

# Interaction Design

## Menu

Prof. Dr. Matthias Rauterberg  
Faculty Industrial Design  
Technical University of Eindhoven  
g.w.m.rauterberg@tue.nl

12-DEC-2005

## Key references/literature:

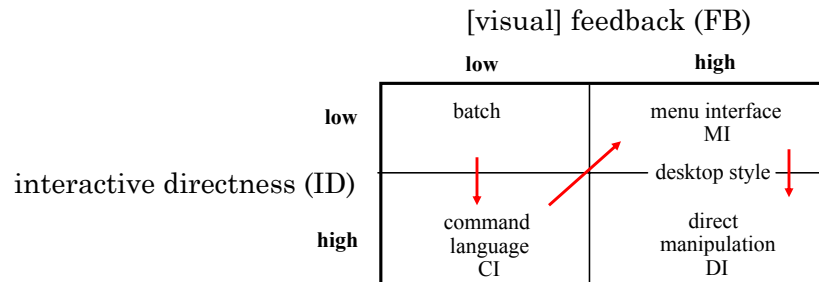
D.J. Mayhew (1992) Principles and guidelines in software user interface design. Prentice Hall.

chapter 4: dialog styles-menus.

ISO/FDIS 9241 (1997) Ergonomic requirements for office work with visual display terminals (VDTs).

Part 14: [menu dialogues](#).

## Two dimensions for interaction

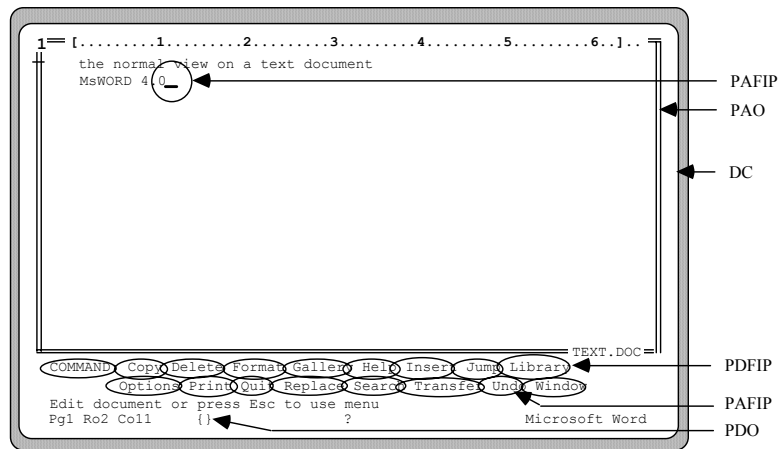


Rauterberg, M. (1998). [About a method to measure the ergonomic quality of user interfaces in a task independent way](#). In: Technical proceedings of the workshop on emerging technologies in human engineering testing and evaluation (T.P. Enderwick & J. Geddie, eds.; pp. 128-144), unclassified Report No. AC/243(Panel 8)TP/17, NATO, Brussels.

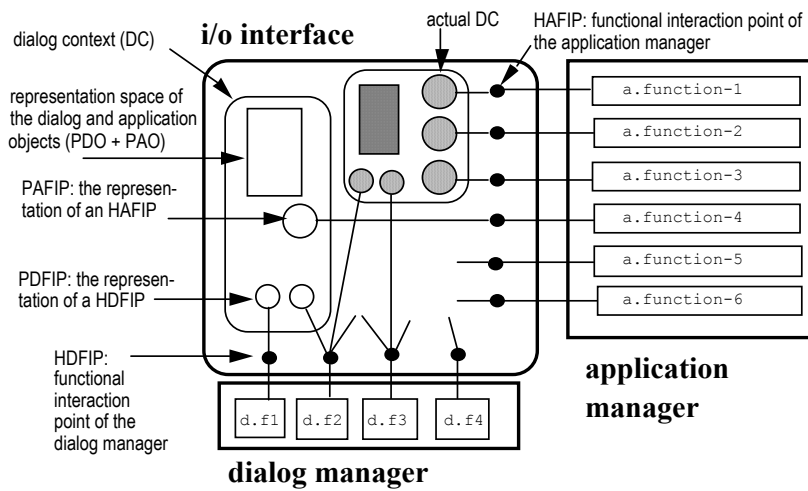
## Reduce Short-Term Memory Load

- People have short-term memory limitations
  - People remember only about seven chunks of information at a time.
  - Interface designer cannot assume the user will remember anything from form to form, or dialog box to dialog box.
  - If user has to stop and ask “Now what was the filename? The customer ID?” then the design is placing a burden on the user’s memory.

