

usability

Anastasia.Bezerianos@lri.fr

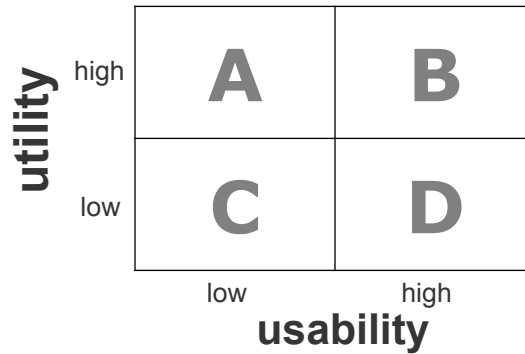
utilisability, usability

« The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use »
(ISO 9241)

A usable system is: easy to learn, efficient, visually pleasing and allows easy error correction

utility

Meet specific needs and support real tasks



is D better than A ?

Usability Criteria: Before starting

Criteria and recommendations

Ten Usability Heuristics, <http://www.useit.com/>, Jakob Nielsen

Advantages

- Helps correct usability problems
- Criteria easy to remember and simple to apply
- Allows quick evaluation of a system

Disadvantages

- Not really a check-list
- Some redundancies
- Have to be careful when applying some criteria
(Can have exceptions or conflicts between criteria)

Guides

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Visibility and feedback

Objective: aid the use and learning of a system

Feed-back and *feed-forward* mechanisms to

- reduce memory load
- prevent errors (more later)
- reassure (e.g.: progression of an operation)

- helps user understand
 - what actions are available
 - what the system is doing
 - how it is interpreting the user's input

... users should always be aware of what is going on

Visibility and feedback

Recommendations: feed-forward

- gray out non-available commands
- make input possibilities clear
- give list of possible inputs instead of typing
- give example of expected input
- give intelligent default values

Recommendations: feed-back

- each user action should be followed by a changed representation in the interface
- inform users of long operations
- indicate currently used modes
- show status of system operations in progress

Visibility and feedback

Give input format, example and default

The image shows two software windows. The left window, titled 'Form1', contains a 'Date:' label followed by a text input field. Below this, there are three small input boxes for 'Month', 'Day', and 'Year', with the values 'May', '22', and '1997' respectively. The right window, titled 'Appointment', has a tabbed interface with 'General', 'Attendees', 'Notes', and 'Planner' tabs. The 'General' tab is active, showing 'Start' and 'End' time fields, a date picker set to 'Wed 5 /14 /97', and an 'All day' checkbox. Below these is a 'Description' field with the text 'Smart Technology Ser'. To the right of the description is a calendar for May 1997, with the 14th highlighted. At the bottom of the calendar is a 'Where:' label and a text input field.

Visibility and feedback

System Response time (time to give feedback)

how users perceive delays

- < 0.1s perceived as “instantaneous”
- 1s user’s flow of thought stays uninterrupted, but delay noticed
- 10s limit for keeping user’s attention focused on the dialog
- > 10s user will want to perform other tasks while waiting

Visibility and feedback

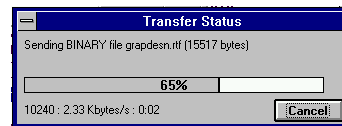
Dealing with long delays

Cursors
for short transactions



Percent done dialogs

- time/work left
- estimated time

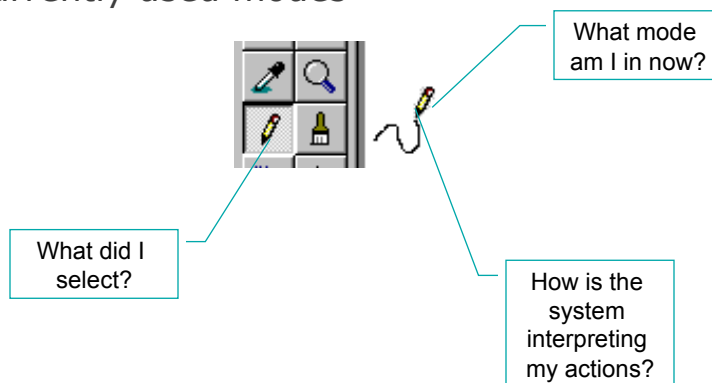


Random
for unknown times



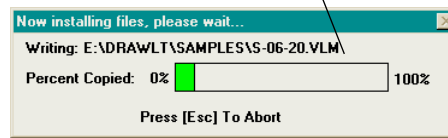
Visibility and feedback

Currently used modes



Visibility and feedback

Multiple files being copied,
but feedback on bar is file by file.



