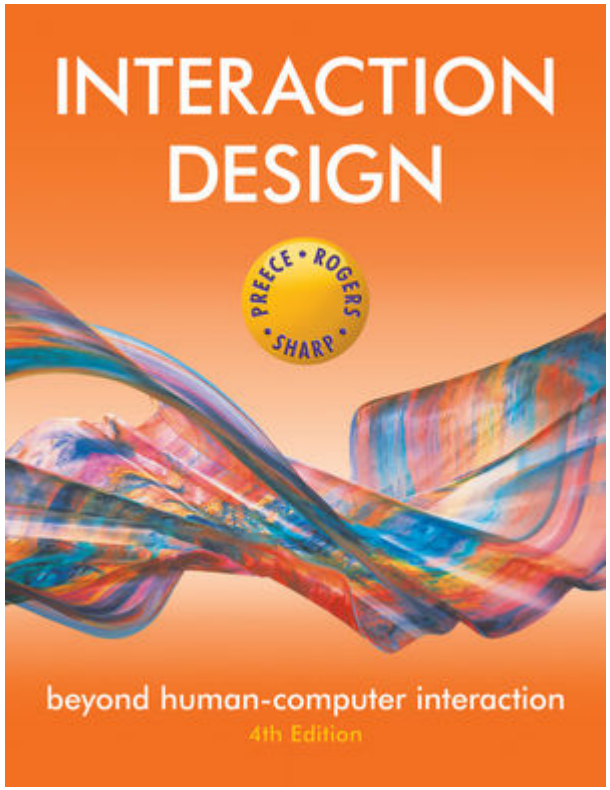
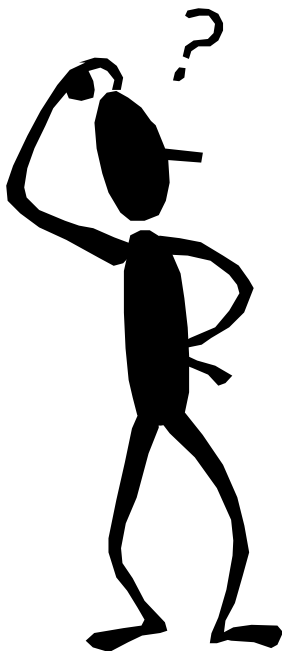


Chapter 11

DESIGN, PROTOTYPING and CONSTRUCTION

Overview

- Prototyping
- Conceptual design
- Concrete design
- Using scenarios
- Generating prototypes
- Construction



Prototyping

- What is a prototype?
- Why prototype?
- Different kinds of prototyping
  - Low fidelity
  - High fidelity
- Compromises in prototyping
  - Vertical
  - Horizontal
- Final product needs to be engineered

What is a prototype?

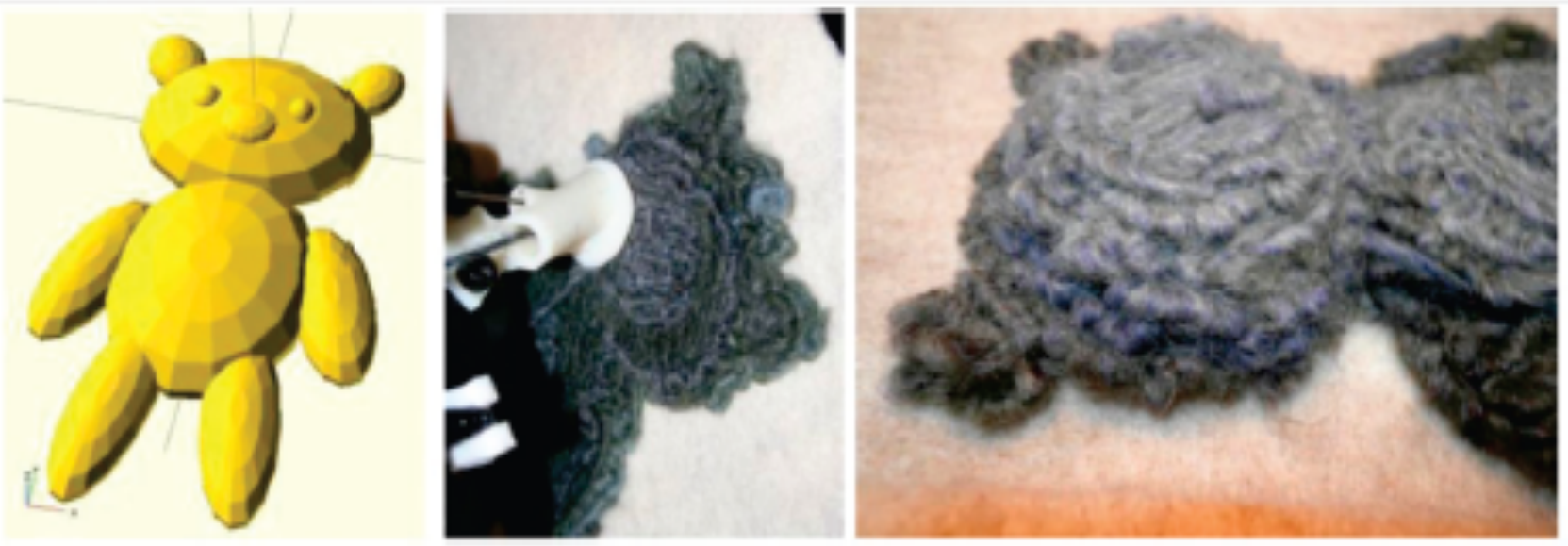
In other design fields a prototype is a small-scale model:

- a miniature car
- a miniature building or town
- the examples here come from a 3D printer

(a)

Figure 11.1 (a) Color output from a 3D printer: all the gears and rods in this model were 'printed' in one pass from bottom to top, and when one gear is turned, the others turn too.

Source: (a) The Computer Language Company, Inc., courtesy of Alan Freedman



(c)

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

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

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

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

Example storyboard

Type	Advantages	Disadvantages
Low-fidelity prototype	Lower development cost Evaluates multiple design concepts Useful communication device Addresses screen layout issues Useful for identifying market requirements Proof of concept	Limited error checking Poor detailed specification to code to Facilitator-driven Limited utility after requirements established Limited usefulness for usability tests Navigational and flow limitations
High-fidelity prototype	Complete functionality Fully interactive User-driven Clearly defines navigational scheme Use for exploration and test Look and feel of final product Serves as a living specification Marketing and sales tool	More resource-intensive to develop Time-consuming to create Inefficient for proof-of-concept designs Not effective for requirements gathering

Table 11.3 Advantages and disadvantages of low- and high-fidelity prototypes

Sketching

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

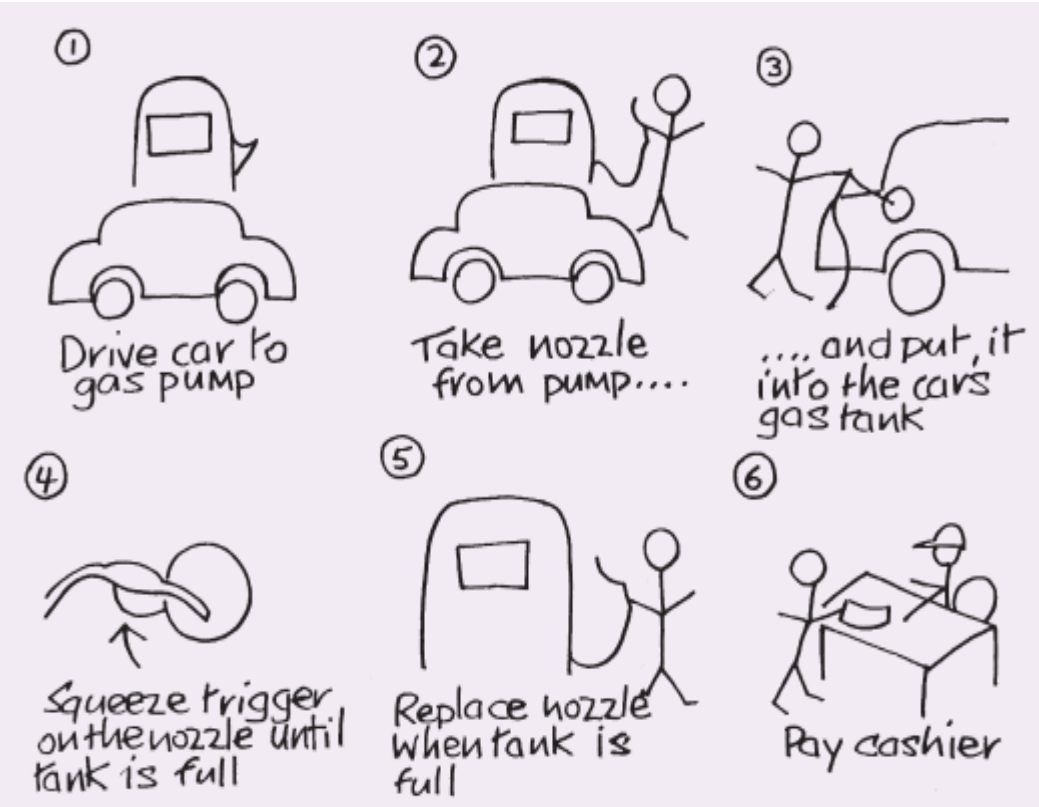
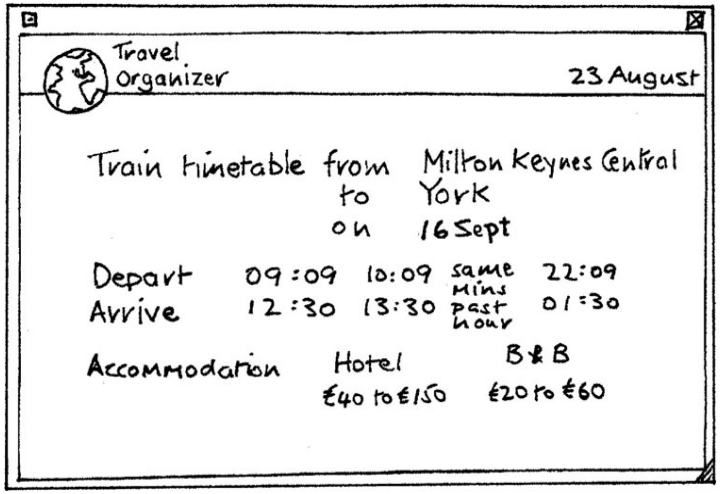


Figure 11.5 A storyboard depicting how to fill a car with gas

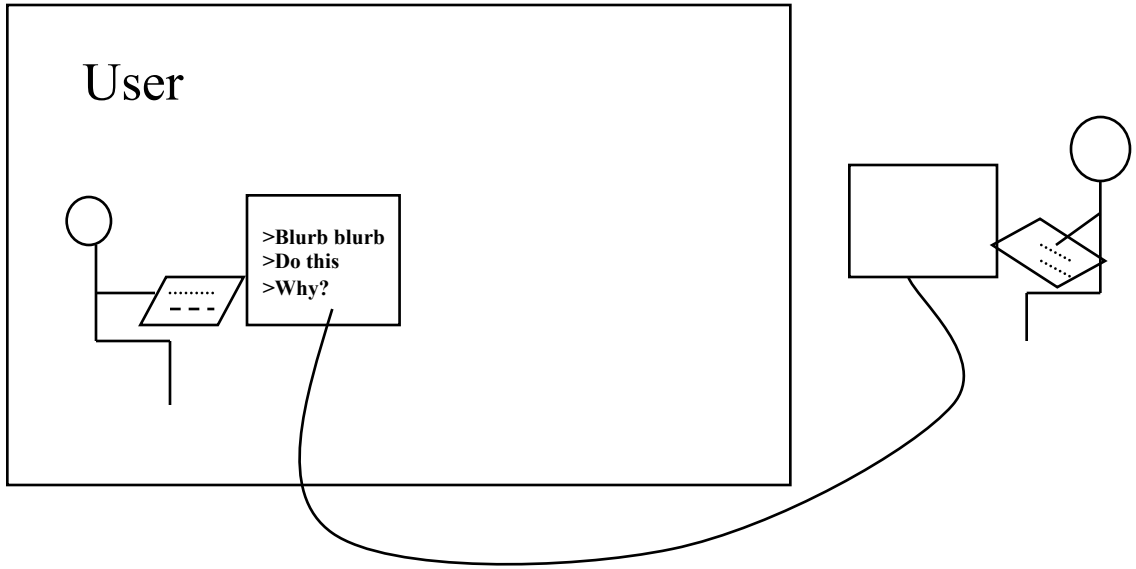
Card-based prototypes



- Index cards (3 X 5 inches)
- Each card represents one screen or part of 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
- High-fidelity prototypes can be developed by integrating existing hardware and software components
- Danger that users think they have a complete 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

Conceptual design

- Transform user requirements/needs into a conceptual model
- A conceptual model is an outline of what people can do with a product and what concepts are needed to understand and interact with it
- Mood board may be used to capture feel
- Consider alternatives: prototyping helps

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?