# Menu Interface Structure



# Menu Interaction Style



# Menu based example

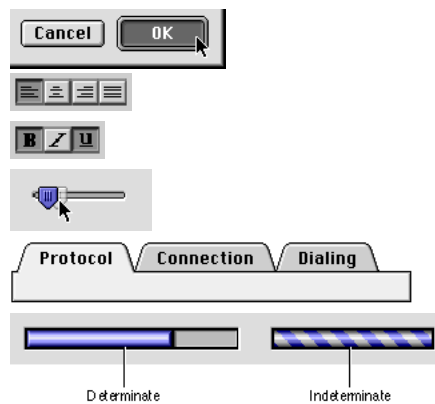


© M. Rauterberg, TU/e

7/68

## Interface Elements (1)

- Other:
  - buttons
  - radio buttons
  - check boxes
  - sliders
  - toolbars
  - tabs
  - progress indicators
  - palettes
  - dialog boxes



© M. Rauterberg, TU/e

8/68

## Interface Elements (2)

- **Menus:**
  - *types:*
    - pull-down, pop-up, fall-down  
(like pull-down, but the bar doesn't have to be explicitly selected),
    - tear-off,
    - pin-up  
(stay on the screen until explicitly requested to go away),
    - pie  
(menu options arranged in a circle).
  - cascading  
(one menu selection opens another menu adjacent to it, and so on)
  - grouping
  - hierarchy
  - keyboard accelerators
  - consistency

- **Example:**

Payment Details
Payment methods:
1. Cash
2. Check
3. Credit card
4. Invoice
9. Cancel transaction
Choose one: _

## Menu Display

- **Permanent display**
  - Frequent use
  - Critical options
  - Screen availability
- **On-demand display**
  - Infrequent use
  - Expert users

# Menus

- Advantages
  - recognize options instead of recalling
  - Structures learning (logical grouping and meaningful naming)
- Disadvantages
  - Dangers of too many menus
  - Can consume lots screen space
    - Consider country selection

## Advantage of Menu Selection

- Affordances
  - Allow users to work by recognition instead of recall
  - Structured decision-making
  - Decomposes a complex interaction into a series of smaller steps
  - Simplified the tasks
    - reduces error rates
    - reduces keystrokes
    - reduces training

## Disadvantages of Menu Selection

- The designer must consider all options beforehand.
  - The user need to follow the designer's logic of organization
  - Sometimes less efficient
    - especially with deep structures
    - Slow for frequent users
  - Menu items may consume too much screen space

## Menu-based systems

- usually hierarchical
  - each menu option opens up another menu
- menus used in many types of system
  - with GUIs
  - with form fill-in
  - with most other software systems
- design of menu structure is crucial

## Menus continued

- Sometimes several versions of the menus are needed based on type of user
  - E.g. mail-order clerks do not need to know many of the options available since they only process new orders
- Menus should also include options not in the event list
  - E.g. options related to controls
    - Backup and recovery of databases
    - User account maintenance
- User preferences are usually provided to allow the user to tailor the interface
- Menus should always include help facilities
- See next slide for the menu hierarchy for customer support system

## Trade-off in menu design

- use of screen space
  - fixed
  - pop-up/drop-down
  - menu mode
- access to menu
  - GUI – mouse point/click
  - non-GUI – F keys, modes (Word Perfect)
  - joy-stick, cursor keys

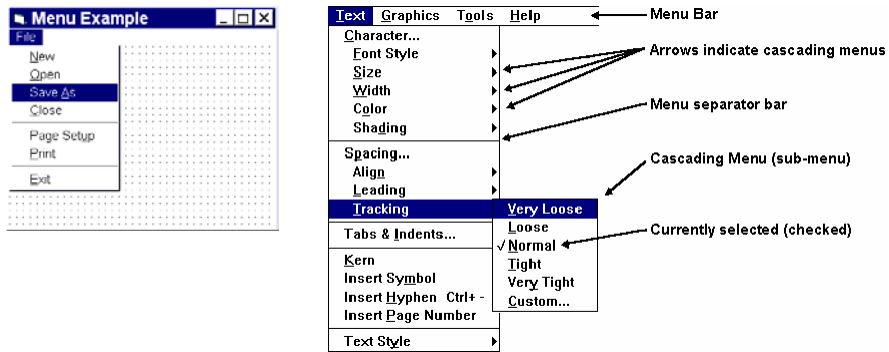


# Pull Down Menus

**Pull down menus are used extensively in Windows applications. They have a number of advantages:**

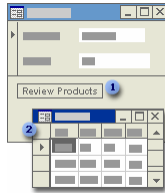
- The menu names are visible textual items on the menu bar, so the user can see the main functions of the software at all times.
- Menu titles, or menu items (such as Open Project on the File menu) should be unambiguous if they are well chosen, whereas, it can be difficult to design satisfactory icons for abstract functions is the standard icon for open.
- The menu bar takes up very little space on the screen.
- The menu options are generally available by means of access keys (holding down the Ctrl key then typing underlined letters of the menu title and menu item required, so Ctrl+O for 'Open Project' on the 'File' menu).
- Functions are grouped together to form a set of menu items that are related, this helps users to know where to look for facilities.

## Pull Down Menus: examples



# Pop-up Menus

A pop-up menu is one that is displayed on top of a form or at the current cursor location, and is not attached to the menu bar.



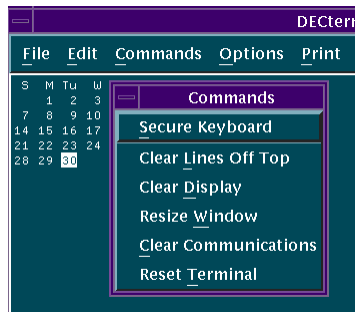
You can create a pop-up form to display information to a user or to prompt a user for data. A pop-up form stays on top of other open forms, even when another form is active. A pop-up form can be modeless or modal. A modal pop-up form is called a custom dialog box.