Drawing Board LT

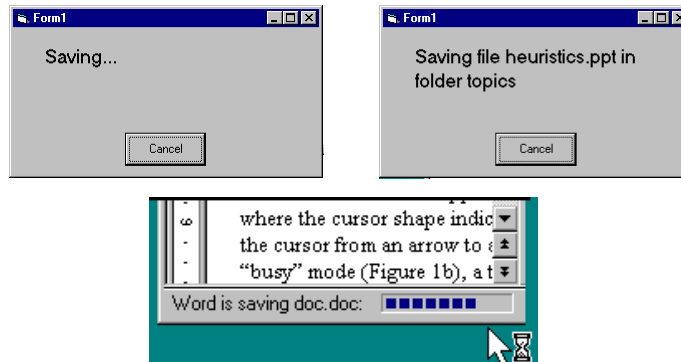
2. Match system & real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.

Follow real-world conventions, making information appear in a natural and logical order.

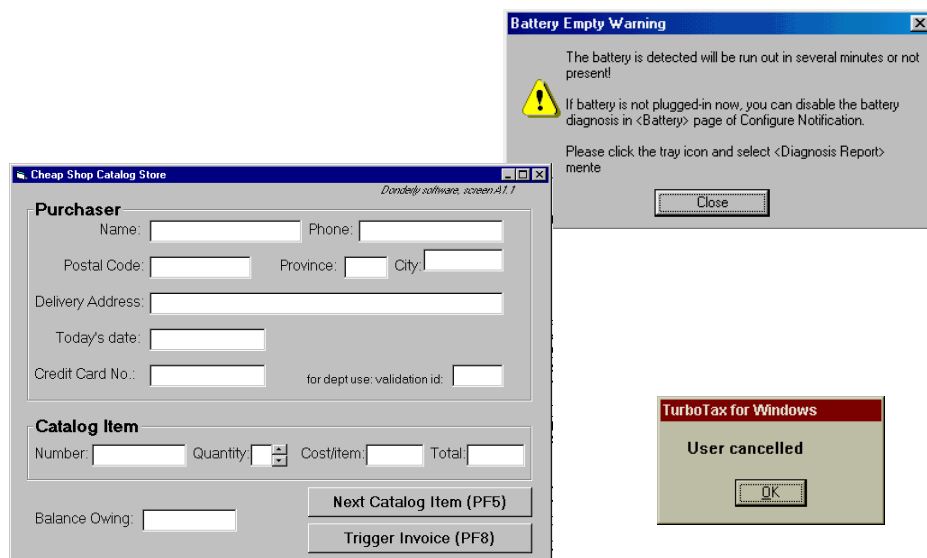
Match between system and real world

Be as specific as possible about operations,
based on user's input



Best within the context of the action

Match between system and real world

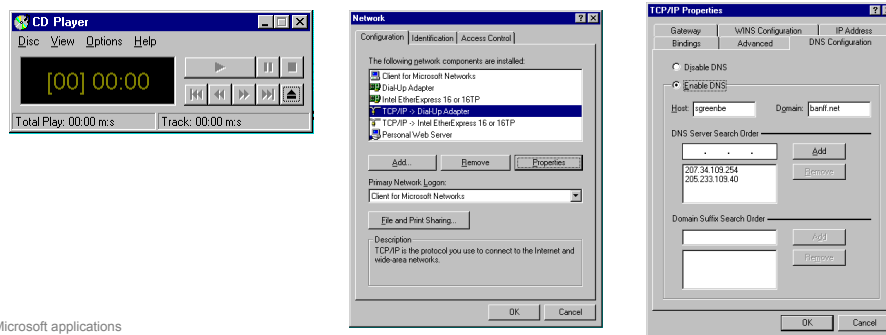


Match between system and real world

Use the user's mental model of other situations

Match the users' task sequence

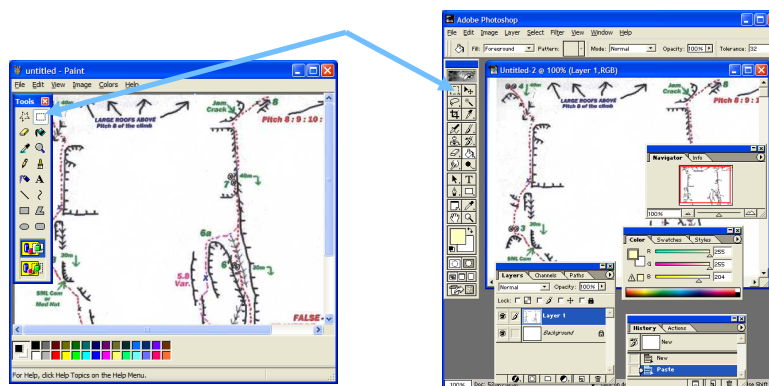
Mapping between interface and task semantics



From Microsoft applications

Match between system and real world

Good use of metaphors and transfers



From Microsoft applications

3. User control and freedom

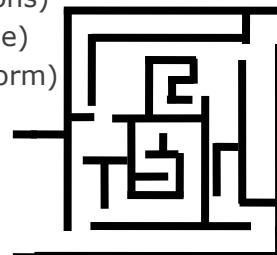
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

User control and freedom

Users don't like to feel trapped by the computer!
should offer an easy way out of as often as possible

Strategies:

- Cancel button (for dialogs waiting for user input)
- Universal Undo and Redo (can get back to previous state)
- Interrupt (especially for lengthy operations)
- Quit (for leaving the program at any time)
- Defaults (for restoring a partially filled form)
- ... consider autosaving



4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing.
Follow platform conventions.

Consistency and standards

Global coherence of interface

internal: inside the application

external: between applications (e.g. icons, shortcuts),
w.r.t. the metaphor of the system (e.g. desktop)

Principle: a system that seems familiar is seen as
easy to use by users

Goal: help learning and use

Risk: block system evolution (rigidity of standards)

