



What the Digerati Know

INFO/CSE100, Spring 2006

Fluency in Information Technology

<http://www.cs.washington.edu/education/courses/100/06sp/>



Readings and References

- Reading
 - Fluency with Information Technology
 - » Chapter 2, What the Digerati Know
- References
 - W3Schools
 - » <http://www.w3schools.com>
 - Webmonkey
 - » <http://www.webmonkey.com/>
 - WebReference
 - » <http://www.webreference.com/>

Learning New Tools

- How do we learn to use new tools?
 - Be taught their use by someone else -- car, bicycle, etc...
 - Reading the owner's manual -- chain saw
 - Watching others -- vending machine
 - Figure them out ourselves -- CD player
- Software designers wanting you to learn the new tool ASAP, so they try for “intuitive” usage
 - Consistent Interfaces -- build in experience
 - Suggestive icons -- bypass terminology
 - Metaphors -- exploit analogous reasoning

Intuitive Design



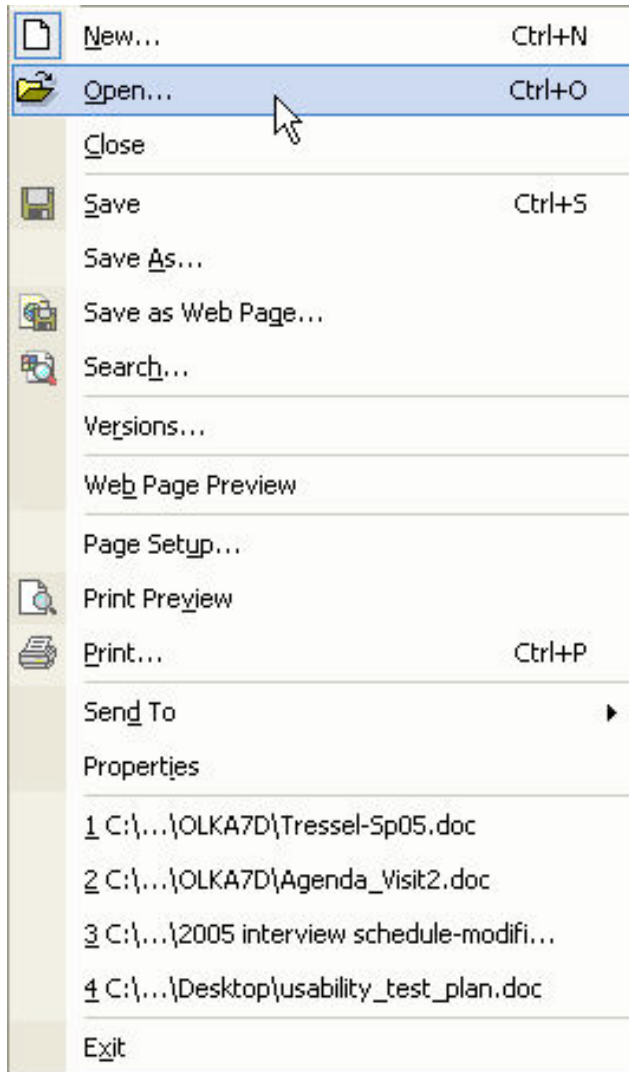
Mar-31-06

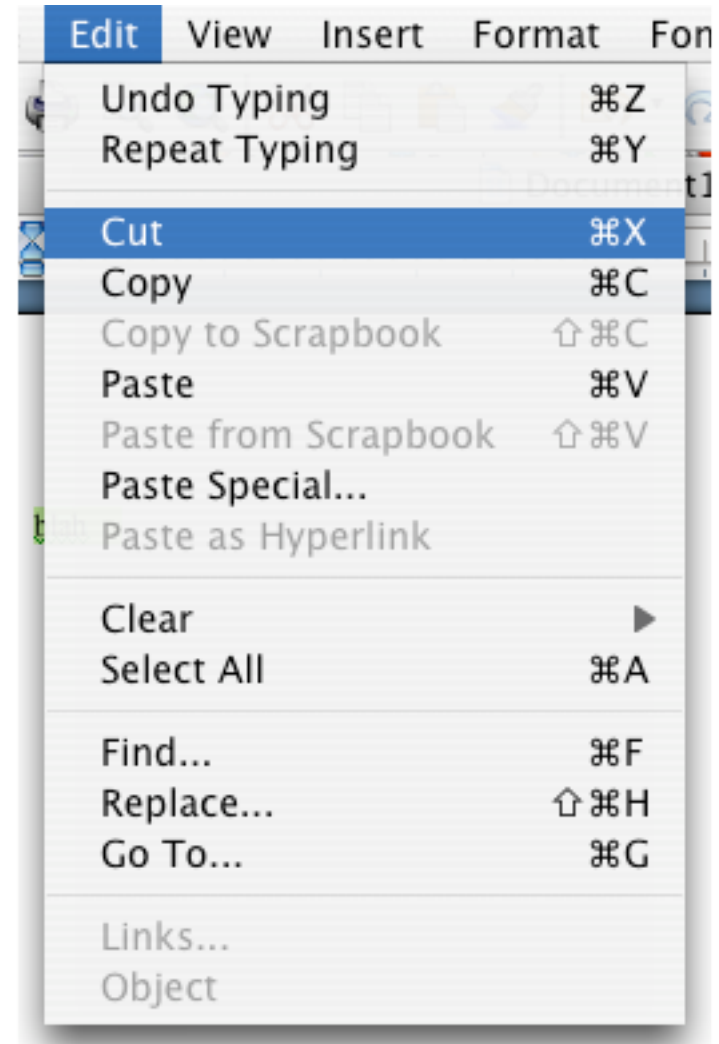
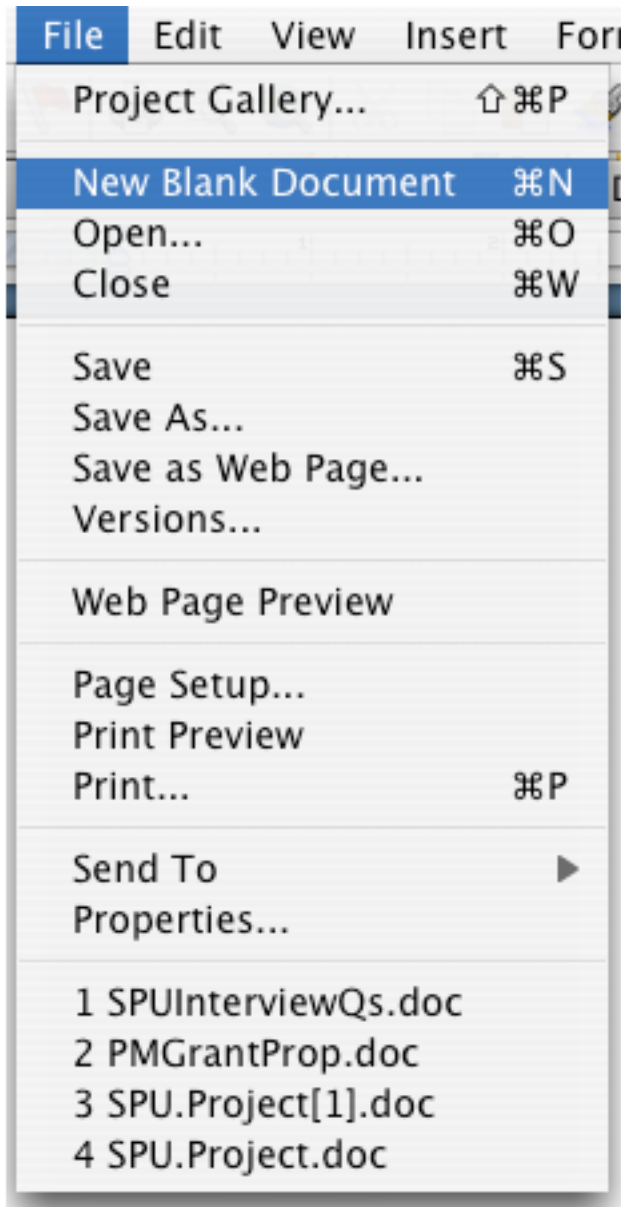
digerati @ university of washington



