Tue 31 Mar

# Lecture 9 & 10: Design, Prototyping and Construction

# Overview

- Prototyping and construction
- Conceptual design
- Using scenarios and storyboard in design

Based on material from:
Sharp, H., Preece, J. and Rogers, Y. (2019). *Interaction Design Beyond Human-Computer Interaction*. 5th Ed. Sussex: John Wiley & Sons Ltd.

# Learning Outcomes

- Describe prototyping and different types of prototyping activities

- Enable you to produce simple prototype from the models developed during the requirement activity

# 1. Prototyping & Construction

- What is a prototype?
- Why prototype?
- What to prototype? Layout, design concept, features
- Different kinds of prototyping
  - low fidelity
  - high fidelity
- Construction

# 1.1 What is Prototype?

- In other design fields a prototype is a small-scale model:
  - a miniature car
  - a miniature building or town
- The example here comes from a 3D printer

(c)

(c) A teddy bear ‘printed’ from a wireframe design http://www.disneyresearch.com/project/printed-teddy-bears/

(c) Courtesy of Scott Hudson, Human–Computer Interaction Institute, Carnegie Mel- Ion University.

# 1.1 What is Prototype? (cont)

- 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

# 1.2 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

# Filtering dimensions of prototyping

| Filtering dimension | Example variables |
|---|---|
| Appearance | size; color; shape; margin; form; weight; texture; proportion; hardness; transparency; gradation; haptic; sound |
| Data | data size; data type (e.g., number; string; media); data use; privacy type; hierarchy; organization |
| Functionality | system function; users' functionality need |
| Interactivity | input behavior; output behavior; feedback behavior; information behavior |
| Spatial structure | arrangement of interface or information elements; relationship among interface or information elements – which can be either two-or three-dimensional, intangible or tangible, or mixed |

# Manifestation dimensions of prototyping

| Manifestation dimension | Definition | Example variables |
|---|---|---|
| Material | Medium (either visible or invisible) used to form a prototype | Physical media, e.g. paper, wood, and plastic; tools for manipulating physical matters, e.g. knife, scissors, pen, and sand-paper; computational prototyping tools, e.g. Macromedia Flash and Visual Basic; physical computing tools, e.g. Phidgets and Basic Stamps; available existing artifacts, e.g. a beeper to simulate a heart attack |
| Resolution | Level of detail or sophistication of what is manifested (corres-ponding to fidelity) | Accuracy of performance, e.g. feedback time responding to an input by a user (giving user feedback in a paper prototype is slower than in a computer-based one); appearance details; interactivity details; realistic versus faked data |
| Scope | Range of what is covered to be mani-fested | Level of contextualization, e.g. website color scheme testing with only color scheme charts or color schemes placed in a website layout structure; book search navigation usability testing with only the book search related inter-face or the whole navigation interface |