Consistency and standards

Recommendations

windows should look similar

e.g.: search box at top right

consistent graphics

e.g. information/controls in same location on all windows

same vocabulary used for commands as other systems

e.g.: open / copy-paste / preferences / ...

syntax of commands coherent across all the interface

e.g.: similar actions have similar effects

Consistency is not only visual consistency

Other examples: syntax, interaction, command result

Consistency and standards

Style guides:

published by system designers

describe the look and feel of a platform

are often too strict: help those who follow them and make life difficult for anyone who wants to deviate ...

Examples :

- OSF Motif
- IBM CUA
- Apple Human Interface Guidelines
- MS Windows



In principle good, but can be hard to follow
Implemented (in part) in interface toolkits

Consistency and standards

Motif style guide v1.1 : MessageDialogs should be used to convey a message to the user. They must not interrupt the user's interaction with the application. They should include a message, and one of the following button arrangements.

OK
OK Help
OK Cancel
OK Cancel Help

Cancel
Cancel Help

Yes No
Yes No Help
Yes No Cancel
Yes No Cancel Help

Retry Cancel
Retry Cancel Help

Consistency and standards

Macintosh Human Interface Guidelines

Figure 5-11 Examples of correct and incorrect window titles

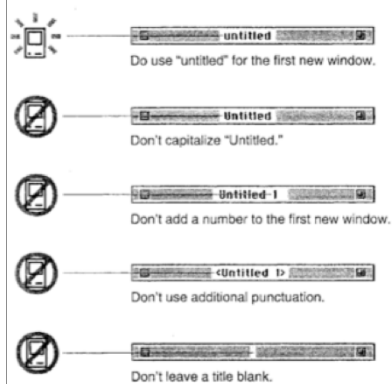


Table 10-1 Pointers

Pointer	Name	Used for
	Arrow	Scroll bar, other controls, size box, title bar, close box, zoom box, menu bar, desktop
	Crosshairs	Drawing, shrinking, or stretching graphic objects
	I-beam	Selecting and inserting text
	Plus sign	Selecting fields in an array
	Wristwatch	Showing that a lengthy operation is in progress

Consistency and standards

These are labels with a raised appearance.

