Design Principles and Usability Heuristics

You can avoid common design pitfalls by following 9 design principles

You can inspect an interface for usability problems with these principles

Design principles and usability heuristics (I)

Broad "rules of thumb" that describe features of "usable" systems

Design principles

- broad usability statements that guide a developer's design efforts
- derived by evaluating common design problems across many systems

Heuristic evaluation

- same principles used to "evaluate" a system for usability problems
- becoming very popular
  - user involvement not required
  - catches many design flaws
- is an "expert review"

Design principles and usability heuristics (II)

Advantages

- the “minimalist” approach
  - a few general guidelines can correct for the majority of usability problems
  - easily remembered, easily applied with modest effort
- discount usability engineering
  - cheap and fast way to inspect a system
  - can be done by usability experts

Challenges (for lack of a better word)

- principles can’t be treated as a simple checklist
  - Note: “If done wrong, that’s bad” is a common “disadvantage”, but it is worth noting here.
- subtleties involved in their use

Discount Usability Engineering

Cheap/Fast/Easy To Use!

- no special labs or equipment needed
  - might even be able to run it on your own machine in your office
  - can even be used on paper prototypes
- can be on order of 1 day to apply
  - standard usability testing may take weeks
- once understood, can use in many scenarios with little additional learning
- the more careful you are, the better it get

Heuristic Evaluation

Developed by Jakob Nielsen (1990)

- seems inspired by Shneiderman's "Eight Golden Rules"

Helps find usability problems in a UI design

Small set (3-5) of evaluators examine UI

- independently check for compliance with usability principles ("heuristics")
- different evaluators will find different problems
- evaluators only communicate afterwards
  - findings are then aggregated

Can perform on working UI or on sketches

Heuristic Evaluation Process

Evaluators go through UI several times

- inspects various dialogue elements
- compares with list of usability principles
- consider other principles/results that come to mind

Usability principles

- Nielsen's "heuristics"
  - there are several slightly different sets (we will see one) of heuristics
- supplementary list of category-specific heuristics
  - competitive analysis & user testing of existing products

Use violations to redesign/fix problems

Phases of Heuristic Evaluation

1) Pre-evaluation training

- give evaluators needed domain knowledge and information on the scenario

2) Evaluation

- individuals evaluate and then aggregate results

3) Severity rating

- determine how severe each problem is (priority)

4) Debriefing

- discuss the outcome with design team

Evan Golub / Ben Bederson / Saul Greenberg

How to Perform Evaluation

At least two passes for each evaluator

- first to get feel for flow and scope of system
- second to focus on specific elements

If system is walk-up-and-use or evaluators are domain experts, then no assistance needed

- otherwise might supply evaluators with scenarios

Each evaluator produces list of problems

- explain why with reference to heuristic or other info.
- be specific and list each problem separately

Evan Golub / Ben Bederson / Saul Greenberg

Examples

Can’t copy info from one window to another

- violates “Minimize the users’ memory load”
- fix: allow copying

Typography uses mix of upper/lower case formats and fonts

- violates “Consistency and standards”
- slows users down
- probably wouldn’t be found by user testing
- fix: pick a single format for entire interface

Severity Rating

Used to allocate resources to fix problems

Estimates of need for more usability efforts

Combination of

- frequency
- impact
- persistence (one time or repeating)

Should be calculated after all evaluations are in

Should be done independently by all judges

Nielsen’s Example Ratings List

0 = I don't agree that this is a usability problem at all.
1 = Cosmetic problem only.
need not be fixed unless extra time is available on project
2 = Minor usability problem.
fixing this should be given low priority
3 = Major usability problem.
important to fix, so should be given high priority
4 = Usability catastrophe.
imperative to fix this before product can be released

Some comments on the above…
• Although Nielsen provides a “0” rating, it is unclear where it would be used
- perhaps on a “second opinion” evaluation
• It is possible for a cosmetic problem to be a usability catastrophe
- imagine a green checkmark meaning “bad/danger”

Debriefing

Conduct with evaluators, observers, and development team members

Discuss general characteristics of UI

Suggest potential improvements to address major usability problems

Development team rates how hard things are to fix

Make it a brainstorming session
• little criticism until end of session

Results of Using HE

Discount: benefit-cost ratio of 48 [Nielsen94]

- cost was $10,500 for benefit of $500,000
- value of each problem ~15K (Nielsen & Landauer)
- how might we calculate this value?
  - in-house –> productivity
  - open market –> sales

Correlation between severity & finding w/ HE

http://www.useit.com/papers/heuristic/heuristic_evaluation.html

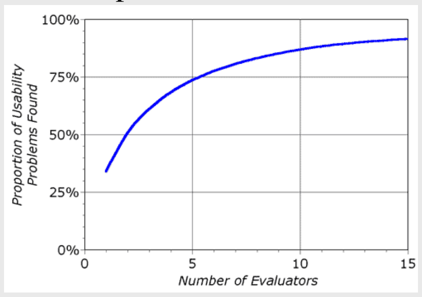
Why Multiple Evaluators?

