Human–Task–Machine

Machines were expensive
Labour was cheap
Human worked harder

Machines are cheap
Labour is expensive
Machines work harder

- Evolution of HCI
  - From labour-intensive to processors-intensive
  - New hardware needs new HCI designs

Why modelling?

- Models help one to focus on matters that are relevant
- Models allow one to reason about the object being observed scientifically, e.g. by using
  - Logic
    - Statistics
    - Functions
    - Graphs
- Models are never perfect
  - Models are often based on assumptions
  - Models are simplifications of the object being observed
- Why modelling then?
  - Being able to focus is important, given limited attention span
  - Being able to reason is better than no reasoning at all

HCI Design Overview

- From machine dominant to user dominant
- User modelling is not mechanical
- HCI is a complex, unforgiving task, more craft than science

What is a model?

- Model building through:
  - Observation
  - Internal processing
  - Verification

Connections

Locations

Orientations

MODELLING THE HUMAN SYSTEM - incomplete knowledge

- HCI requires human modelling
- What does the user know?
- How would the user act?
- Where do the models come from?
- Models based on physiological and psychological experiment
- Experiments are slow and difficult to do
- Hard to interpret the (conflicting) results
- Easier to predict collective behaviour
- But not detailed, individual behaviour
GUI Design Sequence

- Style
  - menu, form, commands, WIMP, English
- Structure
  - Who initiates a dialog? What is the flow?
- Format
  - Screen display layout, message chucking
- Error-handling
  - Data validation, error messages design
- Data structure
  - avoid mismatch between system and user data

GUI DIALOGUES - design guidelines

- Design structure of dialogue
- Investigate requirements of the tasks and of user
- Develop an interaction structure
- Who initiates the dialogue? How should it flow?
- Informal evaluation of choices so far, with feedback from users
- Design forms of messages
  - Consider screen or display behind, and thinking of message data
  - Seek efficient input formats, to minimize user typing
- Design error handling
  - Establish ways of validating the data input by users
  - Consider ways of recovering from errors, or limiting error impact
  - Consider protecting users from errors
- Write error messages that are clear, meaningful and constructive
- Design data structures
  - Map structures using the user’s model of data, if possible. To avoid mismatch between the system and user views of the interaction
- Top-down design should flow from specification of user interface

DIALOGUE STYLES - five main types of GUI

- Menus
  - Choosing from a fixed set of options
- Form-filling
  - Analogous to writing on pre-printed paper forms
- Command languages
  - Also called ‘text box’ or ‘command prompt’
- Direct manipulation
  - Also called WIMP - windows, icons, mouse and pointer
    - Some people think this is the only thing GUI means...
- Natural languages
  - Conversational text, speech, gaze, gesture, etc.
    - As if the user is interacting with another human

GUI DIALOGUES Properties

- Initiative - how an interaction is started and driven forward
- Feedback - essential if there is to be interaction
- Information load - quantity of data the user has to remember or manage
- Power - amount of work done by distinct user actions or commands
- Efficiency - overall rate of working (throughput) of the HCI system
- Flexibility - multiple ways of achieving the same result
- Complexity - how choices and actions in an interaction are seen by users
- Observability - are system functions clear and easy to locate or monitor?
- Controllability - the ease and accuracy of navigation around the system
- Consistency - stable behaviour of the details in methods of interaction
- Balance - overall trade-off between all the HCI properties

Our Task in HCI Evaluation

- Evaluation in an ordinary software project:
  - Given specification (usually signed off by clients)
  - Evaluate software against specification
- In HCI design, we started with:
  - A user model (we build it, but models are never perfect)
  - Tasks (we specify them)
  - Machine (we should know it well)
- We designed our style, structure, format, error-handling, data structure
- Now we’ve got negative feedback from the user
  - What to blame/improve? user model? or the design?

GUI Design can be Frustrating

- HCI is a job that is difficult to please
  - Like the Estate Department
  - One can only get it wrong
- You may have done 99.9% of the things right
  - Users may not appreciate that
    - (actually, some do!)
- Users will pay attention to your 0.1% annoying feature
- The final 0.1% is always hard to fix
  - Just like the final bugs in programming…