Is it any surprise that people try and click on them?

The form is titled 'Subscriber' and 'Contact'. It contains the following fields and buttons:

- Subscriber** (label)
- Name:** (input field)
- Account #:** (input field)
- Tech. Re** (label)
- Status:** (label)
- Contact** (label)
- Telephone:** (input field)
- E-Mail:** (input field)
- Address:** (input field)
- St** (label)
- Save** (button)
- Cancel** (button)

5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Errors: help users

People will make errors!

Errors we make

Mistakes

conscious deliberations lead to an error instead of correct solution

Slips

unconscious behavior gets misdirected en route to satisfying goal

- e.g. drive to store, end up in the office

shows up frequently in skilled behavior

- usually due to inattention

often arises from similar actions



Errors: prevent them

Designing for slips:

General rules

- prevent slips before they occur
- detect and correct slips when they do occur
- user correction through feedback and undo



Errors: prevent them

Capture error

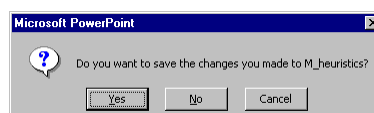
- frequent activities take charge instead of one intended
- occurs when common & rarer actions have same initial steps
 - change clothes for going out and find oneself in pajamas
 - confirm saving of a file when you don't want to replace it

minimize by

- make actions undoable instead of confirmation
- allow reconsideration of action by user
 - e.g. open trash to undelete a file



I can't believe I pressed Yes...



Errors: prevent them

Description error

intended action similar to others that are possible

occurs when right & wrong objects physically near each other

- pour juice into bowl instead of glass
- throw sweaty shirt in toilet instead of laundry basket
- move file to wrong folder with similar name

minimize by

rich feedback

check for reasonable input, etc.

undo

Errors: prevent them

Loss of activation

forget goal while undergoing the sequence of actions

start going to room and forget why you are going there

navigating menus/dialogs & can't remember what you look for

... but continue action to remember (or go back to beginning)!

minimize by

if system knows goal, make it explicit (e.g. title in save mode)

if not, allow person to see path taken (e.g. history)

Errors: prevent them

Mode errors

- do actions in a mode thinking you are in another
- refer to file that's in a different directory
- look for commands / menu options that are not relevant

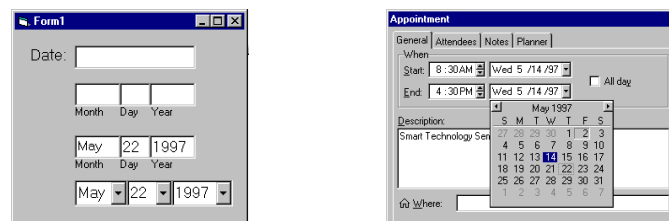
minimize by

- have as few modes as possible (preferably none)
- make modes highly visible

Errors: prevent them

Prevent errors

- try to make errors impossible



Provide reasonable checks on input data

- e.g. if entering order for office supplies
 - 500000 pencils is an unusually large order. Do you really want to order that many?

Errors: help users

Prevent/mitigate continuation of wrongful action:

1. Gag

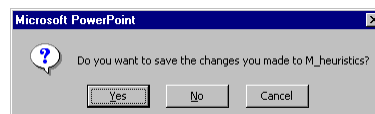
deals with errors by preventing the user from continuing
e.g. cannot get past login screen until correct password entered

2. Warn

warn people that an unusual situation is occurring
... when overused, becomes an irritant

e.g.,

- audible bell
- alert box



Errors: help users

3. Do nothing

illegal action just doesn't do anything

user must infer what happened

e.g. enter letter in numeric-only field (key clicks ignored)

e.g. put a file icon on top of another file icon (returns it to original position)

4. Self-correct

system guesses legal action and does it instead

but leads to a problem of trust

e.g. spelling corrector

Errors: help users

4. Lets talk about it

system initiates dialog with user to come up with solution to the problem

e.g. compile error brings up line in source code

5. Teach me

system asks user what the action was supposed to have meant

action then becomes a legal one

e.g. adding a word in the spelling dictionary

Errors: help users

6. If all else fails provide meaningful error messages

error messages should be in the user's task language

don't make people feel stupid

Try again, bonehead!

