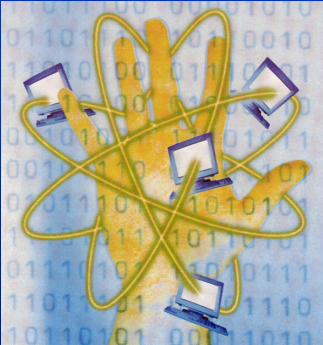


# HUMAN-COMPUTER INTERFACE DESIGN

University of Essex

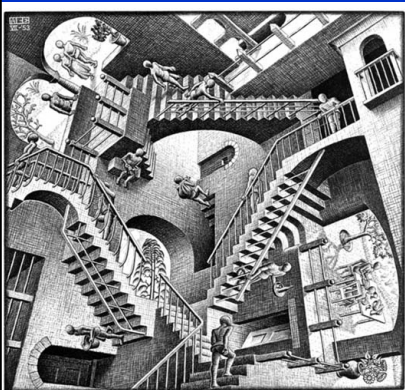
Course EE212  
Part 1, Section 4  
GUI dialogues - properties, principles and guidelines

Computing & Electronic Systems  
Autumn 2008  
John Foster (module supervisor) and Edward Tsang



1

## M C ESCHER - 'Relativity'

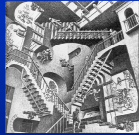


- Makes sense locally... in several different localities
- Makes no sense globally... yet the conflicting localities *match* at their edges
- Our mental models (of world or of HCI) can behave similarly must understand the subtleties

1C

## What is a model?

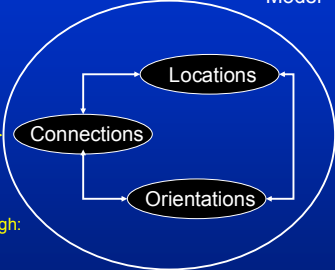
Observed Object



Model building through:

- Observation
- Internal processing
- verification

Model

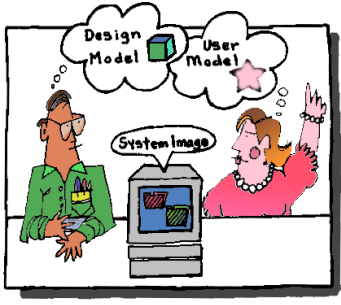


1

## 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

## GUI DESIGN - conceptual (mental) models



- We perceive the world by (mentally) modelling it ...

1A

## GUI DESIGN PRINCIPLES - task characteristics

- Establish what the **system** ought to do ... it may seem obvious, but complex systems can be hard to specify fully such a specification is called a **task taxonomy** there are formal methods to discover and define task taxonomy usually produces the designer's view of the system
- Establish the **user's model** of what the system ought to do should be more than just a wish-list of 'I want a system to ...' survey a range of users, seeking logical knowledge from their viewpoint there are also formal methods to discover and define user views
- System design** not usually a mechanistic process, but needs creativity to reconcile conflicts between views of designers, users and system capability - usually becomes a **subjective** question of balance or compromise tempting to base new designs on earlier ones, but this leaves designer 'blind' to wider or more radical ideas about improvement

2

## GUI DIALOGUES Properties

- Abstract qualities that describe HCI interactions
  - initiative* - how an interaction is started and driven forward by the computer, by the user or by both  
initiative can, and often does, vary with the task in hand
  - feedback* - essential if there is to be interaction  
good feedback helps the user learn what they did right or wrong  
prompt feedback is important, because slowness impairs relevance
  - information load* - quantity of data the user has to remember or manage  
it should be appropriate to the level of user skill or experience
  - power* - amount of work done by distinct user actions or commands
  - efficiency* - overall rate of working (throughput) of the HCI system  
also the amount of work per unit cost of HCI system

3

## GUI DIALOGUES properties

- More qualities that describe HCI interactions
  - flexibility* - multiple ways of achieving the same result
  - complexity* - how choices and actions in an interaction are seen by users (regardless of how complex the underlying software is ...)  
minimise by using logical grouping of related or similar actions  
minimise by creating structure - hierarchical or orthogonal
  - observability* - are system functions clear and easy to locate or monitor ?  
needed for good feedback about system's capacity for interaction

