HCI: Interactive System Design (Prototype Techniques)

Professor Ram Mohana Reddy Guddeti

Information Technology Department

NITK Surathkal, Mangalore, India

Learning Objective

- In the previous lecture, we learned about a method (contextual inquiry) to gather requirements for a design

- Designer can come up with ideas on the basis of this data

  - Typically more than one designs are proposed

- It is necessary to evaluate the alternative designs to find out the most appropriate one

Learning Objective

- Interactive systems are designed following a user-centered design approach

  - Evaluation of the alternative design proposals should be done from the user's perspective

- Employing end users in evaluating designs is not easy

  - It is costly in terms of money, time, effort and manpower

Learning Objective

- In the initial design phase, when the proposed design undergoes frequent changes, it is not advisable to even feasible to carry out evaluation with real users

- An alternative way to collect feedback on proposed design is to develop and evaluate "prototypes"

Learning Objective

- In this lecture, we shall learn about the prototyping techniques used in interactive system design

- In particular, we will learn the following:
  - Why we need prototyping (already discussed in the previous slides)?
  - What are the techniques available (overview)?
  - How these techniques are used (details)?

Prototyping

- A prototype is essentially a model of the system

  - The prototype (model) can have limited or full range of functionalities of the proposed system

- A widely used technique in engineering where the novel products are tested by testing a prototype

Prototyping

- Prototypes can be "throw away" (e.g., scale models which are thrown away after they serve their purpose) or can go into commercial use

- In software development prototypes can be

  - Paper-based: likely to be thrown away after use

  - Software-based: can support few or all functionalities of the proposed system. May develop into full-scale final product

Prototyping in HCI

- Essential element in user centered design

    - Is an experimental and partial design

    - Helps involving users in testing design ideas without implementing a full-scale system

- Typically done very early in the design process

    - Can be used throughout the design life cycle

What to Prototype?

- Any aspect of the design that needs to be evaluated

  – Work flow

  – Task design

  – Screen layout

  – Difficult, controversial, critical areas

Prototypes in HCI

- In HCI, prototypes take many forms

    – A storyboard (cartoon-like series of screen sketches)

    – A power point slide slow

    – A video simulating the use of a system

    – A cardboard mock-up

    – A piece of software with limited functionality

    – Even a lump of wood

Prototypes in HCI

- We can categorize all these different forms of prototypes in the three following groups

  - Low fidelity prototypes

  - Medium fidelity prototypes

  - High fidelity prototypes

Low Fidelity Prototypes

- Basically paper mock-up of the interface look, feel, functionality

  - Quick and cheap to prepare and modify

- Purpose

  - Brainstorm competing designs

  - Elicit user reaction (including any suggestions for modifications)

Low Fidelity Prototypes

Interface of a proposed system

A sketch of the interface

Low Fidelity Prototypes

- In a sketch, the outward appearance of the intended system is drawn
  - Typically a crude approximation of the final appearance

- Such crude approximation helps people concentrate on high level concepts
  - But difficult to visualize interaction (dialog's progression)

Low Fidelity Prototypes: Storyboarding

- Scenario-based prototyping

- Scenarios are scripts of particular usage of the system

- The following (four) slides show an example storyboarding of a scenario of stroller-buying using an e-commerce interface

Low Fidelity Prototypes: Storyboarding

Initial screen. Shows the layout of interface options.

Low Fidelity Prototypes: Storyboarding

Once a stroller is selected by the customer, its tag is scanned with a hand-held scanner. The details of the stroller is displayed on the interface if the scanning is successful. Also, the option buttons become active after a successful scan.

Low Fidelity Prototypes: Storyboarding

However, the customer can choose a different product at this stage and the same procedure is followed. For example, the customer may choose a stroller with different color.

Low Fidelity Prototypes: Storyboarding

Once the customer finalizes a product, a bill is generated and displayed on the interface. The option buttons become inactive again.

Low Fidelity Prototypes: Storyboarding

- Here, a series of sketches of the *keyframes* during an interaction is drawn

  - Typically drawn for one or more typical interaction scenarios

  - Captures the interface appearance during specific instances of the interaction

  - Helps user evaluate the interaction (dialog) unlike sketches

Low Fidelity Prototypes: Pictiv

- Pictiv stands for "plastic interface for collaborative technology initiatives through video exploration"

- Basically, using readily available materials to prototype designs
  - Sticky notes are primarily used (with plastic overlays)
  - Represent different interface elements such as icons, menus, windows etc. by varying sticky note sizes

Low Fidelity Prototypes: Pictiv

- Interaction demonstrated by manipulating sticky notes

  – Easy to build new interfaces "on the fly"

- Interaction (sticky note manipulation) is videotaped for later analysis

Medium Fidelity Prototypes

- Prototypes built using computers

  – More powerful than low fidelity prototypes

  – Simulates some but not all functionalities of the system

  – More engaging for end users as the user can get better feeling of the system

  – Can be helpful in testing more subtle design issues

Medium Fidelity Prototypes

- Broadly of two types

  - **Vertical prototype** where in-depth functionalities of a limited number of selected features are implemented. Such prototypes helps to evaluate common design ideas in depth.