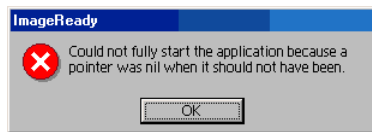
Error 25

Cannot open this document

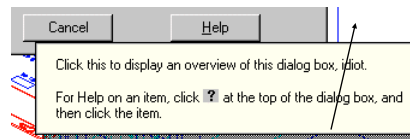
Cannot open "chapter 5" because the application "Microsoft Word" is not on your system

Cannot open "chapter 5" because the application "Microsoft Word" is not on your system. Open it with "OpenOffice" instead?

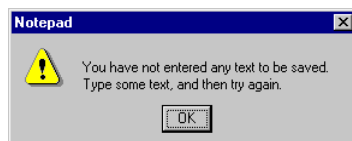
Errors: help users



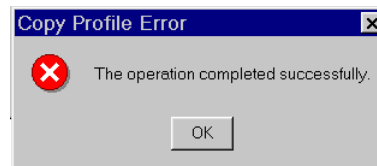
Adobe's ImageReady



AutoCAD Mechanical



Windows Notepad



Microsoft's NT Operating System

6. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible.

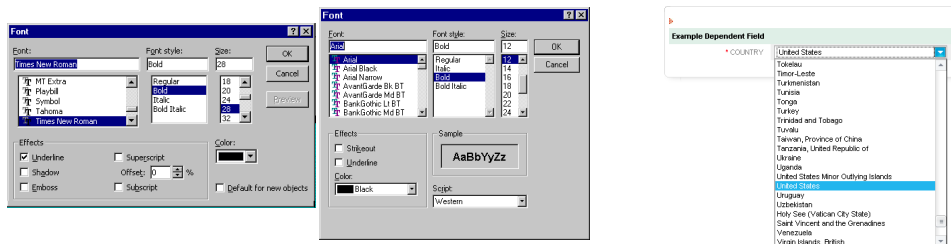
The user should not have to remember information from one part of the dialogue to another.

Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Recognition rather than recall

Computers good at remembering, people not!
Promote recognition over recall

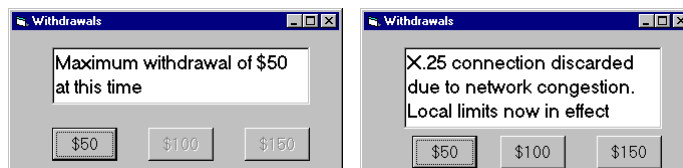
menus, icons vs text commands, field formats
promote visibility of objects (but less is more!)



From Microsoft applications

Recognition rather than recall

Options visible, instructions clear



Use meaningful mnemonics, icons & abbreviations

e.g. File / Save

Ctrl + S

(abbreviation)

Alt FS

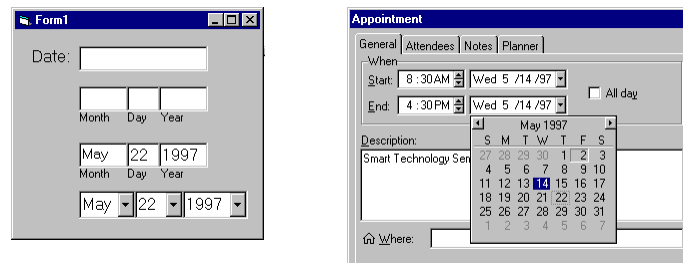
(mnemonic for menu action)



(tooltip icon)

Recognition rather than recall

Give input format, example and default



Reducing memory load

Small number of rules applied universally

generic commands

same command can be applied to many objects

interpreted in context of interface object

e.g. copy, cut, paste, drag 'n' drop, ... for
characters, words, paragraphs, circles, files

contextual menus

7. Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users.
Allow users to tailor frequent actions.

