PV182
Human Computer Interaction

Lecture 11

Heuristic Evaluation

Fotis Liarokapis
liarokap@fi.muni.cz

26th November 2018

Heuristic Evaluation

Usability Heuristics

- Avoid common design pitfalls by following 9 (10, 15, 14. …) design principles
- Inspect an interface for usability problems with these principles

Design principles

- Broad usability statements that guide a developer's design efforts
  – use the users language
  – provide feedback…
- Derived from common design problems across many systems

Heuristic evaluation

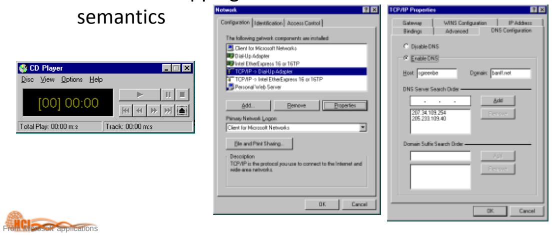
- Systematic inspection to see if interface complies to guidelines
- Method
  – 3-5 inspectors
  – usability engineers, end users, double experts…
  – inspect interface in isolation (~1–2 hours for simple interfaces)
  – compare notes afterwards
    - single evaluator only catches ~35% of usability problems
    - 5 evaluators catch 75%
- Works for paper, prototypes, and working systems

Heuristic evaluation

- Advantages
  – "minimalist" approach
    - a few guidelines identify many common usability problems
    - easily remembered, easily applied with modest effort
  – discount usability engineering
    - end users not required
    - cheap and fast way to inspect a system
    - can be done by usability experts, double experts, and end users
- Problems:
  – principles are more or less at the motherhood level
    - can't be treated as a simple checklist
    - subtleties involved in their use

1 Simple and natural dialogue

– use the user's conceptual model
– match the users' task sequence
– minimize mapping between interface and task semantics



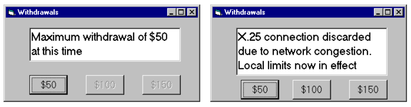
From Microsoft applications

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
  - remove modes
  - use windows frugally
    - don't add unneeded navigation and window management

2 Speak the users' language

- Terminology based on users' language for task
  – e.g. withdrawing money from a bank machine



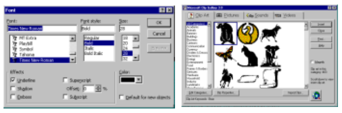
Maximum withdrawal of $50 at this time

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

- Use meaningful mnemonics, icons & abbreviations
  – eg File / Save
    - Ctrl + S (abbreviation)
    - Alt FS (mnemonic for menu action)
    - (tooltip icon)

3 Minimize user's memory load

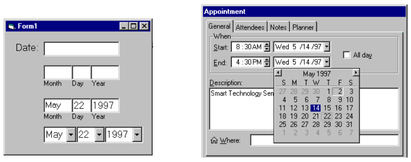
- Computers good at remembering, people are not!
- Promote recognition over recall
  – menus, icons, choice dialog boxes vs commands, field formats
  – relies on visibility of objects to the user (but less is more!)



From Microsoft applications

3: Minimize user's memory load

- Gives input format, example and default

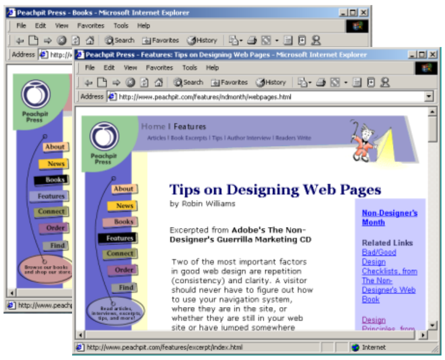


4: Be consistent

- Consistent syntax of input
- Consistent language and graphics
  – same visual appearance across the system (e.g. widgets)
  – same information/controls in same location on all windows

Ok Cancel   Cancel Ok   Ok Cancel   Accept Dismiss

- Consistent effects
  – commands, actions have same effect in equivalent situations
    - predictability