  - Example: working of a single menu item in full

Medium Fidelity Prototypes

- Broadly of two types

  - **Horizontal prototype** where the entire surface interface is implemented without any functionality. No real task can be performed with such prototypes.

  - Example: first screen of an interface (showing layout)

Medium Fidelity Prototypes: Scenarios

- Computer are more useful (than drawing on paper as in storyboarding) to implement scenarios

  - Provide many useful tools (e.g., power point slides, animation)

  - More engaging to end-users (and easier to elicit better response) compared to hand-drawn story-boarding

Hi Fidelity Prototypes

- Typically a software implementation of the design with full or most of the functionalities

  - Requires money, manpower, time and effort

  - Typically done at the end for final user evaluations

Prototype and Final Product

- Prototypes are designed/used in either of the following:

  ❖ **Throw-away:** prototypes are used only to elicit user reaction. Once their purpose is served, they are thrown away.

  – Typically done with low and some medium fidelity prototypes

  ❖ **Incremental:** Product is built as separate components (modules). After each component is prototyped and tested, it is added to the final system

    – Typically done with medium and hi fidelity prototypes

Prototype and Final Product

- Prototypes are designed/used in either of the following:

  ❖ **Evolutionary:** A single prototype is refined and altered after testing, iteratively, which ultimately "evolve" to the final product

  – Typically done with hi fidelity prototypes

Prototyping Tools

- For (computer-based) medium and hi fidelity prototype developed, several tools are available

  - **Drawing tools,** such as Adobe Photoshop, MS Visio can be used to develop sketch/storyboards

  - **Presentation software**, such as MS Power Point with integrated drawing support are also suitable for low fidelity prototypes

Prototyping Tools

- For (computer-based) medium and hi fidelity prototype developed, several tools are available

    - **Media tools,** such as Adobe flash can be to develop storyboards. Scene transition is achieved by simple user inputs such as key press or mouse clicks

Prototyping Tools

- For (computer-based) medium and hi fidelity prototype developed, several tools are available

  - **Interface builders,** such as VB, Java Swing with their widget libraries are useful for implementing screen layouts easily (horizontal prototyping). The interface builders also supports rapid implementation of vertical prototyping through programming with their extensive software libraries

The Wizard of Oz Technique

- A technique to test a system that does not exist

- First used to test a system by IBM called the listening typewriter (1984)

  - Listening typewriter was much like modern day voice recognition systems. User inputs text by uttering the text in front of a microphone. The voice is taken as input by the computer, which then identifies the text from it.
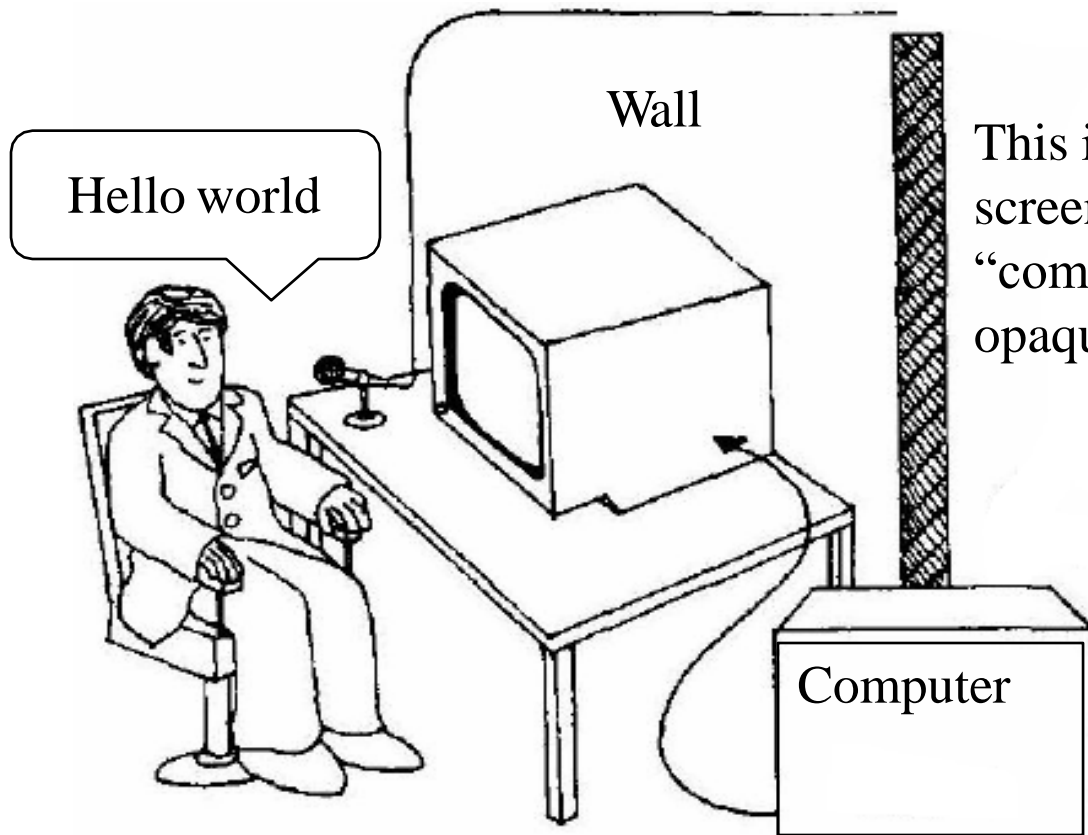
The Wizard of Oz Technique