Single evaluator achieves poor results

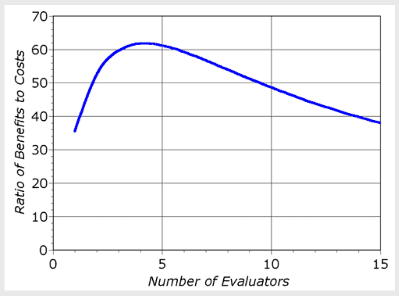
- only finds 35% of usability problems
- 5 evaluators find ~ 75% of usability problems
- why not more evaluators???? 10? 20?
  - adding evaluators costs more
  - many evaluators won't find many more problems

Why Multiple Evaluators (cont)?

problems found



benefits / cost



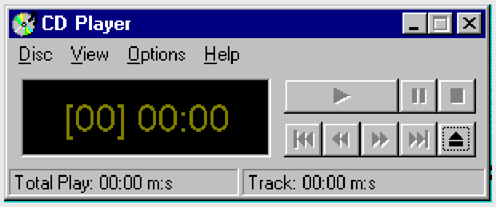
(Graphs for a specific example)

1 Simple and natural dialogue

Conform to the user's conceptual model.

Match the users' task in as natural a way as possible

- maximize mapping between interface and task semantics



Good? Bad?

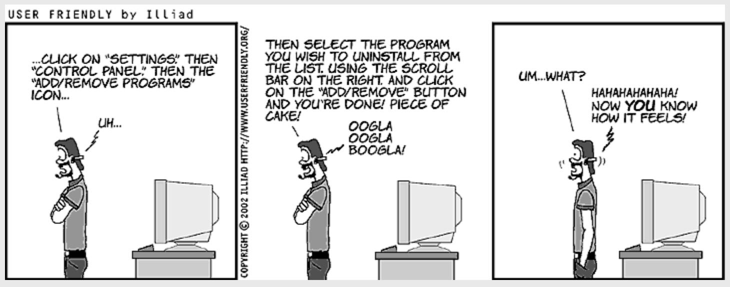
This has changed over time as people went away from audio tape in their lives…

1 Simple and natural dialogue

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
- use windows frugally
  - don’t make navigation and window management excessively complex

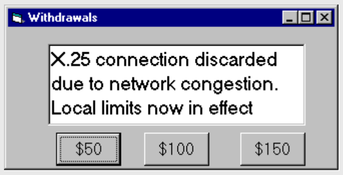
2 Speak the users’ language


USER FRIENDLY by Illiad
...CLICK ON "SETTINGS," THEN "CONTROL PANEL," THEN THE "ADD/REMOVE PROGRAMS" ICON...
UH...
COPYRIGHT © 2002 ILLIAD HTTP://WWW.USERFRIENDLY.ORG/
THEN SELECT THE PROGRAM YOU WISH TO UNINSTALL FROM THE LIST, USING THE SCROLL BAR ON THE RIGHT, AND CLICK ON THE "ADD/REMOVE" BUTTON AND YOU'RE DONE! PIECE OF CAKE!
OOGLA OOGLA BOOGLA!
UM...WHAT?
HAHAHAHAHAHA! NOW YOU KNOW HOW IT FEELS!

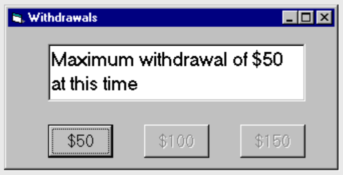
2 Speak the users' language

Use terminology based on users' language for task.

- e.g. withdrawing money from a bank machine



X.25 connection discarded due to network congestion. Local limits now in effect



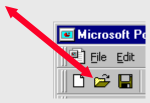
Maximum withdrawal of $50 at this time

Bad **Better**

Use meaningful mnemonics, icons, and abbreviations.

- eg: File / Save

– Ctrl + S	(abbreviation)
– Alt F S	(mnemonic for menu action)
– Open folder	(tooltip icon)



NOTE: This could fall under #7 providing shortcuts.

2 Speak the users' language

Ex: Consider a virus detection program that may have to be occasionally turned off.

One option would be to have an "override mode" that when activated would turn off the virus detection.

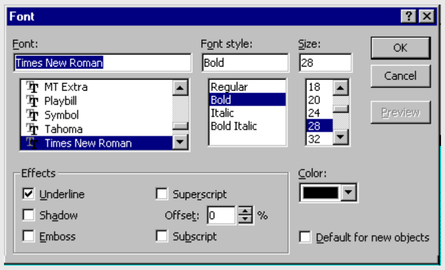
But this would be *on* when the user wanted the utility to be *off* – conflicting with the users' model

Alternatively, a checkbox that was on when the utility would be on would speak the users' language.

3 Minimize user’s memory load

Promote recognition over recall.

- computers are good at remembering thing, people not as much…
- menus, icons, choice dialog boxes vs command lines, field formats
- relies on visibility of objects to the user (but less is more!)