Flexibility and efficiency of use

Capability to adapt to different contexts of use

Recommendations :

- permit command activations from keyboard or mouse
- allow frequently used operations to be activated by every location
- allow users to parameterize their software based on their preferences
- give quick access to frequent commands in menus

Can contradict minimalist design (later)

Flexibility and efficiency of use

Expert users - want to perform frequent operations quickly

Strategies:

keyboard and mouse accelerators

abbreviations

command completion

context menus

function keys

double clicking vs menu selection

type-ahead (entering input before the system is ready for it)

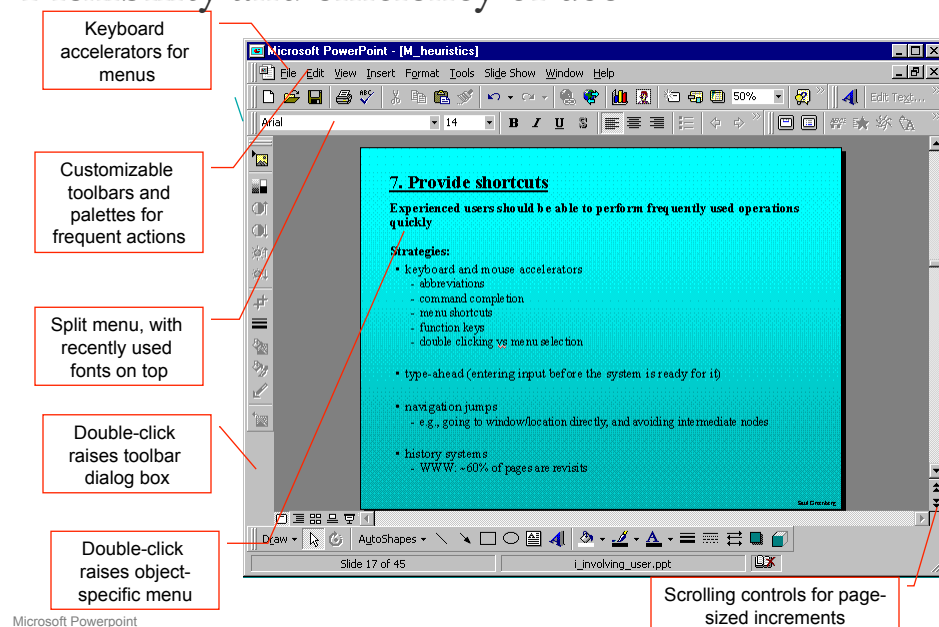
navigation jumps and search

e.g., going to window/location directly, avoiding intermediate nodes

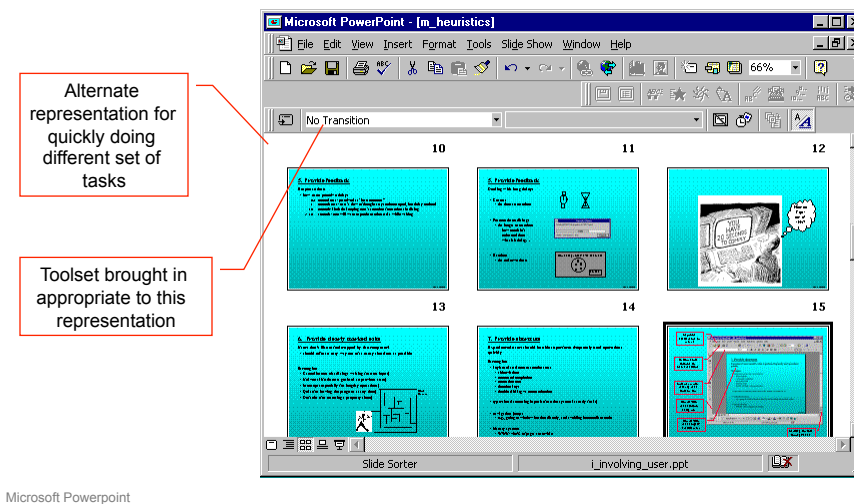
history systems

WWW: ~60% of pages are revisits

Flexibility and efficiency of use



Flexibility and efficiency of use



Flexibility and efficiency of use

Adaptable UI: explicit personalization by user
e.g. dictionary, presentation preferences

Preference configuration is harder in general and complex applications. Don't turn your user into a designer!

Adaptive UI: dynamic personalization without explicit user action
Adaptive interfaces are controversial ...

- can be unpredictable
- can violate coherence

8. Aesthetic and minimalist design

Dialogues (windows) should not contain information which is irrelevant or rarely needed.

