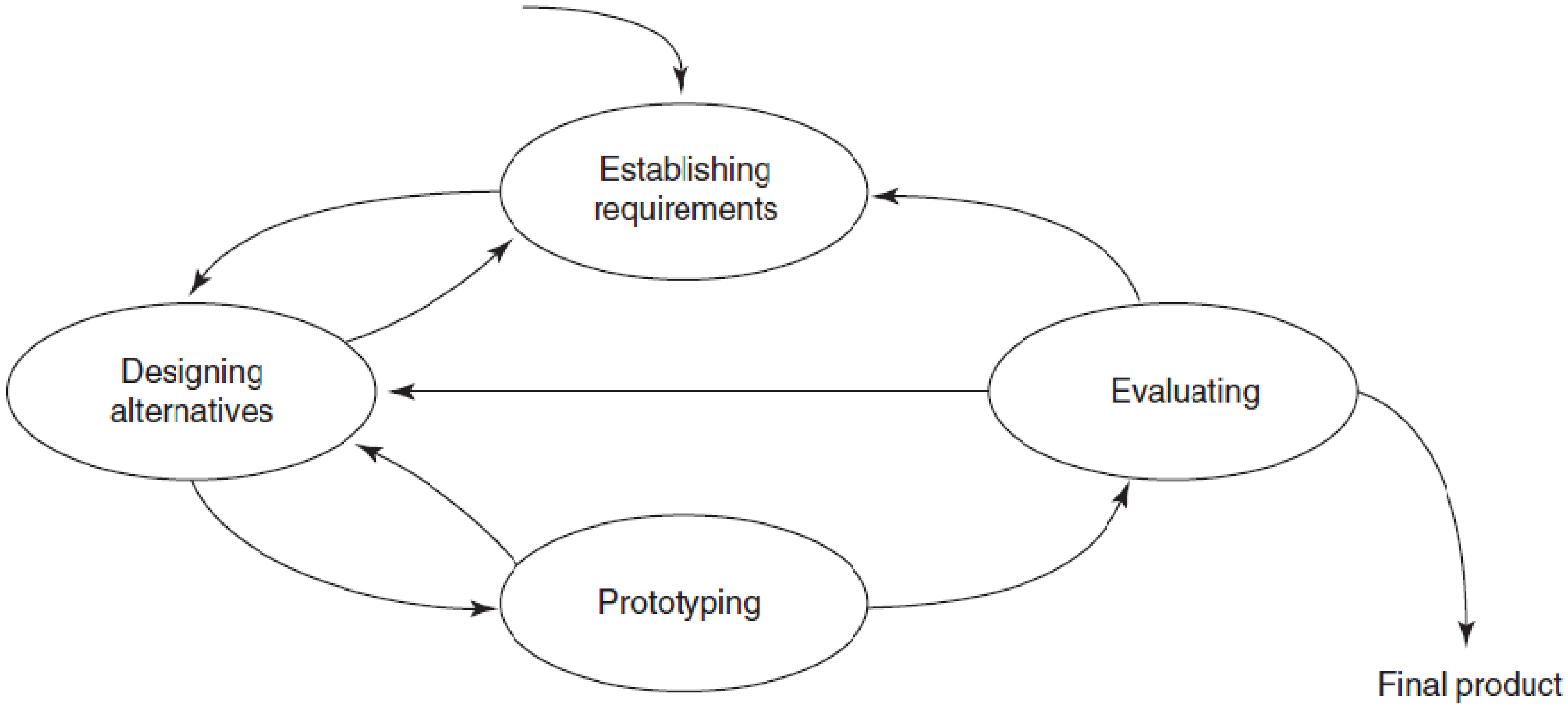
Design and Prototyping

- Conceptual Design
- Concrete Design
- Design Rationale
- Prototyping

Design and Prototyping

Recall the simple HCI lifecycle model:



Design: ideas

Build an interactive version: prototypes to evaluate ideas

Conceptual Design

About transforming requirements into a conceptual model

It is fundamental but difficult to grasp the idea, e.g., conceptual models take many different forms

A conceptual model is an outline of what people can do with a product (obtained from the current functional requirements) and what concepts are needed to understand and interact with it (related to who the user will be, what kind of interface will be used, etc.)

Alternatives are needed

Guiding principles of conceptual design:

- Keep an open mind but never forget the users and context
- Discuss ideas with other stakeholders as much as possible
- Use prototyping to get rapid feedback
- Iterate, iterate and iterate

Conceptual Design

3 perspectives to develop the conceptual model:

1. Which interaction type?

- Type refers to how the user invokes actions when interacting with the system (may not be mutually exclusive)

  Instructing: user gives instructions to the system to perform his task such as typing in commands, selecting options from menus in windows environment, voice control commands, gesturing

  Conversing: user has a dialog with the system via speaking or typing in questions to which the system replies via speech or text output such as ticket booking, e-banking

Conceptual Design

Manipulating: user interacts with objects in a virtual or physical space by manipulating (e.g., moving, placing) them such as moving a file by dragging its icon, and zooming in an image using fingers

Exploring: user moves through a virtual environment or a physical space by exploiting his knowledge of how he moves and navigates through existing spaces such as exploring the virtual 3D environment of the interior of a building, and in a physical space embedded sensors, user is provided with useful information at a particular time

- Can be obtained through requirements activity
- Suggest interface types, e.g., fill-in form or speech recognition for search engines, WIMP or command line for file printing

Conceptual Design

2. Any metaphor?

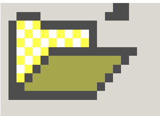
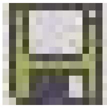
- Help people become familiar with a new system by relating it with real-world activity $\Rightarrow \uparrow$ initial familiarity

  e.g., typewriter was metaphor for word processor: when users saw the keyboard on a computer, they assumed that it behaved similarly to the typewriter’s keyboard

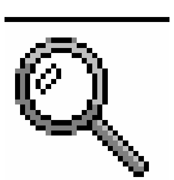
Conceptual Design

e.g., actions of copying, cutting & pasting were well known before computers were invented

- Way to relate a difficult or more abstract concept to a familiar one

  e.g., open file  ,save file 

- Disadvantage: metaphor may not be widely known or correctly understood

  e.g., 

  = find a file?
  = zoom a document?

Conceptual Design

- Three steps are involved: understand functionality, identify potential problem areas, generate metaphors
- To evaluate metaphors, we can ask:
  - How much structure does it provide? A good metaphor will provide structure, and preferably familiar structure
  - How much is relevant to the problem?
  - Is it easy to represent? A good metaphor will be associated with particular visual, audio elements and/or words
  - Will users understand it?
  - How extensible is it? Does it have extra aspects that may be useful later on?

Conceptual Design

Example: Metaphor for an eBook Shelf

It provides the familiar structure of a physical book shelf

Details of book covers are shown and user may know that covers are relevant to the available books

It is easy to associate books with the book covers

It is believed that users can easily understand it

It is easily extended to multiple book shelfs

Conceptual Design

3. Any interaction paradigm?

- Way of thinking about interaction design based on technology advances and creativity
- Not well-defined

Examples of paradigms:

- Time-sharing
  - 1940s-1950s: explosive technological growth
  - 1960s: need to channel computing power

  $\Rightarrow$ single computer that supports multiple users

Conceptual Design

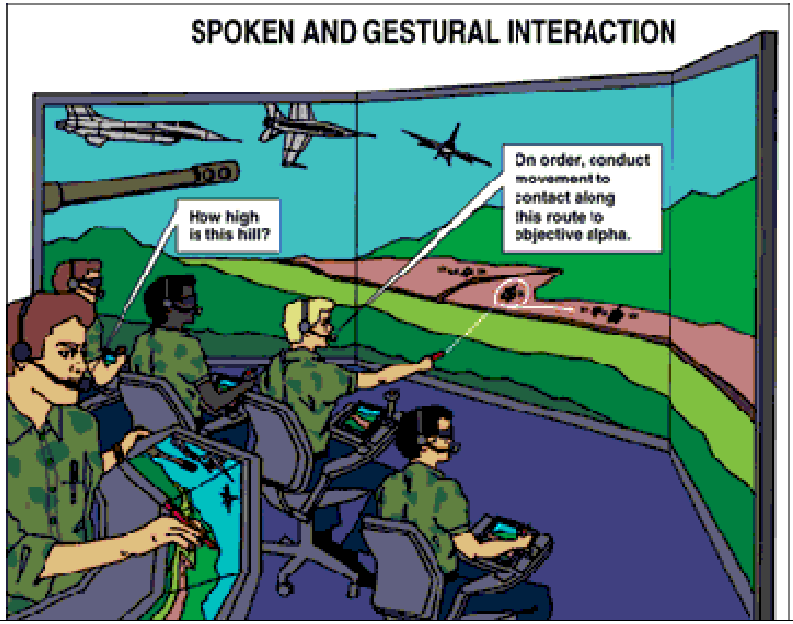
- Window systems & WIMP interface
  - Motivation: human can pursue more than one task at time
  - Windows are used for dialogue partitioning & to “change the topic”
  - 1981: Xerox introduced the first commercial window system
  - Windows, icons, menus & pointers are now familiar interaction mechanisms

Conceptual Design

- Multi-modality
  - Rely on the uses of multiple human communication channels, e.g., visual, touch, speech
  - Each different channel for the user is referred to as a modality of interaction
  - Simple multi-modal systems:
    - Visual + touch
      e.g., editing using a computer
    - Speech + touch
      e.g., telephone banking, systems for visually impaired
  - Complex multi-modal systems:
    - Extract & combine meaningful multimodal information from multiple, rich, multidimensional inputs

Conceptual Design

- Example : QuickSet



- Input modes: pen, speech, hand-held PC
- Unimodal (speech)
  "Attack the chemical plant at the base of the hill, to the west side of the lake, with missiles"
- Multimodal (speech + pen)
  [circles killzone] "missiles strike here"

Conceptual Design

New paradigms include

- Ubiquitous computing ("ubicomp"):
  - Technology is embedded in the environment: when using the system, we are not aware that it is computer
  - Computers would be designed to be embedded in the environment
  - People should be able to access and interact with information any place and any time
  - It will radically change the way people think about and interact with computers
  - Major rethink of what HCI is in this context

Conceptual Design

Example: iPot

上網家電 可監察年邁父母起居

在通訊科技日趨發達下，會發電郵的家電早就耳熟能詳，但最近像松下電業等大公司，想藉監視會發電郵的熱水瓶或煤氣系統的使用率，發揮家居監護作用，這對於人口日趨老化的日本，商機無限。

共同社報道，這些服務的其中一個好處，是避免子女有事沒事也打電話給年邁父母，從而減少衝突。有自動上網功能的家電會傳送資料到子女的手機和電腦，當他們發現家電的使用頻率與平時不同，才聯絡獨自生活的父母也不遲。

松下電業在新年伊始開了幾個新網站，介紹新菜譜和環保資訊外，還包括了一個照顧家居老人的「看護網」，使用者必須先繳付1.5萬日圓（約995港元）費用，然後每月支付3,000日圓（約199港元）服務費，方法是在老人常用的起居室裝置感應器。

每日電郵通報情況 如平安鐘

感應器是曉得自動發出信息的「平安鐘」，可以每日一次用電郵通報房內的情況，不論老人「白天喜歡在睡房內躺着」，抑或「晚上去廁所很頻密」等生活細節，子女都可瞭如指掌。

至於有內置無綫通訊功能的iPot熱水瓶，首次費用和每月服務費，均與松下電業的感應器相同。Vision Machine由前年開始提供「護身符熱綫」服務，藉iPot使用次數監察家居老人狀況，主要是跟NTT DoCoMo的通訊網絡合作。

東京煤氣公司也不甘後人，推出一項「使用情況通知服務」，透過安裝在煤氣錶上的通訊器發出電郵，每天兩次通報煤氣的使用情況，其收費更便宜，入會費5,000日圓（約332港元），月費1,470日圓（約98港元），但使用者必須是煤氣用戶。 ■高碧斯

松下推出這款曉得自動發放電郵的iPot電熱水瓶，可助子女照顧父母。（互聯網圖片）

Thermal pot with monitoring function by NTT DoCoMo & Panasonic (Hong Kong Economics Times, Jan. 03)

Conceptual Design

Example: Interactive Mirror

Display time, weather, heart rate, weight, etc.