When a pop-up form is modeless, you can access other objects and menu commands while the form is open. For example, on a Suppliers form, you could add a command button that displays a Product List pop-up form. The pop-up form displays the products available from the current supplier in the Suppliers form.

## Tear-Off Menu (Menu Type) Tear-Off Choice (Choice Type)

A tear-off menu is a menu that has been torn off using the tear-off choice within it.

A tear-off choice is a kind of action choice that, when activated, tears off the control containing the choice.



### Recommended

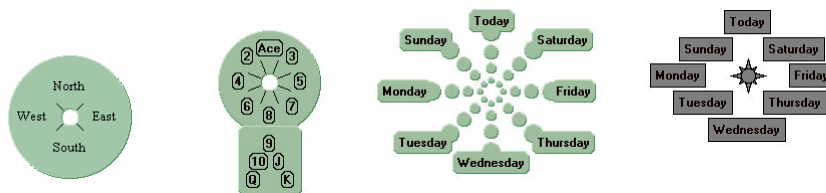
Allow the user to tear off a menu when the user is likely to want to make a number of choices from a pull-down menu.

By default, allow a user to tear off a menu when the user will frequently access the menu.

Allow the user to configure any pop-up or cascaded menu, other than an option menu, to be torn off.

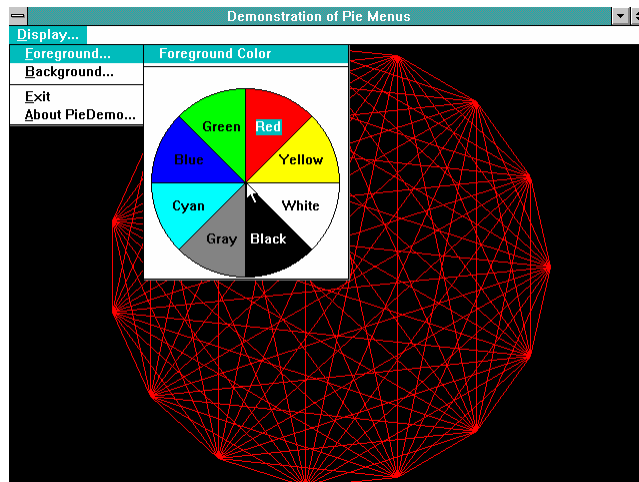
# What Are Pie Menus?

Pie Menus are a naturally efficient user interface technique: directional selection of pie slice shaped targets. The cursor starts out in the center of the pie, so all targets are large, nearby, and in different directions. Fitts' Law explains the advantages of pie menus, relating their **fast selection speed** and **low error rate** to their large target size and small distance. Pie menus are easy for novice users, who just follow the directions, and efficient for experienced users, who can quickly "mouse ahead" once they know the way.



[taken from <http://www.piemenu.com/>]

## Pie menu: example



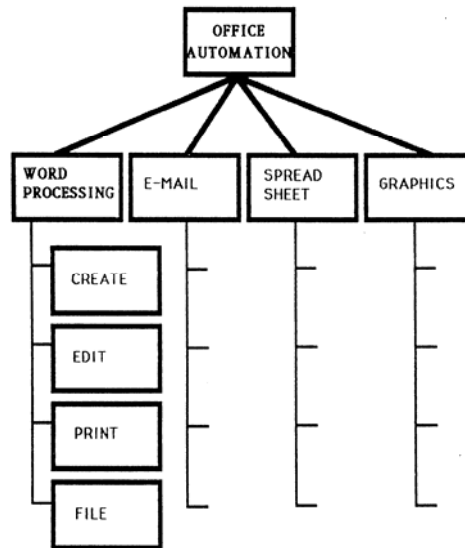
# Command buttons v Menus

- Command buttons
  - Obvious, clear features on form
  - Cover the normal operations
- Menus
  - Less clear - often group functionality
  - Cover less common operations
  - May duplicate Command buttons

## Menu structure (1)

- single menu
  - binary menu
  - multiple-item menu (better: radio button)
  - extended menu
  - pull-down/pop-up menu
  - multiple-selection menu (better: checkboxes)
- linear sequence: same sequence regardless of user's choice
- tree-structured: depth and breadth trade off
- a-cyclic & cyclic menu network
- menu maps: to reduce sense of disorientation & lost