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.

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?

Concrete design

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

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."

Using scenarios

- Express proposed or imagined situations
- Used throughout design in various ways
  - as a basis for overall design
  - 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

Generate storyboard from scenario

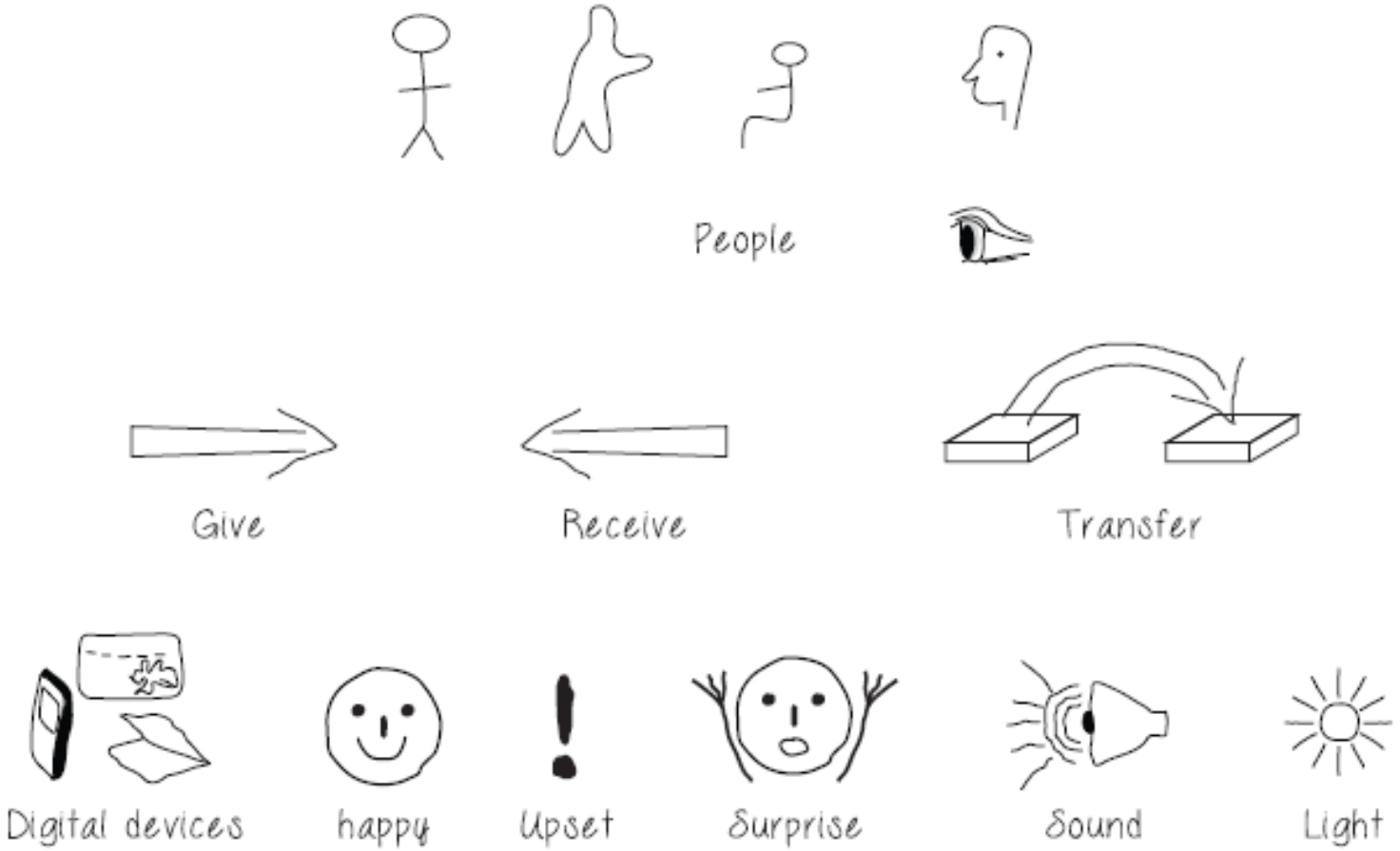


Figure 11.4 Some simple sketches for low-fidelity prototyping

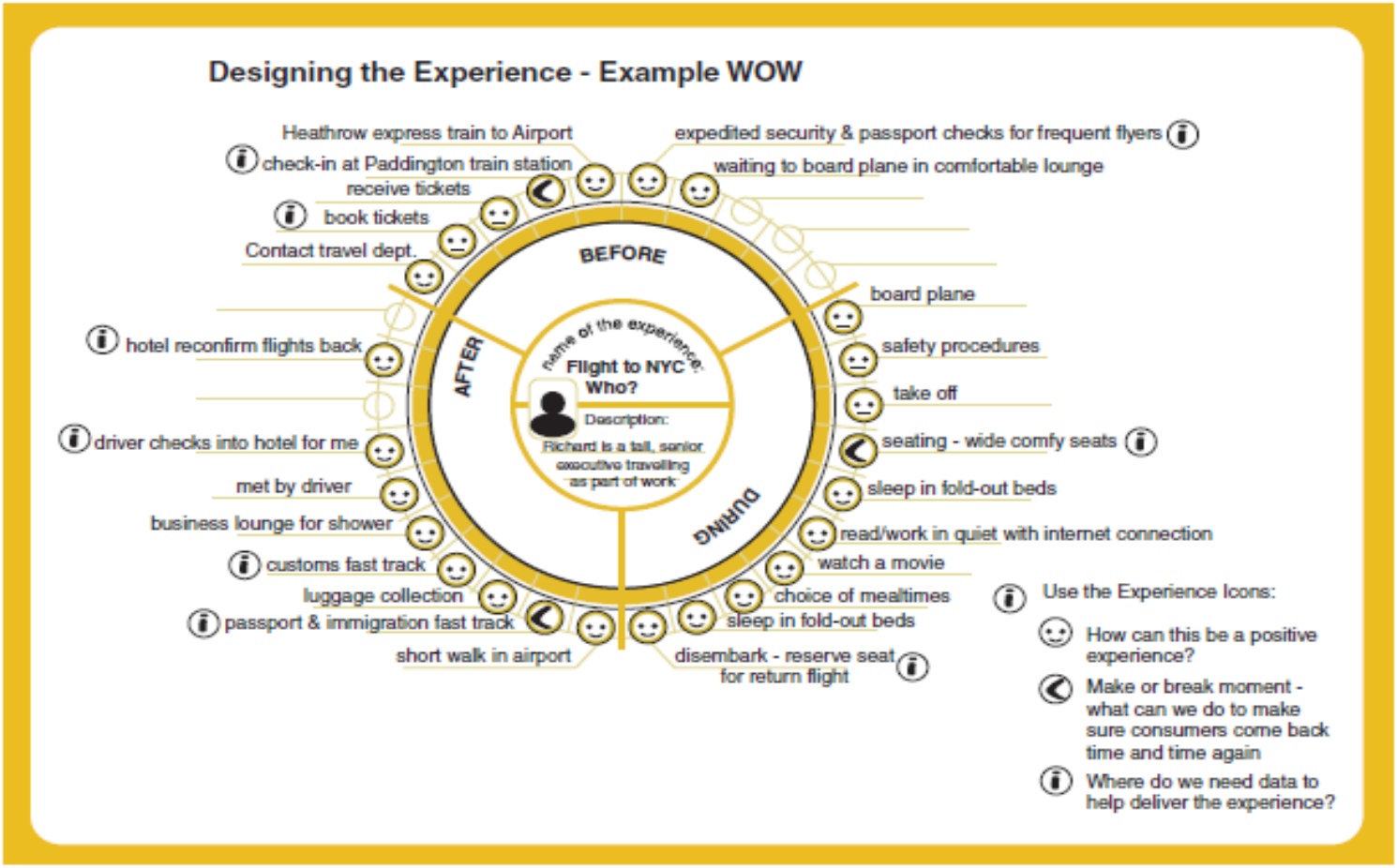
Generate card-based prototype from use case

