

# Understanding and conceptualizing interaction



# Recap

- HCI has moved beyond designing interfaces for desktop machines
- About extending and supporting all manner of human activities in all manner of places
- Facilitating user experiences through designing interactions
  - Make work effective, efficient and safer
  - Improve and enhance learning and training
  - Provide enjoyable and exciting entertainment
  - Enhance communication and understanding
  - Support new forms of creativity and expression

# Understanding the problem space

- What do you want to create?
- What are your assumptions?
- Will it achieve what you hope it will?

# A framework for analysing the problem space

- Are there problems with an existing product?
- Why do you think there are problems?
- Why do you think your proposed ideas might be useful?
- How would you see people using it with their current way of doing things?
- How will it support people in their activities?
- Will it really help them?

# An example

- What were the assumptions made by cell phone companies when developing WAP services?
- Was it a solution looking for a problem?



# Assumptions: realistic or wish-list?

- People want to be kept informed of up-to-date news wherever they are - reasonable
- People want to interact with information on the move - reasonable
- People are happy using a very small display and using an extremely restricted interface - not reasonable
- People will be happy doing things on a cell phone that they normally do on their PCs (e.g. surf the web, read email, shop, bet, play video games) - reasonable only for a very select bunch of users

# From problem space to design space

- Having a good understanding of the problem space can help inform the design space
  - e.g. what kind of interface, behavior, functionality to provide
- But before deciding upon these it is important to develop a conceptual model

# Conceptual model

- Need to first think about how the system will appear to users (i.e. how they will understand it)
- A conceptual model is a high level 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”



# First steps in formulating a conceptual model

- What will the users be doing when carrying out their tasks?
- How will the system support these?
- What kind of interface metaphor, if any, will be appropriate?
- What kinds of interaction modes and styles to use?

Always keep in mind when making design decisions how the user will understand the underlying conceptual model

# Conceptual models

- Many kinds and ways of classifying them
- Here we describe them in terms of core activities and objects
- Also in terms of interface metaphors

# Conceptual models based on activities

- 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

# 1. 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. CAD, word processors, VCRs, vending machines
- Main benefit is that instructing supports quick and efficient interaction
  - good for repetitive kinds of actions performed on multiple objects

## 2. 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 Bob and 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 the question:  
"How many legs does a centipede have?" and the system responds:

**You asked: How many legs does a centipede have?**

**Jeeves knows these answers:**

**Where can I find a definition for the math term leg?**

**Where can I find a concise encyclopedia article on ? centipedes?**

**Where can I see an image of the human appendix?**

**Why does my leg or other limb fall asleep?**

**Where can I find advice on controlling the garden pest ? millipedes and centipedes?**

**Where can I find resources from Britannica.com on leg ?**

# 3. Manipulating and navigating

- Involves dragging, selecting, opening, closing and zooming actions on virtual objects
- Exploit's users' knowledge of how they move and manipulate in the physical world
- Exemplified by (i) what you see is what you get (WYSIWYG) and (ii) the direct manipulation approach (DM)
- Shneiderman (1983) coined the term DM, came from his fascination with computer games at the time



# 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 become screen space 'gobblers'
- Moving a mouse around the screen can be slower than pressing function keys to do same actions

## 4. Exploring and browsing

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

# Conceptual models based on objects

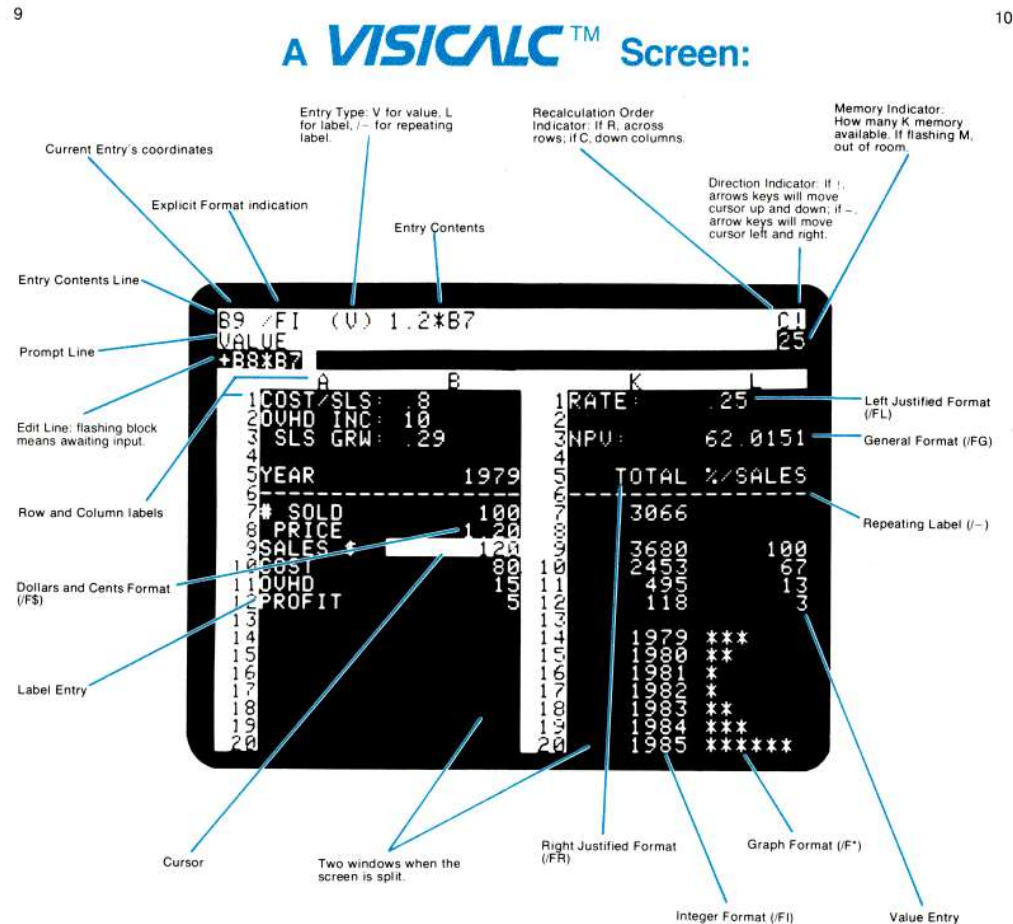
- Usually based on an analogy with something in the physical world
- Examples include books, tools, vehicles
- Classic: Star Interface based on office objects



