Lecture 2
Interaction

2110646 User Interface Design

Chate Patanothai

The Interaction

- interaction models
  - translations between user and system
- ergonomics
  - physical characteristics of interaction
- interaction styles
  - the nature of user/system dialog
- context
  - social, organizational, motivational

Some terms of interaction

domain – the area of work under study
e.g. graphic design

goal – what you want to achieve
e.g. create a solid red triangle

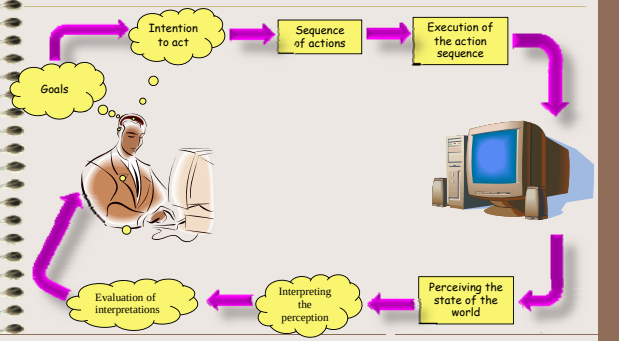
task – how you go about doing it
– ultimately in terms of operations or actions
e.g. … select fill tool, click over triangle

Note …
- traditional interaction …
- use of terms differs a lot especially task/goal !!!

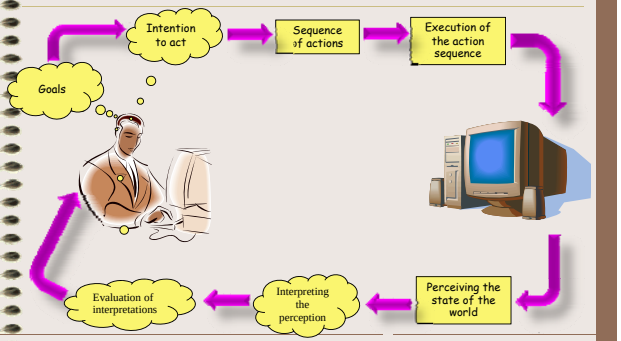
Donald Norman's model

- Seven stages
  - forming the goal
  - forming the intention
  - specifying the actions at interface
  - executing the action
  - perceiving the system state
  - interpreting the system state
  - evaluating the outcome respect to goal
- Norman's model concentrates on user's view of the interface

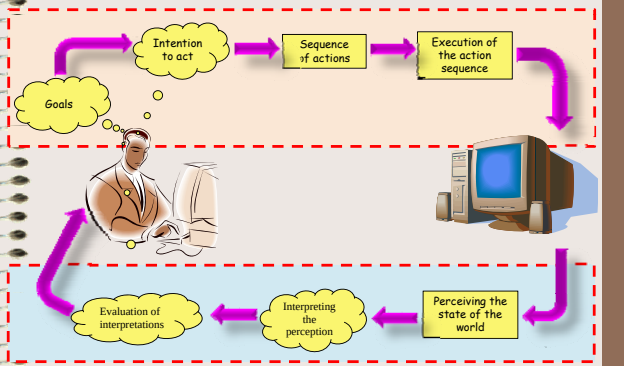
Human Activity Cycle



Human Activity Cycle



Human Activity Cycle



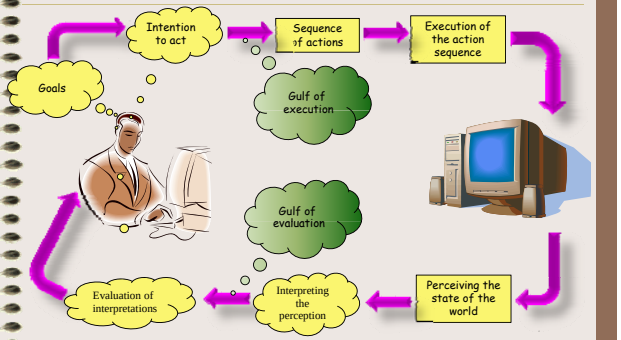
Using Norman's model

Some systems are harder to use than others

Gulf of Execution
user's formulation of actions
≠ actions allowed by the system

Gulf of Evaluation
user's expectation of changed system state
≠ actual presentation of this state

Human Activity Cycle



Human error – slips and mistakes

slip
- understand system and goal
- correct formulation of action
- incorrect action

mistake
- may not even have right goal!