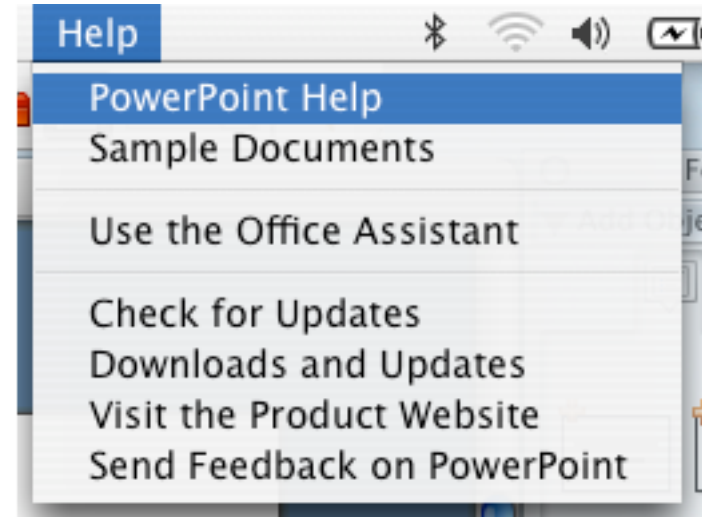
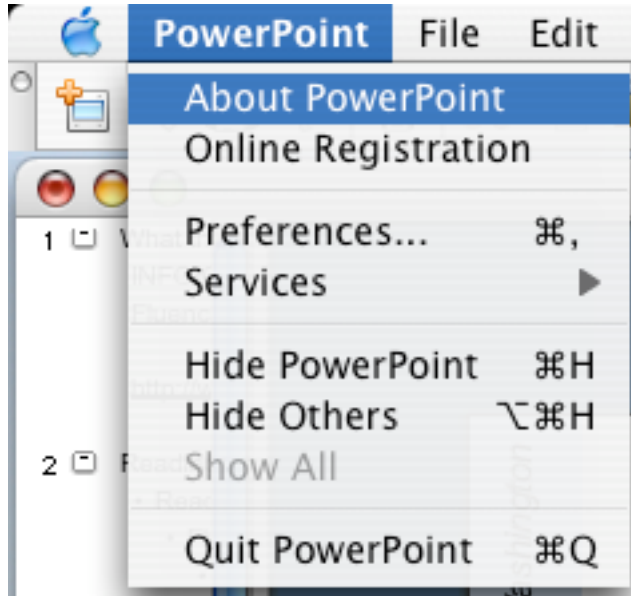
Consistent Interfaces

Most modern applications File and Edit menus with standard commands





Consistent Interfaces





Metaphors

- Some familiar computer metaphors
 - Desktop (folders, inboxes, etc...)
 - Place (rooms, hallways, doors)
 - Tree (root, branches, leaves)
 - Stoplight (green = go, red = stop)



Dance Dance Revolution



Mar-31-06

digerati @ university of washington

What does “New” Mean?

