

# Design, prototyping and construction



# Overview

- Prototyping and construction
- Conceptual design
- Physical design
- Tool support



# *Prototyping and construction*

- What is a prototype?
- Why prototype?
- Different kinds of prototyping
  - low fidelity
  - high fidelity
- Compromises in prototyping
  - vertical
  - horizontal
- Construction

# *What is a prototype?*

In other design fields a prototype is a small-scale model:

- a miniature car

- a miniature building or town

# *What is a prototype?*

In interaction design it can be (among other things):

- a series of screen sketches

- a storyboard, i.e. a cartoon-like series of scenes

- a Powerpoint slide show

- a video simulating the use of a system

- a lump of wood (e.g. PalmPilot)

- a cardboard mock-up

- a piece of software with limited functionality written in the target language or in another language

# *Why prototype?*

- Evaluation and feedback are central to interaction design
- Stakeholders can see, hold, interact with a prototype more easily than a document or a drawing
- Team members can communicate effectively
- You can test out ideas for yourself
- It encourages reflection: very important aspect of design
- Prototypes answer questions, and support designers in choosing between alternatives

# *What to prototype?*

- Technical issues
- Work flow, task design
- Screen layouts and information display
- Difficult, controversial, critical areas

# *Low-fidelity Prototyping*

- Uses a medium which is unlike the final medium, e.g. paper, cardboard
- Is quick, cheap and easily changed
- Examples:
  - sketches of screens, task sequences, etc
  - 'Post-it' notes
  - storyboards
  - 'Wizard-of-Oz'



# *Storyboards*

- Often used with scenarios, bringing more detail, and a chance to role play
- It is a series of sketches showing how a user might progress through a task using the device
- Used early in design

# Sketching

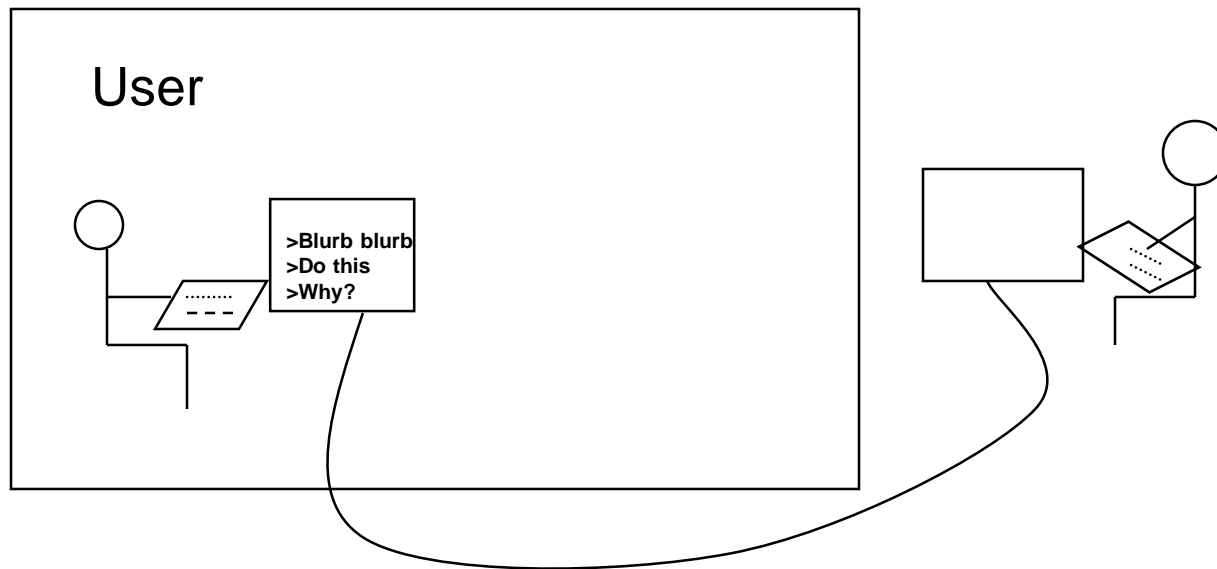
- Sketching is important to low-fidelity prototyping
- Don't be inhibited about drawing ability. Practice simple symbols

# *Using index cards*

- Index cards (3 X 5 inches)
- Each card represents one screen
- Often used in website development

# 'Wizard-of-Oz' prototyping

- The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.
- Usually done early in design to understand users' expectations
- What is 'wrong' with this approach?



# *High-fidelity prototyping*

- Uses materials that you would expect to be in the final product.
- Prototype looks more like the final system than a low-fidelity version.
- For a high-fidelity software prototype common environments include Macromedia Director, Visual Basic, and Smalltalk.
- Danger that users think they have a full system.....see compromises

# *Compromises in prototyping*

- All prototypes involve compromises
- For software-based prototyping maybe there is a slow response? sketchy icons? limited functionality?
- Two common types of compromise
  - 'horizontal': provide a wide range of functions, but with little detail
  - 'vertical': provide a lot of detail for only a few functions
- Compromises in prototypes mustn't be ignored.  
Product needs engineering

# *Construction*

- Taking the prototypes (or learning from them) and creating a whole
- Quality must be attended to: usability (of course), reliability, robustness, maintainability, integrity, portability, efficiency, etc
- Product must be engineered
  - Evolutionary prototyping
  - 'Throw-away' prototyping

# *Conceptual design: from requirements to design*

- Transform user requirements/needs into a conceptual model
- “a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended”
- Don’t move to a solution too quickly. Iterate, iterate, iterate
- Consider alternatives: prototyping helps



# *Three perspectives for a conceptual model*

- Which interaction mode?