Fixing things?
slip – better interface design
mistake – better understanding of system

Ergonomics

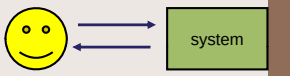
- Study of the physical characteristics of interaction
- Also known as human factors
- Ergonomics good at defining standards and guidelines for constraining the way we design certain aspects of systems

Ergonomics - examples

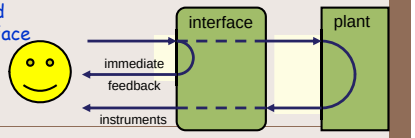
- arrangement of controls and displays
  - e.g. controls grouped according to function or frequency of use, or sequentially
- surrounding environment
  - e.g. seating arrangements adaptable to cope with all sizes of user
- health issues
  - e.g. physical position, environmental conditions (temperature, humidity), lighting, noise,
- use of color
  - e.g. use of red for warning, green for okay, awareness of colour-blindness etc.

Indirect manipulation

- office– direct manipulation
  - user interacts with artificial world



- industrial – indirect manipulation
  - user interacts *with* real world *through* interface
- issues ..
  - feedback
  - delays



Interaction styles

Command line, Menu selection,
Form-fill, Direct manipulation

Command Line

- expressing instructions to the system directly
  - function keys, single characters, short abbreviations, whole-word commands, or a combination
- powerful: offers access to system functionality
- flexible: has a number of options/parameters
- but difficult to learn to use
- better for expert users than novices

Command line interface

- Basic goals of language design
  - Precision
  - Compactness
  - Ease of writing and reading
  - Speed of learning
  - Simplicity to reduce errors
  - Ease of retention over time

Command line interface

- Higher-level goals of language design
  - Close correspondence to reality
  - Convenience in carrying out manipulations relevant to users' tasks
  - Compatibility with existing notations
  - Flexibility to accommodate novices and experts
  - Expressiveness to encourage creativity
  - Visual appeal

Command line interface

- Constraints of a language
  - The capacity for human beings to record the notation
  - The convenience in speaking (vocalizing)
- Successful languages serve the goals within the constraints.

Command line interface

- Computer programming languages
- Scripting languages
- Database query languages
- Operating systems commands
  - users issue a command and watch the output.
  - if correct, next command
  - else some strategy is adopted

Command line interface

- Users must recall notation and initiate action
- require memorization and typing
- example, UNIX command to delete blank lines from a file
  - grep -v ^$ filea > fileb

Simple command list

- Each command for a single task
- # of commands matches # of tasks
- small # of tasks → simple to learn and use
- large # of cmds → danger to confuse (e.g. vi editor of Unix)

Command plus arguments

- Each command is followed by one or more argument that indicate objects to be manipulated
- A blank (or other delimiter) is use to separated the commands from the arguments, and between the arguments

```
COPY FILEA, FILEB
DELETE FILEA
PRINT FILEA, FILEB, FILEC
```


Command plus arguments

- Keyword labels for arguments may be helpful

```
COPY FROM=FILEA TO=FILEB
```

- More chances of type, but readability is improved and order dependence is eliminated

```
COPY TO=FILEB FROM=FILEA
```


Command plus options and arguments

- Commands may have options to indicate special cases, e.g.

```
PRINT/3, HQ FILEA
PRINT (3, HQ) FILEA
PRINT FILEA -3, HQ
```

to print 3 copies of FILEA at the printer in the headquarters buildings

- More options → chance of high error rate, increase in learning

Command plus options and arguments

- Ex. One airline-reservations system
  - check the seat availability on a flight on August 21, from Washington's National Airport (DCA) to New York's La Guardia Airport (LGA) at about 3:00 PM

```
A0821DCALGA0300P
```

- expert or frequent user can take advantage of this type

Hierarchical command structure

- The full set of commands is organized into a tree structure

Action	Object	Destination
CREATE	File	File
DISPLAY	Process	Screen
REMOVE	Directory	Local Printer
COPY		Remote Printer
MOVE		

Hierarchical command structure

- Offers a meaningful structure to a large # of commands
  - 5 x 3 x 4 = 60 tasks with 5 commands
- Command-menu approach can be developed to aid novice and intermittent user
- Difficulty to know what keywords are available

The benefits of structure

- Well designed → recognize the structure into semantic knowledge
- Meaningful structure is beneficial for
  - task concepts
  - computer concepts
  - syntactic details of command languages