**Table 11.2** The definition and variables of each manifestation dimension

# 1.3 What to Prototype?

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

# 1.4 Low-fidelity Prototype?

- 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

# 1.4 Low-fidelity Prototype?

a) <u>Storyboard</u>

- 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

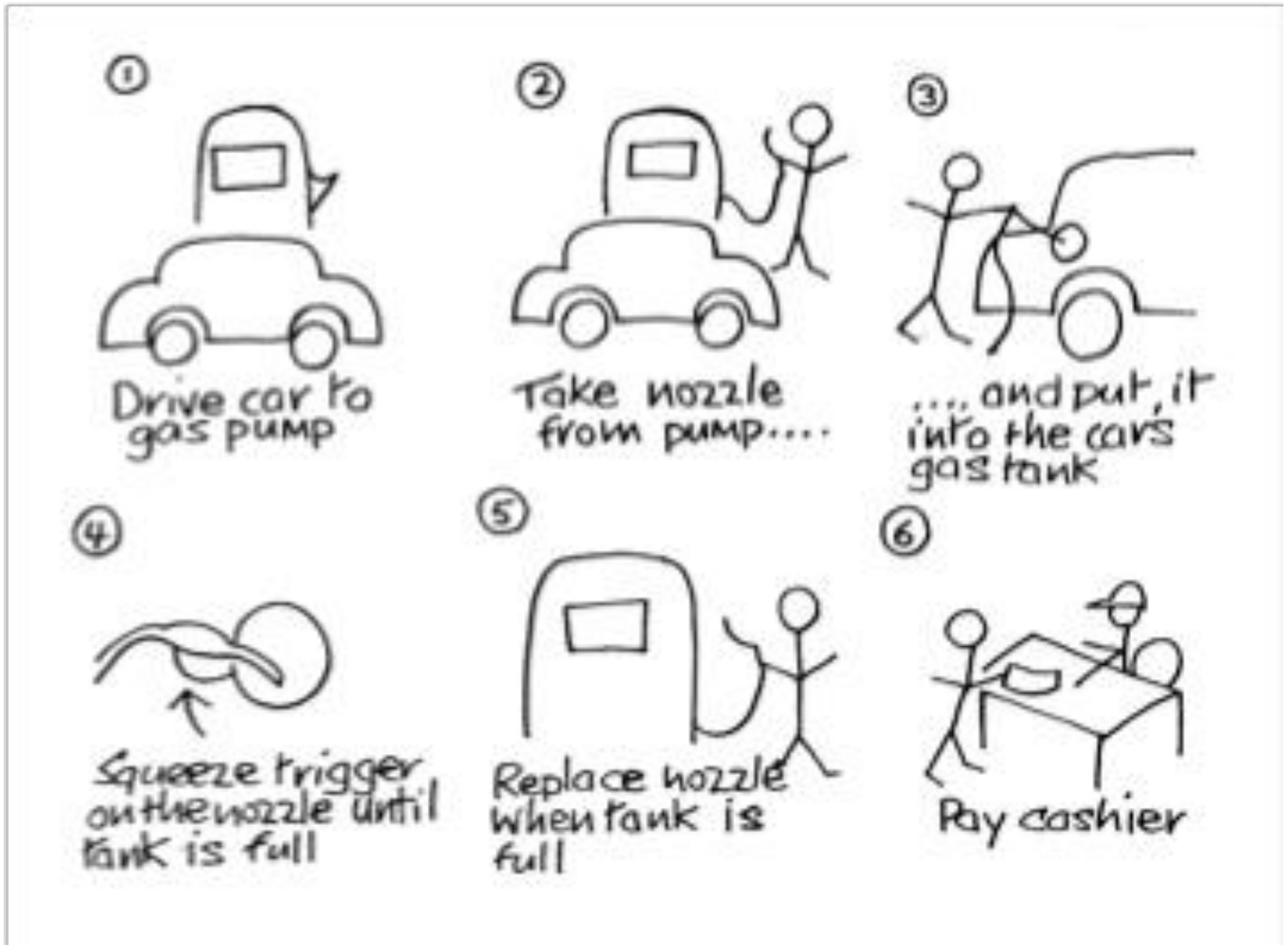
- Used early in design

# 1.4 Low-fidelity Prototype?

## b) Sketching

- Sketching is important to low-fidelity prototyping

- Don’t be inhibited about drawing ability. Practice simple symbols



# 1.4 Low-fidelity Prototype?

## c) Card-based

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

# 1.5 High-fidelity Prototype?

- 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 etc

- Danger that users think they have a full system

# 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
  1. horizontal: provide a wide range of functions, but with little detail
  2. vertical: provide a lot of detail for only a few functions
- Compromises in prototypes mustn't be ignored. Product needs engineering