Font

Font: Times New Roman

Font style: Bold

Size: 28

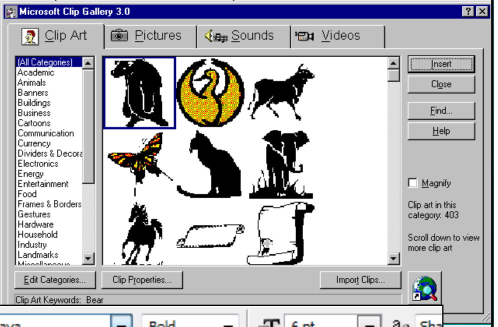
Effects: [✓ Underline] Shadow Emboss Superscript Subscript

Offset: 0 %

Color:

Default for new objects

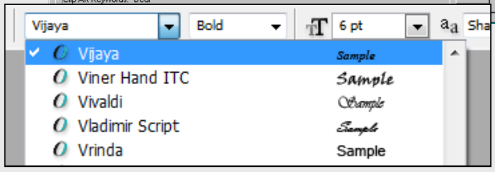
Bad


Microsoft Clip Gallery 3.0

Clip art in this category: 403

Scroll down to view more clip art

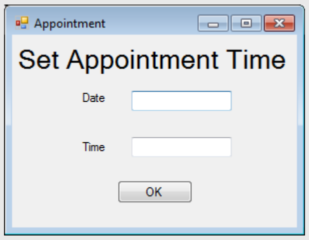
Clip Art Keywords: Bear



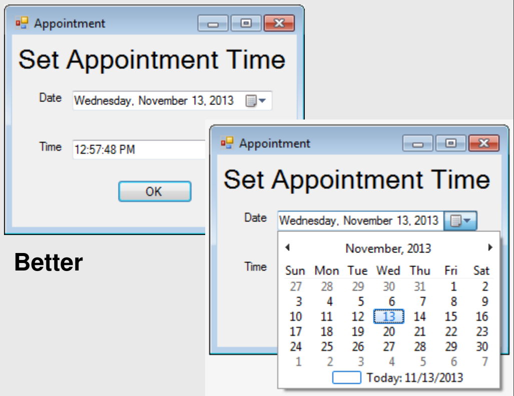
Better

3: Minimize user’s memory load

Describe required input format and provide an example or a default or a selection interface.



Bad


Set Appointment Time

Date: Wednesday, November 13, 2013

Time: 12:57:48 PM

November, 2013						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Today: 11/13/2013

Better

Small number of rules applied universally.

generic commands

– same command can be applied to all interface objects

•*interpreted in context of interface object*

– copy, cut, paste, drag ’n drop, ... for characters, words, paragraphs, circles, files

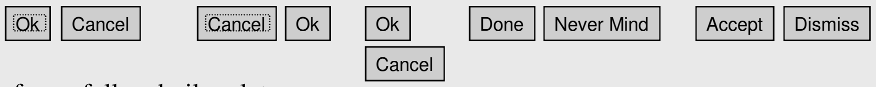
4: Be consistent

Consistency of effects.

- same words, commands, actions will always have the same effect in equivalent situations
  - predictability

Consistency of language and graphics.

- same information/controls in same location on all screens / dialog boxes



- forms follow boiler plate
- same visual appearance across the system (e.g. widgets)
  - e.g. different scroll bars in a single window system!

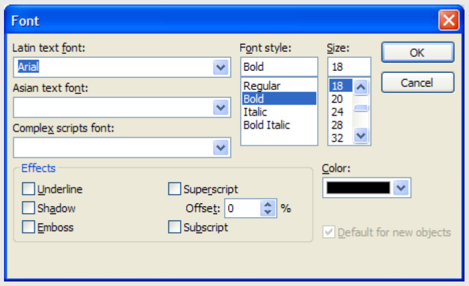
Consistency of input.

- consistent syntax across complete system

4: Be consistent

In application suites, have individual applications consistent with the other individual applications in the suite.

PowerPoint 2003



Font

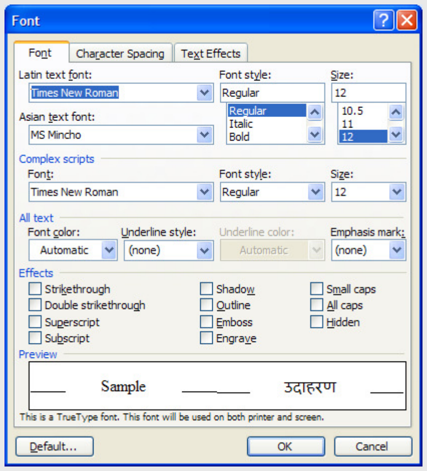
Latin text font: Arial
Asian text font:
Complex scripts font:
Font style: Bold (Regular, Bold, Italic, Bold Italic)
Size: 18 (18, 20, 24, 28, 32)
OK
Cancel

Effects
- [ ] Underline
- [ ] Shadow
- [ ] Emboss
- [ ] Superscript
- Offset: 0 %
- [ ] Subscript

Color:
- [✓] Default for new objects

-vs-

Word 2003



Font

Font | Character Spacing | Text Effects

Latin text font: Times New Roman
Asian text font: MS Mincho
Font style: Regular (Regular, Italic, Bold)
Size: 12 (10.5, 11, 12)

Complex scripts
Font: Times New Roman
Font style: Regular
Size: 12

All text
Font color: Automatic
Underline style: (none)
Underline color: Automatic
Emphasis mark: (none)