Johnson et al (1989)

# Another classic: the spreadsheet (Bricklin)

- Analogous to ledger sheet
- Interactive and computational
- Easy to understand
- Greatly extending what accountants and others could do



# 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)
- Hybrid conceptual models are often employed, where different ways of carrying out the same actions is supported at the interface - but can take longer to learn

# 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'
- Conjures up the essence of the unfamiliar activity, enabling users to leverage of this to understand more aspects of the unfamiliar functionality



# Benefits of interface metaphors

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

# Problems with interface metaphors

- Break conventional and cultural rules
  - e.g. recycle bin placed on desktop
- Can constrain designers in the way they conceptualize a problem space
- 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

# Conceptual models: from interaction mode to 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. speech, menu-based, gesture

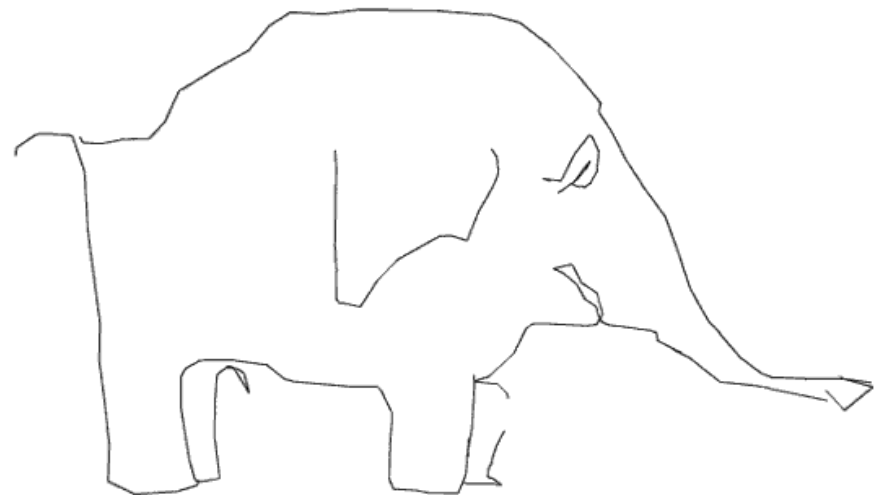
# Many kinds of interaction styles available...