## Menu structure: example



© M. Rauterberg, TU/e

25/68

## Menu structure (2)

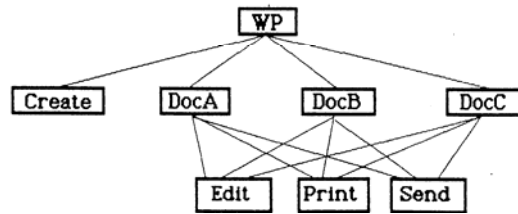
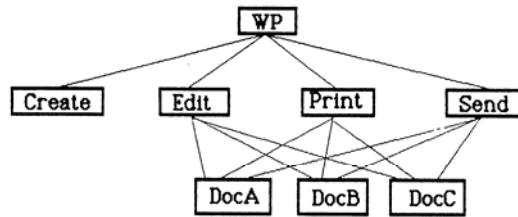
- From the standpoint of the user, the overall system structure is reflected in the menus available
- Each menu contains a hierarchy of options, and they are arranged by sub-system or by actions on objects
  - Example includes:
    - Order entry subsystem
    - Order fulfillment subsystem
    - Customer maintenance subsystem
    - Catalog maintenance subsystem
    - Reporting subsystem
- Alternatively, menus could be arranged based not on subsystems but on objects
  - Customers
  - Orders
  - inventory, and shipments

© M. Rauterberg, TU/e

26/68

## Menu structure (3)

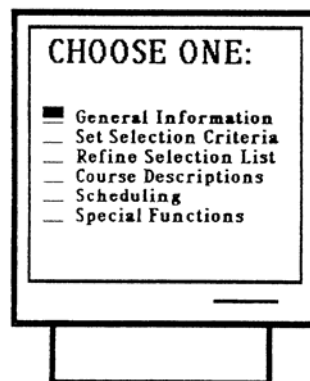
- Match menu structure to task structure



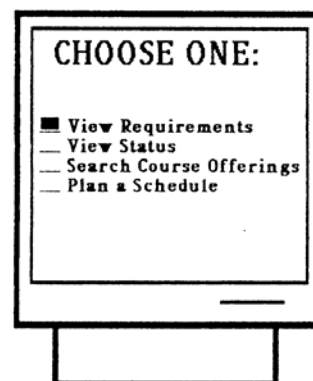
## Menu structure (4)

Create logical, distinctive, and mutually exclusive semantic categories with clear meanings

POOR:



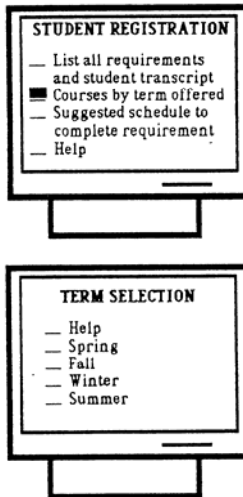
IMPROVED:



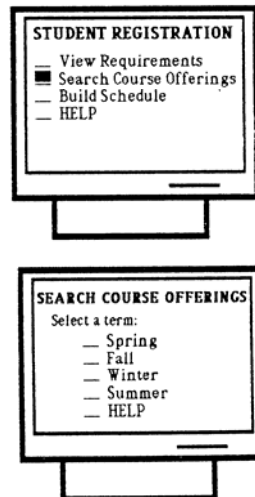
## Menu structure (5)

Menu items should be brief, consistent in grammatical style and placement, and matched with corresponding menu titles.

POOR:



IMPROVED:

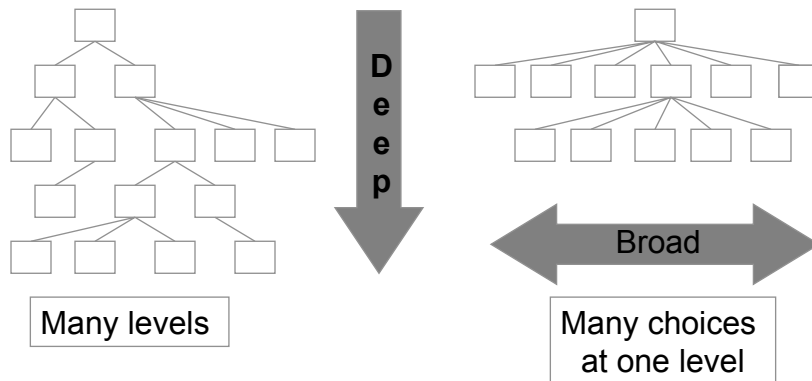


## Menu structure (6)

- Match menu structure to user tasks
  - Not system data or internal structure
- Provide a “main menu”
  - Where in Windows?
- Allow for customization
  - Not all users work the same way

# Menu structure (7)

Depth vs. Breadth

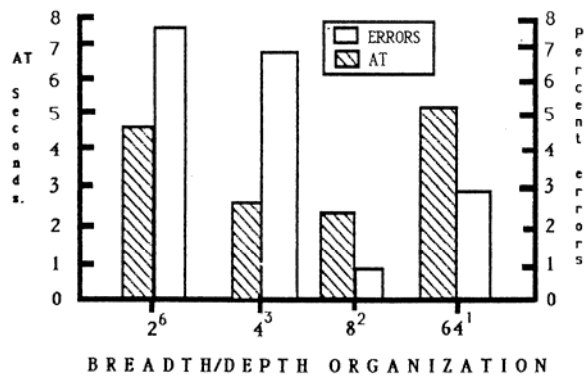


# Menu structure (8)

Four depth/breadth menu structure organisations, all with 64 target items:  
 $[2^6]$  6 levels, 2 item  
 $[4^3]$  3 levels, 4 items  
 $[8^2]$  2 levels, 8 items  
 $[64^1]$  1 level, 64 items

task: access target item that fell into meaningful semantic categories

dependent variable:  
 [AT] access time  
 [PE] percentage errors



[D.P. Miller (1981) [The depth/breadth trade off in hierarchical computer menus](#). Proceedings of the Human Factors Society 25th Annual Meeting, 2, pp. 296-300]



