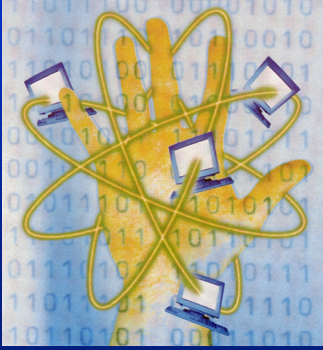


HUMAN-COMPUTER INTERFACE DESIGN

University of Essex



Course EE212 / CE653


Part 1, Section 3
GUI dialogues - design styles

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

1

COMPUTER HISTORY - radical changes

Leo1 at Cadbury Hall 1951



- **Early computers (1940s - 1960s)**
large and very expensive company goods
priority was efficient use of processor and memory - scarce resources
operators trained to serve the machine
'user interface' was punched cards, paper tape, switches and lights
non-interactive, non-real-time (batch mode) response
- **Today**
inexpensive, small and cool, domestic goods - becoming invisible
priority is usability - processor and memory resources are abundant
machine serves users, often just a single user
user interface is graphical, sometimes accepting handwriting or speech
- **Press and popular media - focus is "GigaHertz & Terabits"**
more *significant* is change in HCI, for usability and real-time interactivity

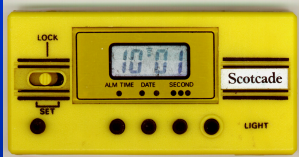

2

COMPUTERS EVERYWHERE - but we don't call them that

- Computer technology embedded into other products -
video recorder, television, radio, mp3 Walkman
microwave oven, washing machine, central heating controller
diary, address book, notepad
toys, hand-held games
mobile telephone, fax machine
photocopier, vending machine
machine tools, industrial process control
cars, bus-stops (live timetables), ship's bridge, aircraft flight-deck
- Computation is not the main purpose (any more)
- Graphical user interfaces (GUI) are everywhere -
low cost of liquid crystal displays and 'conductive rubber' buttons, means that all the products listed above are likely to have a form of GUI

3

GRAPHICAL USER INTERFACE - in a pocket alarm clock ?

7 graphical symbols - colon, bell, alarm, 4 x 7-segment numbers / letters steady or flashing

- **2 modes - read and set**
interactive interface - symbols change significance with mode
- **Read mode options :**
time (hours, minutes), time (seconds), date, alarm time
- **Set mode options:**
time (hours, minutes), date (day, month), alarm time (hours, minutes)
alarm on/off, hourly chime on/off, run clock (seconds start from zero)

4

DESIGN OF GUI DIALOGUES - analysing construction and use

- **GUI means ...**
any interface with interactive graphical or symbolic display
text-based screen counts as GUI, labelled switches and lights do not
- **Different styles of dialogue :**
approaches to **presentation** (data output) and **activation** (data input)
depends on available hardware, the task and on user skills and needs
- **Properties of any dialogue :**
subjective qualities and objective characteristics of a dialogue style
- **GUI dialogue principles and guidelines :**
task characteristics
sequence of design steps

5

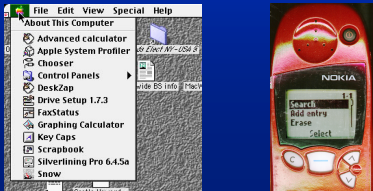
DIALOGUE STYLES - five main types of GUI

- **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

6

DIALOGUE STYLES - menus

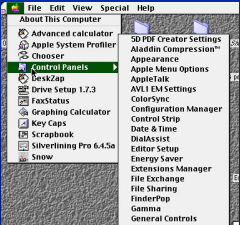
- So widespread, they're hard to avoid - widely used in personal computers and handheld electronic devices 'alarm clock' example was a very simple kind of menu system - choices were implied by mode or symbol behaviour (eg. flashing)
- Mechanism user chooses from a fixed set of options, usually in several stages use keyboard entry, button push or a pointer (mouse, joystick)