Effects
- [ ] Strikethrough
- [ ] Double strikethrough
- [ ] Superscript
- [ ] Subscript
- [ ] Shadow
- [ ] Outline
- [ ] Emboss
- [ ] Engrave
- [ ] Small caps
- [ ] All caps
- [ ] Hidden

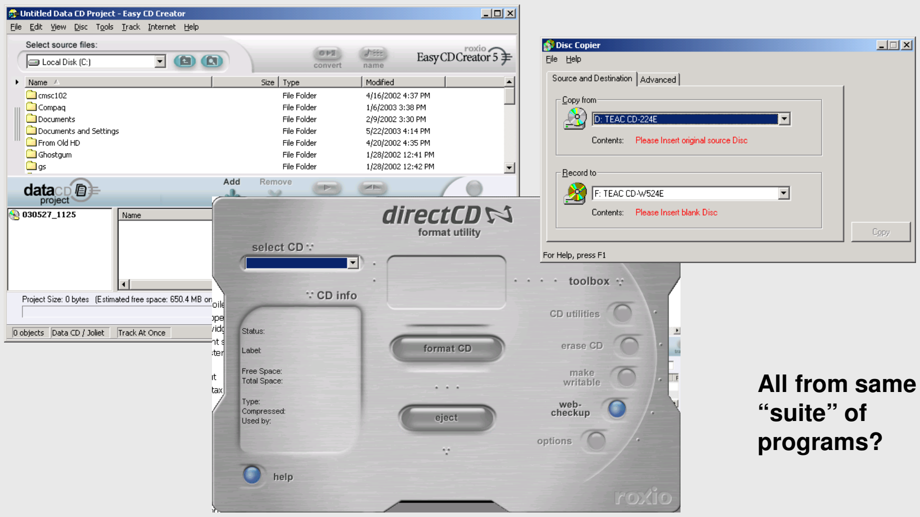
Preview
Sample उदाहरण

This is a TrueType font. This font will be used on both printer and screen.

Default... OK Cancel

4: Be consistent

In application suites, have individual applications consistent with the other individual applications in the suite.



Name	Size	Type	Modified
cmsc102		File Folder	4/16/2002 4:37 PM
Compaq		File Folder	1/6/2003 3:38 PM
Documents		File Folder	2/9/2002 3:30 PM
Documents and Settings		File Folder	5/22/2003 4:14 PM
From Old HD		File Folder	4/20/2002 4:35 PM
Ghostgum		File Folder	1/28/2002 12:41 PM
gs		File Folder	1/28/2002 12:42 PM

Copy from: D: TEAC CD-224E
Contents: Please Insert original source Disc

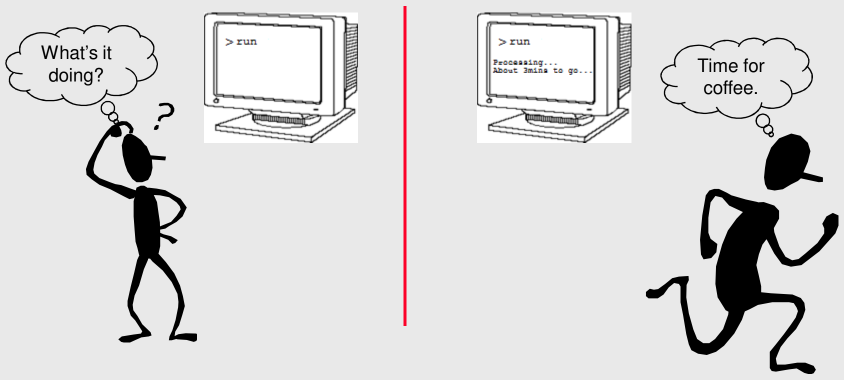
Record to: F: TEAC CD-W524E
Contents: Please Insert blank Disc

All from same "suite" of programs?

5: Provide feedback

Continuously inform the user about.

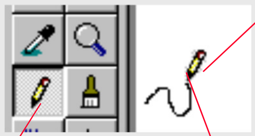
- what it is doing
- how it is interpreting the user's input
- user should always be aware of what is going on



```
> run
```

```
> run
Processing...
About 3mins to go...
```

5. Provide feedback



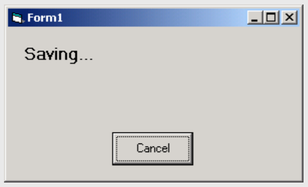
What mode am I in now?

What did I select?

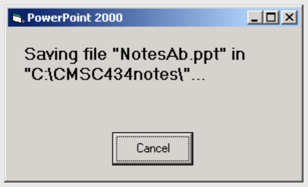
How is the system interpreting my actions?

5. Provide feedback

Should be as specific as possible, based on user's input.



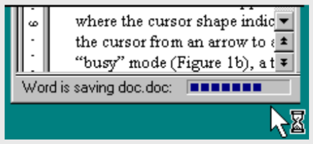
Saving...



Saving file "NotesAb.ppt" in "C:\CMSC434notes\"...