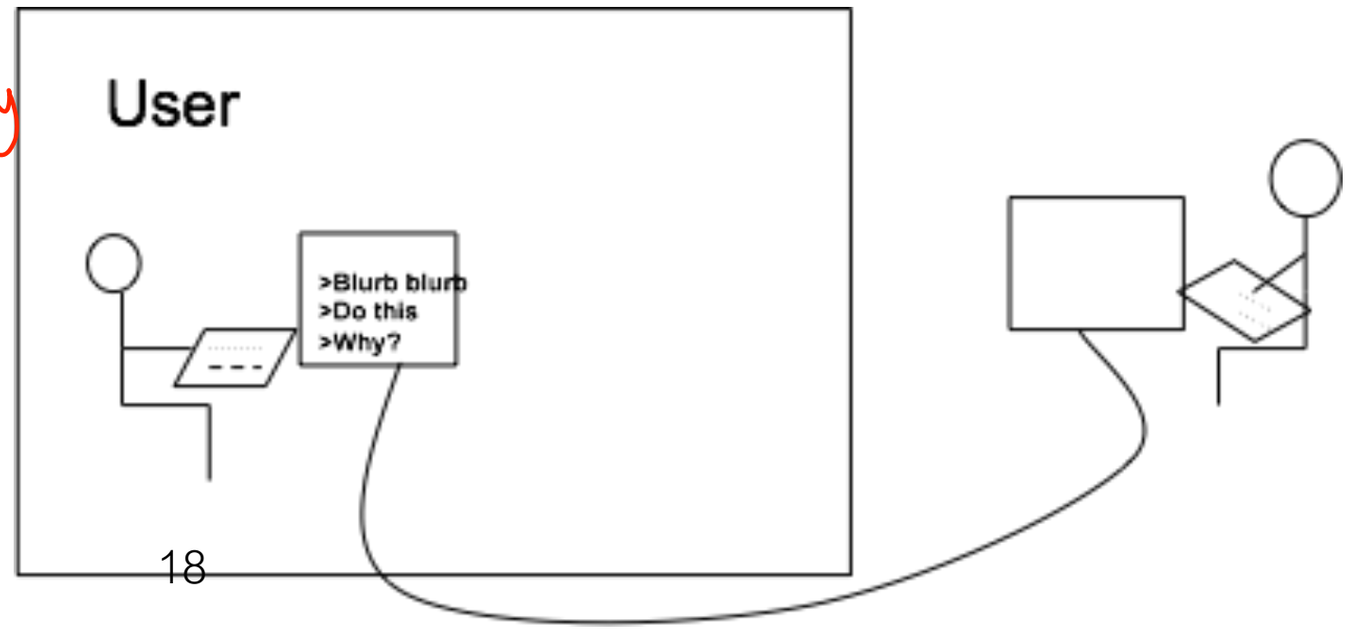
# 1.5 High-fidelity Prototype? (cont)

Wizard-of-Oz

video → interfactor intelligent elevator

- 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?

User think they deal w/ system, but actually



# Low-fidelity VS High-fidelity Prototype?

| Type | Advantages | Disadvantages |
|---|---|---|
| Low-fidelity | - Lower development cost<br>- Evaluate multiple design concepts<br>- Useful communication device<br>- Address screen layout issues<br>- Useful for identifying market requirements<br>- Proof-of-concept | - Limited error checking<br>- Poor detailed specification to code to<br>- Facilitator-driven we walk users thru it<br>- Limited utility after requirements established<br>- limited usefulness for usability tests<br>- Navigational and flow limitations |
| High-fidelity | - Complete functionality<br>- Fully interactive<br>- User-driven → freedom for navigation<br>- Clearly defines navigational scheme<br>- Use for exploration and test<br>- Look & feel of final product<br>- Serves as a living specification<br>- Marketing and sales tool | - More expensive to develop<br>- Time-consuming to create<br>- Inefficient for proof-of-concept design<br>- Not effective for requirements gathering |

# 1.6 Construction

- Taking the prototypes (or learning from them) and creating a whole

- Quality must be attended to:
  - usability (of course) Project
  - reliability
  - robustness
  - maintainability
  - integrity
  - portability
  - efficiency
  - etc

# 2. Conceptual design: from requirements to design

- Transform user requirements/needs into a conceptual model - the first step of design

- "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"

- Mood board may be used to capture feel

- Don't move to a solution too quickly. Iterate, iterate, iterate

# 2. Conceptual design: from requirements to design (cont)

- Which metaphors would be suitable to help users understand the product?

- Which interaction type(s) would best support the users' activities?

- Do different interface types suggest alternative design insights or options?

# 2.1 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?

# 2.2 Considering interaction and interface types

- Which interaction type?
  - How the user invokes actions
    - Instructing, conversing, manipulating or exploring
- Do different interface types provide insight?
  - shareable, tangible, augmented reality, etc.

# 2.3 Expanding the initial 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 privacy on a smartphone

- What information is needed?

  - What data is required to perform the task?

  - How is this data to be transformed by the system?