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

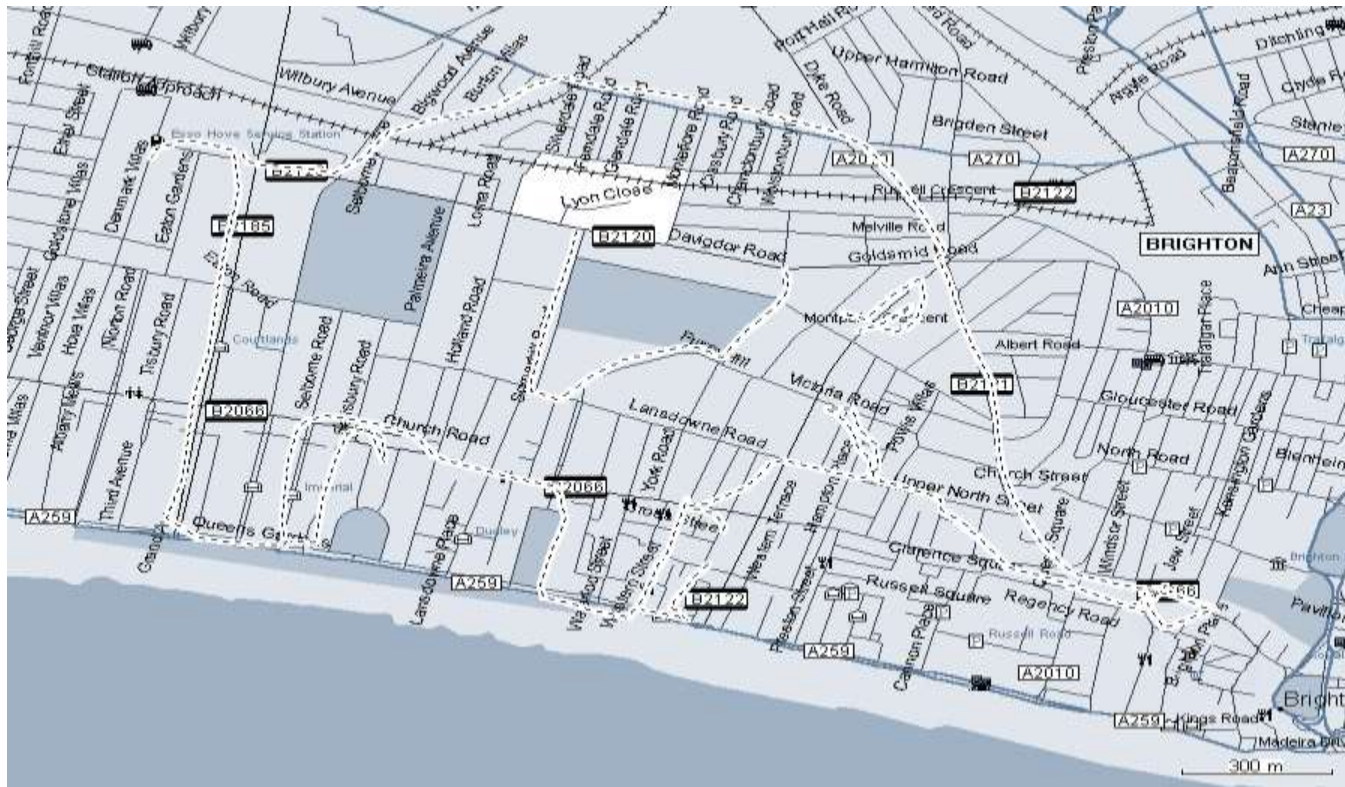


# Interacting via GPRS enabled cell phone...

- 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

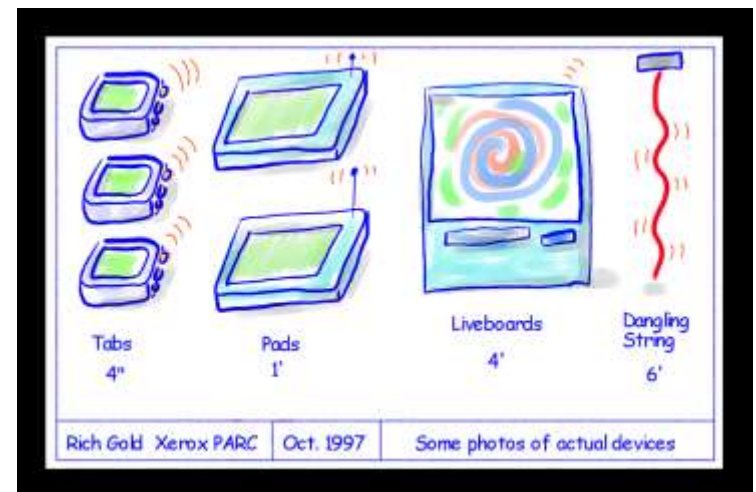


# Which interaction style to choose?

- Need to determine requirements and user needs
- Take the budget and other constraints into account
- Also will depend on suitability of technology for activity being supported
- This topic will be covered more later when discuss how to actually design conceptual models

# Interaction paradigms

- Another form of inspiration for conceptual models
- From the desktop to ubiquitous computing (embedded in the environment)





# Examples of new paradigms

- Ubiquitous computing (mother of them all)
- Pervasive computing
- Wearable computing
- Tangible bits, augmented reality
- Attentive environments
- Transparent computing
  - and many more....

# Two examples: BlueEyes (IBM) and Cooltown (HP)

- Visionary approaches for developing novel conceptual paradigms



[Almaldeen.ibm.com/cs/blueeyes/](http://Almaldeen.ibm.com/cs/blueeyes/)  
[cooltown.hp.com/mpulse/backissues/0601/0601-cooltown.asp](http://cooltown.hp.com/mpulse/backissues/0601/0601-cooltown.asp)

# Summary points

- Important to have a good understanding of the problem space
- Fundamental aspect of interaction design is to develop a conceptual model
- Interaction modes and interface metaphors provide a structure for thinking about which kind of conceptual model to develop
- Interaction styles are specific kinds of interfaces that are instantiated as part of the conceptual model
- Interaction paradigms can also be used to inform the design of the conceptual model