Consistent argument ordering

- Inconsistent

```
SEARCH file no, msg id
TRIM msg id, segment size
REPLACE msg id, code no
INVERT group size, msg id
```

- Consistent

```
SEARCH msg id, file no
TRIM msg id, segment size
REPLACE msg id, code no
INVERT msg id, group size
```


Consistent argument ordering

- Command verb versus object of interest
  - command first
    - **DISPLAY FILE**
    - **INSERT LIST**
  - object first
    - **FILE DISPLAY**
    - **LIST INSERT**

    (more appropriate when using selection by pointing on graphic displays is use)

    (leaving an object selected after an action is complete, if next action can be done without selection)

Consistent argument ordering

- Different thinking patterns
  - visually oriented interface (right brain)
  - syntax-oriented command (left brain)

Symbols versus keywords

- Command structure affects performance

Symbol editor	Keyword editor
FIND:/TOOTH/;-1	BACKWARD TO "TOOTH"
LIST;10	LIST 10 LINES
RS:/KO/,/OK/;*	CHANGE ALL "KO" TO "OK"

Hierarchical structure and congruence

- hierarchical (verb-object-qualifier)
  - **MOVE ROBOT FORWARD**
  - **MOVE ROBOT BACKWARD**
- nonhierarchical (verb only)
  - **ADVANCE**
  - **RETREAT**
- congruent (meaningful pairs of opposites)
  - **LEFE/RIGHT**
  - **FORWARD/BACKWARD**
- noncongruent
  - **GO/BACK**
  - **TURN/LEFT**

Hierarchical structure and congruence

- hierarchical congruent
  - least error
- nonhierarchical congruent
  - ***most efficient***
- hierarchical noncongruent
- nonhierarchical noncongruent
  - most error
  - least efficient

Hierarchical structure and congruence

- Congruent helps to remember the natural pairs of concepts and terms
- Hierarchical structure help to reduce memorization
  - commands + objects + qualifiers → commands x objects x qualifiers

Hierarchical structure and congruence

- In summary, sources of structure that have proven advantageous include:
  - Positional consistency
  - Grammatical consistency
  - Congruent pairing
  - Hierarchical form

* visual or perceptual form of keyboard layout is helpful

Naming and abbreviations

- Inconsistent abbreviation strategies
  - first few letters
  - first and last letters
  - first letter of each word in a phrase
  - first consonants

Specificity vs. generality

- With few names, a command set is easy to master
- With hundred of names, the choice of meaningful, organized set of names becomes more important
- Specific terms – more descriptive, more memorable
- General terms – more familiar, easier to accept

Abbreviation strategies

1. Simple truncation : first, second, third, etc. letters of each command
2. Vowel drop with simple truncation
3. First and last letter
4. First letter of each word in a phrase
5. Standard abbreviations from other contexts : QTY for QUANTITY
   XTALK for CROSSTALK
6. Phonics : focus attention on the sound, XQT for execute

Abbreviation strategies

- Truncation appears to be the most effective mechanism.
- Conflicting abbreviations appear often
- decoding of an unfamiliar abbreviation is no as easy as when vowel dropping is used

Guidelines for abbreviations

1. Simple, primary rule for most items; simple, secondary rule for conflict items
2. Abbreviations from secondary rule should have a marker.
3. # of words abbreviate by the secondary rule should be kept to minimum
4. Users should be familiar with the rules to generate abbreviations

Guidelines for abbreviations

5. Truncation is an easy rule, but may produce conflict
6. Use fixed-length abbreviations in preference to variable-length
7. Abbreviations should not be designed to incorporate endings
8. Unless there is a critical space problem, abbreviations should not be used in messages display

Guidelines for command line

- choose meaningful commands
- follow consistent syntax (grammatical structure)
- support consistent rules for abbreviation
- make command as short as possible to help prevent typing errors
- limit the number of commands
- offer macro

Menu Selection

- Set of options displayed on the screen
- Options visible
  - less recall - easier to use
  - rely on recognition so names should be meaningful
- Selection by:
  - numbers, letters, arrow keys, mouse
  - combination (e.g. mouse plus accelerators)
- Often options hierarchically grouped
  - sensible grouping is needed
- Restricted form of full WIMP system

Menu Selection

- Can eliminate training and memorization of complex command sequences
- Select an item easily with a few keystroke
- Often contrasted with command language