How the user invokes actions

Activity-based: instructing, conversing, manipulating and navigating, exploring and browsing.

Object-based: structured around real-world objects

# *Three perspectives for a conceptual model*

- Which interaction paradigm?
  - desktop paradigm, with WIMP interface (windows, icons, menus and pointers),
  - ubiquitous computing
  - pervasive computing
  - wearable computing
  - mobile devices and so on.
- Is there a suitable metaphor?  
(contd)....

# *Is there a suitable metaphor?*

- Interface metaphors combine familiar knowledge with new knowledge in a way that will help the user understand the product.
- Three steps: understand functionality, identify potential problem areas, generate metaphors
- Evaluate metaphors:
  - How much structure does it provide?
  - How much is relevant to the problem?
  - Is it easy to represent?
  - Will the audience understand it?
  - How extensible is it?

# *Expanding the conceptual model*

- What functions will the product perform?  
What will the product do and what will the human do (task allocation)?
- How are the functions related to each other?  
sequential or parallel?  
categorisations, e.g. all actions related to telephone memory storage
- What information needs to be available?  
What data is required to perform the task?  
How is this data to be transformed by the system?

# *Using scenarios in conceptual design*

- Express proposed or imagined situations
- Used throughout design in various ways
  - scripts for user evaluation of prototypes
  - concrete examples of tasks
  - as a means of co-operation across professional boundaries
- Plus and minus scenarios to explore extreme cases

# *Using prototypes in conceptual design*

- Allow evaluation of emerging ideas
- Low-fidelity prototypes used early on, high-fidelity prototypes used later

# *Physical design: getting concrete*

- Considers more concrete, detailed issues of designing the interface
- Iteration between physical and conceptual design
- Guidelines for physical design
  - Nielsen's heuristics
  - Shneiderman's eight golden rules
  - Styles guides: commercial, corporate
    - decide 'look and feel' for you
    - widgets prescribed, e.g. icons, toolbar

# *Physical design: getting concrete*

- Different kinds of widget (dialog boxes, toolbars, icons, menus etc)

- menu design

- icon design

- screen design

- information display



# *Menu design*

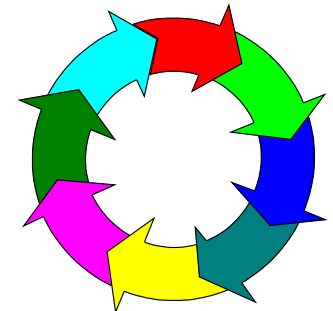
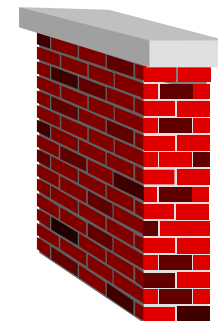
- How long is the menu to be?
- In what order will the items appear?
- How is the menu to be structured, e.g. when to use sub-menus, dialog boxes?
- What categories will be used to group menu items?

# *Menu design*

- How will division into groups be denoted, e.g. different colors, dividing lines?
- How many menus will there be?
- What terminology to use? (results of requirements activities will indicate this)
- How will any physical constraints be accommodated, e.g. mobile phone?

# *Icon design*

- Good icon design is difficult
- Meaning of icons is cultural and context sensitive
- Some tips:
  - always draw on existing traditions or standards
  - concrete objects or things are easier to represent than actions
- From clip art, what do these mean to you?



# *Screen design*

Two aspects:

- How to split across screens
  - moving around within and between screens
  - how much interaction per screen?
  - serial or workbench style?
- Individual screen design
  - white space: balance between enough information/interaction and clarity
  - grouping items together: separation with boxes? lines? colors?

# *Screen design: splitting functions across screens*

- Task analysis as a starting point
- Each screen contains a single simple step?
- Frustration if too many simple screens
- Keep information available: multiple screens open at once

# *Screen design: individual screen design*

- Draw user attention to salient point, e.g. colour, motion, boxing
- Animation is very powerful but can be distracting
- Good organization helps: grouping, physical proximity
- Trade off between sparse population and overcrowding

# *Information display*

- Relevant information available at all times
- Different types of information imply different kinds of display
- Consistency between paper display and screen data entry

# *Summary*

- Different kinds of prototyping are used for different purposes and at different stages
- Prototypes answer questions, so prototype appropriately
- Construction: the final product must be engineered appropriately
- Conceptual design (the first step of design)
- Physical design: e.g. menus, icons, screen design, information display
- Prototypes and scenarios are used throughout design