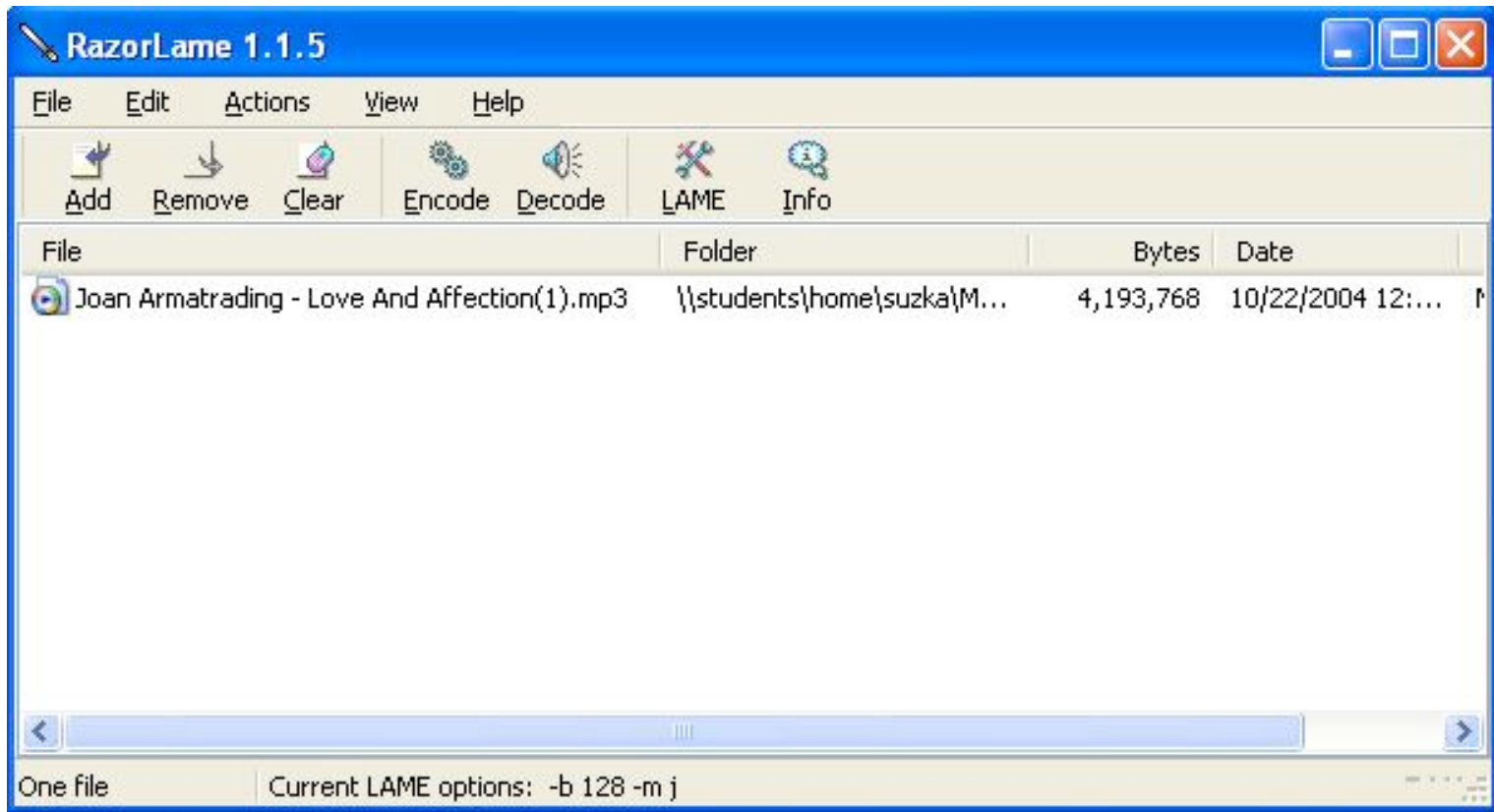
- Most applications have the concept of a *document* with:
 - Attributes: date created, date modified, creator, ...
 - Content: image, text, sound
- “New” means create a ‘blank instance’ of a document for this application
 - A document has attributes as well as content
 - » All stored in one file with a place for anything
 - » A ‘blank instance’ is simply the structure with some of the attributes filled in but without any of the content

“Click Around”

- Software designers use standard ideas to make applications intuitive
- To learn a new application, check it out by clicking around
 - Take a minute to ...
 - » Look under all menus to see operations
 - » Follow the “...” for menu operations
 - » Try to recognize what the icons mean
- Clicking around is Exploration! :)