Peachpit Press - Books - Microsoft Internet Explorer

Peachpit Press - Features: Tips on Designing Web Pages - Microsoft Internet Explorer

http://www.peachpit.com/features/ndmonth/webpages.html

Home | Features

Articles | Book Excerpts | Tips | Author Interview | Readers Write

Tips on Designing Web Pages

by Robin Williams

Excerpted from **Adobe's The Non-Designer's Guerrilla Marketing CD**

Two of the most important factors in good web design are repetition (consistency) and clarity. A visitor should never have to figure out how to use your navigation system, where they are in the site, or whether they are still in your web site or have jumped somewhere

Non-Designer's Month

Related Links

Bad/Good Design

Checklists, from The Non-Designer's Web Book

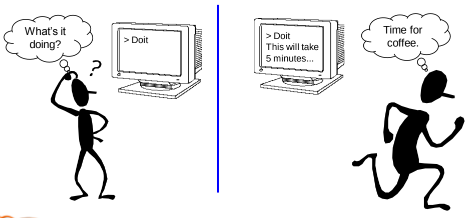
Design Principles, from

http://www.peachpit.com/features/excerpt/index.html

From Peachpit website

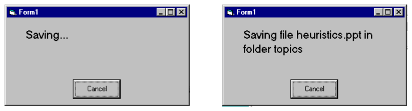
5: Provide feedback

- 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
-
-

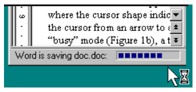


5. Provide feedback

- Be as specific as possible, based on user's input



Best within the context of the action


where the cursor shape indic
the cursor from an arrow to
"busy" mode (Figure 1b), a t
Word is saving doc.doc:

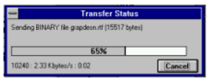
5. Provide feedback

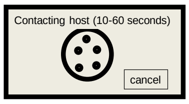
- Response time
  - how users perceive delays

<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

5. Provide feedback

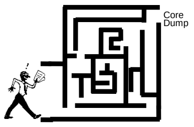
- Dealing with long delays
  - Cursors
    - for short transactions
  - Percent done dialogs
    - time left
    - estimated time
  - Random
    - for unknown times


Transfer Status
Sending BINARY file gradesn.rtf (15517 bytes)
65%
10240 : 2.33 Kbytes/s : 0:02
Cancel



6. Provide clearly marked exits

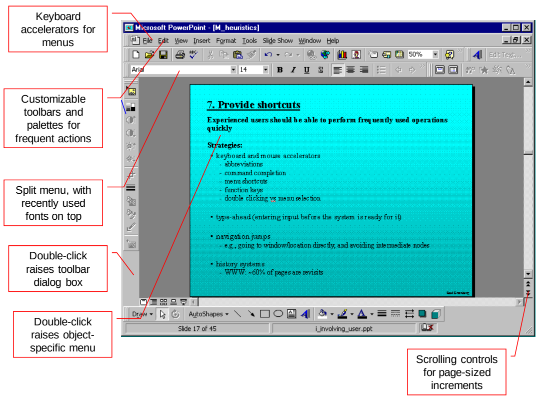
- Users don't like to feel trapped by the computer!
  - should offer an easy way out of as many situations as possible
- Strategies:
  - Cancel button (for dialogs waiting for user input)
  - Universal Undo (can get back to previous state)
  - Interrupt (especially for lengthy operations)
  - Quit (for leaving the program at any time)
  - Defaults (for restoring a property sheet)



MASARYKOVA UNIVERZITA

7. Provide shortcuts

- Experienced users - 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
    - e.g., going to window/location directly, and avoiding intermediate nodes
  - history systems
    - WWW: ~60% of pages are revisits

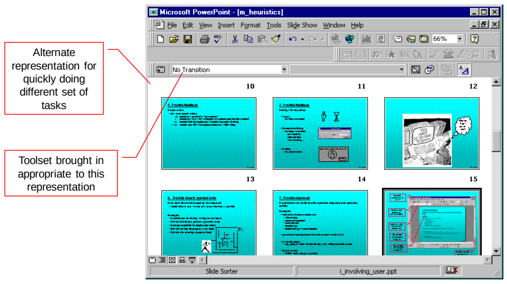

Keyboard accelerators for menus
Customizable toolbars and palettes for frequent actions
Split menu, with recently used fonts on top
Double-click raises toolbar dialog box
Double-click raises object-specific menu
Scrolling controls for page-sized increments
Microsoft PowerPoint - [M_heuristics]
File Edit View Insert Format Tools Slide Show Window Help
50%
Arial
14
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