Semantic organization

Primary goal is to create

- Sensible
- Comprehensible
- Memorable
- Convenient semantic organization

Types

- Single menus
- Linear sequence menu
- Tree structure menu
- Acyclic network
- Cyclic network

Single menus

- Two or more items
- May require two or more screen
- May allow multiple selections
- Pop up or permanent

Single menus

- Binary menus
- Multiple item menus and radio button
- Multiple selections or check boxes
- Pull-down and pop-up menus

Binary menus

- Yes/no
- True/false
- Improved by good abbreviation
- Offer clear and specific choices that are predictable and give sense of control

Binary menu (example)

Do you want instruction (Y, N)?

Your choices are
1 – Get 12 lines of brief instruction
2 – Get 89 lines of complete instruction
3 – Go on to play the game
Type 1, 2, or 3 and press ENTER

Which one do you prefer?

Multiple-item menus and radio buttons

- Have more than two items

Who invented the telephone?
Thomas Edison
Alexander Graham Bell
Lee De Forest
George Westinghouse
Touch your answer.

Multiple-item menus and radio buttons

- In GUI, called radio buttons

Paper: ○ US Letter ◉ A4 Letter
○ US Legal ○ B5 Letter
○ No. 10 Envelope

Multiple-selection menus and check boxes

- Users can select more than one choice.

☑ **Bold**
☑ *Italic*
☐ Underline

Pull-down and pop-up menus

- Pull-down – from menu bar
- Pop-up – appears in display area in response to some keystroke

Linear sequences and multiple menus

- Series of independent menus → series of choices can be seen in sequence of command
- Ex
  - Online multiple choice exam
  - Wizards

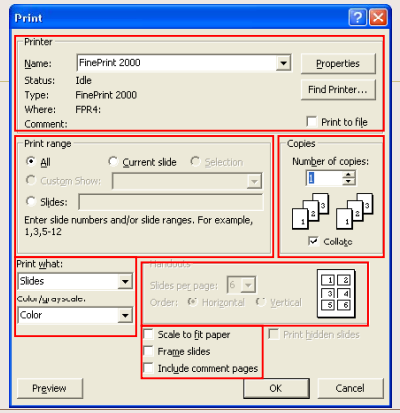
Linear sequence menus

Movement through the menu

- One decision at a time
- If cannot fit in one page, there should be some way of go back to the previous page.
- Display previous choices users have made
- How many and which menus users have not seen?

Linear sequence menu

- Guide users through decision-making
- Give clear sense of progress
- Ability to go backward or terminate or restart the sequence
- Good strategy is to place easy decision first (or upper left corner)



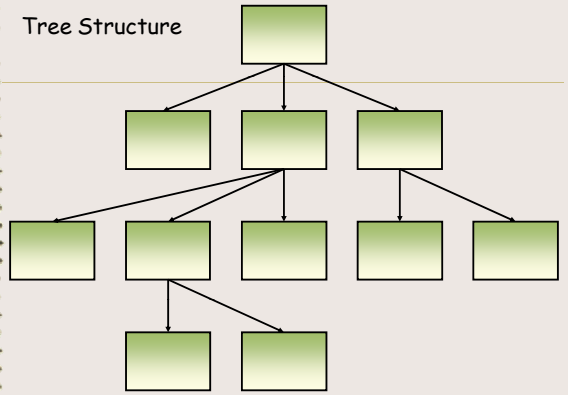
Tree structured menus

- More collections
- Groupings are complex
- No single solution (accepted by everyone)
- Improve by feedback
- Terminology to task domain

Tree structured menus

- If grouping are natural and comprehensible by user (user know what they are looking) → quick
- Example
  - Male, Female
  - Animal, Vegetables, Mineral
  - Font, Size, Style, Space

Tree Structure



Depth versus Breadth

- Depth -- # of level
- Breadth -- # of item/level
- In general
  - 4 - 8 items/level
  - 3 - 4 level
- With large menu, one or both must be compromised
- Breadth preferred over depth

Task-related grouping

- Create groups of logically similar items
  - Countries at level 1,
  - states or provinces at level 2,
  - and cities at level 3
- Form group that cover all possibilities
  - Age ranges
  - < 10, 10-19, 20-29, 30-39, > 40