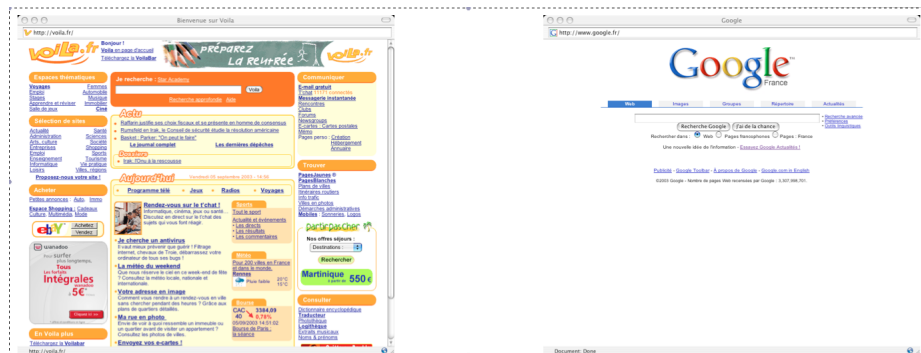
Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Aesthetic and minimal design

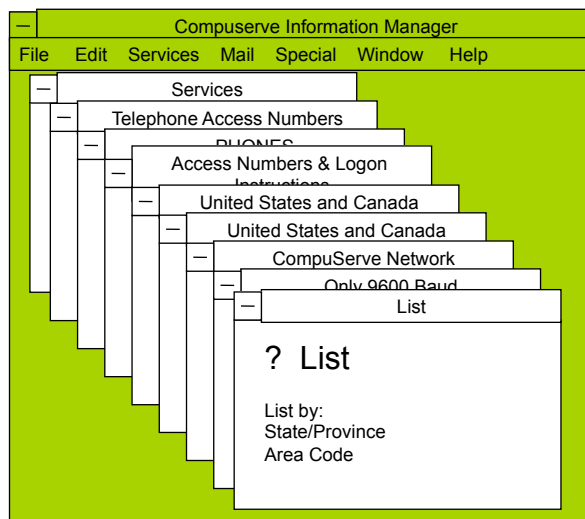
Ways to reduce visual clutter and focus user attention

Recommendations (be concise):

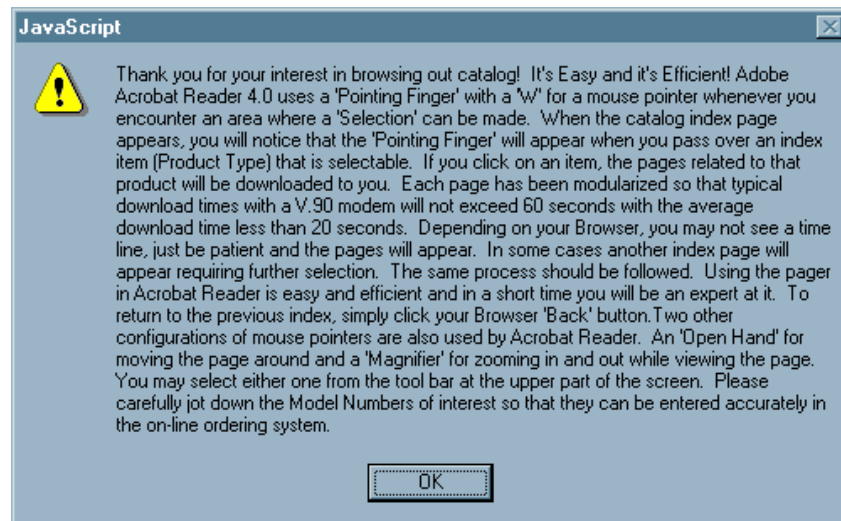
- only display important information (for what the user needs)
- reduce number of actions needed to perform an objective
- minimize input and reading instructions
- avoid too much text
- don't ask for input that you can infer automatically
- avoid users having to remember information
- don't ask users to perform calculations



Minimal design



Aesthetic and minimal design



Reducing memory load

Externalize cognition when possible

- externalizing to reduce cognitive load
- computational offloading
- annotating and cognitive tracing



Minimal design

Present exactly the information the user needs

- less is more: less to learn, to get wrong, to distract...