## Menu structure (9)

- Breadth yields:
  - Fewer steps, shorter access times
  - Fewer paths to get lost in
  - Easier learning, since relationships are visible
- Disadvantages
  - More crowded
  - May confuse similar choices

## Menu structure (10)

- Depth yields:
  - Less crowding on menus
  - Fewer choices to scan
  - Easier to hide unavailable choices
  - Similar choices unlikely to be presented together
- Disadvantages
  - More steps, clicks, choices
  - Can't see relationships between choices
  - Can get lost
  - Higher error rates

# Depth vs. Breadth Design Guidelines

- Fewer levels is usually better
- Breadth within a level
  - 4 - 8 choices without grouping
  - 9 or more with grouping
  - Larger numbers in special cases
    - Expert user; simple choices; logical grouping
- Decreasing direction menus are usually better
  - Fewer choices at deeper levels

## Menu ordering (1)

Order menu items according to functional groups, frequency of use, order of use and/or alphabetic order.

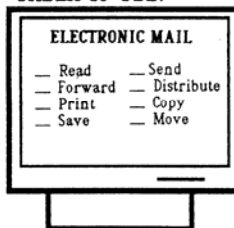
### FUNCTIONAL:



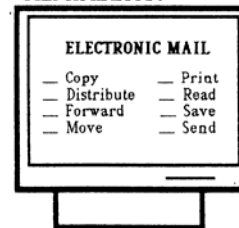
### FREQUENCY OF USE:



### ORDER OF USE:



### ALPHABETIC:



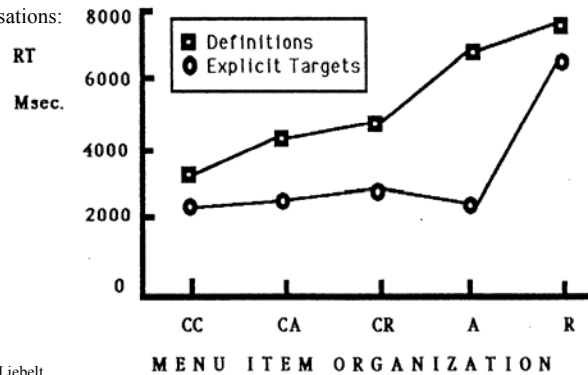
# Menu ordering (2)

1 level menu, 4 columns of 16 items each

five menu structure organisations:  
 [CC] category-category  
 [CA] category-alphabetic  
 [CR] category-random  
 [A] alphabetic  
 [R] random

two target types:  
 [] target definition  
 () explicit target

dependent variable:  
 [RT] reaction time



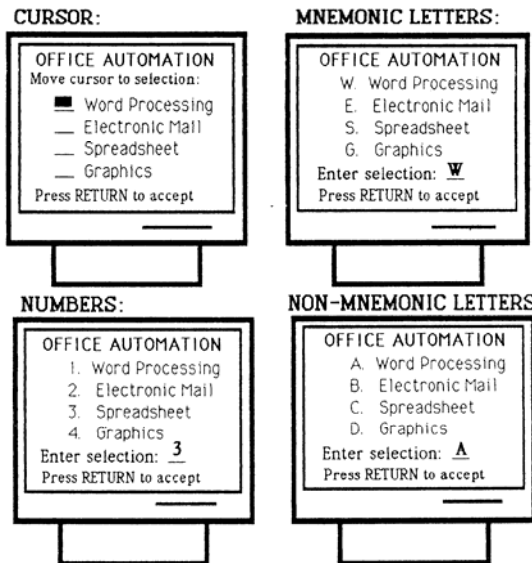
[J.E. McDonald, J.D. Stone & L.S. Liebelt (1983) [Searching for items in menus: the effects of organization and type of target](#). Proceedings of the Human Factors Society 27th Annual Meeting, 2, pp. 834-837]

# Menu selection (1)

Cursor selection is preferred for shorter menus;

mnemonically lettered selections for longer menus;

provide defaults.



## Menu selection (2)

1 level menu, 8 items

4 selection codes:

[CL] compatible letters

[IL] incompatible letters

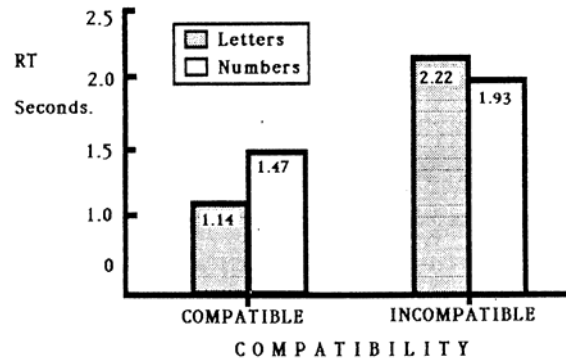
[CN] compatible numbers

[IN] incompatible numbers

dependent variable:

[RT] reaction time

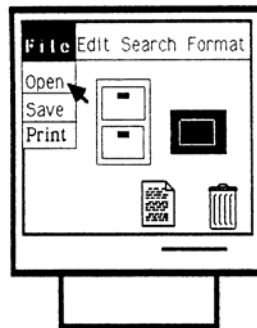
[G. Perlman (1984) [Making the right choices with menus](#). Proceedings INTERACT'84, IFIP, pp. 291-297]



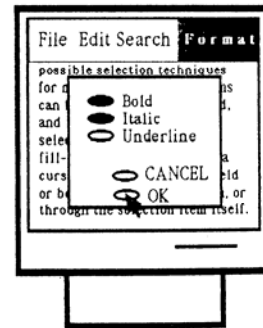
## Menu selection (3)

Distinguish between 'choose one' and 'choose many' menus.

CHOOSE ONE:



CHOOSE MANY:

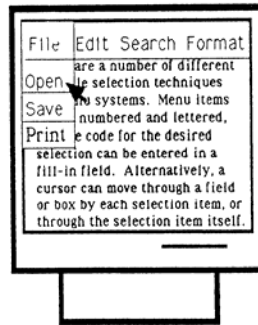


# Menu invocation

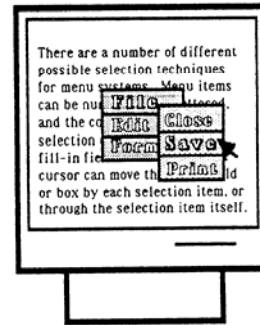
Reserve 'pop-up' or 'user-invoked' menus for high frequency users, and situations where screen real estate is scarce.

Permanent menus are preferred.

PERMANENT:



USER INVOKED:



# Menu navigation (1)

Establish conventions for menu design and apply them consistently on all menu screens.

POOR:



IMPROVED:



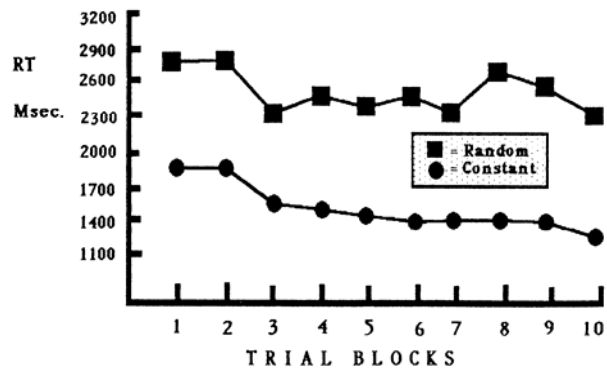
## Menu navigation (2)

5 screen parts:  
title page, number, entries,  
topic heading, instructions

2 positional conditions:  
[R] random  
[C] constant

dependent variable:  
[RT] reaction time

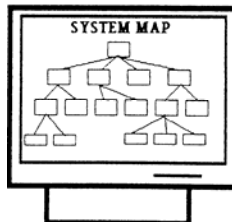
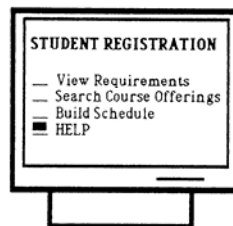
[R.C. Teitelbaum & R.E. Granada  
(1983) [The effects of positional  
constancy on searching menus for  
instructions](#). Proceedings CHI'83,  
ACM, pp. 150-153]



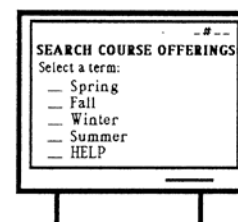
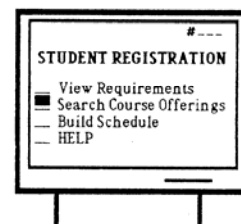
## Menu navigation (3)

Consider the use of  
'menu maps' and 'place  
markers' as navigational  
aids in complex menu  
structures.

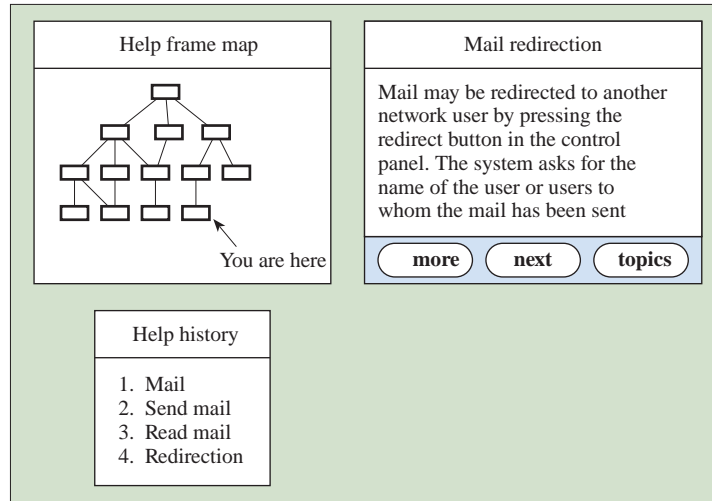
### MENU MAPS:



### PLACE MARKERS:



# Menu navigation: help system

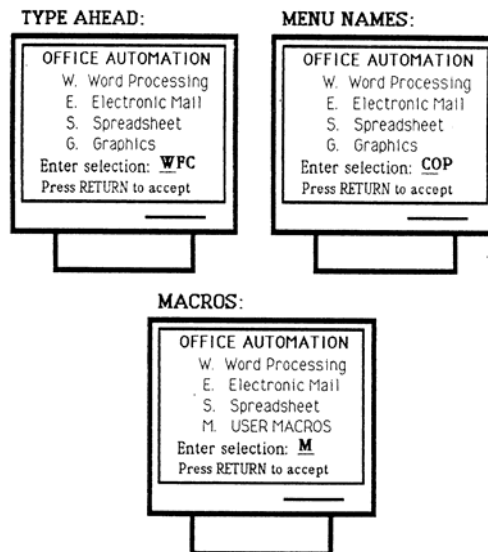


## Menu navigation (4)

Consider the use of direct access through

- \* type-ahead,
- \* menu screen names,
- \* user-created macros

to facilitate navigation for expert users.



# Menu: when to use (1)

## Advantages

- Self-explanatory
  - reduces need for manuals
  - requires little or no training
  - makes both semantics and syntax explicit
- requires little memory
  - recognition versus recall
- few keystrokes
  - less opportunity for user input error
- easy error handling
  - only limited valid inputs at any point
- enhancements are visible

## Disadvantages

- Inefficient for experts and high frequency users
- inflexible
  - system-controlled
  - forced-choice
- impractical for numerous choices
- take up screen 'real estate'

# Menu: when to use (2)

## • knowledge and experiences of the user

- low typing skills
- little system experience
- low task experience
- low application experience
- frequent use of other systems
- low computer literacy

## • job and task characteristics

- low frequency of use
- little or no training
- discretionary use
- high turnover rate
- low task importance
- high task structure



# Menu design guidelines [Shneiderman 1992]

- use task semantics to organize menu
- prefer broad and shallow to narrow and deep
- show position by graphics, numbers, or titles
- use item names as titles for trees
- use meaningful groupings of items
- use meaningful sequencing of items
- make item brief, begin with keyword
- use consistent grammar, layout, terminology
- allow typeahead, jumpahead, or other shortcuts
- allow jumps to previous and main menus
- consider online help, novel selection mechanism, response time, display rate, and screen size

## Menu selection

- Principle: the computer displays a list of items from which the user selects
- Types: numbered, lettered, mnemonic lettered, unlabelled
- Advantages:
  - shortens training
  - reduces keystrokes
  - structures decision making
  - permits use of dialog management tools
  - easy to support error handling
- Disadvantages:
  - Danger of many menus
  - May slow frequent users
  - Requires screen space
  - Requires rapid display rate

# Menu selection, guidelines

- Use task semantics to organize menus (single, linear sequence, tree structure, a-cyclic and cyclic networks)
- Prefer broad/shallow trees
- Show position by graphics, numbers or titles
- Items become titles for trees - i.e. for underlying items, sub-menus
- Meaningful groupings of items
- Brief items, begin with keyword
- Consistent grammar, layout, terminology
- Type-ahead, jump-ahead, or other short-cuts
- Jumps to previous and main menu
- Consider: on-line help, novel selection mechanism, response time, display rate, screen size

## Menu Selection

- Emphasize on
  - Structures
    - The designer predefines the structure.
      - Comprehensible and natural to use
    - The computer displays options.
      - display options and accept inputs
  - Selections
    - The user makes choices.
      - Task-oriented
      - System matches the user's choice with system's functions.

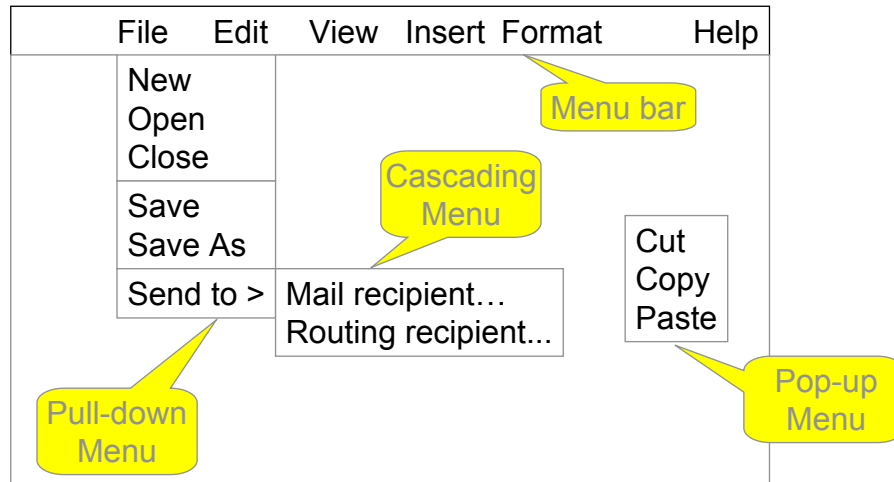
# Menu Selection

- With menu-based systems,
  - users can only select.
  - users can not change the structures.
  - user can navigate through the structure.
  - users can not add any new controls.