Cartoon to guide teeth brushing

Conceptual Design

Example: Smart Window

With AR technology, information such as weather will be provided

https://unwire.hk/2018/10/03/ar-windshield/life-tech/auto/

Conceptual Design

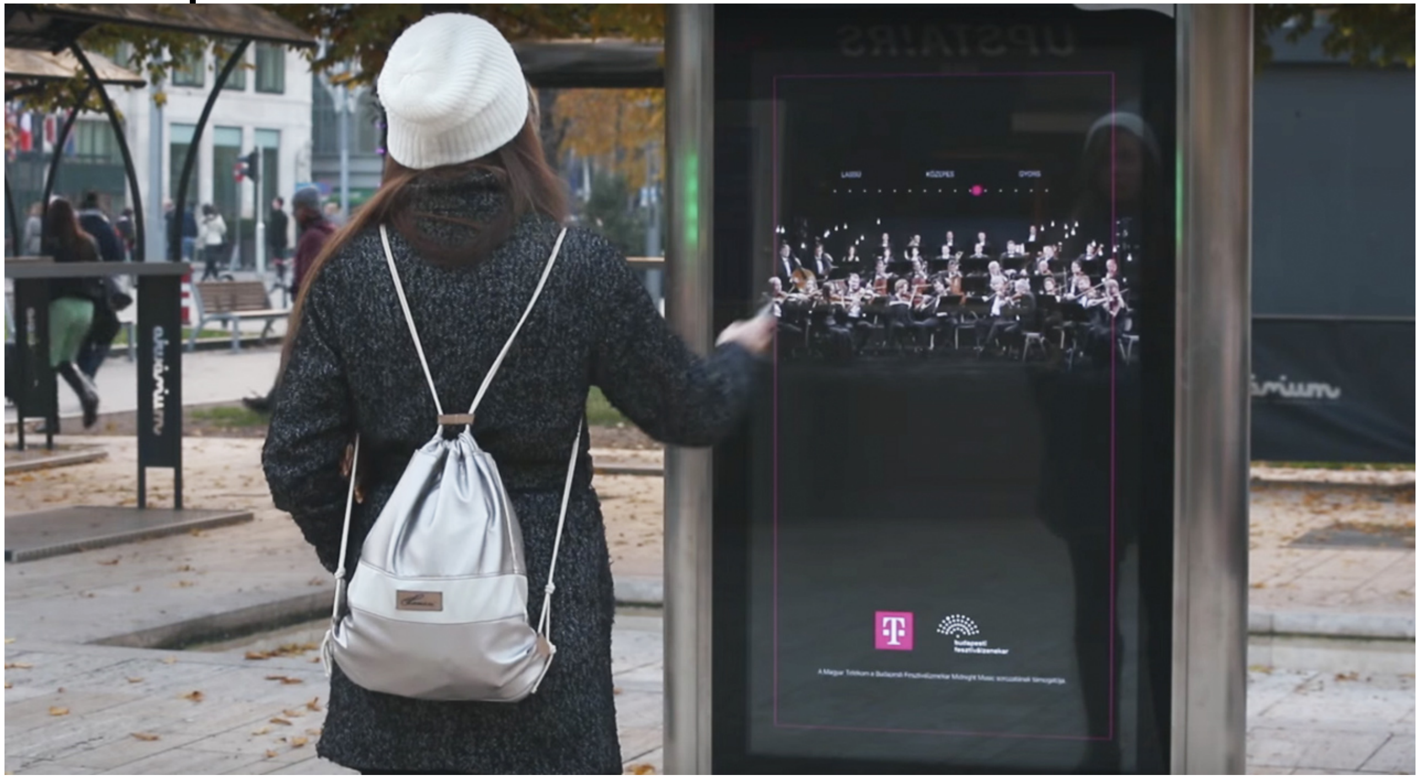
Example: Smart Windshield

With AR technology, better navigation, traffic information, and even advertisement will be provided in 2020

https://www.youtube.com/watch?v=2bcvvjai5EM

Conceptual Design

Example: Interactive Poster



Passersby can conduct the ensemble via their mobile phone in Budapest Festival Orchestra
https://www.springwise.com/interactive-poster-lets-passersby-conduct-orchestra/

Conceptual Design

Example: Smart Bed

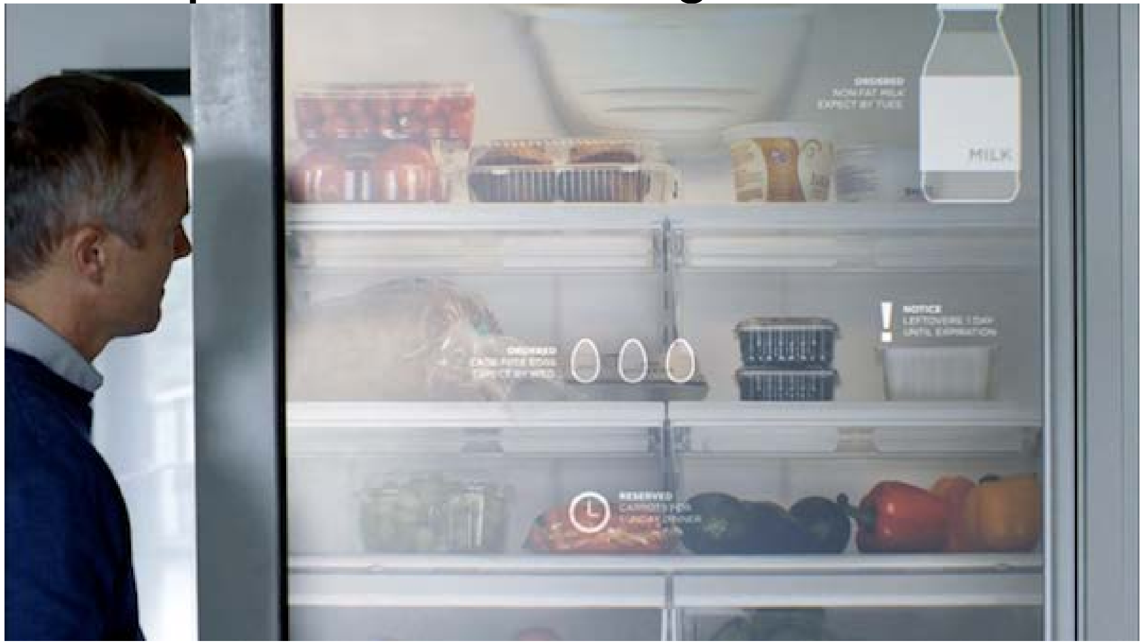
More than a bed:

- Sense user movements and then automatically adjust firmness, comfort and support to keep him sleeping blissfully
- Gently raise user head when mild snoring is detected
- Track how well the user sleeps each night

https://www.sleepnumber.com/

Conceptual Design

Example: Smart Fridge



It will calculate the freshness of food and automatically create a shopping list when we're missing or are low on essential items. It will even be able to order these items automatically. It will suggest recipes based on ingredients in the fridge and help us maintain our diets

https://uxmag.com/articles/the-internet-of-things-and-the-mythical-smart-fridge

Conceptual Design

New thinking/consideration in ubicomp:

- How to enable people to access and interact with information in their work, social, and everyday lives
- How to design user experiences for people using interfaces that are part of the environment with no obvious controlling devices
- How and in what form to provide contextually-relevant information to people at appropriate times and places
- How to ensure that information, that is passed around via interconnected displays, devices, and objects, is secure and trustworthy

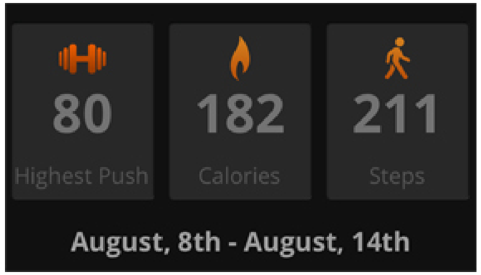
Conceptual Design

- Wearable computing:
  - Follow-on ubiquitous computing
  - Embed technologies on people's clothing such as glasses, jackets, watches, etc.

Fitbit Surge: Watch + GPS Tracking + body (e.g., activities, heart rate, sleep) monitoring

https://www.fitbit.com/us/surge

Conceptual Design



PoloTech Smart Shirt by Ralph Lauren can measure heart rate, breathing rate and calorie expenditure

http://robbreport.com/pmc-lst-gallery-item/ralph-lauren-polotech-smart-shirt/

Conceptual Design

Purple is more than a necklace - It connects wirelessly to your social networks and receives images and messages from friends and family

https://www.artefactgroup.com/work/purple-a-wearable-locket-for-the-21st-century

Conceptual Design

- Attentive environments computing:
  - System “estimates” the user needs without user’s instructions or controls
  - Computer interfaces that can respond to user’s expressions, gestures, etc.

