

Questions

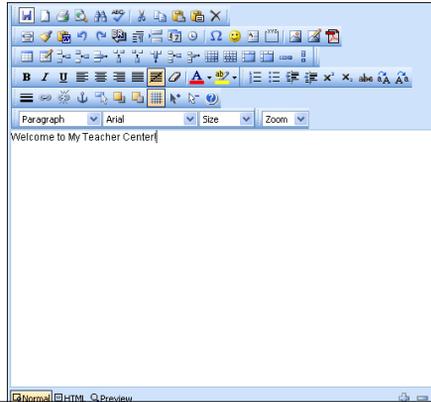
- Deployment issues for this homework
- Apple Magic Mouse
- B&N Nook
- Midterm next week

Menus

- The trouble with hierarchies
 - Mac solution for finding items
- Organize by function
(help support simple mental model)
- Group similar functions visually
(Hierarchical scanning more visually efficient)

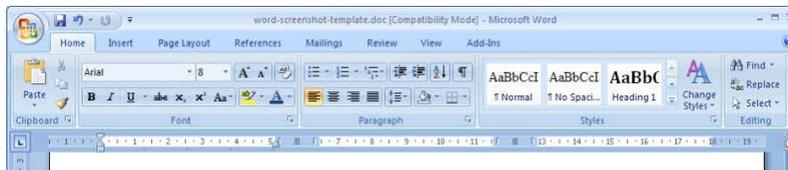
Customization

- Personalizing visual style good (i.e., themes)
- Changing available functionality or structure risky



Customization

- Personalizing visual style good (i.e., themes)
- Changing available functionality or structure risky



Customization

- Personalizing visual style good (i.e., themes)
- Changing available functionality or structure risky

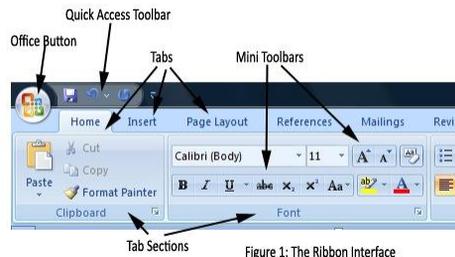
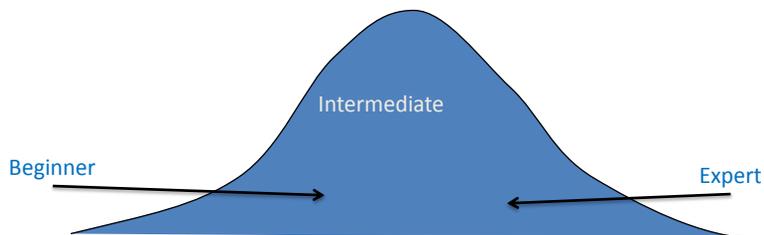


Figure 1: The Ribbon Interface

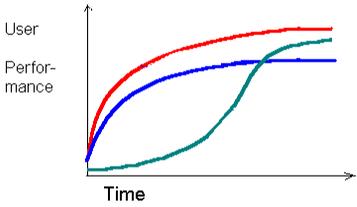
Perpetual Intermediates



Novice and Expert Support

- Best to support natural migration from natural to expert rather than different products
- But in practice, there often *are* different products
- Different kinds of expertise:
 - General computer experience
 - Experience with application
 - Domain expertise

Novice and Expert Support

- Differing needs:
 - Learnability
 - Efficiency
- 
- Expert modes
 - Layered interfaces
 - Undo
 - Efficiency mechanisms (i.e., keyboard shortcuts)

Novice and Expert Support

- Expert modes

The screenshot shows a 'Profile' page with a 'Personal Options' section. Under 'Visual Editor', there is a checkbox for 'Disable the visual editor when writing'. Under 'Admin Color Scheme', there are two radio buttons: 'Blue' (selected) and 'Gray'. Under 'Keyboard Shortcuts', there is a checkbox for 'Enable keyboard shortcuts for comment moderation. [More information](#)'. Under 'Expert mode', there is a checked checkbox for 'Enable expert mode if you are familiar with WordPress and don't need the inline documentation in the admin.' Below this section is a 'Name' field.

Layered Interface

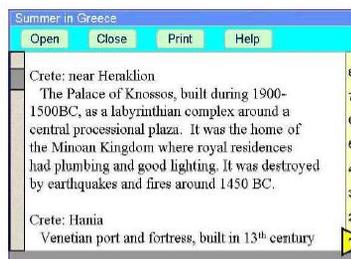


Figure 2a: Layer 1 has typing and a few buttons.