# 2.4 Concrete design

- Many aspects to concrete design
  - Color, icons, buttons, interaction devices etc.
- User characteristics and context
  - Accessibility, cross-cultural design
- Cultural website guidelines → ex. date language

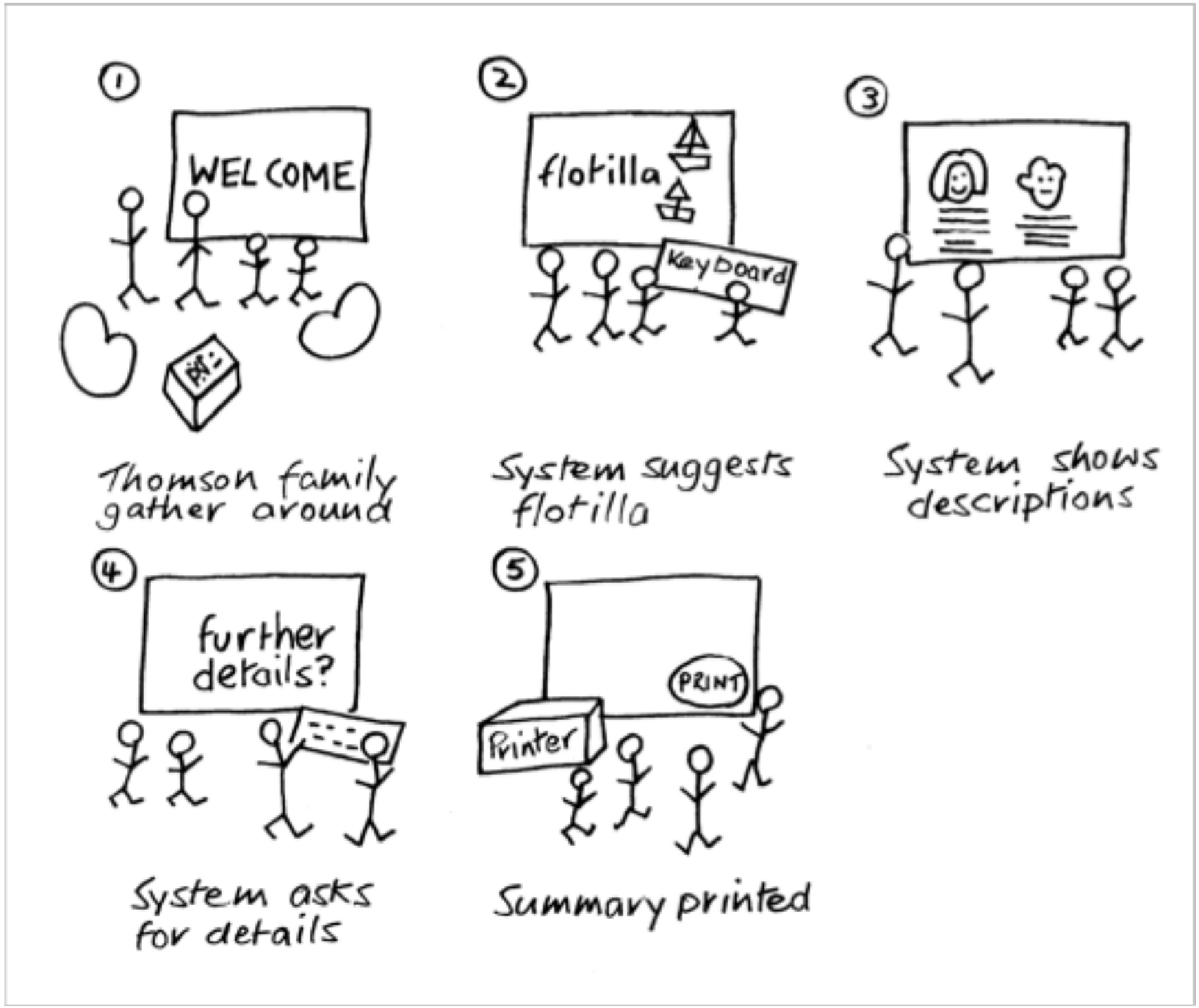
  successful products “are … bundles of social solutions. Inventors succeed in a particular culture because they understand the values, institutional arrangements, and economic notions of that culture.”