4

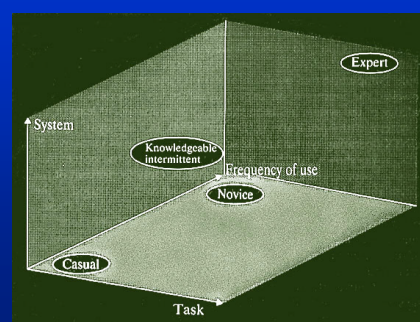
## More on GUI Dialogues

- More qualities that describe HCI interactions
  - controllability* - the ease and accuracy of navigation around the system  
where has the user been ?  
where is she now ?  
where can he go from here ?
  - consistency* - stable behaviour of the details in methods of interaction  
important for fast learning and to develop a mental model of system  
rewards extrapolation and encourages exploration  
applies to layout, display methods, data entry and parameter syntax
  - balance* - overall trade-off between all the HCI properties  
abstract and intangible, but important in complex systems; e.g. OS  
part of HCI 'look and feel' that experienced users become fond of

6

## GUI DIALOGUES - balance should allow for human variability

- Users vary in HCI:  
Approaches  
Expectation  
Strengths  
Needs



7

## GUI DESIGN PRINCIPLES - matching human capabilities

- Screen design for text
  - use logical, task-related *sequence* of display for collection of text items  
put related items close together, creating *groups* of information  
helps to 'recode' information into fewer but bigger 'chunks' in STM
  - avoid clutter and emphasise *spaciousness* - "spaces matter"  
helps to reduce visual search time, by separating different 'chunks'
  - ensure *consistency* of layout between different text screens  
helps users to learn how to operate the system
  - maintain *simplicity*, displaying appropriate level and amount of text data  
avoids overloading the user's STM
  - text data must be *relevant* to the task, but displaying all the data that might occasionally be relevant can mean displaying too much - conflicting with the need for simplicity

8

## GUI DESIGN PRINCIPLES - matching human capabilities

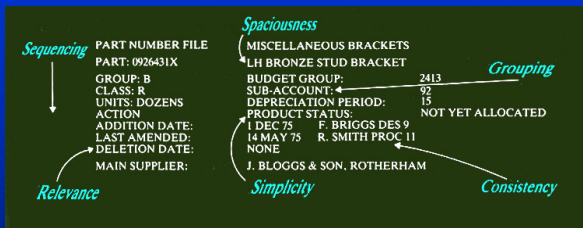
```

PART NUMBER FILE SUB-FILE MISC BKTS
SUPPLIER J. BLOGGS & SON, ROTHERHAM
PART 0926431X DESCRIPTION LH BRONZE STUD BRACKET
GROUP B CLASS R STATUS NOT YET ALLOCATED
SUB-ACCOUNT 92 BUDGET GROUP 2413
QUANTITY UNIT DOZENS DEPRECIATION PERIOD 15 ACTION
DATE OF ADDITION 1/12/75 ADDED BY F. BRIGGS DES9
DATE LAST AMENDED 14/5/75 AMENDED BY PROC 11 R. SMITH
DATE OF DELETION
COMPONENTS NONE
SUB ASSEMBLIES NONE
  
```

- Example of bad screen design for text
  - no visual structure - a mass of characters
  - 'chunked' at word or character level - tends to overload STM
  - hard to search for a particular category of information

9

### GUI DESIGN PRINCIPLES - matching human capabilities



- Example of a better text screen design (Stewart 1976)  
visual structure - groups of information  
'chunked' at category, or type of information, level  
easier to search for a particular category of information