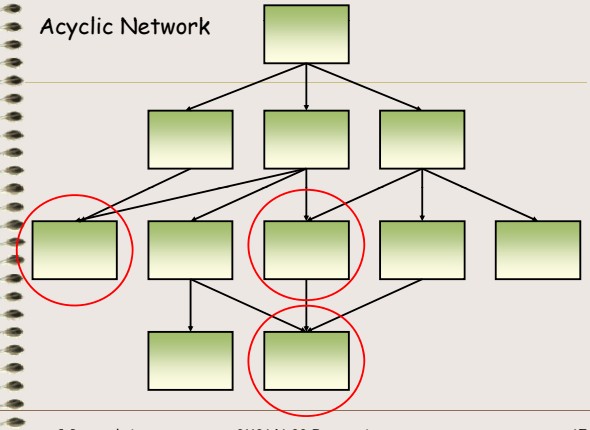
Task-related grouping

- Make sure that items are non overlapping
- Use familiar terminology, but ensure that items are distinct from one another

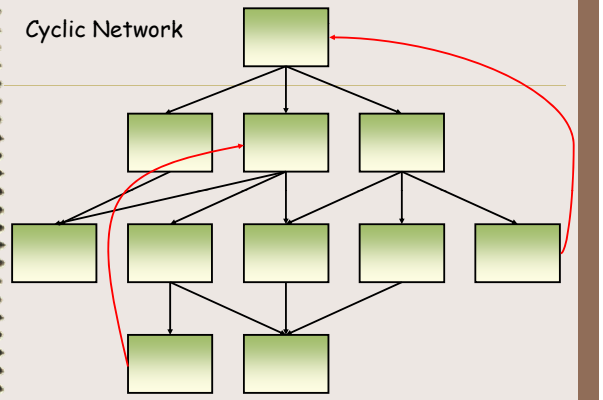
Acyclic and cyclic menu networks

- Sometimes it makes sense to go to a choice from two or more parts of a tree
- Go across sections of a tree rather than restart from main menu
- Potential for getting lost increase
- Need stack to keep visiting history

Acyclic Network



Cyclic Network



Item presentation sequence

Natural sequence

- Time
- Numerical ordering
  - Ascending or descending order
- Physical Properties
  - Increasing or decreasing length, area, volume, weight, etc.

Item presentation sequence

When no task-related ordering

- Alphabetic sequence of terms
- Grouping of related items
- Most frequently used item first
- Most important items first

Response time and display rate

Response time – the time it takes for the system to begin displaying information in response to a user selection

Display rate – the speed at which the menu are displayed

- Deep menu trees or complex traversals become annoying to the user if the response time is slow

Fast movement through menus

- Menu with type ahead
- Menu name or bookmarks for direct access
- Menu macro

Menu with type ahead

- User does not have to wait to see the menu
- When menus are familiar
- When response time/display rate are slow

Menu names or bookmarks for direct access

- Simple naming scheme allows users to directly access pages
- Useful only if there is a small # of possible destinations that each user need to remember
- WWW browsers provide bookmarks for quick access to specific pages

Menu macro

- Users defined their own commands

Phrasing of menu items

- use familiar and consistent terminology
- ensure that items are distinct from one another
- use consistent and concise phrasing
- bring the keyword to the left

Menu selection guidelines

- Use task semantics to organize menu (single, linear sequence, etc.)
- Prefer broad and shallow to narrow and deep
- Show position by graphics, numbers, titles
- Use item names as titles for tree

Menu selection guidelines

- Use meaningful groupings of items
- Use meaningful sequencing of items
- Make items brief, begin with keyword
- Use consistent grammar, layout, terminology
- Allow typeahead, jumpahead, or other shortcuts

Menu selection guidelines

- Allow jumps to previous and main menus
- Provide online context-sensitive help

Menu Structures

- Single menus
- Sequential linear menus

Sequential Linear Menus

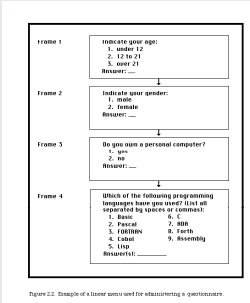


Figure 2.2 Example of a linear menu used for administering a questionnaire.

- only one path
- the user may need to go back and change an answer in a previous menu
- the user may want different order

Simultaneous Menus

- all options are available simultaneously
- but, might not be fitted in one screen

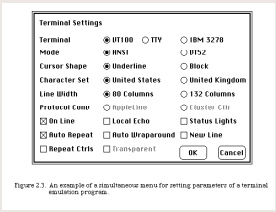


Figure 2.3. An example of a simultaneous menu for setting parameters of a terminal emulation program.