# 3. 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

# 3.1 Using prototype in design

- Generating storyboards from scenarios



# Generate storyboard from scenario

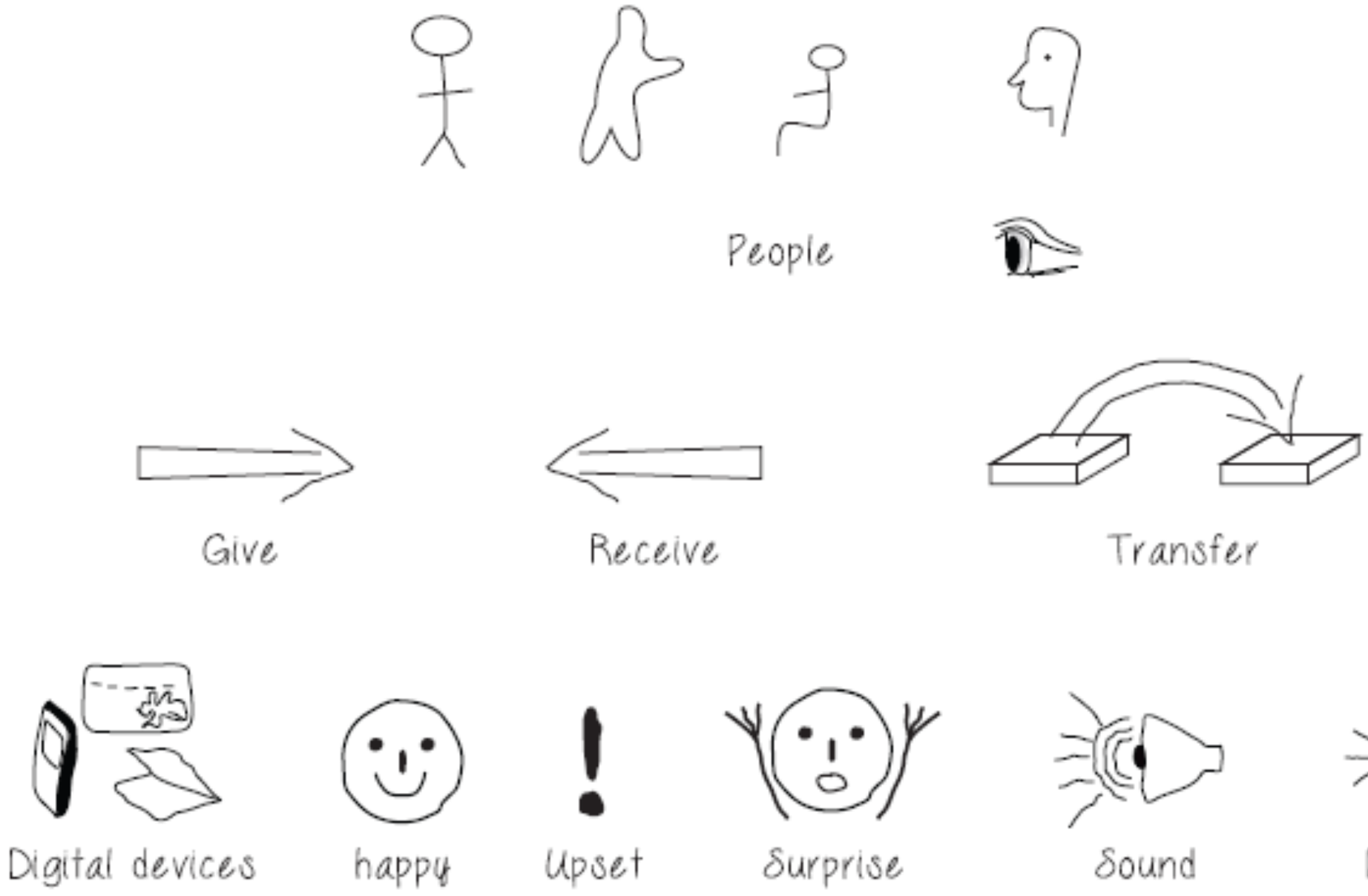



**Figure 11.4** Some simple sketches for low-fidelity prototyping

# Generate card-based prototype from use case

**Figure 11.6** Prototype developed for cell phone user interface

# Construction: Physical Computing

- Build and code prototypes using electronics
- Toolkits available include
  - Arduino
  - LilyPad (for fabrics)
  - Senseboard
  - MaKey MaKey
- Designed for use by wide range of people