聰明電話懂按心情接來電 ✽美國卡內基美隆大學的科學家，正研製一種會自動因應主人心情而決定接聽與否的「聰明電話」，可以擔當接綫生的角色。這個名爲Sensay的電話，利用放置於身體上的感應器，可以知道主人是否忙碌或心情不佳，自動播出聲帶叫來電者稍後再致電。	Smart telephone which helps us to receive a call or not (Hong Kong Economics Times, Sept. 03)

Conceptual Design

豐田「表情汽車」獲專利

汽車會哭會笑，相信大家只會想到卡通片的情節，但原來豐田（Toyota）真有這個構想，生產有表情的汽車，讓駕駛者有多些娛樂，而這個構思已取得專利。

駕駛者駕車時，經常會遇到一些氣憤或愉快的事，如後車胡亂切綫，會令人感到不滿；但獲其他人讓綫，又會希望表示謝意。不過，身在車中的駕駛者難以表達這些不滿或謝意，故四位來自豐田的發明家決定要令汽車表達感情。

有表情的汽車一如卡通片中的汽車，以車頭燈作眼睛，有眼簾和眼眉，亦有一條模仿動物尾巴的天綫。汽車裝有電腦及軟件系統，可探測路面及汽車狀況，如軚盤角度、煞車效率及車速等，駕駛者或乘客可向系統輸入「心情」。

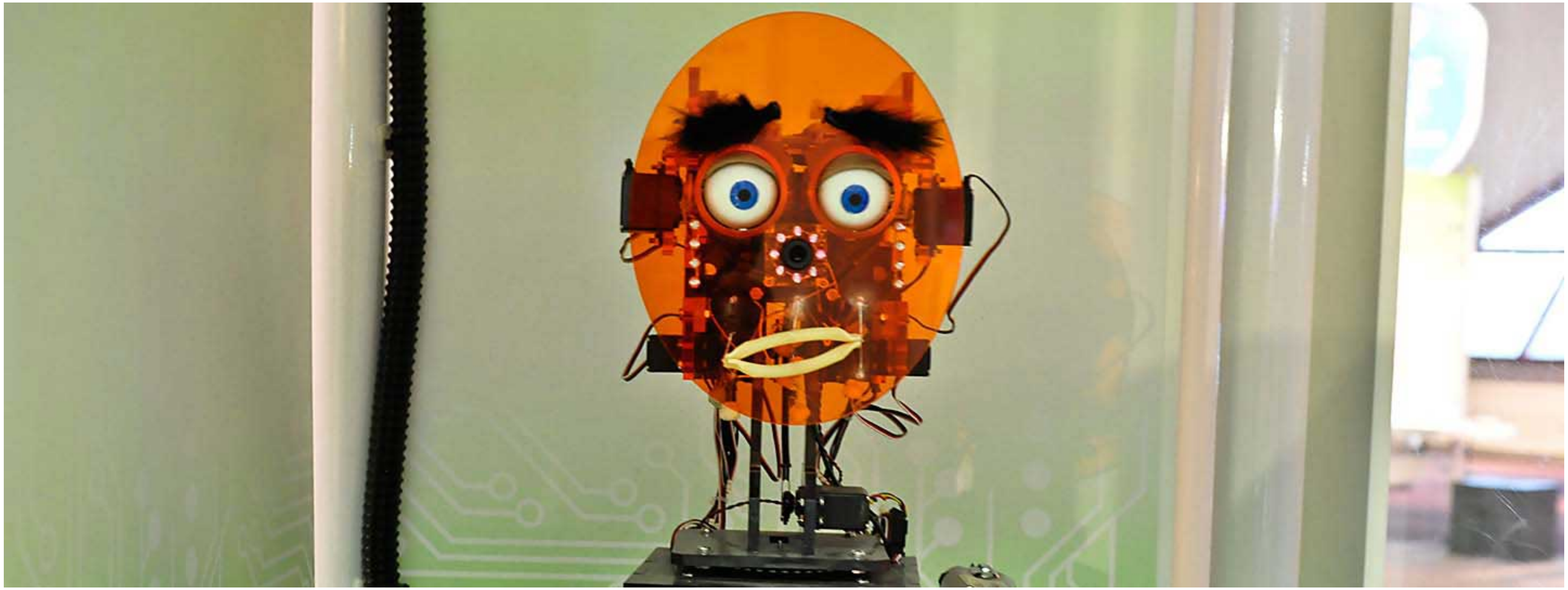
天綫會擺尾 車頭燈顏色表達心情

駕駛者想表示善意讓其他車先行，車頭燈及眼眉會轉為橙色，眼簾半開（模擬笑着），車尾的天綫會如動物擺尾般左右搖擺。汽車出現故障，車頭燈會變為憂鬱的藍色，眼簾低垂，以及「淚」燈閃爍。

設計師在設計汽車表情時，以傳統上各種顏色予人的感覺為依歸，如橙色表示心情好，紅色表示憤怒或驚訝，綠或藍等冷色表示憂愁和哭泣等。豐田表示，已取得的專利亦可用於電單車、船或飛機上。

Future car which has different expressions according to driver's emotion
(Hong Kong Economics Times, July 04)

Conceptual Design



IBM robot, Pong, demonstrates how robots can be used to gauge and evaluate human interest in products in a store environment. Pong consists of a vision system that can track visitor movement and software capable of interpreting visual focus and recognition of facial gestures

http://www.carnegiesciencecenter.org/exhibits/roboworld-meet-robots-pong/

Conceptual Design

Can be expanded further:

- What functions will the product perform?
  - How the task is divided between user and system? What will the product do? What will the human do?
- How are the functions related to each other?
  - Procedure in doing the task, e.g., sequential or parallel?
  - Can different tasks be grouped as a categorization, e.g., cut, copy, paste
- What information needs to be available?
  - What data are required to perform the task?
  - How are the data to be transformed by the system? e.g., display the data using a graph or list of numbers

Concrete Design

Consider more concrete, detailed issues of the design:

- How information is presented and interacted in the interface? e.g., fill-in forms, command line
- Combination of media? e.g., sound and animations
- What kind of feedback? e.g., voice, colour change
- What combinations of input and output devices? e.g., speech, keyboard+mouse, handwriting recognition
- Controlled by software or hardware? e.g., use WIMP to start scanning or use a button on scanner to do so
- Whether to provide agents and in what format?
- How to design menu, icon, screen and display information?
- Iterations are required between concrete and conceptual design

Concrete Design

Shneiderman's eight golden rules:

- Strive for consistency: e.g., consistency in layout, font
- Enable frequent users to use shortcuts: e.g., "Ctrl+S" is faster than using WIMP to save
- Offer informative feedback
- Acknowledge task completion: make it clear when an action has been successfully completed, e.g., "printing completed"
- Offer error prevention and simple error handling
- Permit easy reversal of action, e.g., provide "undo"
- Support control principle of usability: users feel more comfortable if they feel in control of the interaction
- Reduce short-term memory load: e.g., use recognition instead of recall

Concrete Design

A number of design rules, guidelines, principles exist, e.g.,

- D.Mayhew, *Principles and Guidelines in Software User Interface Design*: Design of interface types: menu, fill-in forms, etc.
- D.D.McCracken and R.H.Wolfe, *User-Centered Website Development: A Human-Computer Interaction Approach*: Web design
- S.Weinschenk and D.T.Barker, *Designing Effective Speech Interfaces*: Speech interface design

Standards also help:

- Set by national or international bodies to ensure compliance by a large community of designers
- High authority and low level of detail
  e.g., ISO 14915: user interface of multimedia applications

Concrete Design

Scenarios can help:

- Express proposed or imagined situations
- Used throughout design in various ways
  - As a basis for overall design (e.g., storyboard creation)
  - Scripts for user evaluation of prototypes
  - Concrete examples of tasks
  - For technical implementation
  - As a means of co-operation within design teams
  - As a means of co-operation across professional boundaries, i.e., basis of communication in a multidisciplinary team
- Plus (most positive consequences of a particular design solution) and minus (most negative consequences) scenarios to explore extreme cases, e.g.,

Concrete Design

Scenario 3: Hyper-wonderland

This scenario addresses the positive aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspection of one of the caissons[1] Kurt takes his portable PC, switches it on and places the cursor on the required information. He clicks the mouse button and gets the master file index together with an overview of links. He chooses the links of relevance for the caisson he is inspecting.

Kurt is pleased that he no longer needs to plan his inspections in advance. This is a great help because due to the 'event-driven' nature of inspection, constructors never know where and when an inspection is taking place. Moreover, it has become much easier to keep track of personal notes, reports etc. because they can be entered directly on the spot.

The access via the construction site interface does not force him to deal with complicated keywords either. Instead, he can access the relevant information right away, literally from where he is standing.

A positive side-effect concerns his reachability. As long as he has logged in on the computer, he is within reach of the secretaries and can be contacted when guests arrive or when he is needed somewhere else on the site. Moreover, he can see at a glance where his colleagues are working and get in touch with them when he needs their help or advice.

All in all, Kurt feels that the new computer application has put him more in control of things.

[1] Used in building to hold water back during construction.

Concrete Design

Scenario 4: Panopticon

This scenario addresses the negative aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspecting one of the caissons Kurt starts talking to one of the builders about some reinforcement problem. They argue about the recent lab tests, and he takes out his portable PC in order to provide some data which justify his arguments. It takes quite a while before he finds a spot where he can place the PC: either there is too much light, or there is no level surface at a suitable height. Finally, he puts the laptop on a big box and switches it on. He positions the cursor on the caisson he is currently inspecting and clicks the mouse to get into the master file. The table of contents pops up and from the overview of links he chooses those of relevance – but no lab test appears on the screen. Obviously, the file has not been updated as planned.

Kurt is rather upset. This loss of prestige in front of a contractor engineer would not have happened if he had planned his inspection as he had in the old days.

Sometimes. he feels like a hunted fox especially in situations where he is drifting around thinking about what kind of action to take in a particular case. If he has forgotten to log out, he suddenly has a secretary on the phone: "I see you are right at caisson 39, so could you not just drop by and take a message?"

All in all Kurt feels that the new computer application has put him under control.

Design Rationale

A technical term for the documentation used in choosing among alternative designs