7

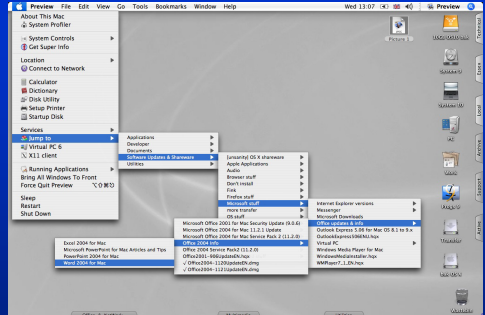
DIALOGUE STYLES - menus

- Structure
 - simple systems have single menu, using 'annunciator-type' symbols
 - more complex systems have multiple menus - in a linear sequence, eg. menu 1 -> menu 2 -> menu 3 -> menu 1 in a hierarchy, either by replacement or cascade in combination with other styles, especially form-filling
- Number of choices
 - typically > 4 but < 12 choices per menu
 - too many choices becomes confusing



8

CASCADING MENUS



- Advantages and disadvantages ...
 - quick visibility of folders & files - easy to locate where to put something
 - requires much screen space
 - later menus can *hide* earlier ones, reducing the visibility of choices

9

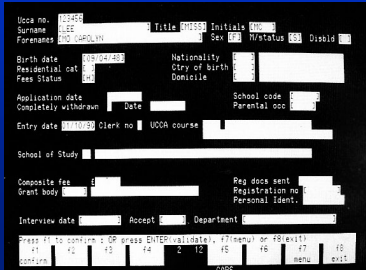
DIALOGUE STYLES - menus

- Advantages
 - minimal typing required
 - well-defined, logical structure
 - easy learning by user
 - low computer memory load
 - straightforward software design
 - CAD design tools available
- Disadvantages
 - sometimes slow or unwieldy
 - not suitable for data entry
 - can be frustrating for experts
 - consumes much screen space
- Conclusions about menus :
 - strongly recommended for inexperienced users
 - experts need 'type-ahead bypass' or direct keyboard short-cuts
 - menus in WIMP systems appeal to both novices and experts

10

DIALOGUE STYLES - form-filling

- Very widely used in industry and business - hotels, banks, travel, personnel records, purchasing, government screen resembles paper form, good match with 'paper ticket' systems
- Keyboard data entry
 - optional pointer (mouse, joystick)
 - cursor prompts user for next entry
 - data errors can be checked in each field



11

DIALOGUE STYLES - form-filling problems

- Syntax of entries within the form can create difficulties -
 - text or number too big for field
 - text entered into number field, or vice versa
 - some characters illegal in some fields, eg. commas in numbers
 - data entered in wrong sequence, when fields used depend on data fields with defaults that cannot be changed after form entry begins
- Other difficulties
 - how to provide on-line help ?
 - how to allow for correction of entry errors ?
 - in each field - requires use of a 'field terminator' eg. <tab>, <return> at form end - must allow for data-dependencies in valid field choice
 - how to provide navigation (and short-cuts) through multi-page forms

12

DIALOGUE STYLES - form-filling

- Advantages
 - familiar method for users
 - simple entry of data
 - some training required
 - well-defined, logical structure
 - low computer memory load
 - straightforward software design
 - CAD design tools available
- Disadvantages
 - sometimes slow or unwieldy
 - consumes much screen space
 - not ideal for selecting commands
 - no easy navigation between forms
 - requires high-resolution display
 - customised forms often needed
- Conclusions about form-filling :
 - suitable for all levels of user skill
 - best for parameter (words, numbers) entry, rather than commands
 - careful attention to high-quality form organisation and design is *vital*
 - syntax and layout of form is often uniquely tailored to each system

13

DIALOGUE STYLES - command language

- Used to be common-place
 - in IBM PC-DOS, BBC computer, Apple II, Sinclair, Atari and lots more
 - still is common in Unix and at basic level of Linux and Mac OSX

```

Last login: Sun Oct 7 11:43:42 on tttyl
Welcome to Darwin!
PB-AL:~ Home$ cd /Applications
PB-AL:/Applications Home$ ls
Audacity 1.2.6      RealPlayer.app
Camino 1.0.3       Safari 2.0.4.app
Democracy IP-TV    Skype 1.4.0.49.app
Dictionary.app     Sophos 4.6.5
Firefox 1.5.0.8.app System Preferences.app
Graphic Converter 5.7.3 Text Editors
Juice (v2.1).app  Thunderbird 1.5.0.8.app
Keynote 2.0.2.app Utilities
NeoOffice 1.2.2.app Virtual PC
Office             WireTap Pro
Other Applications iDraw 1.3
Other Applications iTunes 4.9.app
PrintKit.app
PB-AL:/Applications Home$
    
```

- Snobbish view -
 - "Real programmers do it on the command line"
 - ... but how many people still use (or know) the assembly-level instruction set of their computer ?