10

### GUI DESIGN PRINCIPLES - technological progress

- Screen design for graphics  
display technology continues to improve, in fixed and mobile types  
screen resolution and response time are no longer serious limitations for many GUI applications, giving greater freedom to graphic design  
only low-cost applications use monochrome, low-resolution displays - these need care in graphic design to exploit limited resources well
- Graphics technology can be limiting for high-quality systems  
that use real-time animation and rendering of 3-D scenes  
that use real-time animation and rendering of user-controlled avatars  
- games industry is driving down the cost of such systems

11

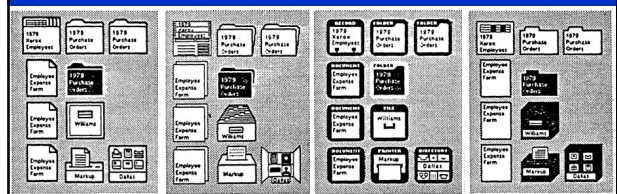
### GUI DESIGN PRINCIPLES - five principles of good graphic design

- Manipulable objects  
now a well-established part of window, icon, menu, pointer (WIMP) systems
- User focus and visual order  
gives feedback about which objects are 'selected' for use
- Revealed structure  
what you see should sometimes be more than what you get
- Appropriate and consistent graphic vocabulary  
for good information flow
- A match with the medium  
graphics should reflect limits of simple displays e.g. mobile phones

12

### GOOD GRAPHIC DESIGN - user focus and visual order

- Which icon is 'selected' for use ?

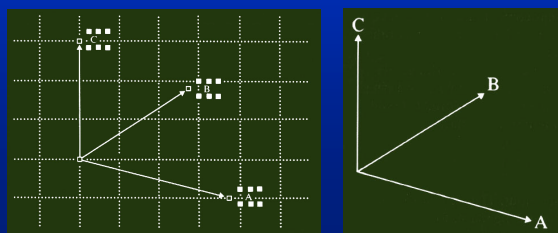


reverse-video is often used to indicate selection of GUI objects  
should give strong visual cues, without creating confusion or doubt  
needs care in the graphic design of icons and background  
- choice of brightness, colour, shape  
(from **Star system**, late 1970s - monochrome and low-resolution)

13

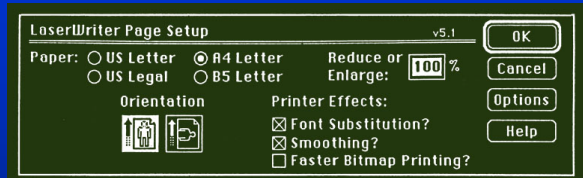
### GOOD GRAPHIC DESIGN - revealed structure

- WIMP systems are famous for being 'WYSIWYG' -  
what you see is what you get (on hard copy or output file)  
often useful to show *more* than what you get, and reveal inner structures  
'handles' to manipulate objects or data, grids for optional alignment  
common in drawing, image-editing, CAD and page-layout applications



14

### GOOD GRAPHIC DESIGN - consistent graphic vocabulary



- Standardised graphics - exploits subconscious perception  
'radio buttons' - for mutually exclusive choices, 1 out of N  
'check boxes' - for inclusive, independent choices  
icons - for easy recognition of simple choices, without words  
buttons - to activate choices or navigate further options  
default button - the action of <return> key is indicated by 'heavy' lines  
numeric boxes - 'favourite' box *pre-selected* for instant data entry
- Fast learning - understand one dialogue, then use them all

15

## GUI DESIGN PRINCIPLES - responding to human needs

- **Response time**
  - slow response is always a disadvantage, but the meaning of 'slow' varies with the task in hand -
    - > 15 seconds - **not** an interactive interface, users get bored waiting
    - 4 to 15 secs - too slow for effective STM, acceptable for major actions
    - 2 to 4 secs - causes a loss of concentration if happens frequently
    - < 2 secs - acceptable for many interactive actions and commands
    - almost instantaneous - needed for mouse-tracking and text entry
- **Variability of response time**
  - user concentration and performance tend to decline if the system response time keeps changing by a large amount