Bad **Better**

Best within the context of the action rather than with a dialog box.



where the cursor shape indic
the cursor from an arrow to
"busy" mode (Figure 1b), a t

Word is saving doc.doc:

5. Provide feedback

Response time is important…

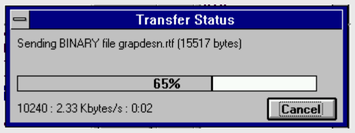
- how users perceive delays

0.1	second max: perceived as “instantaneous”
1	seconds max: user’s flow of thought stays uninterrupted, but delay noticed
10	seconds: limit for keeping user’s attention focused on the dialog
> 10	seconds: user will want to perform other tasks while waiting and might think that the application has failed

5. Provide feedback

Dealing with long delays…

- Cursors
  - for short transactions
- Percent-done dialogs
  - for longer transactions
    - how much left
    - estimated time
    - what it is doing


Transfer Status

Sending BINARY file grapdesn.rtf (15517 bytes)

65%

10240 : 2.33 Kbytes/s : 0:02

NOTE: When giving this type of feedback, take care to do so in a meaningful fashion based upon percent of time. For example, if doing a progress bar for an e-mail client, rather than the % of messages sent, use % of size of messages.

- “Still Working”
  - for unknown/changing times

6. Provide clearly marked exits

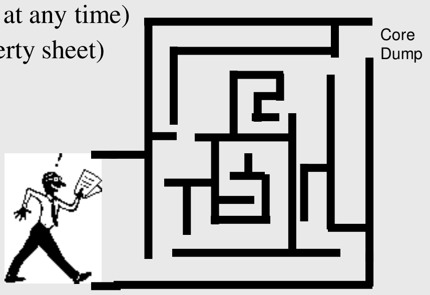
6. Provide clearly marked exits

Users don't like to feel trapped by the computer!

- should offer an easy way out of as many situations as possible

Strategies:

- Cancel button (for dialogs waiting for user input)
- Universal Undo (can get back to previous state)
- Interrupt (especially for lengthy operations)
- Quit (for leaving the program at any time)
- Defaults (for restoring a property sheet)



7. Provide shortcuts

Experienced users should be able to perform frequently used operations quickly!

Strategies:

- keyboard and mouse accelerators
  - abbreviations
  - command completion
  - menu shortcuts
  - function keys
  - double clicking vs menu selection

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

- navigation jumps
  - e.g., going to window/location directly, and avoiding intermediate nodes

- history systems
  - WWW: ~60% of pages are revisits

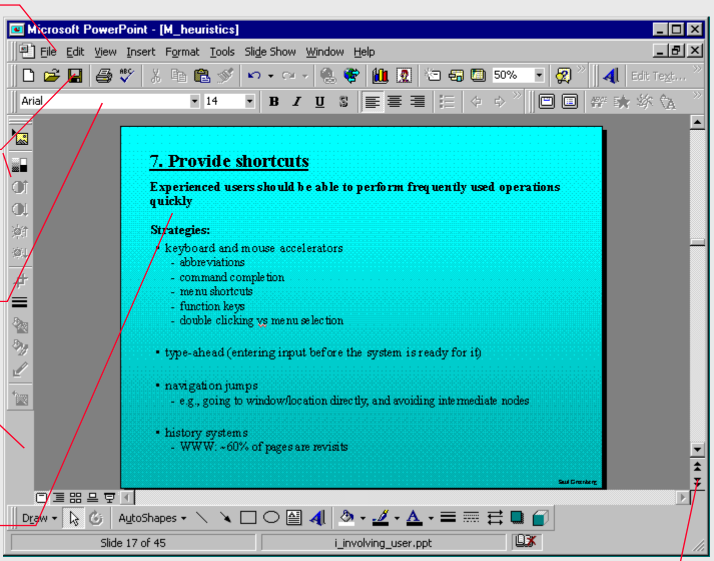
Keyboard accelerators for menus

Customizable toolbars and palettes for frequent actions

Split menu, with recently used fonts on top

Right-click raises toolbar dialog box

Right-click raises object-specific menu


Microsoft PowerPoint - [M_heuristics]
File Edit View Insert Format Tools Slide Show Window Help
50%
Edit Text...
Arial
14
Draw
AutoShapes
Slide 17 of 45
i_involving_user.ppt

7. Provide shortcuts

Experienced users should be able to perform frequently used operations quickly

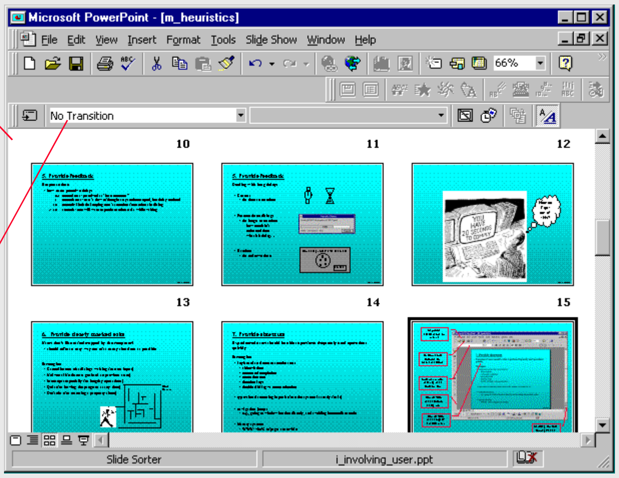
Strategies:
- keyboard and mouse accelerators
  - abbreviations
  - command completion
  - menu shortcuts
  - function keys
  - double clicking vs menu selection
- type-ahead (entering input before the system is ready for it)
- navigation jumps
  - e.g., going to window/location directly, and avoiding intermediate nodes
- history systems
  - WWW: ~60% of pages are revisits