14

DIALOGUE STYLES - command language

- Structure
 - initiative entirely with the user -
 - expected to enter syntactically-correct command, without prompting
 - command grammar can be -
 - flat : every command is described and operated in isolation
 - hierarchical : similar functions grouped in one command - eg. <edit>
 - orthogonal : eg. 'number' commands different from 'string' commands
 - mixtures of the above
- Syntax
 - can be implemented by -
 - keyword - eg. command verb or noun controls syntax of parameters
 - position of command word, and of parameters, in command string
 - combinations of the above
 - design syntax for benefit of user, not for ease of command processing

15

DIALOGUE STYLES - command language

- Advantages
 - fast and powerful
 - precise
 - concise
 - efficient
 - appeals to experts
 - flexible and user-initiated
 - very low computer memory load
- Disadvantages
 - needs long training
 - puts high memory load on user
 - very poor handling of user errors
 - needs regular use to maintain skill
 - tends to intimidate novice users
 - hard to document and debug
- Conclusions about command language :
 - very high degree of power and flexibility possible
 - needs printed manual, pocket-size command summary, on-line help
 - hard to design a language that is optimum for both user and computer

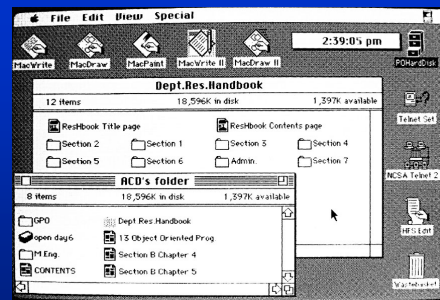
16

DIALOGUE STYLES - designing a command language

- Strategies to reduce user memory load
 - use easily memorable command words
 - use command words that are hard to confuse with each other
 - apply consistent formats to commands and parameters
 - use short command strings
 - put optional or seldom-used parameters at end of command string
 - apply defaults for missing parameters
 - provide clear error messages
 - provide built-in help facility or command
 - abandon further processing if too many errors - eg. in compilation
- This is probably why 'Basic' is still around ...

17

DIALOGUE STYLES - direct manipulation / WIMP




- Invented by Xerox PARC, in 1970s
- Pioneered by the Apple Macintosh in 1980s

18

DIALOGUE STYLES - windows, icons, menus and pointer


- Structure - use of metaphors maps onto tangible experience of users with real-world objects
- Objects in the computer - physical and logical objects are represented by image metaphors
 - files have icons and so do directories (folders)
 - disks (root folders of volumes) have special 'non-delete' icons
 the desktop, and its contents, is a metaphor for the entire computer
- Windows show contents of folders, nesting hierarchically, with arbitrary overlap allow use of non-GUI elements (buttons and indicators), as well as integrating the use of menu and form-filling interfaces



19

DIALOGUE STYLES - windows, icons, menus and pointer

- Movement metaphors - by the user and by computer
- Actions by the user command processes are initiated by manipulating metaphors (icons) aim for 'universal' consistency of action, so the same kind of action initiates an appropriate command on different types of object - double-click opens a file (run program) and a folder (list directory)
- Actions by the computer - animation conveys feedback about interface - inverse for icon selection conveys feedback on computer processing - 'system busy' watchglass gives visual focus - active window appears 'on top of' other windows



20

DIALOGUE STYLES - more about WIMP success

- WIMP widely used outside computing applications - programming video recorders and DVB set-top boxes video games, both handheld and console types 'mimic' displays for industrial process control and plant monitoring air and sea traffic control operational management of telecommunications networks aircraft cockpits
- Skill is needed to create a good WIMP HCI good-quality graphic design is essential in the choice of icons, shapes, shades and colours part of creating 'look and feel' in a WIMP GUI eg. Mac -v- Sun