$\Rightarrow$ Assist in understanding the design of the system

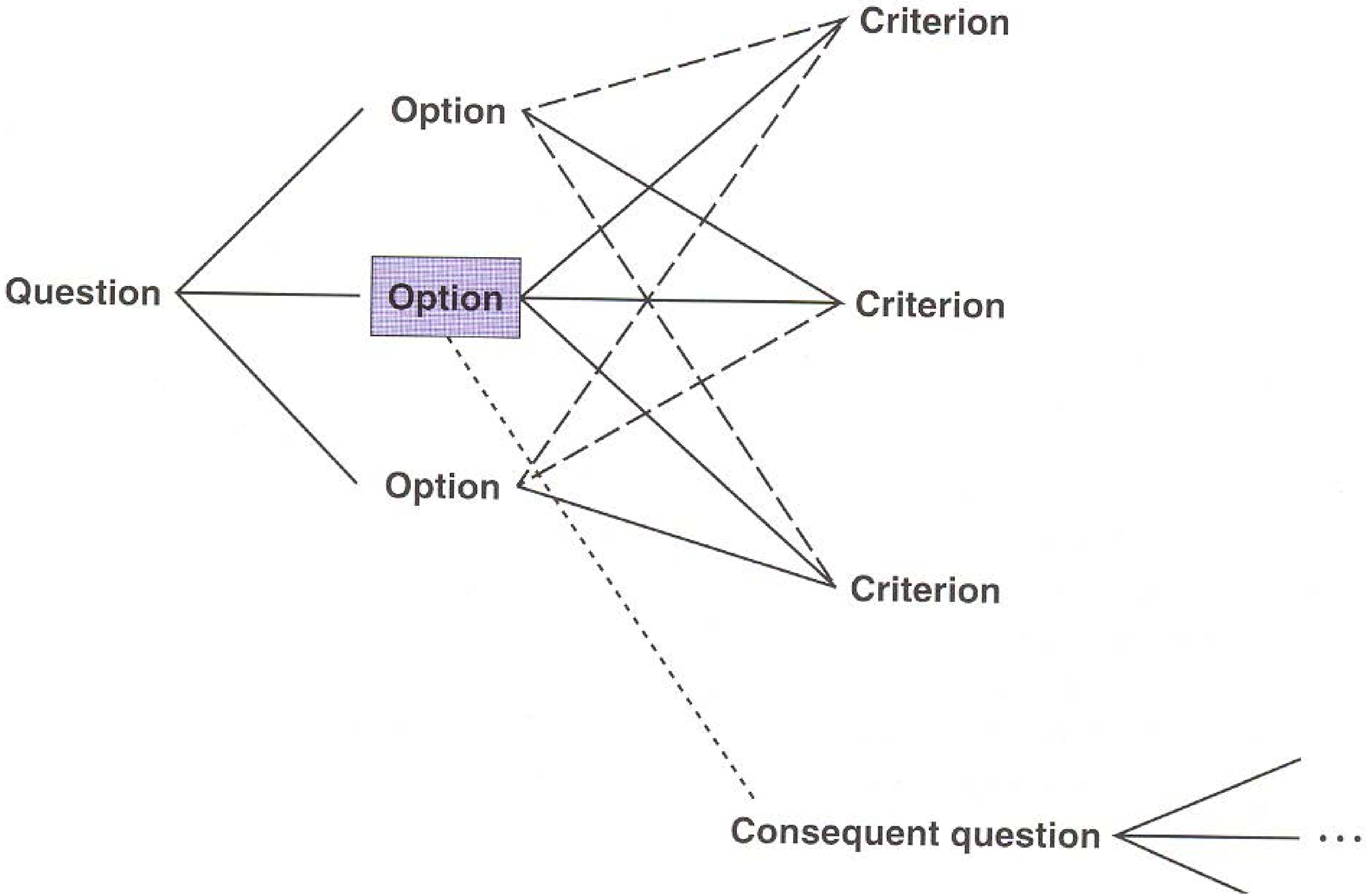
- Benefits
  - Communication among design team members throughout life cycle
  - Reuse of design knowledge across products (e.g., WORD $\rightarrow$ PowerPoint)
  - Enforce design discipline (the designer should deliberate more carefully)
  - Present arguments for design trade-offs

Design Rationale

A common approach: Design Space Analysis

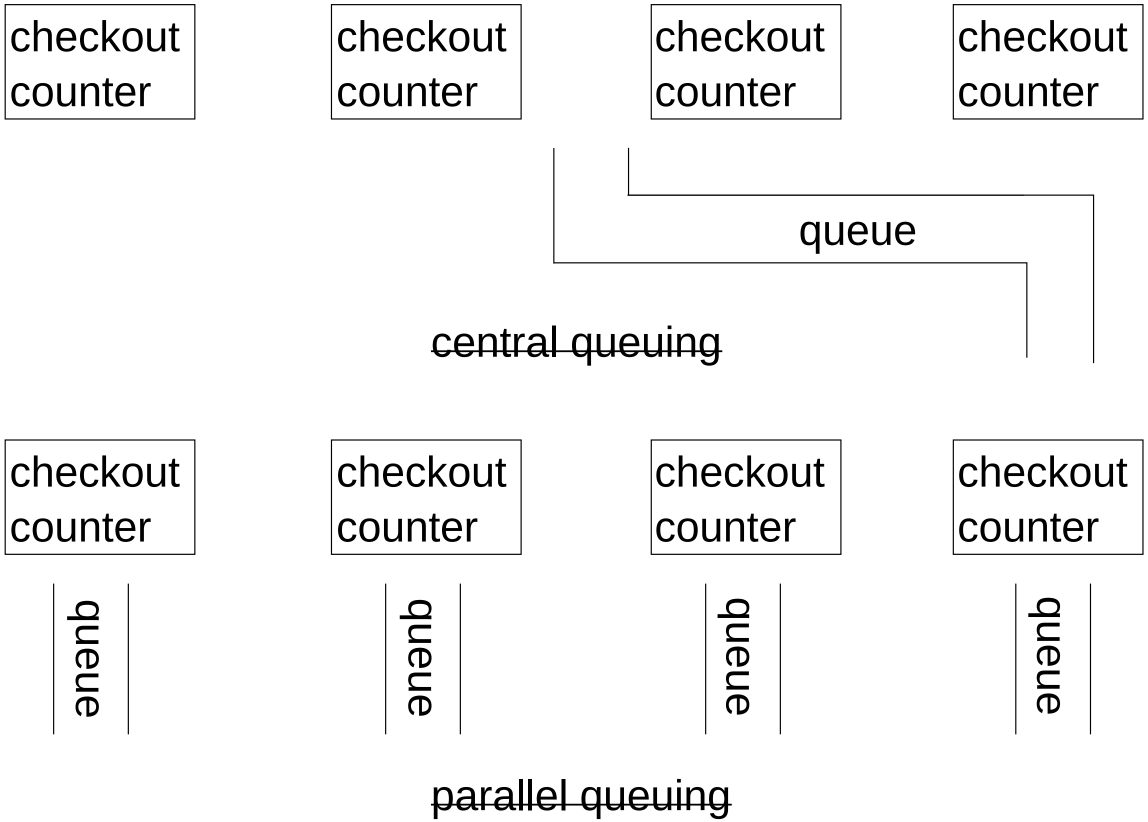
- Notation: Questions, Options, Criteria (QOC)
  - Questions (& sub-questions) represent major issues of a design - key issues in design
  - Options provide alternative solutions to the question - possible design decisions
  - Criteria are the means of assessing the various options in order to make a choice - bases for choosing between options
- Design space includes all possible options, including those rejected
- Structure-oriented

Design Rationale



Design Rationale

- Example: Question: Which queuing-up system should be used? option #1 : central queuing
  option #2 : parallel queuing

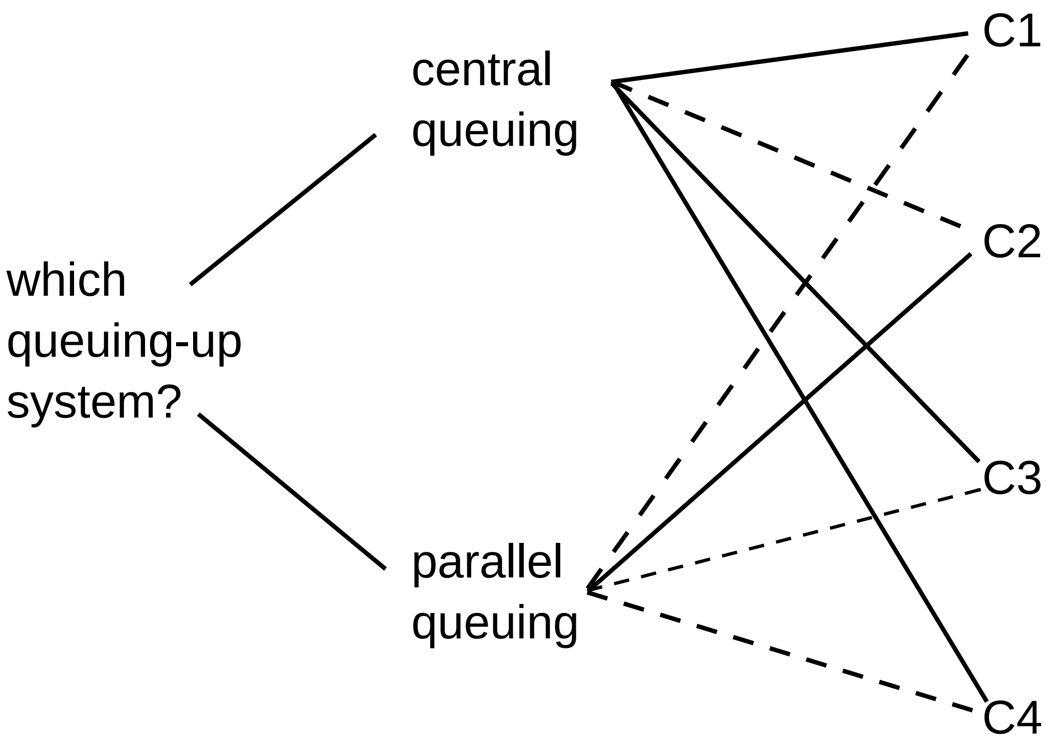


What are the possible criteria?

Design Rationale

Possible criteria are:

- C1: Maximize checkout counter efficiency
- C2: Customer freedom of choices
- C3: Minimize average queuing-up time
- C4: Minimize queuing space

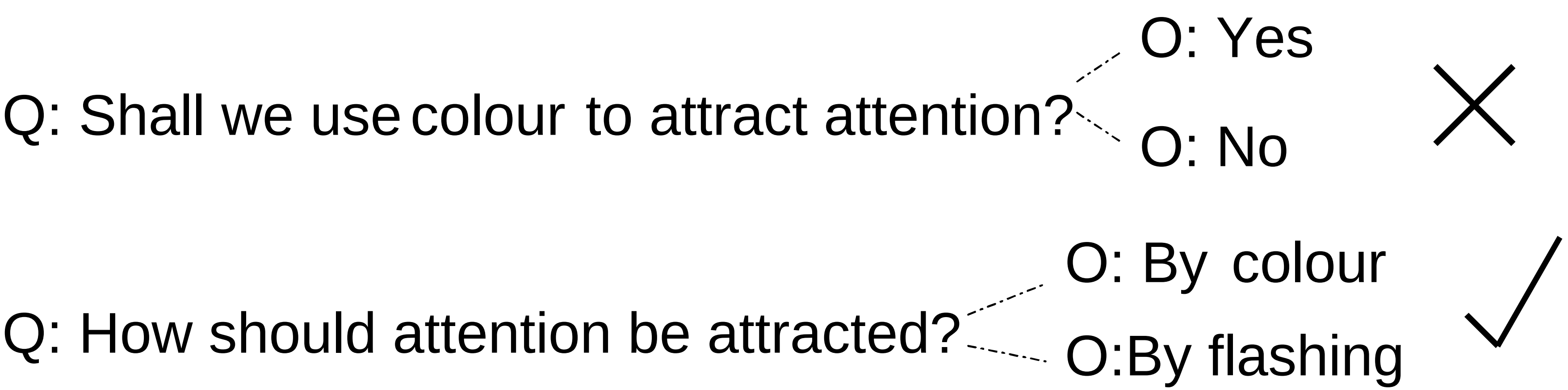


Design Rationale

- Effective design space analysis can be achieved from
  - Raising the right questions
  - Deciding the correct criteria to judge the options
  - Initial questions raised must be sufficiently general that they cover a large enough portion of the possible design space but specific enough that a range of options can be clearly identified
- Hints for doing QOC:
  - Options: - Use criteria to generate options
    - Consider extreme, distinctive or novel options
    - Look for novel combinations of options
  - Criteria: - Design a set of criteria
    - Represent positive and negative criteria