Saul Greenberg

Scrolling controls for page-sized increments

Alternate representation for quickly doing different set of tasks

Toolset brought in appropriate to this representation


Microsoft PowerPoint - [m_heuristics]
File Edit View Insert Format Tools Slide Show Window Help
66%
No Transition
10
11
12
13
14
15
Slide Sorter
i_involving_user.ppt

8: Deal with errors in a positive and helpful manner

People will make errors – plan for it!

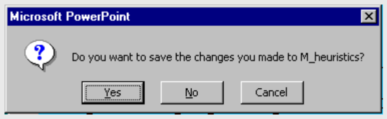
Errors we make

- Mistakes
  - arise from conscious deliberations that lead to an error instead of the correct solution
- Slips
  - unconscious behavior that 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 similarities of actions

Types of slips

Capture error (habit)

- a frequently performed activity takes charge "on autopilot" instead of the one intended at the time
  - occurs when common and rarer actions have same initial sequence
    - change clothes for dinner and find oneself in bed (William James, 1890)
    - confirm saving of a file when you don't want to replace it



Do you want to save the changes you made to M_heuristics?

Types of slips

Description error

- intended action has much in common with others that are possible
  - usually occurs when right and wrong objects physically near each other
    - pour juice into bowl instead of glass
    - go jogging, come home, throw sweaty shirt in toilet instead of laundry basket
    - move file to trash instead of to folder

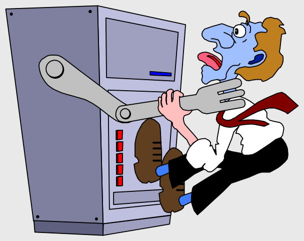
Loss of activation

- forgetting what the goal is while undergoing the sequence of actions
  - start going to room and forget why you are going there
  - navigating menus/dialogs and can't remember what you are looking for
  - but continue action to remember (or go back to beginning)!

Mode errors

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

Designing for slips



General rules

- Prevent slips before they occur
- Detect and correct slips when they do occur
- User correction through feedback and undo

Examples

- capture errors
  - instead of confirmation, make actions undoable
  - allows reconsideration of action by user
    - e.g. Mac trash can can be opened and "deleted" file taken back out
- description errors
  - in icon-based interfaces, make sure icons are not too similar,
  - check for reasonable input, etc.
- loss of activation
  - if system knows goal, make it explicit
  - if not, allow person to see path taken
- mode errors
  - have as few modes as possible (preferably none)
  - make modes highly visible

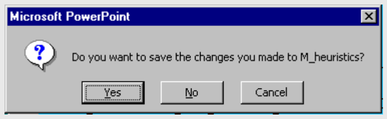
Generic system responses for errors

Interlock

- deals with errors by preventing the user from continuing
  - eg cannot delete an object if none are selected

Warn

- warn people that an unusual situation is occurring
- when overused, becomes an irritant
  - e.g.,
    - audible bell
    - alert box



Microsoft PowerPoint

Do you want to save the changes you made to M_heuristics?

Generic system responses for errors continued...

Do nothing

- illegal action just doesn't do anything
- user must infer what happened
  - enter letter into a numeric-only field (key clicks ignored)
  - put a file icon on top of another file icon (returns it to original position)

Self-correct

- system guesses legal action and does it instead
- but leads to a problem of trust
  - spelling corrector

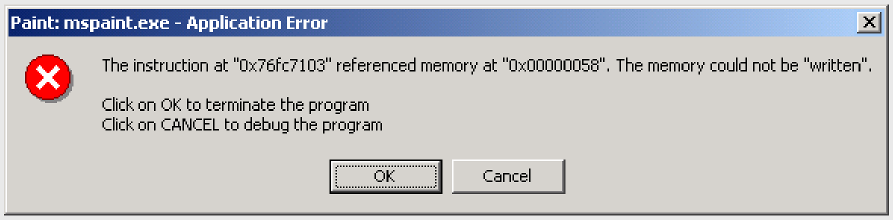
Lets talk about it

- system initiates dialog with user to come up with solution to the problem
  - compile error brings up offending line in source code

Teach me

- system asks user what the action was supposed to have meant
- action then becomes a legal one

8 Deal with errors in a positive and helpful manner


Paint: mspaint.exe - Application Error

The instruction at "0x76fc7103" referenced memory at "0x00000058". The memory could not be "written".

Click on OK to terminate the program
Click on CANCEL to debug the program

HUH ?!?

8 Deal with errors in a positive and helpful manner

>Shut down reactor
Bus error – core dump

A problematic message to a nuclear power plant operator

8 Deal with errors in a positive and helpful manner

Provide meaningful error messages!

- error messages should be in the user's language (preferably task language)
- don't make people feel stupid

Bad

Try again…

Error 25

Cannot open this document.

