

# User Interface Design

## Project Assignment PA #1

due date 05/04/2007 (worth 10%)

March 26, 2007

### Paper Prototyping

Prototype Testing: Tutorials 02/04/ and 03/04/.

Final Hand-In: Due at 11:59pm, on Thursday, April 5th, 2007, by blackboard.

### Overview

In this group assignment, you will do your first implementation of your term project, which will be a paper prototype. Your paper prototype should be able to handle at least 3 scenarios. These scenarios could be the scenarios you described in the task from Tutorial #4.

The tutorials in this week (26/03/ and 27/03/) offer you an opportunity to start building a paper prototype. The tutorials next week (02/04/ and 03/04/) offer you an opportunity to test a paper prototype on your classmates.

The testing time will be divided into three 20 minute periods, and the projects will likewise be divided into 3 groups. During each period, two-thirds of the class will run their prototypes, while the remaining third serve as users. Overall, you should get 40min worth of testing of your prototype, and 20min serving as a user. Participation in testing is required. Since your classmates are too much like you (they're taking SENG3300/SENG6300, they're mostly CS students, etc., etc.), you should consider them merely pilot users, who help you find the most obvious usability problems and help you practice running your paper prototype. Additionally, you should find some realistic users. First, revise your paper prototype to address the critical usability problems and explore possible design alternatives. Then, test it on at **least 3 users from your target population, all from outside the class.**

## Choosing What to Prototype and Test

You may need to adjust your scenarios so that they explore the riskiest parts of your interface. A part of your interface is risky if its usability is hard to predict, or if its usability strongly affects the usability of the whole system. For example:

- **Novel design:** Parts of your interface that are new and different are potentially risky. By contrast, a username/password form isn't risky at all, because it's a familiar and well-understood idiom.
- **Frequent use:** A frequently-used feature might be risky, because the efficiency of the whole interface depends strongly on it.
- **Error danger:** A feature in which user errors might be common or hard to recover from is risky.
- **Complexity:** A complicated or subtle part of your system is risky. The configuration interface for a firewall, for example, would be risky because it involves system-level concepts that users may not know or understand. A font selection dialog might be risky simply because of the number of choices it offers.

Risky parts need the most design iteration, so they'll give you the most payoff from prototyping. In other words, don't waste your effort on prototyping a login screen, but do make sure to prototype a novel, complicated, frequently-used dialog box. Not every risky part can be easily tested with paper prototyping, but if you make sure your scenarios cover the risky parts now, you'll be able to plan your subsequent (computer-based) prototypes better.

## Preparing for testing

Before testing your prototype, you should:

- **Build your prototype.** Draw the static background, menus, dialog boxes, and other windows. Decide how to implement the dynamic parts of your interface. Hand-sketching is encouraged. You don't have to prepare every possible screen in advance; it may be much easier to write responses on the fly.
- **Prepare a briefing for test users.** This should be at most a page of information about the purpose of your application and any background information about the domain that may be needed by your test users (who may be classmates) to understand it. These are your notes for the briefing, so make them short, simple and clear, not dense wordy paragraphs. This is not a manual or quick-reference card. It should not describe how to use the interface.

- **Write your 3 scenario tasks on separate index cards.** Just write the concrete goal(s) of the task (e.g. "buy milk, tomatoes, and bread"). Don't write the specific steps to follow, since that's for your users to figure out. The tasks should be brief, roughly 5 minutes to run.
- **Choose roles for your team members.** One person must play the computer. The other team members (if any) will be observers. We won't bother with a facilitator for these pilot tests. It may be useful for you to swap roles after every user on Testing Day, so that each of you gets a chance to try each role, but decide how you'll do it in advance.
- **Practice running your paper prototype.** Every team member should practice playing the computer, learning the steps involved in making the prototype functional, such as rearranging pieces and writing responses. It isn't important to be fast, just competent and confident. A few trials are enough. Make sure your prototype can handle the 3 scenario tasks you chose.

## Running the Tests

When you run your prototype on a user, you should do the following things:

- **Brief the user.** Use the briefing you wrote up to describe orally the purpose of the application and background information about the domain. Don't waste too much time on this: 1 minute should be enough.
- **Present one task.** Hand the index card to the user and let them read it. Make sure they understand the task.
- **Watch the user do the task.** Take notes from your observations.
- **Repeat with the other tasks.** Run as many tasks on the user as you have time for.

Bring extra materials for testing. Having extra blank Post-it notes, correction tape, and index cards on hand will help you improvise if a user does something unexpected, or help you make small fixes to your prototype between users.

## Playing Test Users for Your Classmates

On Testing Day, when you are serving as a user, you should:

- **Relax and enjoy yourself.** You're not being tested – the interface is. Part of the point of this experience is to feel what it's like to be the user in a user test, so that you can empathize with them.
- **Be cooperative.** Don't be intentionally dense, e.g. looking for Exit everywhere but the File menu. Interact with the interface as you would if you were really using it.
- **Think aloud.** Help the observers understand what you're thinking by verbalizing your thought process. "Let's see, I want to enter this bottle of milk, so where's the scanner... oh, here it is. I'll scan the bottle like this, oops that didn't work, let me find the bar code..." You get the idea.

## Written Report

You should hand in an electronic report in PDF format with the following parts:

- **Risk assessment.** List the parts of your interface that you consider risky, state which scenarios will test each one, and state how you plan to prototype them (paper, computer, or other) to mitigate the risk.
- **Prototype photos.** Digital photos of the pieces of your prototype. Show the prototype in interesting states; don't just show a blank window. (Although you will iterate your paper prototype during this assignment, the photos only need to show one iteration.)
- **Briefing.** The briefing you gave to users.
- **Scenario tasks.** The tasks you gave to users, as you wrote them on the cards.
- **Observations.** Usability problems you discovered from the testing. Describe what users did, but don't record users' names. Your observations should include both your classmates on Testing Day, and at least 3 users from outside the class.
- **Prototype iteration.** Describe how your prototype changed between your Testing Day users and the real users.
- **Risk resolution.** What did you learn about the risky parts of your interface from this prototype? Propose design solutions for the usability problems you found.

## What to Hand In

Your written report, completed by the deadline. Submit your report as **PDF file** to the digital dropbox, and don't use the # character in the file name.