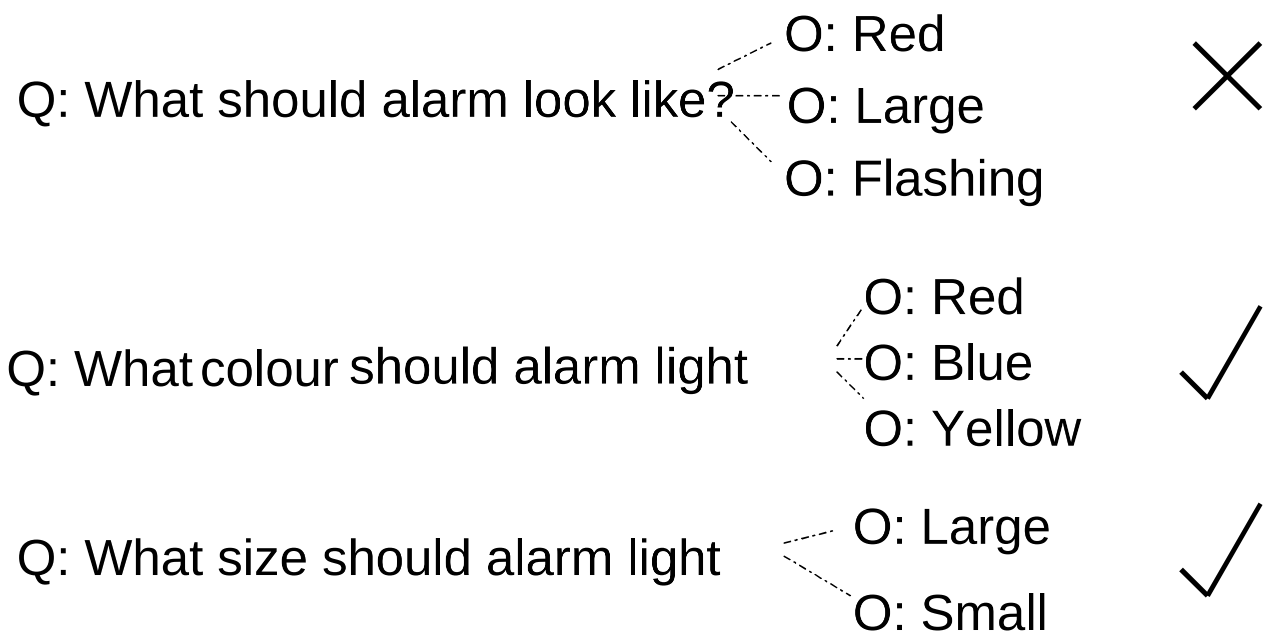
Design Rationale

- Good practice for questions:
  - Does question compare at least two alternatives?



Design Rationale

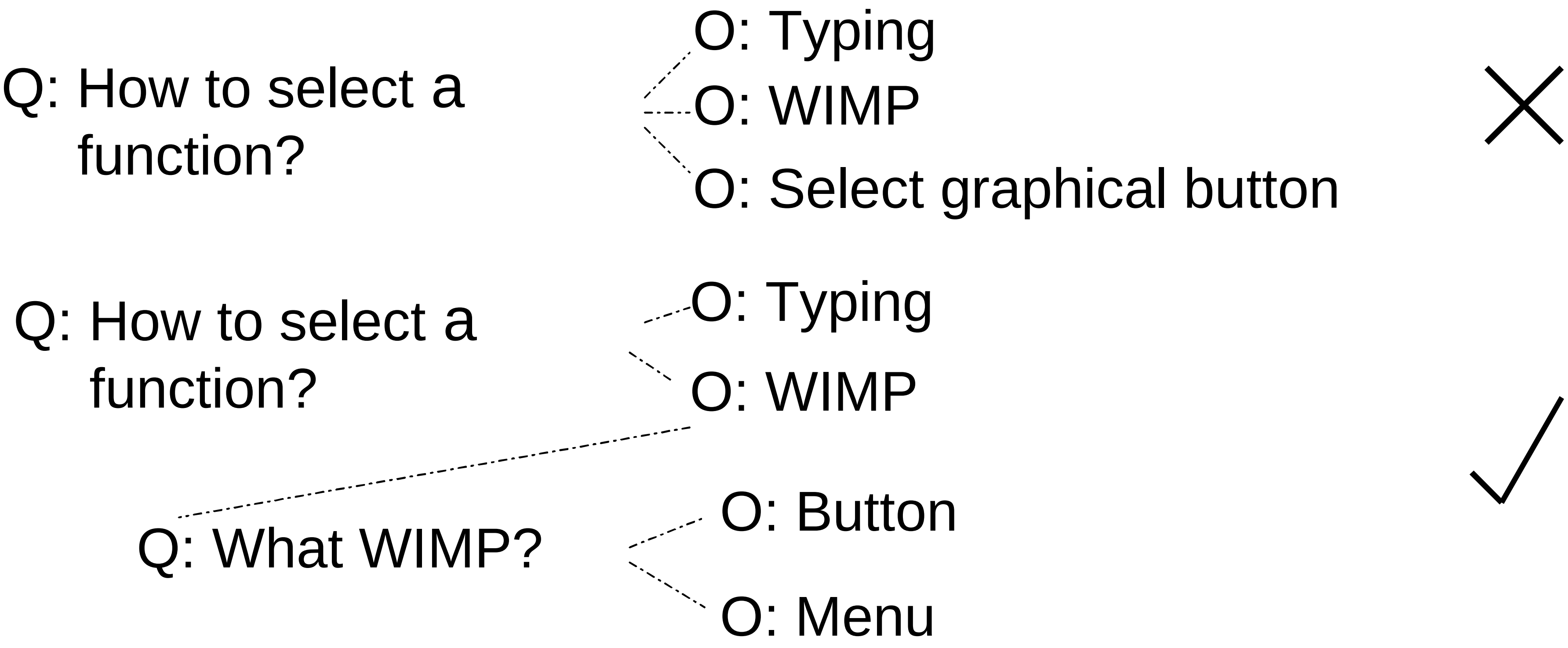
- Does question focus on a single issue?



- Questions can be at different levels:
  e.g., What color should the alarm lights be?
  e.g., How should attention be attracted? (more abstract)

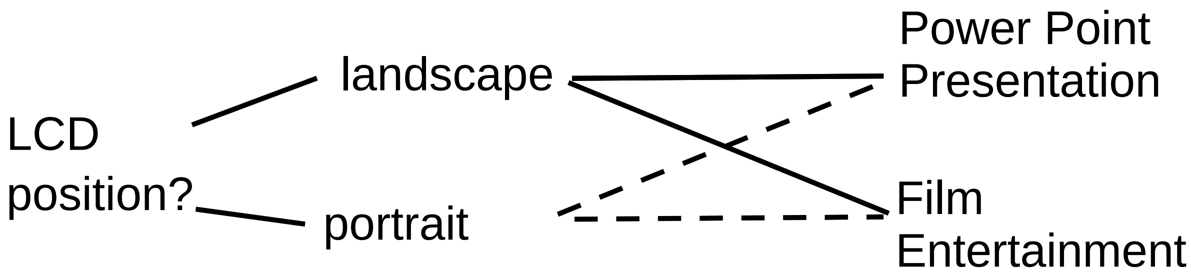
Design Rationale

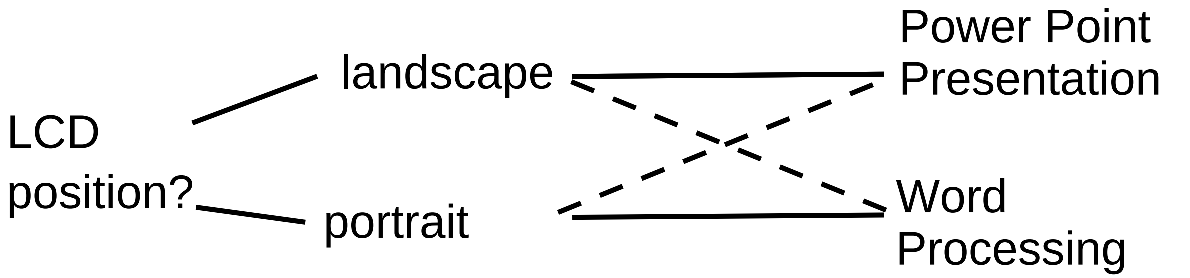
- Good practice for options:
  - Are options at the same level of abstraction?



Design Rationale

- Good practice for criteria:
  - Are criteria useful for assessing & comparing appropriate features of options?





Prototyping

What is prototype?

In other design fields it is a small-scale model, e.g.:

Architecture: a miniature building or town
Electronics: a workable electronic board

Increasing use in design due to the advent of 3D printing:

(a)

(c)

Prototyping

In HCI, prototype is used to simulate or animate (some) features of the intended system during design process

A physical representation of conceptual/concrete design, e.g.

- a series of screen sketches
- a storyboard, i.e., a cartoon-like series of scenes
- a cardboard mock-up
- a piece of wood (e.g., PalmPilot)
- a piece of software with limited functionality written in the target language or in another language
- a video simulating the use of a system
- a Powerpoint slide show

Prototyping

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
- Effective communication device among team members
- Effective way to test designer's ideas (e.g., feasibility)
- Prototypes answer questions, and support designers in choosing between alternatives
- Implementation is expensive, if no prototyping

$\Rightarrow$ any design errors are built in to the first implementation you can test, and it is expensive to make changes

$\Rightarrow$ design errors may be left in the final product

Prototyping

Benefits of prototyping:

- Improve the chances of creating a usable product, since mistakes and omissions are flushed out early in the development process, not after implementation
- Users are good at criticizing an existing system; they are much less able to imagine how specifications would translate into a product
- Prototyping brings the users into the process at a much earlier stage
- Higher user satisfaction and acceptance on the developed interactive products

Recognized ~30 years ago: Maryam Alavi, “An Assessment of the Prototyping Approach to Information Systems Development,” *Communications of the ACM*, June 1984

Prototyping

Lim, Stolterman and Tenenburg (2008) focus on filtering and manifestation, i.e., emphasize specific aspects of a product being explored by prototype, and help designers to develop their ideas through external representations

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

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

Prototyping

What to prototyping?

- Technical issues (e.g., capture video using a watch)
- Work flow, task design (e.g., procedure of taking a photo using a digital camera)
- Screen layouts and information display (e.g., layout of a Web)
- Difficult, controversial, critical areas, e.g., control types on portable media player (PMP)?

Prototyping

Low-fidelity prototyping:

- Not look very much like the final product
- Use a medium which is unlike the final medium, e.g. paper, cardboard, etc.
- Quick, cheap and easily changed
- Can be sketches of screens, storyboards, etc.