Better

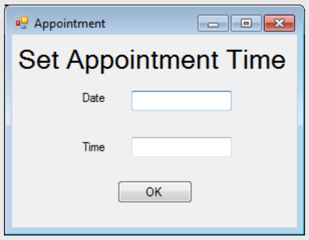
Cannot open "chapter 5" because the application "Microsoft Word" is not on your system

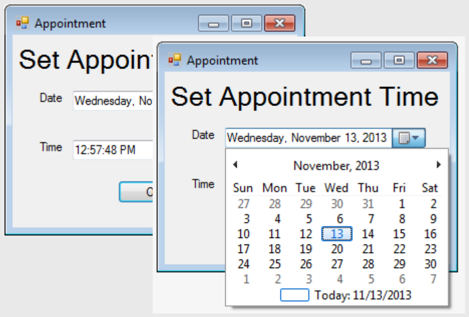
Cannot open "chapter 5" because the application "Microsoft Word" is not on your system. Open it with "OpenOffice" instead?

8 Deal with errors in a positive and helpful manner

Prevent errors.

- try to make errors “impossible” to make
- modern widgets: only “legal commands” selected, or “legal data” entered (which if these might allow you to enter February 29th, 2014?)





Provide reasonableness checks on input data.

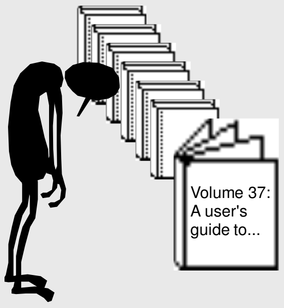
- on entering order for office supplies
  - 5000 pencils is an unusually large order. Do you really want to order that many?

Consumer Manuals...

C'MON.
I'M SO AFRAID.
CONQUER YOUR FEARS.
I FEEL INADEQUATE.
C'MON... I'LL BE RIGHT NEXT TO YOU THE WHOLE TIME...
SKRCHCHHH...
NEW VCR!
"PROGRAMMING INSTRUCTIONS..."
10,000 FUNCTION
..."AND CONNECT THE H-73D8 DEEBLE SIGNAL AMP TWIXT THE VCR AND THE CBD DESCRAMBLER BOX ZH-3108...
"THEN INITIALIZE THE NEW, HANDY, SIMPLIFIED REMOTE"...
REMOTE?
WHERE'S THE HANDY SIMPLIFIED REMOTE?!
OVER HERE!
"...ENTER CHANNEL... PUSH 'START TIME'... ENTER DATE...
"PUSH 'WEEK'... ENTER CODE... PUSH 'DAY', ENTER 'PM' OR 'AM'... PUSH 'TIME'... ENTER 'END TIME.' PUSH THIS PUSH THAT PUSH IT AGAIN
HANG GLIDING

9. Provide help

Help is not a replacement for bad design!



Simple systems:

- walk up and use; minimal instructions

Most other systems:

- feature rich
- some users will want to become “experts” rather than “casual” users
- intermediate users need reminding, plus a learning path

Documentation and how it is used

NOTE: Many users do not read manuals.

- prefer to spend their time pursuing their task

Usually used when users are in some kind of panic, they will want (and perhaps need) immediate help.

- indicates need for online documentation, good search/lookup tools
- online help can be specific to current context
- Kindle “Mayday” option?

NOTE: paper or CD manuals unavailable in many business environments
  – e.g. single copy locked away in system administrator’s office

Sometimes documentation is used for quick reference in advance.

- syntax of actions, possibilities...
- list of shortcuts ...

Types of help

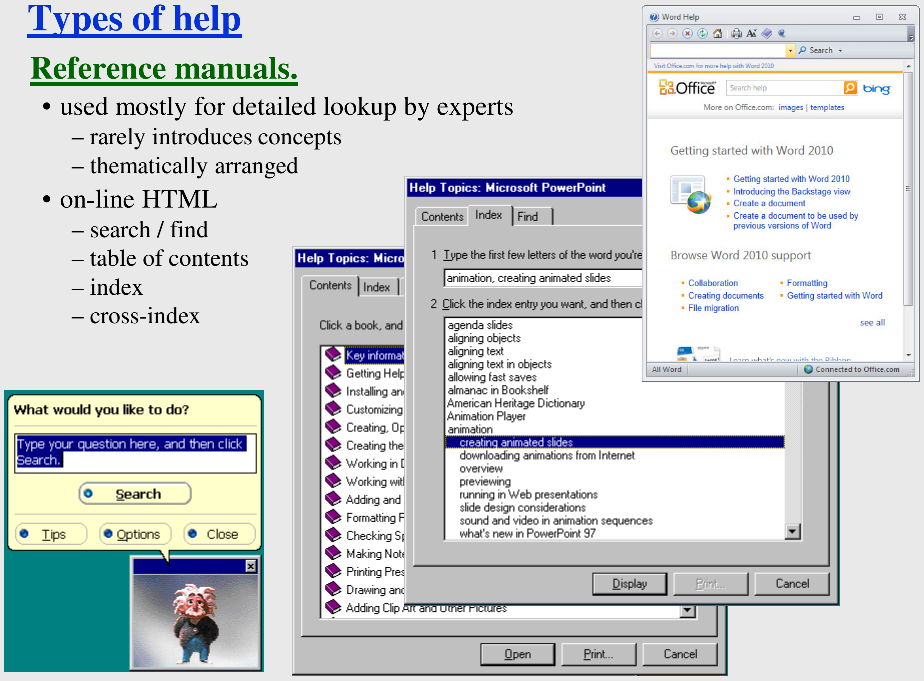
Tutorial and/or getting started manuals.