21

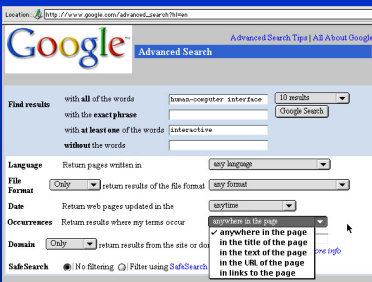
DIALOGUE STYLES - direct manipulation / WIMP

<ul style="list-style-type: none"> Advantages <ul style="list-style-type: none"> uses physical analogy fast learning easy to remember syntax encourages exploration concise and comprehensive powerful design tools available 	<ul style="list-style-type: none"> Disadvantages <ul style="list-style-type: none"> needs large, complex software needs high-resolution display needs extra device (mouse, joystick) skilled graphic design required consumes screen space slows as more windows are opened
--	---

- Conclusions about direct manipulation / WIMP : effective and appealing for all levels of user skill mixes well with other modes, such as form-filling and natural language strong trend towards wider use, due to low cost / high benefit ratio likely to expand further, as display and pointer technologies improve

22

DIALOGUE STYLES - what type of dialogue is this ?



- Hybrid or mixed-style dialogue combination of form-filling, direct-manipulation and menu elements
- Peter Norvig (Google) is an AI expert in Natural Language processing

23

DIALOGUE STYLES - natural language

- 'about to happen' for the last 20 years, now happening! task is much harder than was at first thought
- Structure any system *accepting* human communication with little or no constraint using text, speech, gesture or picture input any system *producing* human communication with little or no constraint using text, speech, gesture or picture (avatar or manikin) output
- Mechanism most often means speech input and/or output, with speech recognition and/or synthesis software potential for facial expression and body language output, with avatars tends to be used with WIMP or display, hence usually treated as a GUI

24

DIALOGUE STYLES - natural language

- Partial success - with speech systems
 - very good performance is possible (needs top-end PC) if :
 - a limited vocabulary of words is used, or
 - the computer is trained in user's speech and accent and the user is trained in computer's feedback dialogues
- Fundamental problem ...?
 - human-human language works because both have 'world model'
 - human-human communication is therefore contextually *symmetric*
 - can make a 'limited scenario' model within computer system, but
 - human-computer communication tends to be contextually *asymmetric*
 - user knows more than the computer, but is often ambiguous or vague
 - computer knows less than the user, but is always exact and precise

25

DIALOGUE STYLES - human -v- computer characteristics

- | | |
|---|--|
| <ul style="list-style-type: none"> • Human characteristics <ul style="list-style-type: none"> estimation intuition creativity adaptation subconscious concurrency processing exceptional cases associative memory access non-deterministic decisions pattern recognition in noise wide-world knowledge liable to make errors | <ul style="list-style-type: none"> • Computer characteristics <ul style="list-style-type: none"> accurate calculation logical deduction repetitive activity without tiredness consistent behaviour multitasking processing routine cases addressed storage and retrieval deterministic decisions rapid processing of data restricted domain knowledge freedom from errors |
|---|--|

26

DIALOGUE STYLES - natural language

- | | |
|---|---|
| <ul style="list-style-type: none"> • Advantages <ul style="list-style-type: none"> no special syntax needed little training required powerful flexible mixed initiative possible | <ul style="list-style-type: none"> • Disadvantages <ul style="list-style-type: none"> meaning can be ambiguous very complex software required consumes processor resources inefficient and imprecise user dialogue can be verbose opaque feedback when errors occur |
|---|---|
- Conclusions about natural language :
 - expensive, complex and inefficient for some systems
 - useful in niche applications eg. in telephony and users with disabilities
 - language vocabulary should be limited to reflect processes required
 - need to discourage novices from attributing understanding to system
 - expanding quickly, as technology advances processing costs fall

27

DIALOGUE STYLES - conclusions

- Menus
 - acceptable to novices and experienced users, good for simple systems
- Form-filling
 - well-suited to many kinds of factual, transaction-based application
 - often require customisation and are hard to design in complex systems
- Command languages
 - liked by experts, tolerated by others for the sake of power and flexibility
- Direct manipulation / WIMP
 - has become the norm in HCI
 - well-supported by high-level GUI (and application) design toolkits
- Natural languages
 - usually taken to mean speech input and / or output, with WIMP display
 - growing in popularity

28