High-fidelity prototyping:

- Use materials that expect to be in the final product
- Look much more like final system than 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

Prototyping

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

Prototyping

Can be classified into 2 types:

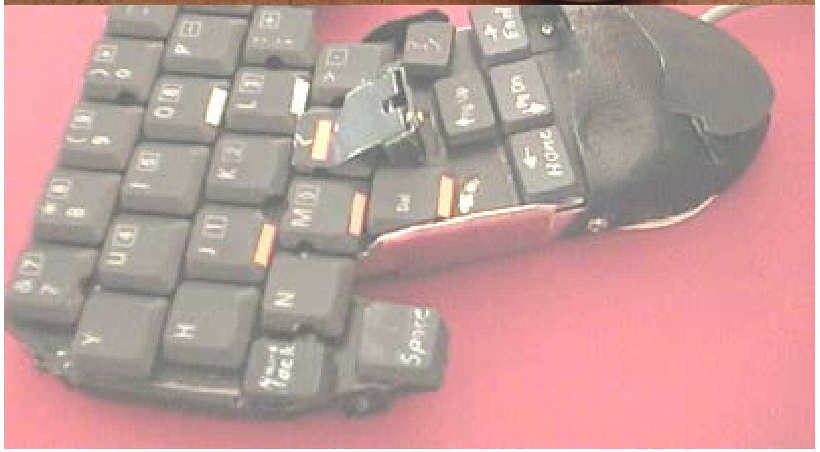
- Evolutionary: the prototype eventually becomes the product
- Revolutionary or throwaway: the prototype is used to get the specifications right, then discarded

Can also be classified into:

- Horizontal prototype: provide a wide range of functions, but with little detail
- Vertical prototype: provide a lot of detail for only a few functions

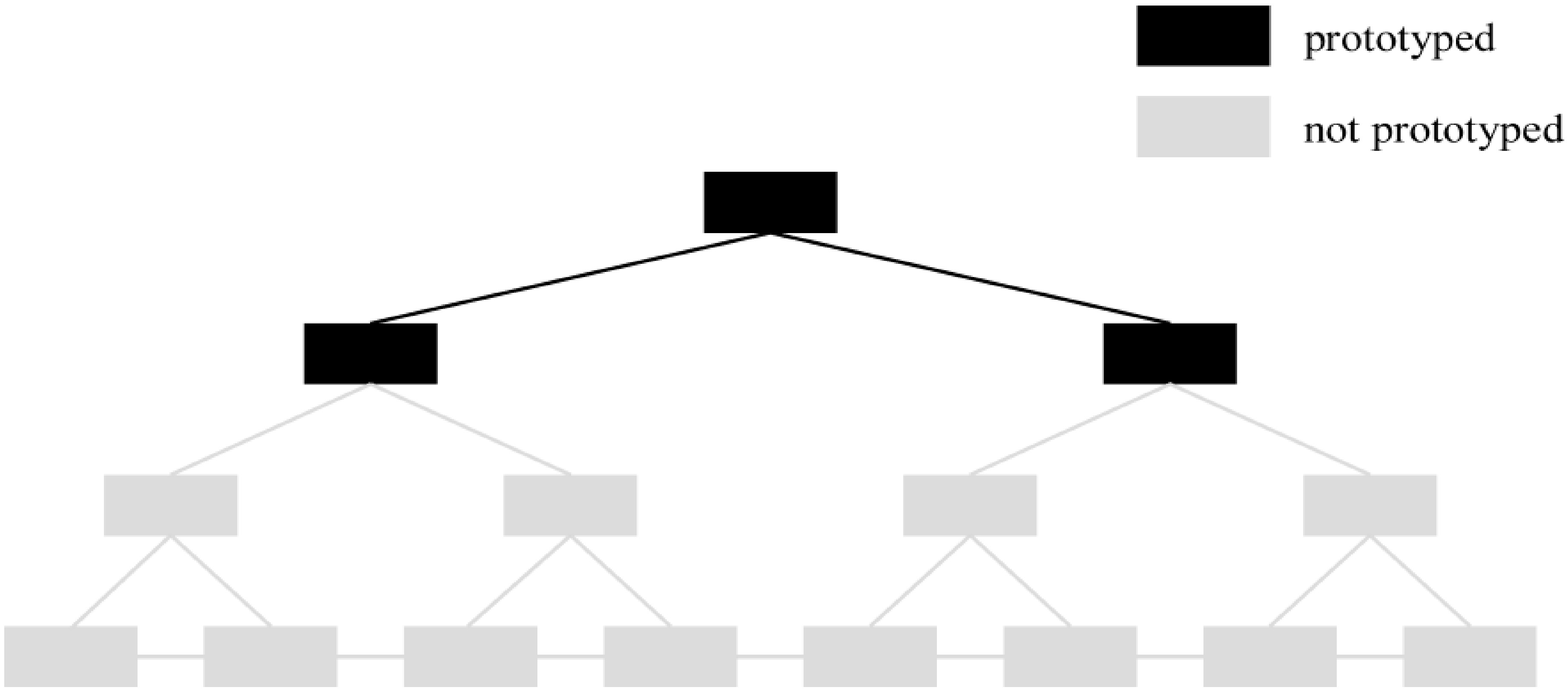
$\Rightarrow$ Compromises are needed in prototyping

Prototyping



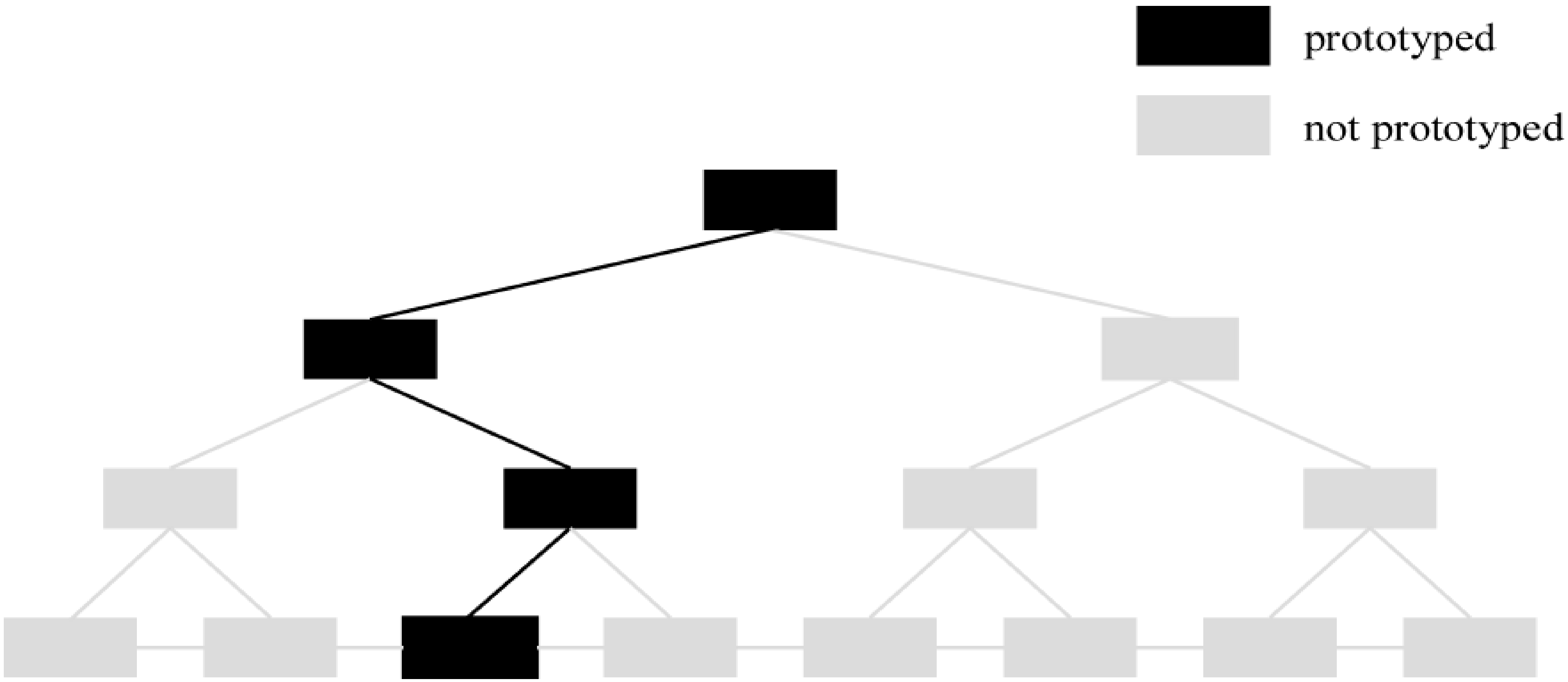
Prototyping

Horizontal prototype: broad but only top-level



Prototyping

Vertical prototype: deep, but only some functions



Prototyping Techniques

Storyboards – a graphical representation of outward appearance of intended system, without any accompanying system functionality

- Often used with scenarios, bringing more detail, and a chance to role play
- Low-fidelity prototyping and used early in design
- A series of sketches showing how a user might progress through a task using the device being developed. Sketching is important to low-fidelity prototyping
- Drawing ability is not the concern. On the other hand, practicing simple symbols is needed:

Prototyping Techniques

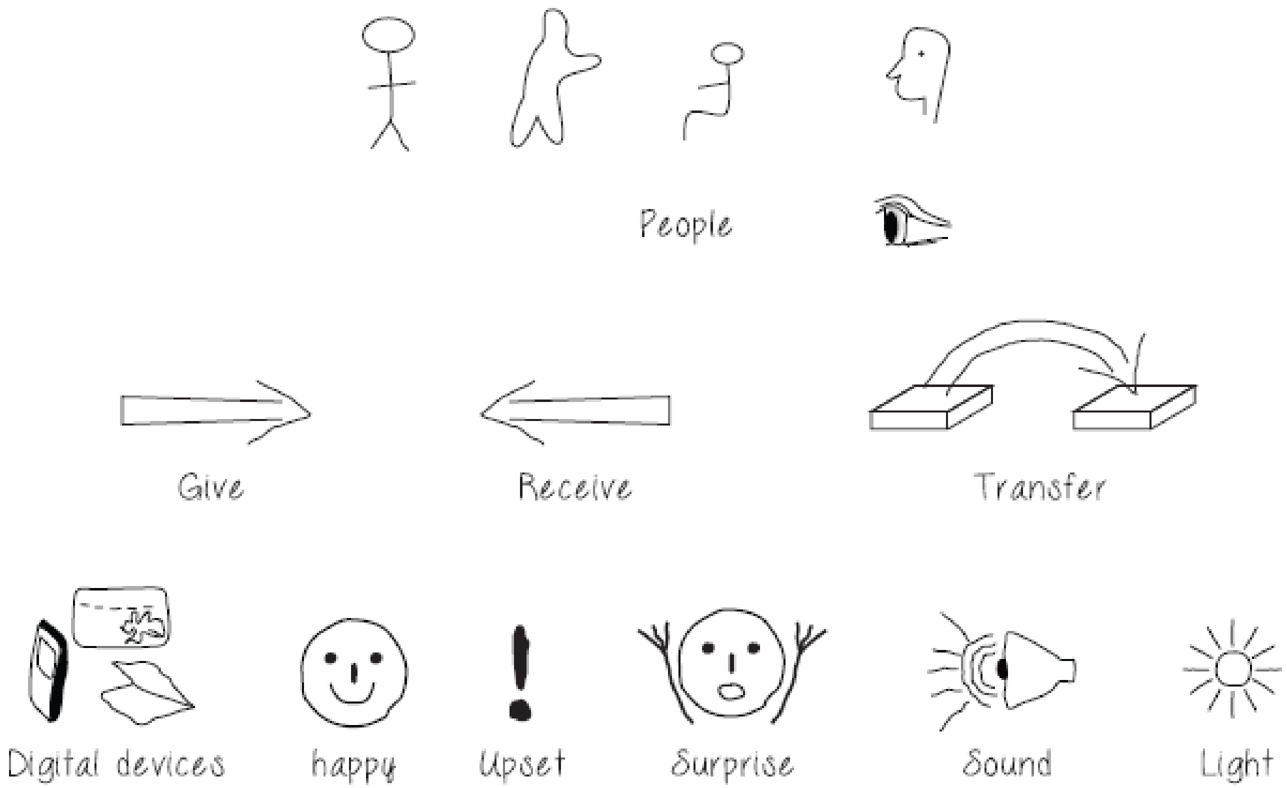


Figure 11.4 Some simple sketches for low-fidelity prototyping

- In fact, there are many tools to polish our drawings, e.g., https://www.autodraw.com/