- short guides that people are likely to read when first obtaining their systems
  - encourages exploration and getting to know the system
  - tries to get conceptual material across and essential syntax
- on-line "tours", exercises, and demos
  - demonstrates very basic principles through working examples

Types of help

Reference manuals.

- used mostly for detailed lookup by experts
  - rarely introduces concepts
  - thematically arranged
- on-line HTML
  - search / find
  - table of contents
  - index
  - cross-index



Getting started with Word 2010

- Getting started with Word 2010
- Introducing the Backstage view
- Create a document
- Create a document to be used by previous versions of Word

Browse Word 2010 support

- Collaboration
- Creating documents
- File migration
- Formatting
- Getting started with Word

see all

1 Type the first few letters of the word you're

animation, creating animated slides

2 Click the index entry you want, and then c

agenda slides
aligning objects
aligning text
aligning text in objects
allowing fast saves
almanac in Bookshelf
American Heritage Dictionary
Animation Player
animation
  creating animated slides
  downloading animations from Internet
  overview
  previewing
  running in Web presentations
  slide design considerations
  sound and video in animation sequences
  what's new in PowerPoint 97

Click a book, and

Key informat
Getting Help
Installing an
Customizing
Creating, Op
Creating the
Working in D
Working wit
Adding and
Formatting F
Checking Sp
Making Note
Printing Pres
Drawing and
Adding Clip Art and Other Pictures

Type your question here, and then click Search.

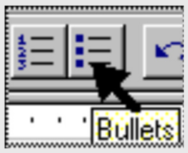
Types of help

Reminders to the user.

- short reference cards used to be VERY popular
  - expert user who just wants to check facts
  - novice who wants to get overview of system's capabilities

- keyboard templates used to be VERY popular
  - shortcuts/syntactic meanings of keys; recognition vs. recall; capabilities

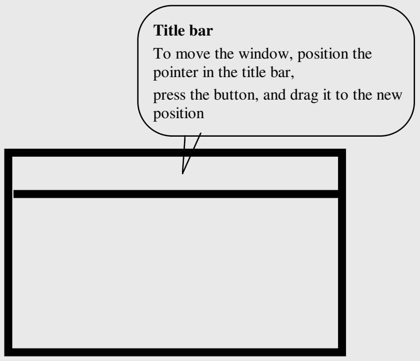
- tooltips are STILL very popular!
  - text over graphical items indicates their meaning or purpose
  - No way to do this with touch interfaces ☹



Types of help

Context-sensitive help.

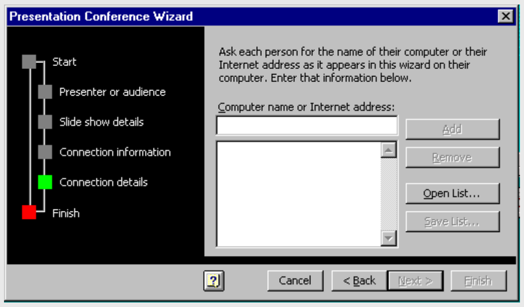
- system provides help on the interface component the user is currently working with
  - Macintosh "balloon help"
  - Microsoft "What's this" help
    - brief help explaining whatever the user is pointing at on the screen


Title bar
To move the window, position the pointer in the title bar,
press the button, and drag it to the new position

Types of help

Wizards specific to task.

- walks user through typical tasks
- *but* dangerous if user gets stuck

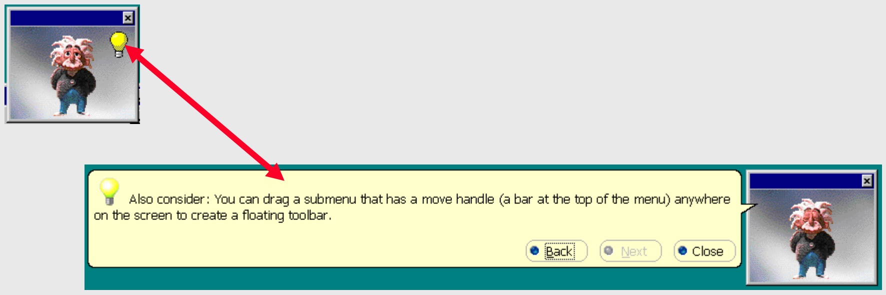


Ask each person for the name of their computer or their Internet address as it appears in this wizard on their computer. Enter that information below.

Types of help

Tips to the user.

- provides migration path to learning system features
- also context-specific tips on being more efficient
- must be "smart", otherwise boring and/or tedious and/or interrupts user's work flow (ie: Office Assistant had good and bad elements)



Also consider: You can drag a submenu that has a move handle (a bar at the top of the menu) anywhere on the screen to create a floating toolbar.