with the rest of the class.

As usual, after the conclusion of the in-class part of the exercise each team should summarize its designs and the lessons learned in a written report which will be due **Wednesday, March 19, 1997, in class**.

Your report should include:

1. a cover-sheet listing all of the participants in your group;
2. a list of the usability problems in the interface with reference to some established usability principle that has been violated;
3. the severity rating of each usability problem;

Severity Ratings

- `0` = this is not a usability problem at all
- `1` = cosmetic problem only – need not be fixed unless extra time is available on the project
- `2` = minor usability problem – fixing this should be given low priority
- `3` = major usability problem – important to fix, so should be given high priority
- `4` = usability catastrophe – imperative to fix this before product can be released

Guidelines for Usability

1. **Simple and Natural Dialog**
   - Match the user's task in as natural a way as possible
   - Present exactly the information the user needs
     - "less is more"
     - information should appear in natural and logical order
     - remove or hide irrelevant or rarely needed information
     - use windows frugally
2. **Speak the User's Language**
   - Use mappings and metaphors between a user's conceptual model and computer display
   - Terminology based on user's language for task, not on system oriented terms
   - View interactions from the user's perspective
   - Use meaningful mnemonics, icons, and abbreviations
3. **Minimize the user's memory load**
   - Computers are good at remembering things; people aren't!
   - Promote recognition over recall
   - Describe the required input format and example, and default
   - Small number rules apply universally
4. **Be consistent.**
   - Consistency of effects
   - Consistency of language and graphics
   - Consistency of input
5. **Provide feedback.**
   - Continuously inform the user about: what the system is doing; how it is interpreting the user's input; minimize the "gulf of evaluation"
   - Should be as specific as possible, based on user's input
   - Be careful with long response times

6. **Provide clearly marked exits.**
    - Don't let user feel trapped by the computer
    - Strategies
        - Cancel button (for dialogs waiting for input)
        - Universal Undo (can get back to previous state)
        - Interrupt (especially for lengthy operations)
        - Quit (for leaving the program at any time)
        - Defaults (for restoring a property sheet)
7. **Provide shortcuts.**
    - Experienced users should be able to perform frequently used operations quickly
    - Strategies
        - keyboard and mouse accelerators
        - type-ahead
        - navigation jumps
        - reuse
8. **Deal with errors in a positive and helpful manner.**
    - People will make errors!
    - System should be robust, should not crash due to invalid input
    - Provide meaningful error messages
    - Prevent errors
    - Have as few modes as possible; make modes visible
    - Make actions reversible
9. **Provide help and documentation.**