Prototyping Techniques

- e.g., a storyboard depicting how to fill a car with gas

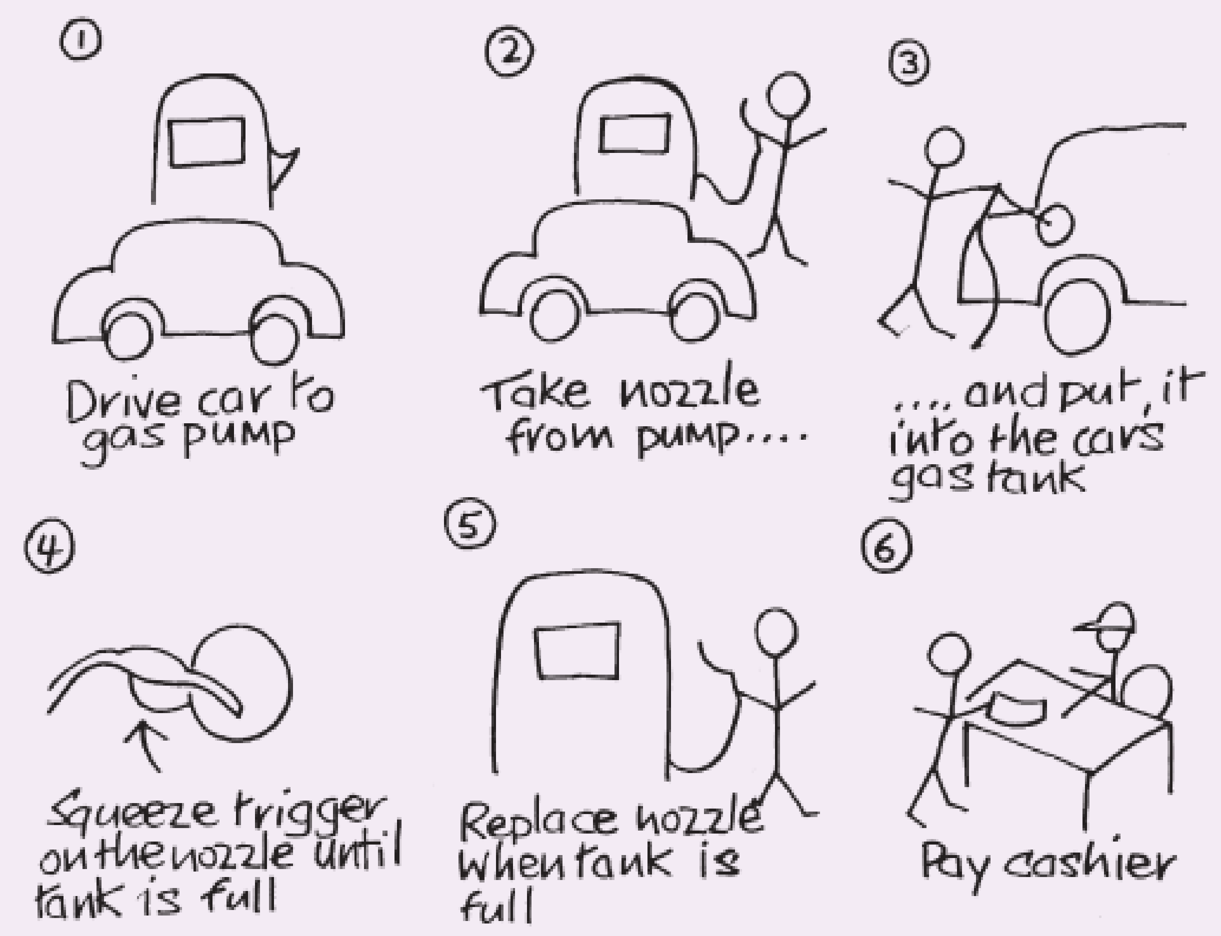
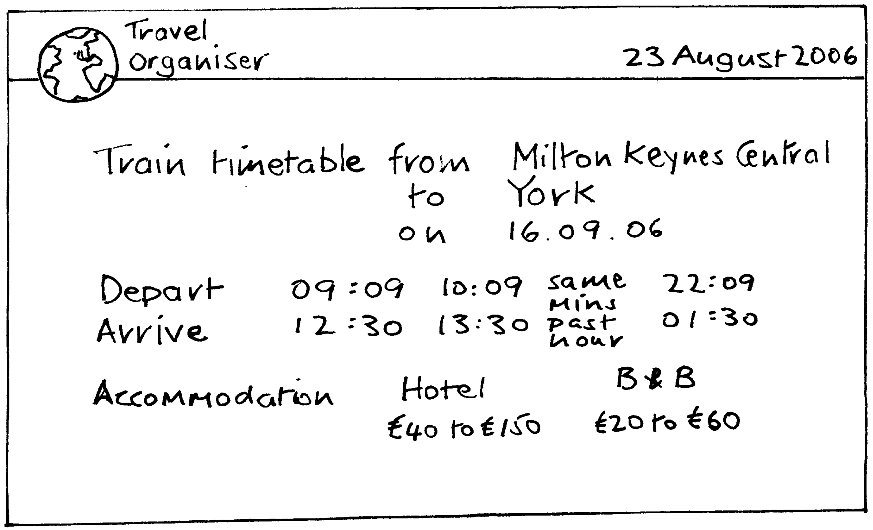


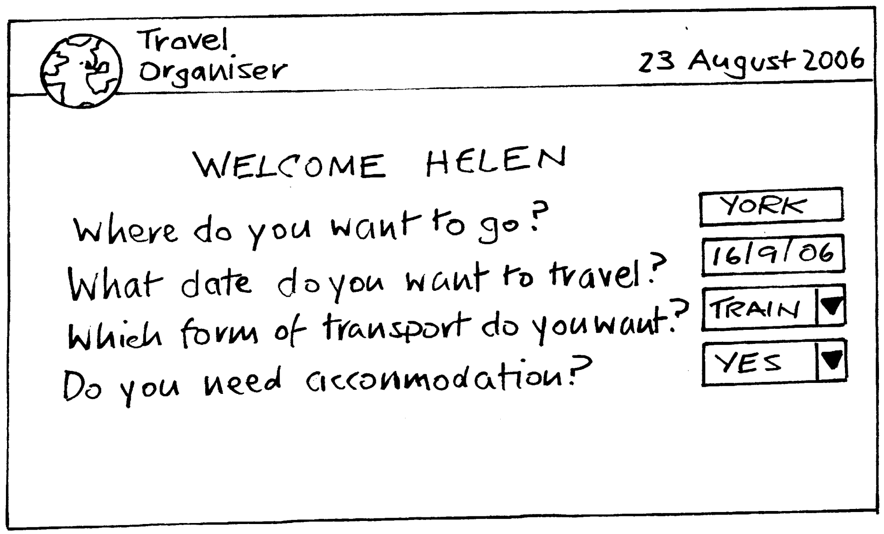
Figure 11.5 A storyboard depicting how to fill a car with gas

Prototyping Techniques

Card-based

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

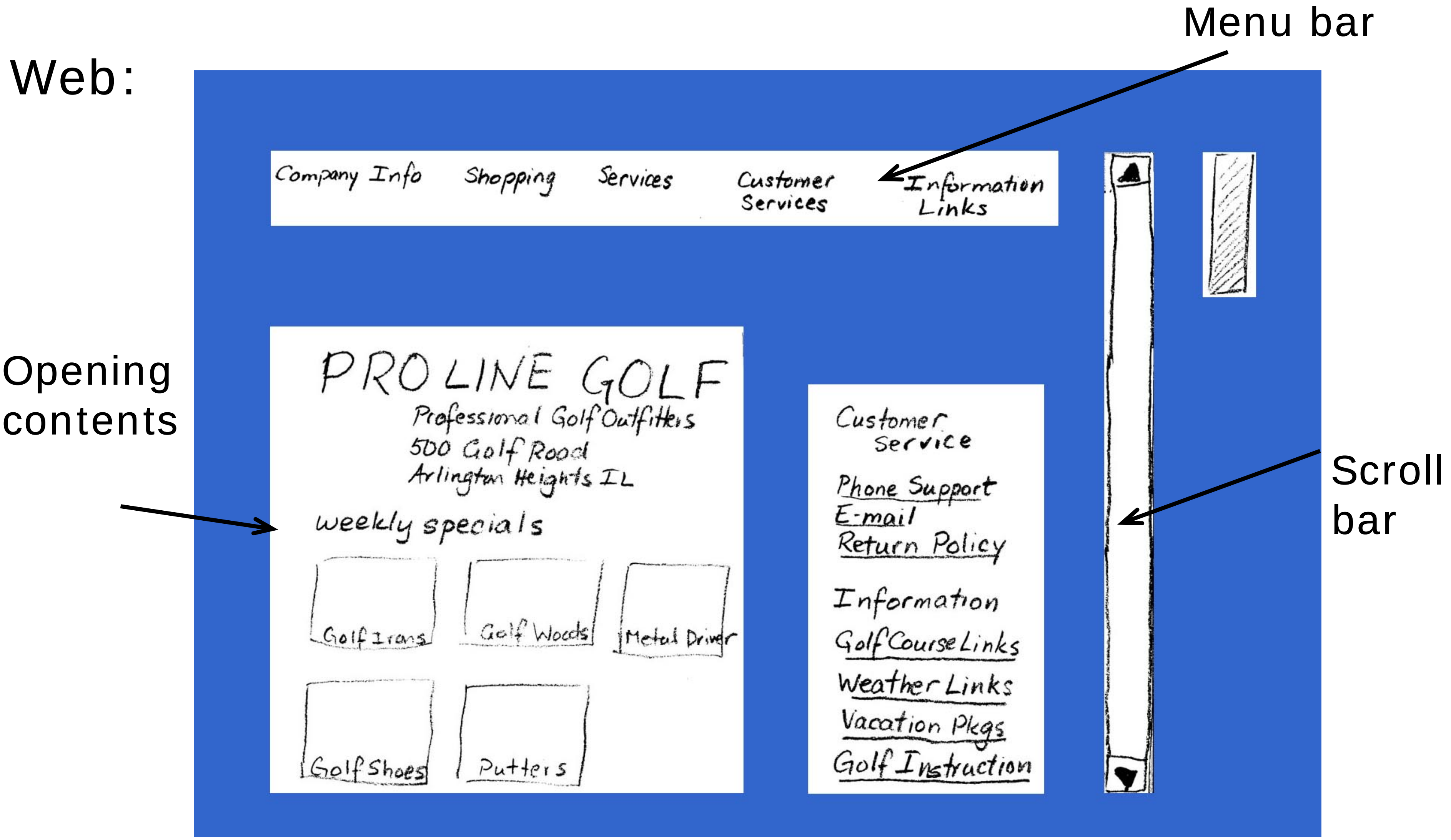




Prototyping Techniques

Paper – also important to low-fidelity prototyping

Web:



Prototyping Techniques

Ashford Charleus, Shahana Dewan, Khairul Kabir, Libai Qu (Computer Science, Spring 2003, City College of New York)

First level:

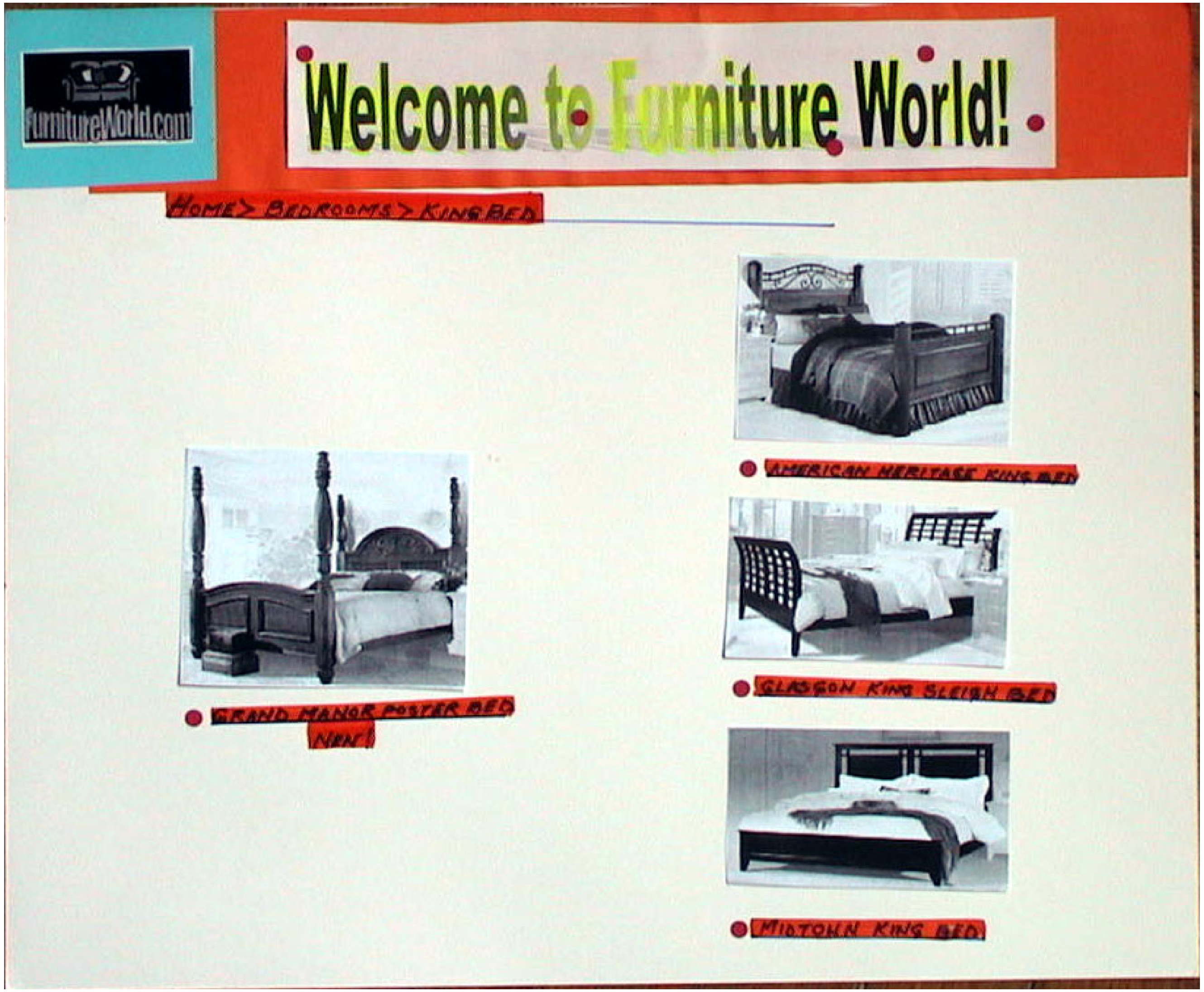
Pulldown menu

Prototyping Techniques

Second level:

Prototyping Techniques

Another second level:



Prototyping Techniques

After prototyping and user testing, this is what their home page looked like:

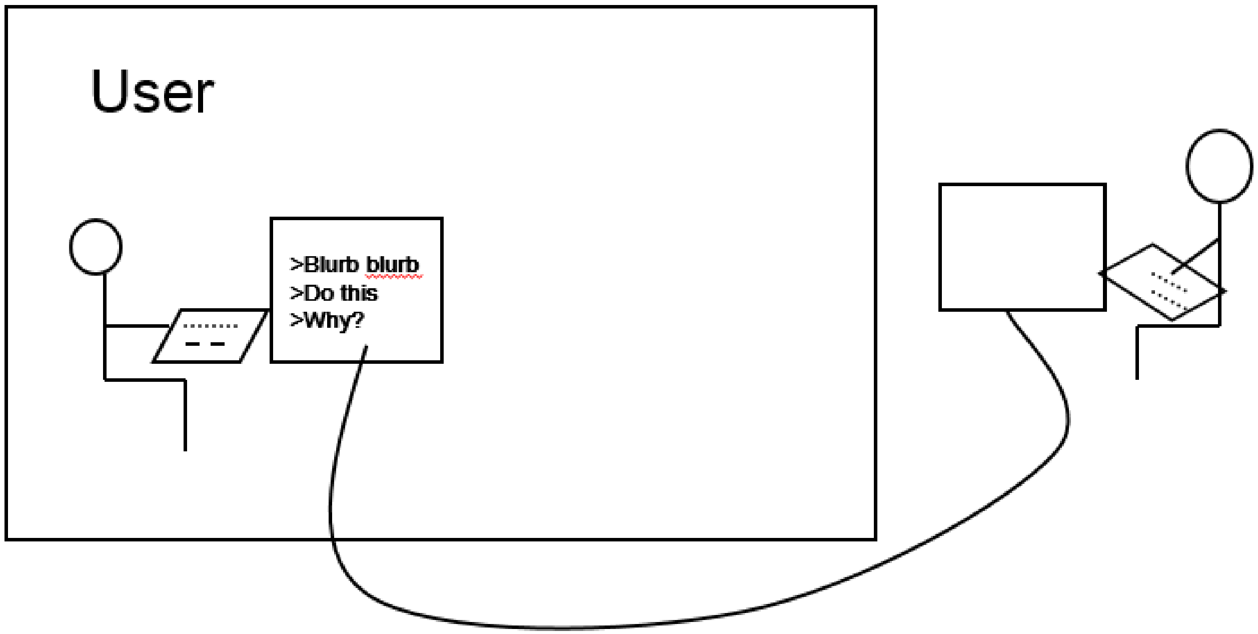
Prototyping Techniques

Figure 11.6 Prototype developed for cell phone user interface

Prototyping Techniques

Wizard-of-Oz

- Another example of low-fidelity 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



Prototyping Techniques

Limited functionality simulations – simulate part of system functionality provided by designers

- An example of high-fidelity prototyping
- Useful for selling ideas to people and for testing out technical issues

  e.g., use Visual Basic to produce a DVD panel on screen; allow people to press mouse button to “experience” its functions

Prototyping Techniques

Physical Computing – Build and code prototypes using electronics

- Toolkits available include Arduino, Senseboard, and MaKey MaKey, which are designed for use by wide range of people

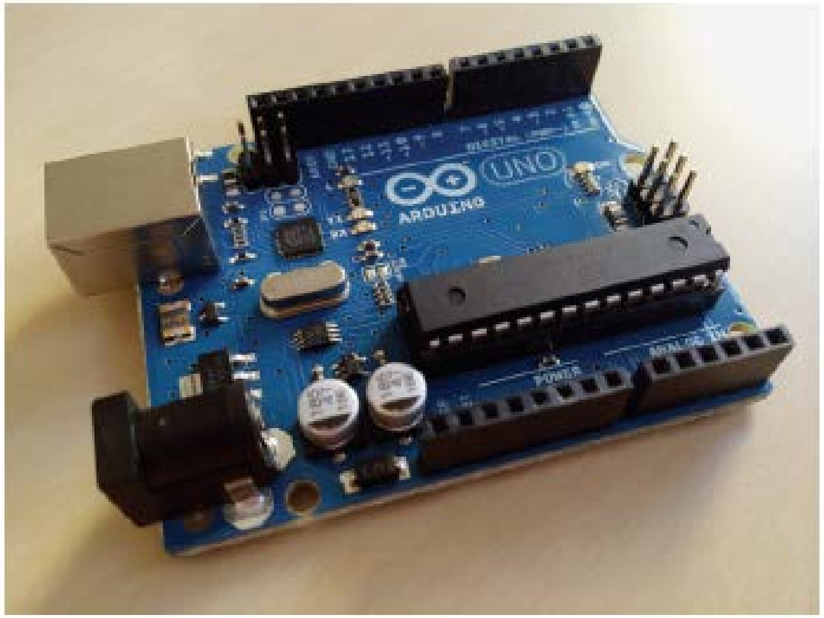


Figure 11.22 The Arduino board

Source: Courtesy of Nicolai Marquardt

Arduino toolkit consists of:

  - Arduino board which is a piece of hardware that is used to build objects
  - Arduino IDE which is a piece of software that makes it easy to program and upload a sketch to the board

Prototyping Techniques

Software Development Kit (SDK) – Package of programming tools and components that supports development of applications for a specific platform, e.g., iOS, Android, Windows

- Include IDE, documentation, drivers, sample code, application programming interfaces (APIs)

- Make development much easier