# Menus

- Structured access to system's functionality
  - The designer predefines the structure.
    - Comprehensible and natural to use
  - The computer displays options.
    - display options and accept inputs
  - The user makes choices.
    - Task-oriented

## Menus: examples



## Menu-driven Applications

Whiz Co Reporting System
<ol style="list-style-type: none"><li>1. Purchasing Reports</li><li>2. Sales Reports</li><li>3. Purchasing Data Entry</li><li>4. Sales Data Entry</li><li>5. System Administration</li></ol>
Type a selection and press <Enter> __ (1 - 5)

# Menu Ordering

- By natural order
- For small lists by:
  - Sequence of occurrence
  - Frequency of occurrence
  - Importance
- Alphabetic order for:
  - Long lists
  - Short lists with no obvious patten or frequency
- Separate destructive choices
- Maintain consistent ordering

# Menu Choices

- Present text choices vertically
  - Left-justify text
- Provide choice descriptors for complex systems
  - Look ahead
    - Shows next lower menu when cursor passes over choice
  - Micro-help
    - Brief menu description in pop-up or status bar

# Wording Menu Choices

- Clear, common meaning
  - Vocabulary of the user
- Single words or very short phrases
- Grammatical consistency
  - Key word first
  - Parallel construction

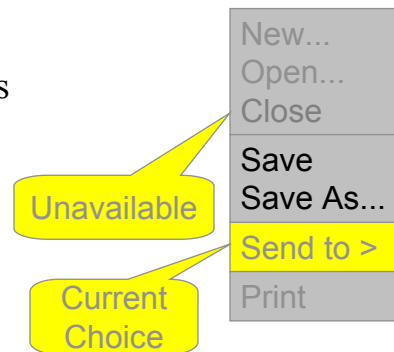
## Which is better?

List all requirements for major  
 Courses offered by term  
 Suggested schedule to  
complete requirement  
 Help

List requirements  
 Search courses offerings  
 Build schedule  
 Help

## Choice Selection (1)

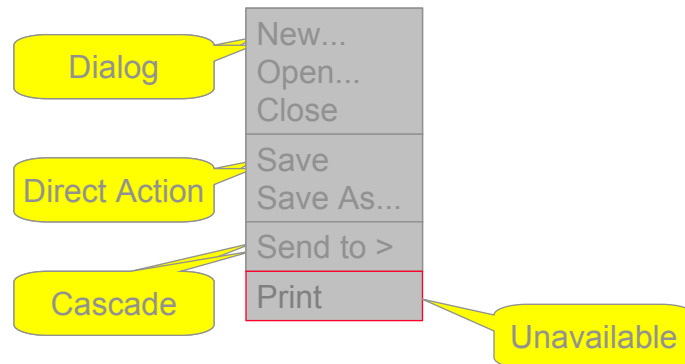
- Provide defaults if sensible
  - Last or most frequent choice
- Highlight current choice
- Indicate unavailable choices
  - “Grayed out”
  - Removed from menu
    - General rule: do NOT remove choices
- Make choices large enough to select



## Choice Selection (2)

- Keyboard selection
  - Up & Down Arrows move cursor up and down
  - Right & Left Arrows move cursor left & right for horizontal menus
- Selection/Execution
  - Provide separate selection and execution actions
  - Highlight selected choice or modify cursor shape
  - Allow alternative techniques for selection & execution if possible

## Additional Indicators



Do NOT use Cascade or Dialog indicators if selection sometimes produces direct action

## Arranging Menu Items

- Use Columns
  - Top-to-bottom reading
  - Left justify descriptions
- For horizontal menus
  - Left-to-right reading
- Intent Indicators
  - Use arrow or triangle to indicate cascading menus
  - Use elipsis (...) to indicate option resulting in a window
  - Use **NO** indicator for items resulting in an action



## Line Separators

- Separate vertical groupings with solid lines
- Separate sub-groupings with dashed lines
- Left-justify lines under the 1st letter of the description
- Right-justify under the last character of the longest description

## Enable Frequent Users to Use Short Cuts

- Users who work with one application all the time are willing to invest time to learn short cuts
- They begin to lose patience with long menu sequences when they know exactly what they want to do
- Short-cut keys can reduce the number of interactions for a given task
- Designers can provide macro facilities for users to create their own short cuts
- E.g. mail order entry clerks wouldn't want long multiple menus to slow them down, but instead short-cuts would make them more productive

## Keyboard Accelerator (1)

- Provide keyboard accelerator
  - Expert users
  - Motor skill problems
  - Cramped or grimy places
    - Factory floor
    - Portables
- Keyboard equivalents
  - “Access keys”

## Keyboard Accelerator (2)

- May be a function key or combination of keys
  - Function key easier to learn than modifier + key
  - No more than 2 keys together in any case
  - Use “+” sign to indicate multiple keys required
- Don’t use accelerators for pop-ups or cascaded menus

