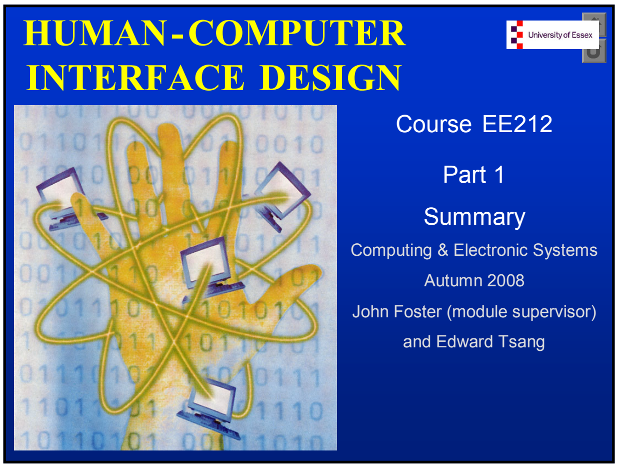


HUMAN-COMPUTER INTERFACE DESIGN

University of Essex

Course EE212

Part 1

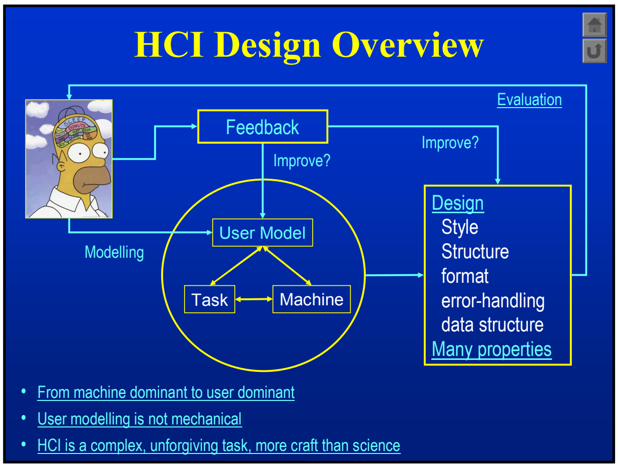
Summary

Computing & Electronic Systems

Autumn 2008

John Foster (module supervisor)

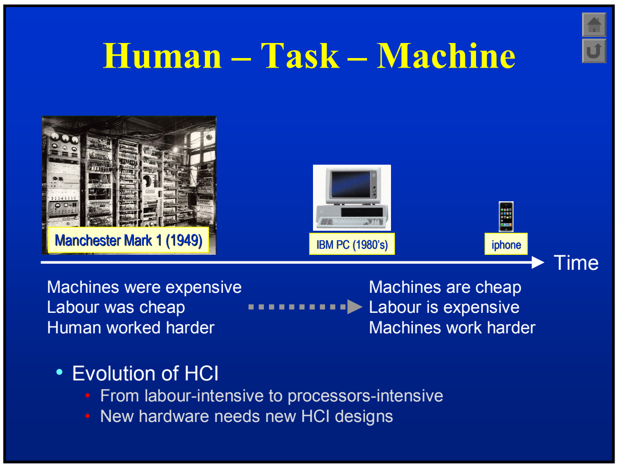
and Edward Tsang



HCI Design Overview

Evaluation
Feedback
Improve?
Improve?
Modelling
User Model
Task
Machine
Design
Style
Structure
format
error-handling
data structure
Many properties

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



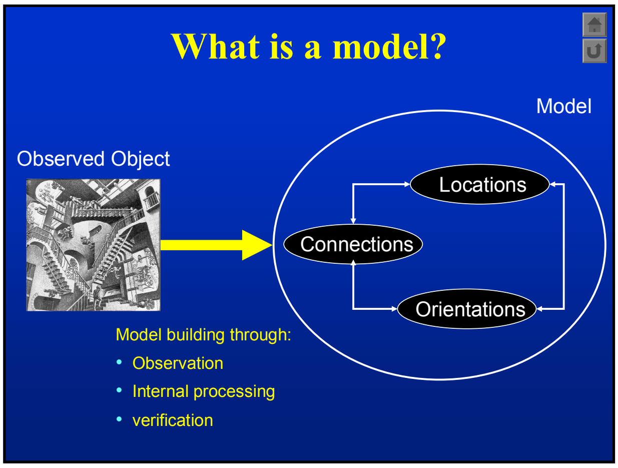
Human – Task – Machine

Manchester Mark 1 (1949)
IBM PC (1980's)
iphone
Time

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

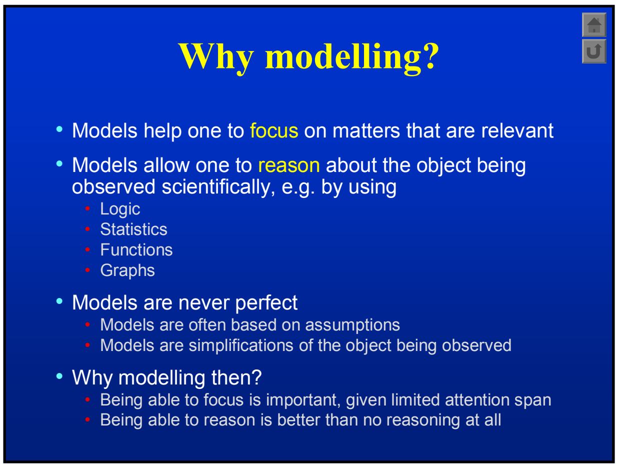


What is a model?

Model
Observed Object
Locations
Connections
Orientations

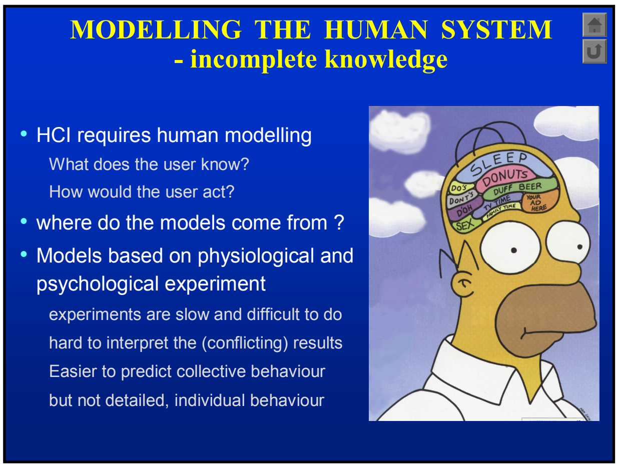
Model building through:

- Observation
- Internal processing
- verification



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



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 *formats* of messages
- consider screen or display layout, and ‘chunking’ of message data
- seek efficient input formats, to minimise 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…