Hierarchical Menus

- tree structure
- symmetric or asymmetric
- depth
  - the number of levels that one must traverse to reach the terminal node
- Breadth
  - the number of alternatives at each level
- May not match user flow

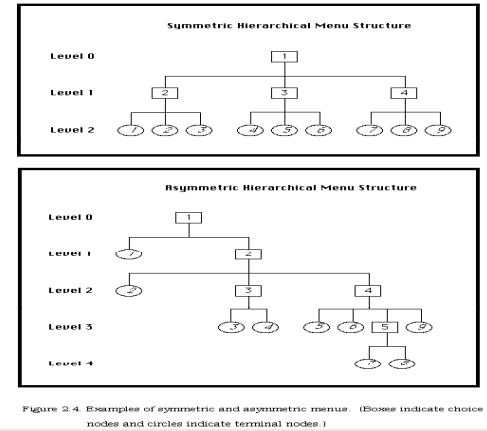


Figure 2.4. Examples of symmetric and asymmetric menus. (Boxes indicate choice nodes and circles indicate terminal nodes.)

Asymmetric menus

- Average depth
  - equal weighting of all possible paths
  - weighting by probability of path given equal choice probability
  - weighting by probability of path given observed choice probabilities

Asymmetric Menus

$$Avg.D = \frac{\sum w_i n_i}{\sum w_i}$$

$$Avg.B = \frac{\sum w_i k_i}{\sum w_i}$$

Asymmetric Menus

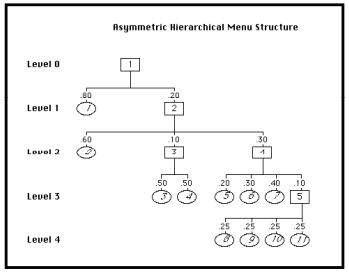


Figure 2.5. An asymmetric menu varying in depth and breadth. (Number on the lines indicate hypothetical choice probabilities. Numbers in the boxes are frame numbers and numbers in the circles are path numbers.

Asymmetric Menus

Table - Measures of Depth in Asymmetric Menus: Probability Weights

Path	Depth	Equal Path	Equal Choice	Observed Choice
1	1	1/11	1/2	0.8
2	2	1/11	1/6	0.12
3	3	1/11	1/12	0.01
4	3	1/11	1/12	0.01
5	3	1/11	1/24	0.012
6	3	1/11	1/24	0.018
7	3	1/11	1/24	0.024
8	4	1/11	1/96	0.0015
9	4	1/11	1/96	0.0015
10	4	1/11	1/96	0.0015
11	4	1/11	1/96	0.0015
Average		3.09	1.87	1.29
Std.Dev		0.9	0.97	0.62

Connected Graph Menus

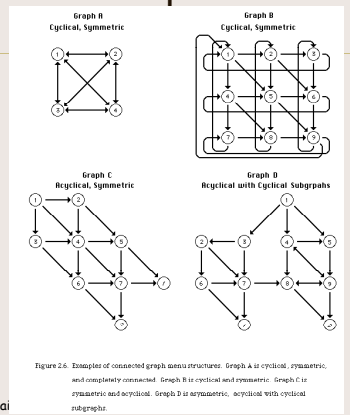


Figure 2.6. Examples of connected graph menu structures. Graph A is cyclical, symmetric, and completely connected. Graph B is cyclical and symmetric. Graph C is symmetric and acyclical. Graph D is asymmetric, acyclical with cyclical subgraphs.

Event Trapping Menus

- special function key
- pull-down
- pop-up

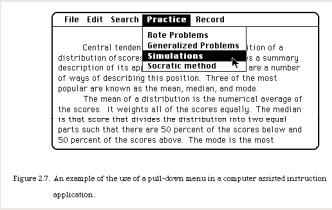


Figure 2.7. An example of the use of a pull-down menu in a computer assisted instruction application.