Microsoft Powerpoint


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

Microsoft Powerpoint

MASARYKOVA UNIVERZITA

8: Deal with errors in a positive manner

- People will make errors!
- Errors we make
  - Mistakes
    - conscious deliberations lead to an error instead of correct solution
  - Slips
    - unconscious behaviour gets misdirected en route to satisfying goal
      - e.g. drive to store, end up in the office
    - shows up frequently in skilled behaviour
      - usually due to inattention
    - often arises from similar actions

MASARYKOVA UNIVERZITA

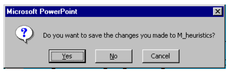
Designing for slips

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

MASARYKOVA UNIVERZITA

Types of slips

- Capture error
  - frequently done activity takes charge instead of one intended
  - occurs when common & 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 delete it
  - minimize by
    - make actions undoable instead of confirmation
    - allows reconsideration of action by user
      - e.g. open trash to undelete a file



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

Types of slips

- Description error
  - intended action similar to others that are possible
    - usually 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

Types of slips

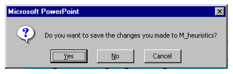
- Loss of activation
  - forget what the goal is while undergoing the sequence of actions
    - start going to room and forget why you are going there
    - navigating menus/dialogs & can't remember what you are looking for
    - but continue action to remember (or go back to beginning)!
  - minimize by
    - if system knows goal, make it explicit
    - if not, allow person to see path taken

Types of slips

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

Generic system responses for errors

- General idea: Forcing functions
  - prevent / mitigate continuation of wrongful action
- Gag
  - deals with errors by preventing the user from continuing
    - eg cannot get past login screen until correct password entered
- Warn
  - warn people that an unusual situation is occurring
  - when overused, becomes an irritant
    - e.g.,
      - audible bell
      - alert box



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

Generic system responses for errors

- 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

Generic system responses for errors

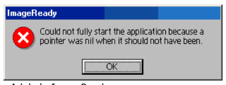
- Lets talk about it
  - system initiates dialog with user to come up with solution to the problem
    - compile error brings up offending line in source code
- Teach me
  - system asks user what the action was supposed to have meant
  - action then becomes a legal one

8: Deal with errors in a positive manner

>Shut down reactor
Bus error – core dump

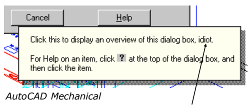
A problematic message to a nuclear power plant operator

8: Deal with errors in a positive manner


ImageReady

Could not fully start the application because a pointer was nil when it should not have been.

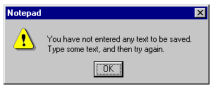
Adobe's *ImageReady*



Click this to display an overview of this dialog box, idiot.

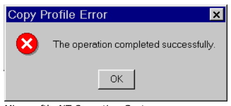
For Help on an item, click ? at the top of the dialog box, and then click the item.

AutoCAD Mechanical


Notepad

You have not entered any text to be saved. Type some text, and then try again.

Windows Notepad


Copy Profile Error

The operation completed successfully.

Microsoft's *NT Operating System*

8: Deal with errors in a positive manner

- 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

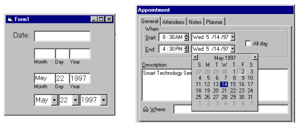
Cannot open this document

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 "Teachtext" instead?

8: Deal with errors in a positive manner

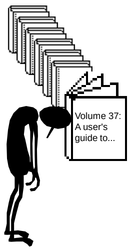
- Prevent errors
  - try to make errors impossible
  - modern widgets: can only enter legal data



- 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?

9. Provide help

- 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



Documentation and how it is used

- Many users do not read manuals
  - prefer to spend their time pursuing their task
- Usually used when users are in some kind of panic
  - paper manuals unavailable in many businesses!
    - e.g. single copy locked away in system administrator's office
  - 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 ...

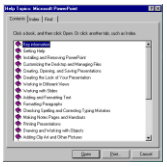
Types of help

- 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

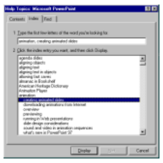
Types of help

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

What would you like to do?
Type your question here, and then click Search.
Search
Tips
Options
Close


Help Topics: Microsoft PowerPoint
Contents
Index
Find
Key information
Getting Help
Installing and Removing PowerPoint
Customizing the Desktop and Managing Files
Creating, Opening, and Saving Presentations
Creating the Look of Your Presentation
Working in Different Views
Working with Slides
Adding and Formatting Text
Formatting Paragraphs
Checking Spelling and Correcting Typing Mistakes
Making Notes Pages and Handouts
Printing Presentations
Drawing and Working with Objects
Adding Clip Art and Other Pictures
Open
Print...
Cancel

Click a book, and then click Open. Or click another tab, such as Index.