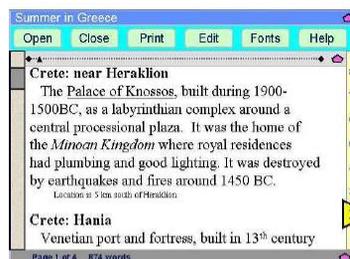


Figure 2c: Layer 2 has additional buttons, fonts, ruler, status bar.

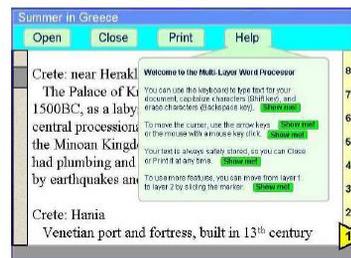


Figure 2b: Layer 1 has simple help with animated instruction.



Figure 2d: Layer 3 has pull-down menus with more features.

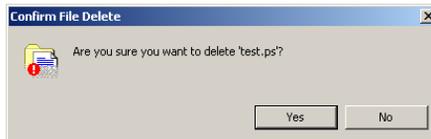
Error Types

- Mistakes
 - Conscious decision with unforeseen consequences
- Slips
 - Automatic behaviors kicking in
 - Drive to the store, end-up in the office
 - Press enter one time too many...
 - Mode errors
 - Forget the mode the application is in
 - Loss of activation
 - Forget what your goals were

Designing for slips

An ounce of prevention is worth more than a pound of cure!

- Examples
 - Design modeless interfaces
 - Instead of confirmations provide undo mechanisms



- Check for reasonable input
 - Be prepared to handle several formats
 - Make entering a incorrect format impossible
- Make the current goal clear
 - Prevent lost of activations

Avoid Errors – Tolerant Designs

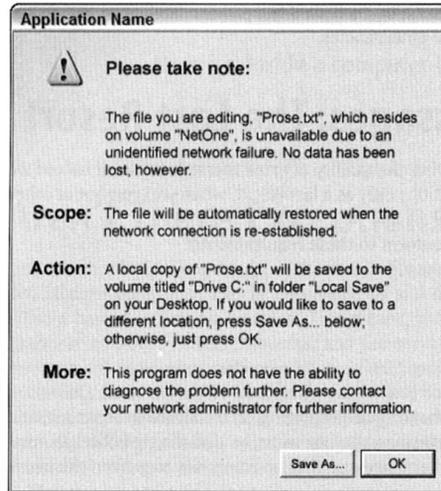
- Systems engineering calls it “fault tolerance”
- Avoid errors in the first place
- Allow easy undo and redo to minimize cost of errors

- Examples:
 - Good date entry (i.e., Outlook)
 - As opposed to a constrained system (airline sites)

Dealing with errors

- People will make errors!
 - You can ignore them
 - Generally very confusing
 - You can correct them automatically
 - Spelling corrector
 - But is the system right 100% of the time?
 - You can discuss it
 - But novice/expert tradeoff
 - You can try to teach the user what to do
 - Office assistant
- Respect users feelings!
 - The user is never wrong

Good error messages



From Cooper's "About Face 2.0"

Good error messages

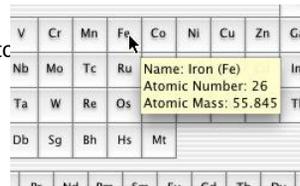
- Provide meaningful error messages
 - Explain the problem in terms of the user conceptual model
 - Don't make the user feel stupid
 - Offer a way to correct the problem
 - Compare
 - Error 25: access denied
 - Cannot open "chapter 5" because "Microsoft Word" is not installed. Do you want to use Notepad instead?

Provide help and documentation

- Providing help is not an excuse for poor design!
 - Saving a couple of line of code or writing several pages of documentation?
 - Users don't like to read manuals
 - They prefer to learn while making progress toward their goals
- Most users will stay at the intermediate level
 - Need reminders and a clear learning path
 - Need a quick way to access critical information
 - Online documentation and good search tool

Types of help (I)

- Tutorial and/or getting started manuals
 - Presents the system conceptual model
 - Basis for successful explorations
 - Provides on-line tours and demos
 - Demonstrates basic features
- Reference manuals
 - Designed with experts in mind
- Reminders
 - Short reference cards, keyboard templates, etc
- “Show me” videos



| | | | | | | | | |
|----|----|----|----|----|----|----|----|----|
| V | Cr | Mn | Fe | Co | Ni | Cu | Zn | Ga |
| Nb | Mo | Tc | Ru | Rh | Pd | Ag | Cd | In |
| Ta | W | Re | Os | Ir | Pt | Au | Hg | Tl |
| Db | Sg | Bh | Hs | Mt | | | | |

Types of help (II)

- Wizards

- Walks user through typical tasks
 - Users feel they are losing control
 - What if I do not have the information requested?



- Tips

- Migration path to learning new features
- Can become boring and tedious

Types of help (II)

- Context sensitive help