A New Application

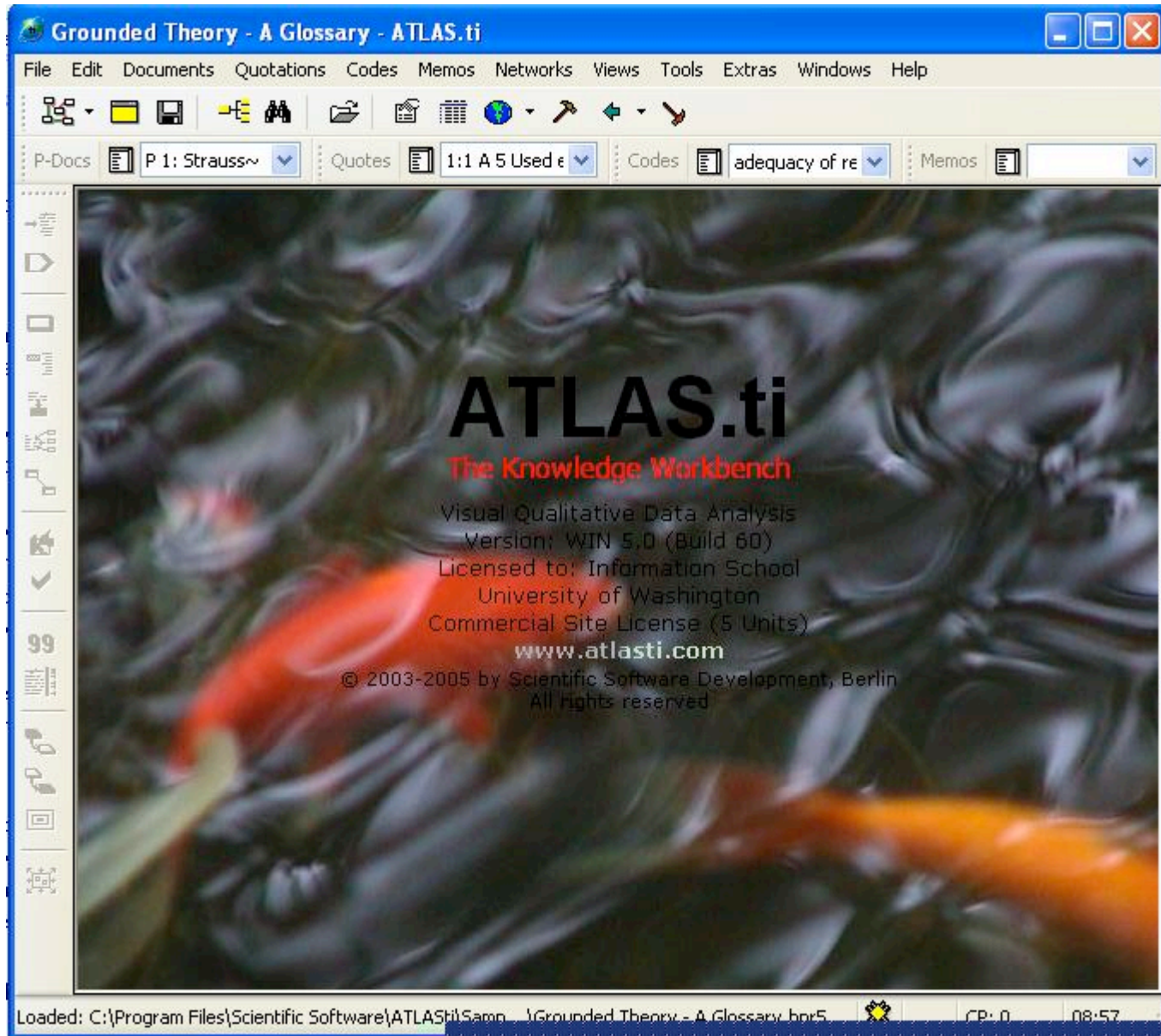




Mar-31-06

digerati @ university of washington

13



“Blazing Away”

- Learn an application fast by trying it!
- Beginning with a new instance, assertively try menu items
 - *Expect to fail* and make a mess
 - Exit the application and if you are asked to “Save?” reply “No”
 - Try repeatedly until becoming familiar



To Learn A New Tool

- Software systems build on a consistent interface, standard metaphors, etc...
 - Expect to teach yourself applications
 - Do so by familiarizing yourself with the features by “Clicking Around”
 - Assertively try out the features, ,”Blaze Away” observing what they do
 - » Be efficient -- stay focused, don't type a lot when you expect to exit
- You also might want to skim the user manual/help file/reference book
 - <http://www.oreilly.com/>



Online Tutorials

- Often provided by expert organizations
- May provide interactive tools, examples
- Good place to find references
- Might support a user community
- Things to consider:
 - Are they up-to-date?
 - Are they selling something?

Example Online Tutorials

- W3Schools
 - <http://www.w3schools.com/html/default.asp>
- Webmonkey
 - http://www.webmonkey.com/webmonkey/authoring/html_basics/
- WebReference
 - <http://www.webreference.com/authoring/languages/html/>

Differences & Similarities

- Different vendors will produce similar software for the same task
 - Superficially, the GUIs use similar features
 - Fundamentally, the task largely determines how the software must work... they *must* be similar
- Implications
 - If you know one word processor, you can learn others fast
 - Software differences: mostly glitz and convenience
 - Don't accept lousy tools, consider switching to other applications

Another Implication

- If applications are similar at their core, techniques can be learned without learning a specific vendor's software
 - » Example: programming languages



Mac or PC???

- Arguments about which is better, Mac or PC create only create heat, no light
 - They are more alike than different
 - Any fluent person can use both
- I first bought a PowerMac, then a PC, then a Windows laptop, and now this Mac PowerBook
 - And I've owned a Sun Sparc 1+ Unix workstation

Summarizing

- Humans must learn to use tools
 - Software designers want you to learn easily
 - » Good software uses consistent interfaces and menus
- Teach yourself applications by “Clicking Around” and “Blazing Away”
 - I don’t know all the features, you don’t know all the features, and that’s okay!
- Software for a task must share core features
- Learn applications independent of vendors



Homework 2

- Print out homework 2
- You will have to complete some tasks on 2 different computer systems (Mac, Windows, or Linux)
- Due in class on Friday, April 7th