# Physical computing kits

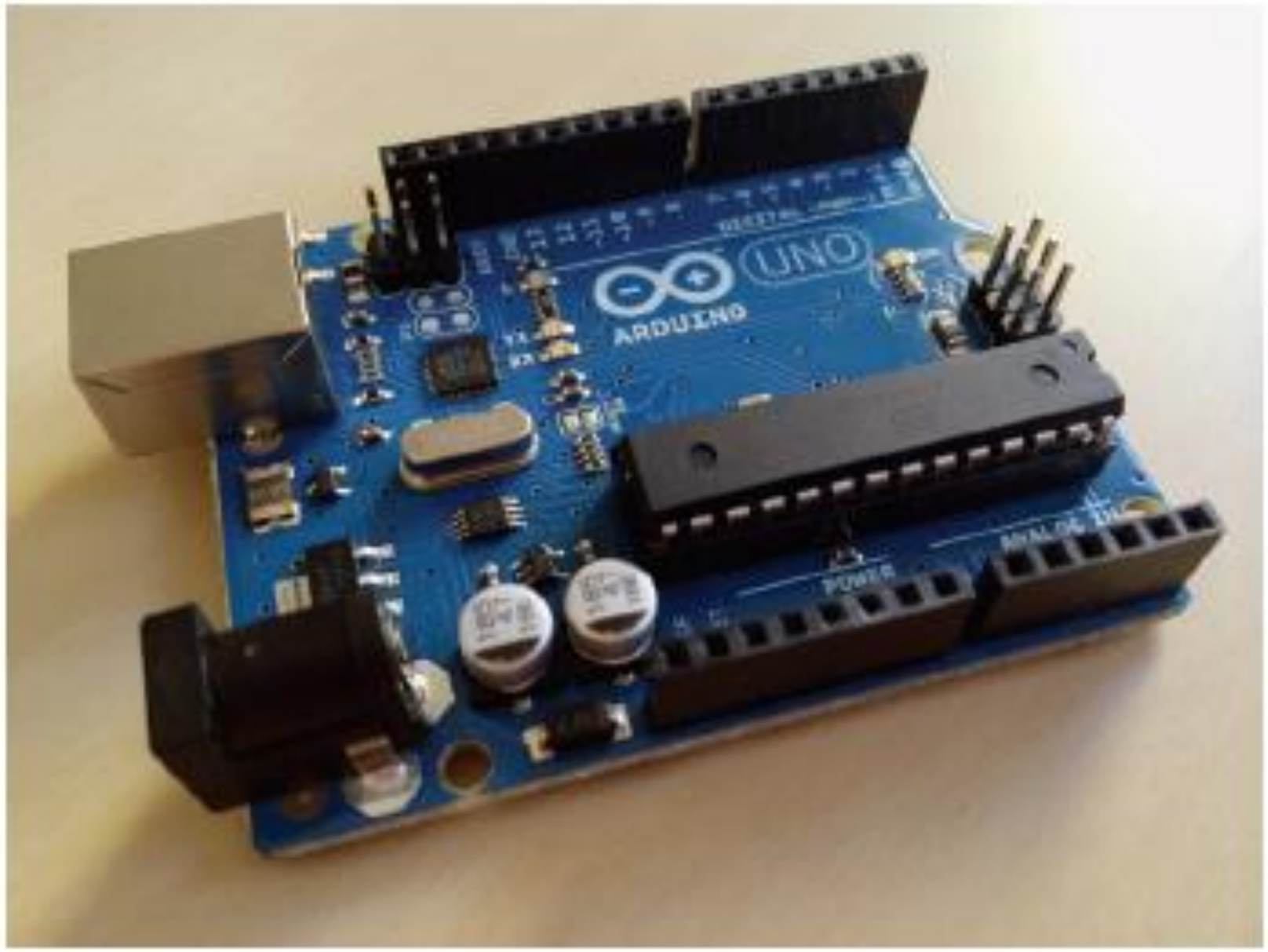



**Figure 11.22** The Arduino board

*Source:* Courtesy of Nicolai Marquardt

# Physical computing kits

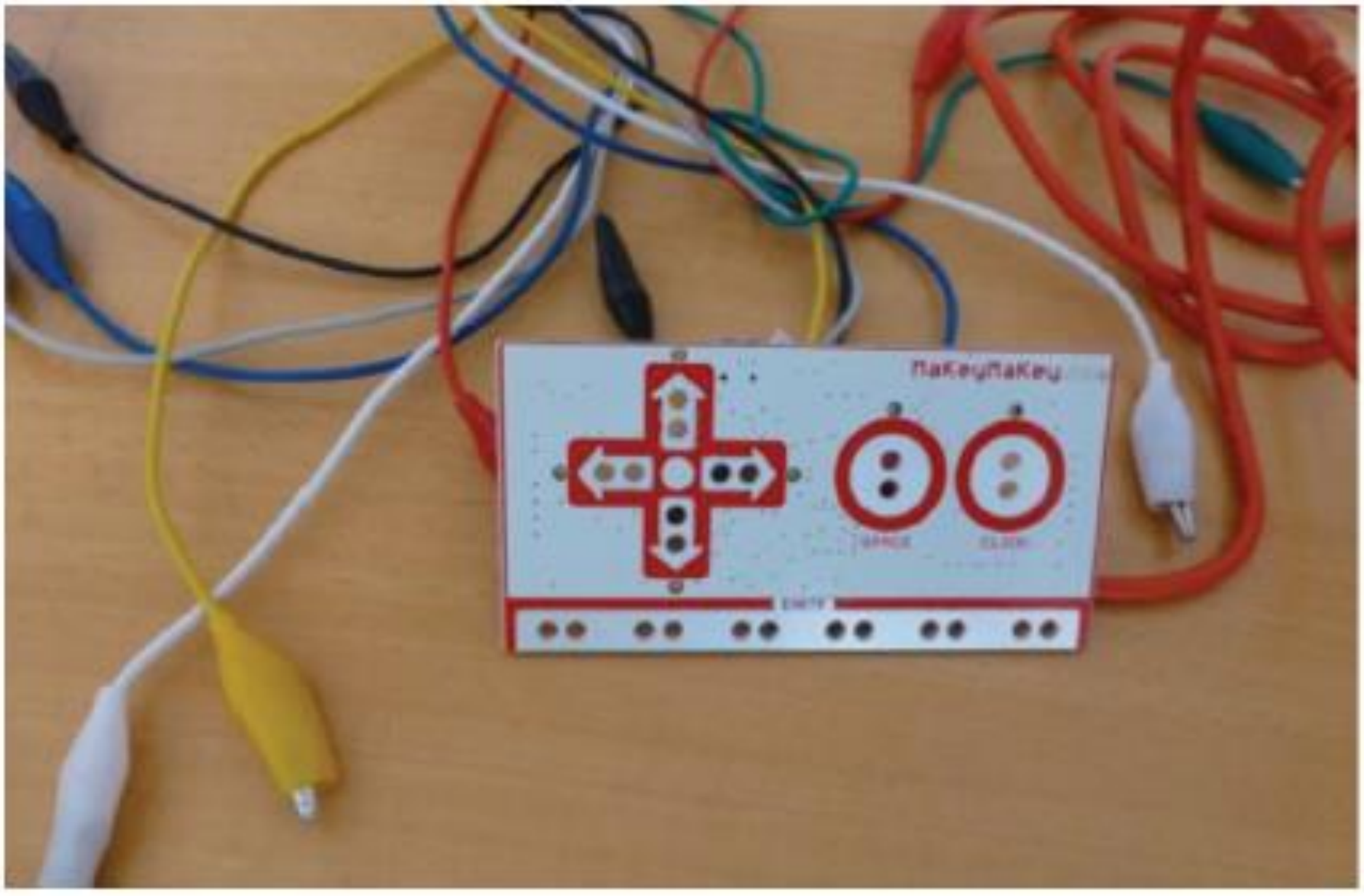



**Figure 11.24** The MaKey MaKey toolkit

# Physical computing kits

**Figure 11.25** A group of retired friends playing with a MaKey MaKey toolkit

# 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)
- Consider interaction types and interface types to prompt creativity
- Storyboards can be generated from scenarios
- Card-based prototypes can be generated from use cases