17

## GUI DESIGN PRINCIPLES - friendly and forgiving ?

- **Error handling**
  - errors in data entry or with commands are common, even with experts need to protect users from system errors, other users and themselves
  - minimise 'error propagation' by aiming to limit the impact of errors
  - ideally, design reversible processes that give the user an 'undo' action
  - write clear and constructive error messages, in a positive tone
    - not < Fatal error 7037 @ A23F#, user "Sarah" illegal parameter >
    - or <segmentation error>
- **Avoid rejecting pseudo-errors** (or use on-screen 'hints')
  - interfaces that are pedantic or fussy about data formats are bad design
  - < 4th October 2003, 4 Oct 2003, 4 10 2003 > all **rejected**
  - < 04 October 2003 > accepted, because day has **two** digits

18

## GUI DESIGN PRINCIPLES - meeting operational needs

- **Documentation**
  - very important subject commercially
  - well-designed, easy-to-use and high-quality documentation is vital for :
    - designers - to avoid errors and ensure specification is understood
    - trainers - to create teaching materials and examples in the system
    - users - to reassure the knowledgeable and to guide the novices
    - maintenance - to fix 'bugs' and install upgrades
  - a different level and style of documentation is required for each level
- **For more information about documentation**
  - "Human performance engineering - a guide for system designers" Bailey R W, Prentice-Hall 1982 (Chapter 19)
  - "Designing the user interface - strategies for effective human-computer interaction" Shneidermann B, Addison-Wesley 1987 (Chapter 9)

19

## IS THIS GOOD GUI DESIGN ? - does it meet a driver's needs ...



- **Information overload ?**
  - inconsistent layout, hard to learn, variable response time and distracting
  - what I'd call being "flashy without function" ?

20

## GUI DIALOGUES - design guidelines

- **Sequence of five main steps, in top-down sequence**
  - choose **style** of dialogue (menu, form, commands, WIMP, English)
    - based on user models and task characteristics
    - eliminate unlikely styles of dialogue
    - select or combine best choices
  - 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

21

## GUI DIALOGUES - design guidelines

- **Sequence of five main steps continued**
  - 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

22

## GRAPHICAL USER INTERFACE DESIGN - design components

- **Support for the designer** (enforces design guidelines too...)
  - most WIMP systems provide ready-made GUI software components, called widgets
  - these widgets match the style, guidelines and 'look and feel' of system
- **Using widget sets**
  - designer works at a higher conceptual level, for greater productivity, compared with lower-level 'library-type' support software
  - automatically conforms to some important aspects of design guidelines
  - widget sets can often be extended, by the designer, to create additional actions or functions
  - using widgets can generate inefficient, slow code

23

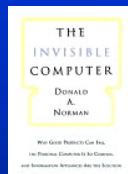
## GUI Design can be Frustrating

- 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...



## REFERENCE ON DESIGN - provocative stuff, not needed for course

- 50 year old adolescent rebel ?
  - many powerful views on computers and technology
  - worked a lot for Apple but criticises Macintoshes
  - someone who wants to change the world
  - radical fervour



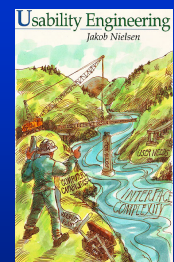
"The invisible computer" Donald A Norman, MIT Press 1998  
and <http://www.jnd.org/dn.pubs.html>  
- Norman's WEB site - lots of essays, *free* book Chapters

- **Recommended -**  
but be prepared to spend hours getting both delighted and annoyed

25

## FURTHER READING ON DESIGN - provocative stuff, not needed for course

- Co-founded UI company with D. Norman
  - also has powerful views on technology design
  - also worked for Apple, yet criticises Macintoshes
  - some radical fervour, but cooler than Norman
  - see <http://www.useit.com/>
  - lots of papers, essays and a few rants ...
  - such as : "Splash screens must die"
  - "The death of file systems"
  - "Use and misuse of focus groups"
  - "Useability Engineering" Jakob Nielsen, Morgan Kaufmann 1994.
- **Recommended, but**  
he's a consultant - may use eccentric language for commercial motives!



26