- Implementing voice recognition system is too complex and time consuming

- Before the developers embark on the process, they need to check if the "idea" is alright; otherwise the money and effort spent in developing the system would be wasted

- Wizard of oz provides a mechanism to test the idea without implementing the system

The Wizard of Oz Technique

- Suppose a user is asked to evaluate the listening typewriter

- He is asked to sit in front of a computer screen

- A microphone is placed in front of him

- He is told that "whatever he speaks in front of the microphone will be displayed on the screen"

The Wizard of Oz Technique

Hello world

Wall

Computer

This is what the user sees: a screen, a microphone and the "computer" in front of an opaque wall.

The Wizard of Oz Technique

This is what happens behind the wall. A typist (the wizard) listen to the utterance of the user, types it, which is then displayed on the user's screen. The user thinks the computer is doing everything, since the existence of the wizard is unknown to him.
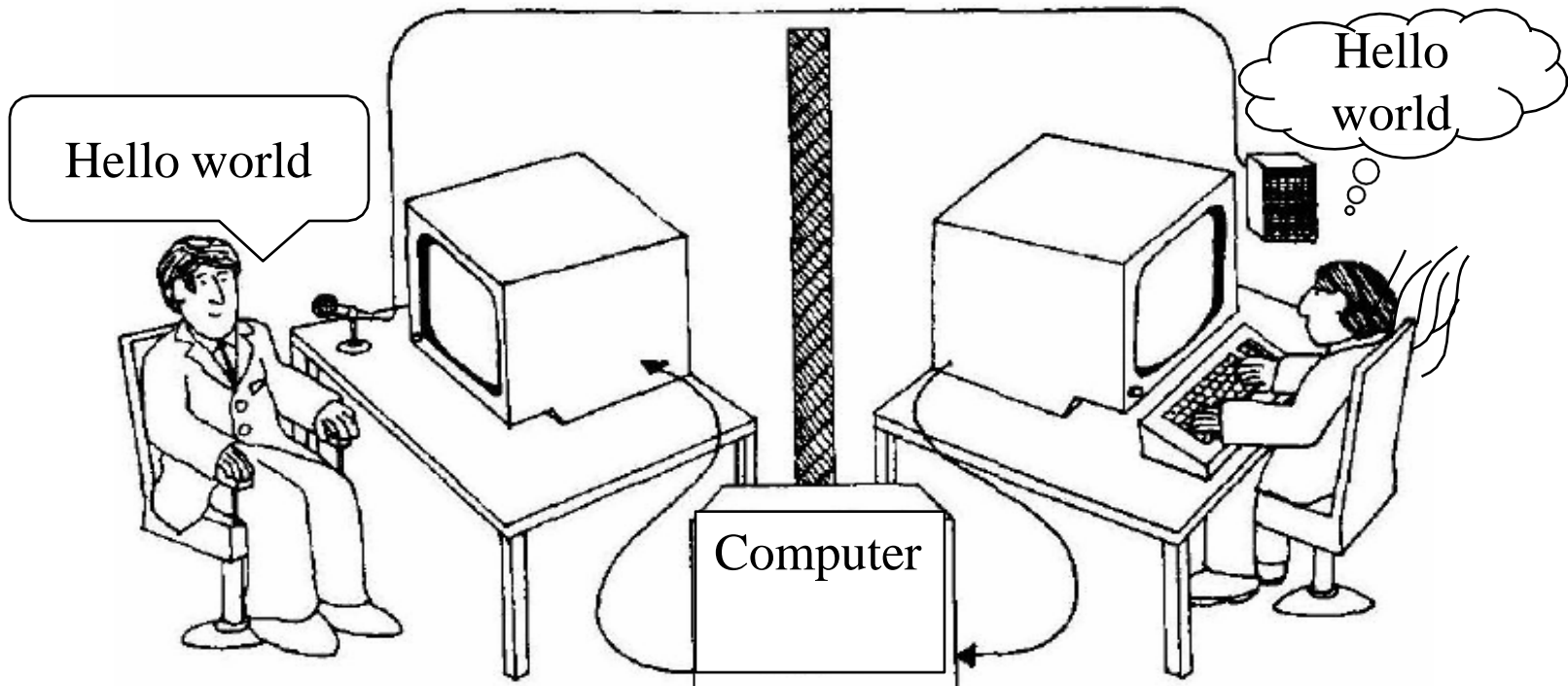
The Wizard of Oz Technique

- Human 'wizard' simulates system response

  – Interprets user input

  – Controls computer to simulate appropriate output

  – Uses real or mock interface

  – Wizard is typically hidden from the user; however, sometimes the user is informed about the wizard's presence

The Wizard of Oz Technique

- The technique is very useful for

    – Simulating complex vertical functionalities of a system

    – Testing futuristic ideas