- information should appear in natural order

 - related information is graphically clustered

 - order of accessing information matches user's expectations

- remove or hide irrelevant or rarely needed information

 - competes with important information on screen

- remove modes or make them clearly visible

- use windows frugally

 - don't add unneeded navigation and window management

10. Provide help & documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.

Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Provide help and documentation

Help is not a replacement for bad design!

Simple systems:

walk up and use; minimal instructions

Most other systems

feature rich

simple things should be simple

learning path for advanced features



Provide help and documentation

Many users do not read manuals

prefer to spend their time pursuing their task

Usually used when users are in some kind of panic

online documentation better

good search/lookup tools

online help specific to current context

Sometimes used for quick reference

syntax of actions, possibilities...

list of shortcuts ...

Provide help and documentation: Types

1. Tutorial and/or getting started manuals

short guides that people are likely to read when
first obtaining their systems
encourages exploration & getting to know the system
tries to get across essential conceptual material

on-line “tours”, exercises, and demos
demonstrates basic principles through working examples

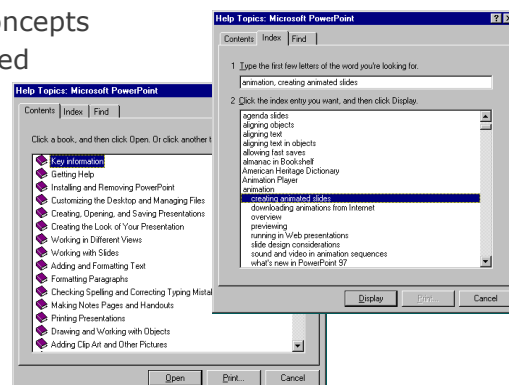
Provide help and documentation: Types

2. Reference manuals

used mostly for detailed lookup by experts
rarely introduces concepts
thematically arranged

on-line hypertext
search / find
table of contents
index
cross-index

Microsoft Help



Provide help and documentation: Types

3. Reminders

short reference cards

expert user who just wants to check facts

novice who wants overview of system's capabilities

keyboard templates & icons

shortcuts/syntactic meanings of keys

recognition vs. recall

tooltips and other context-sensitive help

text over graphical items indicates meaning or purpose

Microsoft Word

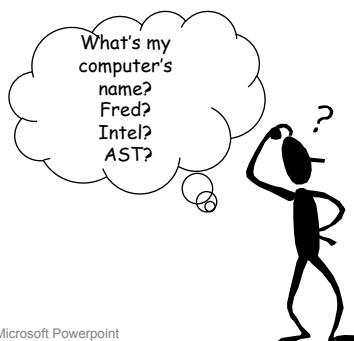


Provide help and documentation: Types

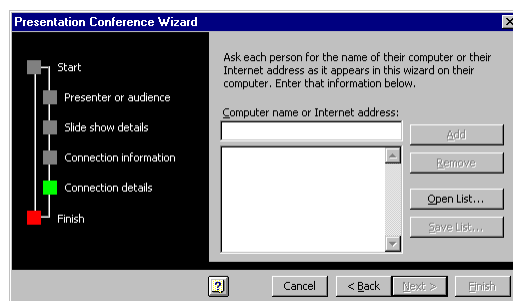
4. Wizards

walks user through typical tasks

... *but* dangerous if user gets stuck



Microsoft Powerpoint



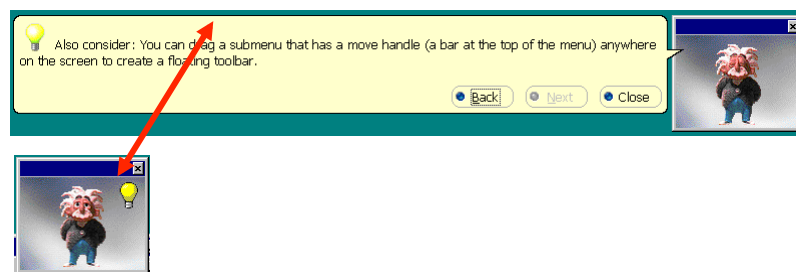
Provide help and documentation: Types

5. Tips

migration path to learning system features

context-specific tips on being more efficient

must be "smart", otherwise boring and tedious



Microsoft Word