Figure 11.6 Prototype developed for cell phone user interface

Explore the user's experience

- Use personas, card-based prototypes or stickies to model the user experience
- Visual representation called:
  - design map
  - customer/user journey map
  - experience map
- Two common representations
  - wheel
  - timeline

An experience map drawn as a wheel

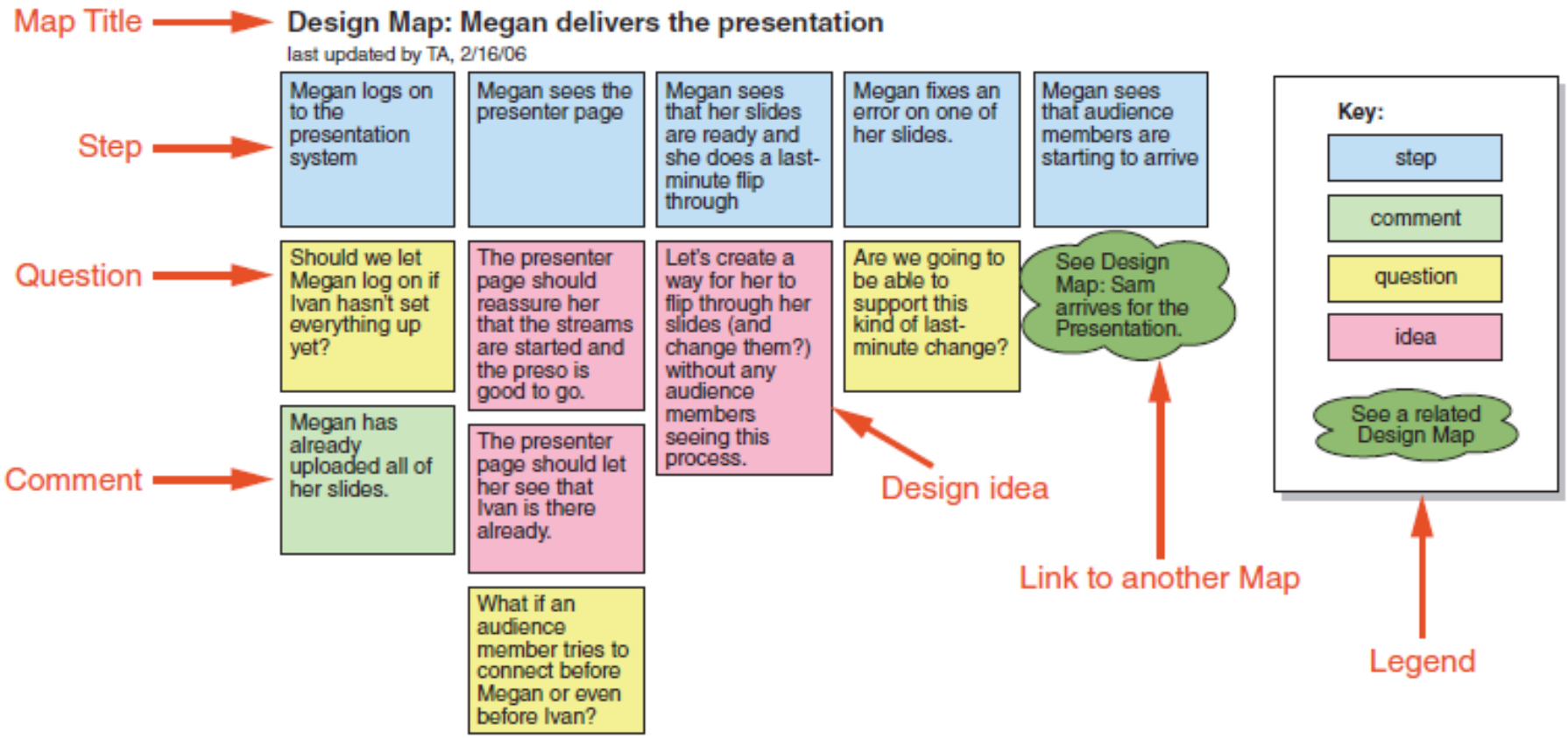


(a)

Figure 11.19 (a) An experience map using a wheel representation. (b) An example timeline design map illustrating how to capture different issues.