Form-fill

- Primarily for data entry or data retrieval
- Screen like paper form.
- Data put in relevant place
- Requires
  - good design
  - obvious correction facilities

Form-fills

Users must be familiar with

- keyboards
- use of TAB or mouse to move cursor
- error correction methods
- field label meanings
- permissible field contents
- use of ENTER/RETURN key

Form-fill guidelines

- Meaningful title
  - avoid computer terminology
- Comprehensible instructions
  - describe tasks in familiar terminology
- Logical grouping and sequencing of field
  - adjacent related field
  - sequence reflect common patterns

Form-fill guidelines

- Visually appealing layout of the form
  - uniform distribution of fields
  - Alignment of field labels
  - match the paper form
- Familiar field labels
  - common terms
- Consistent terminology and abbreviations
  - prepare a list of terms and acceptable abbreviations

Form-fill guidelines

- Visible space and boundaries for data-entry fields
  - # of characters in the fields
- Convenient cursor movement
  - TAB or arrow key
- Error correction for individual characters and entire fields

Form-fill guidelines

- Error message for unacceptable values
  - indicate permissible values
- Optional fields clearly marked
- Explanatory messages for fields
  - status bar
- Completion signal

Coded fields

- Telephone number:
  Telephone: _ - _ _ _ _ - _ _ _ _ or
  Telephone: (_ _) _ _ _ - _ _ _ _
- Personal ID:
  _ - _ _ _ _ - _ _ _ _ _ - _ _ - _ or
  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ or
  □-□□□□-□□□□□-□□-□ or
  □ □□□□ □□□□□ □□ □

Coded fields

- Times:
  _ _ : _ _ : _ _ (HH:MM:SS) (AM/PM)
- Dates:
  Date: _ _ / _ _ / _ _ (DD/MM/YY)
  Date: _ _ / _ _ / _ _ _ _ (DD/MM/YYYY)
- Currency
  $ _ _ _ _.00
  ฿ _ _ _ _.00

Direct Manipulation

WIMP Interface

Windows
Icons
Menus
Pointers

... or windows, icons, mice, and pull-down menus!

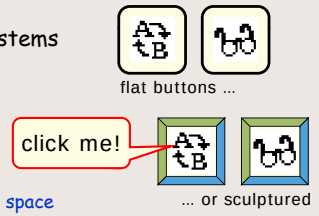
- default style for majority of interactive computer systems, especially PCs and desktop machines

Point and click interfaces

- used in ..
  - multimedia
  - web browsers
  - hypertext
- just click something!
  - icons, text links or location on map
- minimal typing

Three dimensional interfaces

- virtual reality
- 'ordinary' window systems
  - highlighting
  - visual affordance
  - indiscriminate use just confusing!
- 3D workspaces
  - use for extra virtual space
  - light and occlusion give depth
  - distance effects



elements of the wimp interface

windows, icons, menus, pointers

+++

buttons, toolbars, palettes, dialog boxes

Windows

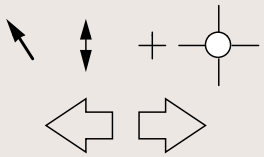
- Areas of the screen that behave as if they were independent
  - can contain text or graphics
  - can be moved or resized
  - can overlap and obscure each other, or can be laid out next to one another (tiled)
- scrollbars
  - allow the user to move the contents of the window up and down or from side to side
- title bars
  - describe the name of the window

Icons

- small picture or image
- represents some object in the interface
  - often a window or action
- windows can be closed down (iconized)
  - small representation fi many accessible windows
- icons can be many and various
  - highly stylized
  - realistic representations.

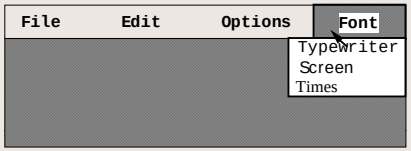
Pointers

- important component
  - WIMP style relies on pointing and selecting things
- uses mouse, touchpad, joystick, trackball, cursor keys or keyboard shortcuts
- wide variety of graphical images



Menus

- Choice of operations or services offered on the screen
- Required option selected with pointer



problem – take a lot of screen space
solution – pop-up: menu appears when needed

Kinds of Menus

- Menu Bar at top of screen (normally), menu drags down
  - pull-down menu - mouse hold and drag down menu
  - drop-down menu - mouse click reveals menu
  - fall-down menus - mouse just moves over bar!
- Contextual menu appears where you are
  - pop-up menus - actions for selected object
  - pie menus - arranged in a circle
    - easier to select item (larger target area)
    - quicker (same distance to any option)
      ... but not widely used!

