Chapter 5
Designing Interactive Systems

- 5.1 Design vs. Requirements
- 5.2 Design and development process
- 5.3 Creativity methods
- 5.4 Tools and methods in the early design phase
  - 5.4.1 Scenario Development and Persona
  - 5.4.2 Sketches and Storyboards
  - 5.4.3 Concept Videos
- 5.5 Prototyping
- 5.6 Wizard of Oz
- 5.7 Approaches to making systems interactive
- 5.8 Describing and specifying interactive systems

How to design an interactive system?

- Activity based
  - Giving instructions
    - issuing commands using keyboard and function keys and selecting options via menus
  - Conversing
    - interacting with the system as if having a conversation
  - Manipulating and navigating
    - acting on objects and interacting with virtual objects
  - Exploring and browsing
    - finding out and learning things
- Based on (physical) objects or artefacts, e.g.
  - Office equipment
  - Tool
  - Book

Giving instructions

- Where users instruct the system and tell it what to do
  - e.g. tell the time, print a file, save a file
- Very common conceptual model, underlying a diversity of devices and systems
  - e.g. Unix shells, CAD, word processors, DVD player, vending machines
- Main benefit is that instructing supports quick and efficient interaction
  - good for repetitive kinds of actions performed on multiple objects

Conversing

- Underlying model of having a conversation with another human
- Range from simple voice recognition menu-driven systems to more complex 'natural language' dialogues
- Examples include timetables, search engines, advice-giving systems, help systems
- Recently, much interest in having virtual agents at the interface, who converse with you, e.g. Microsoft’s Agents (e.g. Clippy)

Pros and cons of conversational model

- Allows users, especially novices and technophobes, to interact with the system in a way that is familiar
  - makes them feel comfortable, at ease and less scared
- Misunderstandings can arise when the system does not know how to parse what the user says
  - e.g. child types into a search engine, that uses natural language (http://www.askids.com/, http://news.ask.com) the question:

  "How many legs does a centipede have?"

  and the system responds:
Manipulating and Navigating

- Involves dragging, selecting, opening, closing and zooming actions on virtual objects
- Exploits users’ knowledge of how they move and manipulate in the physical world
- Examples
  - what you see is what you get (WYSIWYG)
  - the direct manipulation approach (DM)
- Shneiderman (1983) coined the term DM, came from his fascination with computer games at the time
- Common model in the desktop world

Core principles of DM

- Continuous representation of objects and actions of interest
- Physical actions and button pressing instead of issuing commands with complex syntax
- Rapid reversible actions with immediate feedback on object of interest

Why are DM interfaces so enjoyable?

- Novices can learn the basic functionality quickly
- Experienced users can work extremely rapidly to carry out a wide range of tasks, even defining new functions
- Intermittent users can retain operational concepts over time
- Error messages rarely needed
- Users can immediately see if their actions are furthering their goals and if not do something else
- Users experience less anxiety
- Users gain confidence and mastery and feel in control

What are the disadvantages with DM?

- Some people take the metaphor of direct manipulation too literally
- Not all tasks can be described by objects and not all actions can be done directly
- Some tasks are better achieved through delegating
  - e.g. spell checking
- Can waste extensive screen space
- Moving a mouse around the screen can be slower than pressing function keys to do same actions
Exploring and browsing

- Similar to how people browse information with existing media (e.g. newspapers, magazines, libraries)
- Information is structured to allow flexibility in the way user is able to search for information
  - e.g. multimedia, web

Conceptual models based on objects

- Usually based on an analogy with something in the physical world
- Examples include books, tools, vehicles
- Classic: Star Interface based on office objects

Which conceptual model is best?