Source: (a) http://www.ux-lady.com/experience-maps-user-journey-and-more-exp-map-layout/ (b) Adlin, T. and Pruitt, J. (2010) *The Essential Persona Lifecycle: Your guide to building and using personas*. Morgan Kaufmann p. 134.

An experience map drawn as a timeline



(b)

Figure 11.19 Continued

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

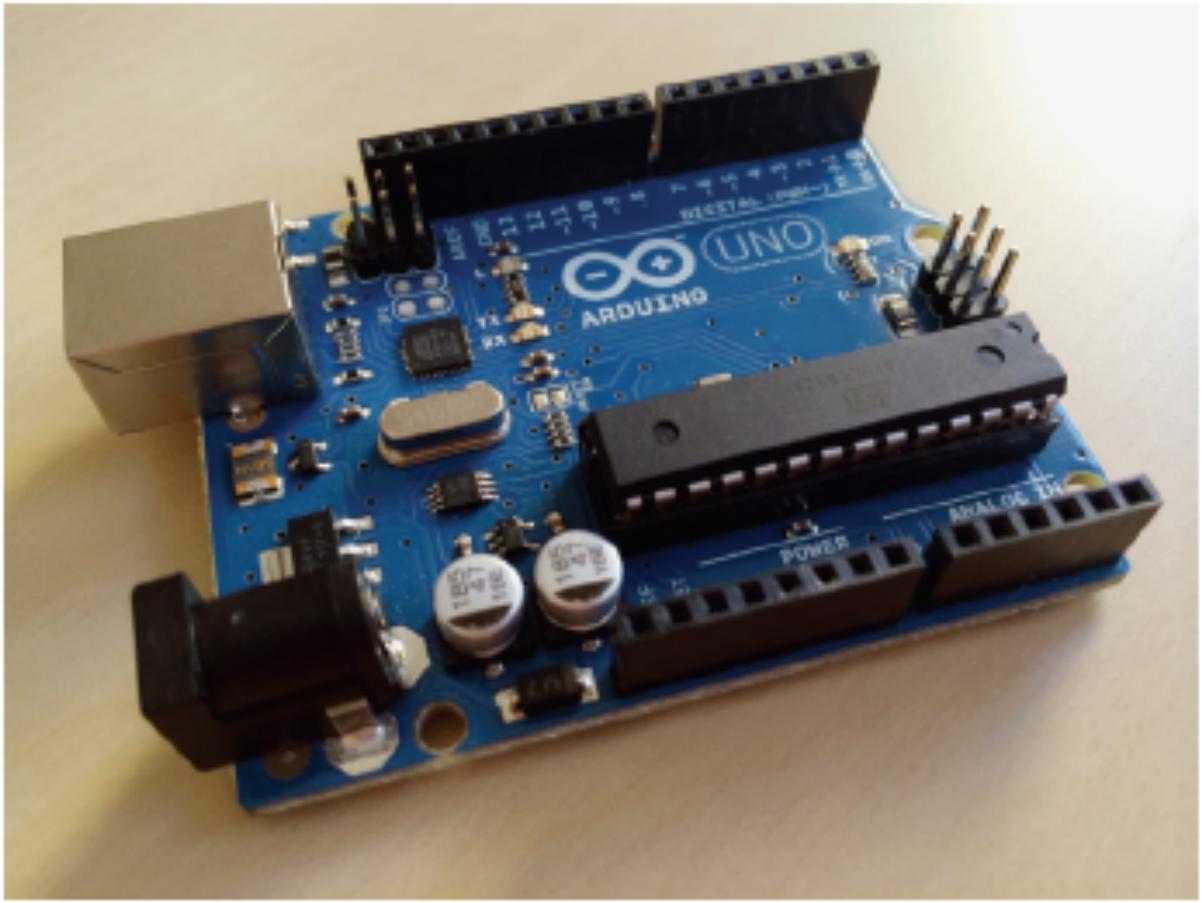


Figure 11.22 The Arduino board

Source: Courtesy of Nicolai Marquardt

Physical computing kits

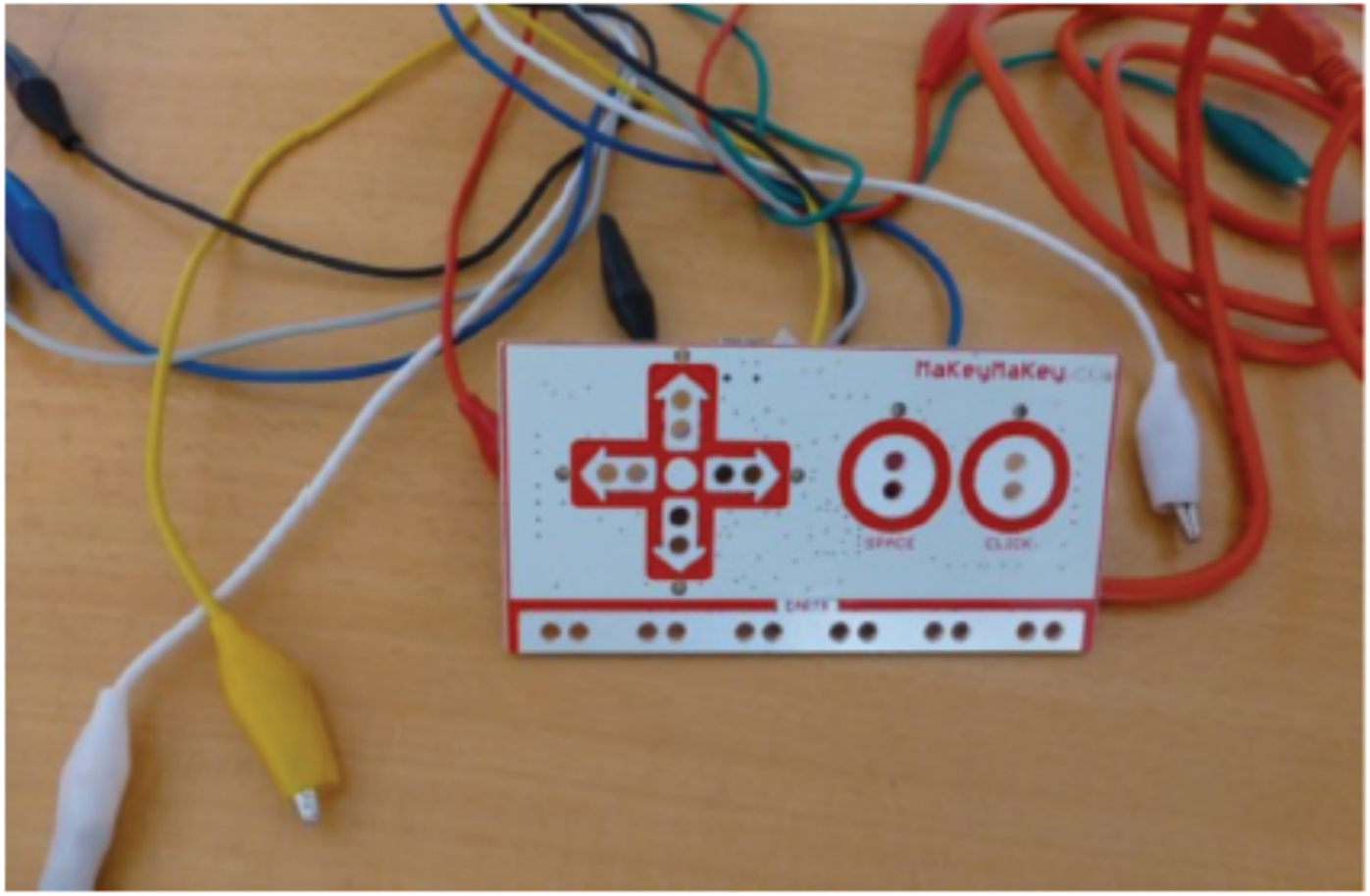


Figure 11.24 The MaKey MaKey toolkit

Physical computing kits

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

Construction: SDKs

- Software Development Kits
  - programming tools and components to develop for a specific platform, e.g. iOS
- Includes: IDE, documentation, drivers, sample code, application programming interfaces (APIs)
- Makes development much easier
- Microsoft's Kinect SDK has been used in research

Summary

- Different kinds of prototyping are used for different purposes and at different stages
- Prototypes answer questions
- The final product must be engineered appropriately
- Two aspects of design: conceptual and concrete
- To generate conceptual design, consider interface metaphors, interaction types and interface types
- Storyboards can be generated from scenarios
- Card-based prototypes can be generated from use cases
- Physical computing kits and SDKs facilitate transition from design to construction