Menus extras

- Cascading menus
  - hierarchical menu structure
  - menu selection opens new menu
  - and so in ad infinitum
- Keyboard accelerators
  - key combinations - same effect as menu item
  - two kinds
    - active when menu open – usually first letter
    - active when menu closed – usually Ctrl + letter

    usually different !!!

Menus design issues

- which kind to use
- what to include in menus at all
- words to use (action or description)
- how to group items
- choice of keyboard accelerators

Buttons

- individual and isolated regions within a display that can be selected to invoke an action

- Special kinds
  - radio buttons
    - set of mutually exclusive choices
  - check boxes
    - set of non-exclusive choices

Toolbars

- long lines of icons …
  … but what do they do?
- fast access to common actions
- often customizable:
  - choose *which* toolbars to see
  - choose *what* options are on it

Palettes and tear-off menus

- Problem
  menu not there when you want it
- Solution
  palettes – little windows of actions
  - shown/hidden via menu option
    e.g. available shapes in drawing package

  tear-off and pin-up menus
  - menu 'tears off' to become palette

Dialogue boxes

- information windows that pop up to inform of an important event or request information.

e.g: when saving a file, a dialogue box is displayed to allow the user to specify the filename and location. Once the file is saved, the box disappears.

interactivity

easy to focus on look
what about feel?

Speech-driven interfaces

- rapidly improving …
  … but still inaccurate
- how to have robust dialogue?
  … interaction of course!

e.g. airline reservation:
reliable "yes" and "no"
+ system reflects back its understanding
"you want a ticket from New York to Boston?"

Look and … feel

- WIMP systems have the same elements:
  windows, icons., menus, pointers, buttons, etc.
- but different window systems
  … *behave* differently

  e.g. MacOS vs. Windows menus

appearance + behaviour = look and feel

Initiative

- who has the initiative?
  - old question–answer – computer
  - WIMP interface – user
- WIMP exceptions …
  - *pre-emptive* parts of the interface
- modal dialog boxes
  - come and won't go away!
  - good for errors, essential steps
  - but use with care

Physical design

- many constraints:
  - ergonomic – minimum button size
  - physical – high-voltage switches are big
  - legal and safety – high cooker controls
  - context and environment – easy to clean
  - aesthetic – must look good
  - economic – … and not cost too much!

Design trade-offs

constraints are contradictory … need trade-offs

within categories:

e.g. safety – cooker controls

front panel – safer for adult

rear panel – safer for child

between categories

e.g. ergonomics vs. physical – MiniDisc remote

ergonomics – controls need to be bigger

physical – no room!

solution – multifunction controls & reduced functionality

inverse actions

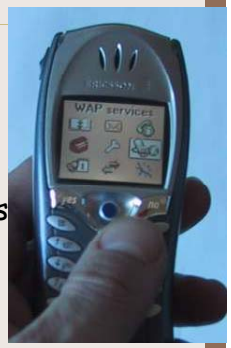
- yes/no buttons
  - well sort of
- 'joystick'
- also left side control

spring back controls

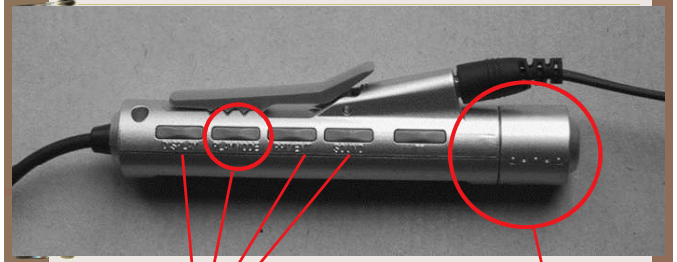
- one-shot buttons
- joystick
- some sliders

good – large selection sets

bad – hidden state



a minidisk controller



series of spring-back controls
each cycle through some options
- natural inverse back/forward

twist for track movement
pull and twist for volume
- spring back
- natural inverse for twist

physical layout

controls:

logical relationship

~ spatial grouping

compliant interaction

state evident in mechanical buttons

rotary knobs reveal internal state and can be controlled by both user and machine

Managing value

people use something

ONLY IF

it has perceived value

AND

value exceeds cost

BUT NOTE

- exceptions (e.g. habit)
- value NOT necessarily personal gain or money

Weighing up value

value

- helps me get my work done
- fun
- good for others

cost

- download time
- money £, $, €, ฿
- learning effort

General lesson …

if you want someone to do something …

- make it easy for them!
- understand their values