- Direct manipulation is good for ‘doing’ types of tasks, e.g. designing, drawing, flying, driving, sizing windows
- Issuing instructions is good for repetitive tasks, e.g. spell-checking, file management
- Having a conversation is good for children, computer-phobic, disabled users and specialised applications (e.g. phone services)
- Exploring and browsing is good if the task is explorative
- Hybrid conceptual models are often employed, where different ways of carrying out the same actions are supported at the interface
  - Toolbar, Menus and Keyboard short cut offer same function
  - Can replace Expert-Mode and Novice-Mode in the UI

Interface Metaphors

- Interface designed to be similar to a physical entity but also has own properties
  - e.g. desktop metaphor, web portals
- Can be based on activity, object or a combination of both
- Exploit user’s familiar knowledge, helping them to understand ‘the unfamiliar’

Benefits
- Makes learning new systems easier
- Helps users understand the underlying conceptual model
- Can be very innovative and enable the applications to be made more accessible to a greater diversity of users

Problems with Interface Metaphors

- Sometimes break conventional and cultural rules
  - e.g. recycle bin placed on desktop
- Can constrain designers in the way they conceptualize a problem space
- Can conflict with design principles
- Forces users to only understand the system in terms of the metaphor
- Designers can inadvertently use bad existing designs and transfer the bad parts over
- Limits designers’ imagination in coming up with new conceptual models
**Data Mountain**  
(Robertson, UIST’98, Microsoft)

**“Pile” metaphor**  
(Mander et al., CHI’92, Apple)

Interaction Mode vs. Interaction Style

- **Interaction mode:**
  - what the user is doing when interacting with a system, e.g. instructing, talking, browsing or other
- **Interaction style:**
  - the kind of interface used to support the mode
    - E.g. Command, Speech, Data-entry, Form fill-in, Query, Graphical, Web, Pen, Augmented reality, Gesture

Many kinds of interaction styles available...

- Command
- Speech
- Data-entry
- Form fill-in
- Query
- Graphical
- Web
- Pen
- Augmented reality
- Gesture...and even...

Interacting via GPS and cell phone...

- Drawing an elephant by walking round the streets of a city (or other mode of transport) and entering data points along the way via the cell phone
- Example: Brighton and Hove(UK) by J. Wood by foot...track length 11.2km (see [www.gpsdrawing.com](http://www.gpsdrawing.com) for more examples)
Making art by recording where walking in a city

Interaction paradigms
- "a particular philosophy or way of thinking about interaction design" Preece, Rogers & Sharp, 2002, Interaction Design, Wiley, p60
- Past: The Desktop – intended for single user sitting in front of standard PC
- Present: “Beyond the Desktop”
- Alternative interaction paradigms
  - Ubiquitous computing
  - Pervasive computing
  - Wearable computing
  - Augmented reality
  - Tangible bits
- See advanced topics in MMI

Chapter 5
Designing Interactive Systems
- 5.1 Design vs. Requirements
- 5.2 Design and development process
- 5.3 Creativity methods
- 5.4 Tools and methods in the early design phase
  - 5.4.1 Scenario Development and Persona
  - 5.4.2 Sketches and Storyboards
  - 5.4.3 Concept Videos
- 5.5 Prototyping
- 5.6 Wizard of Oz
- 5.7 Describing and specifying interactive systems

Interactive Systems
What can be described?
- System functionality with regard to interaction
- Overall interaction concepts (metaphors, styles)
- Layout of key screens, sketches
- Layout of user interface elements (e.g. buttons, icons)
- Navigation and interaction details
- Interactive behavior of a system
- Platform requirements
- Functional assertions (e.g. login will take on average 7 seconds, average time per case is 2 minutes)
- User groups
- …

Interactive Systems
How to describe them?
- Informal
  - System descriptions in plain text
  - Scenarios and use cases
  - Sketches and designs
  - Task-action-mappings
- Semi-formal
  - Task-action-grammar
  - Abstract UI description languages
  - UML
- Implementation languages
  - XML based languages (e.g. XUL)
  - Can be used to generate a concrete UI for the target platform
- … more next term

References
- Task-action-mapping
  http://www.psy.gla.ac.uk/~steve/HCI/cscln/trail1/Lecture8.html