Help Topics: Microsoft PowerPoint
Contents
Index
Find
agenda slides
aligning objects
aligning text
aligning text in objects
allowing text 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
Display
Print
Cancel

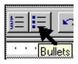
1 Type the first few letters of the word you're looking for.

animation, creating animated slides

2 Click the index entry you want, and then click Display.

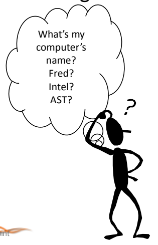
Types of help

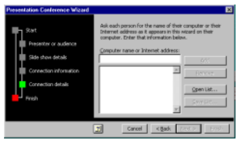
- Reminders
  - short reference cards
    - expert user who just wants to check facts
    - novice who wants to get overview of system's capabilities
  - keyboard templates
    - shortcuts/syntactic meanings of keys; recognition vs. recall; capabilities
  - tooltips and other context-sensitive help
    - text over graphical items indicates their meaning or purpose



Types of help

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



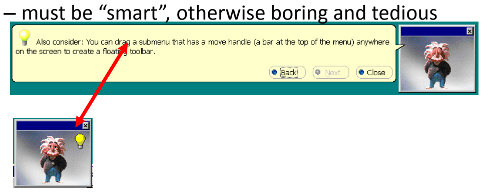


Ask each person for the name of their computer or their Internet address as it appears in this wizard on their computer. Enter that information below.

Computer name or Internet address:

Types of help

- Tips
  - migration path to learning system features
  - also context-specific tips on being more efficient
  - must be "smart", otherwise boring and tedious



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.

Nielsen's 10 heuristics

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

Usage of heuristics

- Heuristic evaluation
  - Principles can be used to systematically inspect the interface for usability problems
  - Principles can be expressed as a set of domain specific questions
  - Evaluation may use checklists

Evaluating Heuristic evaluation

- Problems found by a single inspector
- Problems found by multiple inspectors
- Individuals vs. teams
- Self guided or scenarios?

Problems found by a single inspector

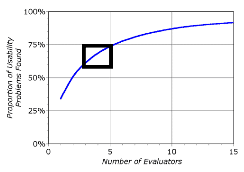
- Average over six case studies
  - 35% of all usability problems;
  - 42% of the major problems
  - 32% of the minor problems
- Not great, but
  - finding some problems with one evaluator is *much* better than finding no problems with no evaluators!

Problems found by a single inspector

- Varies according to
  - difficulty of the interface being evaluated
  - the expertise of the inspectors
- Average problems found by:
  - novice evaluators - 22%
    - no usability expertise
  - regular specialists - 41%
    - expertise in usability
  - double specialists - 60%
    - experience in both usability and the particular kind of interface being evaluated
    - also find domain-related problems
- Tradeoff
  - novices poorer, but cheaper!

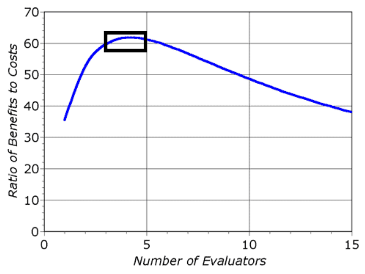
Problems found by multiple evaluators

- 3-5 evaluators find 66-75% of usability problems
  - different people find different usability problems
  - only modest overlap between the sets of problems found



Problems found by multiple evaluators

- Where is the best cost/benefit?



Self Guided vs Scenario Exploration

- Self-guided
  - open-ended exploration
  - Not necessarily task-directed
  - good for exploring diverse aspects of the interface, and to follow potential pitfalls
- Scenarios
  - step through the interface using representative end user tasks
  - ensures problems identified in relevant portions of the interface
  - ensures that specific features of interest are evaluated
  - but limits the scope of the evaluation - problems can be missed

Questions

Acknowledgements

- Prof. Ing. Jiří Sochor