

# Usability Principles

John Stasko

Spring 2007

This material has been developed by Georgia Tech HCI faculty, and continues to evolve. Contributors include Gregory Abowd, Al Badre, Jim Foley, Elizabeth Mynatt, Jeff Pierce, Colin Potts, Chris Shaw, John Stasko, and Bruce Walker. Permission is granted to use with acknowledgement for non-profit purposes. Last revision: January 2007.

## Agenda

- Usability Principles
  - Why?
  - System of principles
    - Learnability
      - Support for learning for users of all levels
    - Flexibility
      - Support for multiple ways of doing tasks
    - Robustness
      - Support for recovery
  - Style guides
- Project preparation



## Good Design (our goal!)

“Every designer wants to build a high-quality interactive system that is admired by colleagues, celebrated by users, circulated widely, and imitated frequently.” (Shneiderman, 1992, p.7)

*...and anything goes!...*



## Why Principles & Guidelines?

- ...Because, well, not everything goes...
- Intended to prevent many bad designs, before they begin, or evaluate existing designs on a scientific basis
- Guidelines based on previous designs, experimental findings
- Rules can all be “broken” (but usually in order to satisfy another principle)



## Concepts, Principles, Guidelines

- No “cookbooks”
- No simple, universal checklists
- There are many concepts, principles, and guidelines
- **Understand** the higher level principles that apply across situations, display types, etc.
- **Implement** the standards and guidelines

*...a few details...*



## Many Sets of Design Principles

- Shneiderman, *Designing the User Interface*
- Dix, Finlay, Abowd, Beale, *Human-Computer Interaction*
- Foley et al, *Computer Graphics: Principles and Practice*
- And many more - including in styleguides, discussed later



## Levels of Consideration

### 1. Meta-display level

- Apply to the whole system, across media & across displays
- Focus on this in Basic Layout Stage

### 2. Display Layout

- Apply to groups of elements in a display
- Focus on this in Prototyping and Redesign

### 3. Element level

- Details about specific parts of a display
- Colors, sound attributes, symbols



## UI Design Principles (Dix et al)

- Categories
  - Learnability
    - Support for learning for users of all levels
  - Flexibility
    - Support for multiple ways of doing tasks
  - Robustness
    - Support for recovery
- Always think about these in terms of meta-display, display, and element levels



# 1. Learnability Principles

- Ease with which new users can begin effective interaction and achieve maximal performance
  - Predictability
  - Synthesizability
  - Familiarity
  - Generalizability
  - Consistency



## 1.1 Predictability

- I think that this action will do....

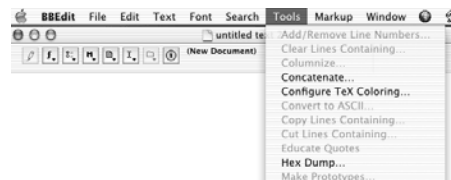
Done

vs.

Submit data, Go to Step 2

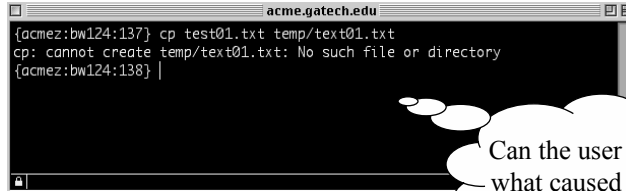


- Operation visibility - Can see avail actions
  - e.g. menus vs. command shell
  - grayed menu items



## 1.2 Synthesizability

- Support for user in assessing the effect of past operations on current system state



```
acme.gatech.edu
{acmez:bw124:137} cp test01.txt temp/text01.txt
cp: cannot create temp/text01.txt: No such file or directory
{acmez:bw124:138} |
```

Can the user figure out what caused this error?

- Moving a file in UNIX shell vs. Mac/Windows
- Is same feedback needed for all users, all apps?



## 1.3 Familiarity

- Does UI task leverage existing real-world or domain knowledge?
  - Really relevant to first impressions
  - Use of metaphors
    - Potential pitfalls
  - Are there limitations on familiarity?



## Metaphors at the UI - What

- *Metaphor* - Application of name or descriptive term to another object which is not literally applicable
  - Use: Natural transfer - apply existing knowledge to new, abstract tasks
  - Problem: May introduce incorrect mental model



## 1.4 Generalizability

- Can knowledge of one system/UI be extended to other similar ones?
  - Example: cut & paste in different applications
  - Does knowledge of one aspect of a UI apply to rest of the UI?
  - Aid: UI Developers guidelines



## 1.5 Consistency

- Likeness in behavior between similar tasks/operations/situations
  - In different things
    - interacting
    - output
    - screen layout
- Is this always desirable for all systems, all users?



## (In)Consistency Example - Macintosh

Drag a file icon to:	Result:
Folder on same physical disk	→ File is moved to folder
Folder on another physical disk	→ File is copied there
Different disk	→ File is copied there
Trash can	→ File is discarded





## 2. Flexibility Principles

- Multiplicity of ways that users and system exchange information
  - Dialog Initiative
  - Multithreading
  - Task migratability
  - Substitutivity
  - Customizability



### 2.1 Dialog Initiative

- Not hampering the user by placing constraints on how dialog is done
  - User pre-emptive
    - User initiates actions
    - More flexible, generally more desirable
  - System pre-emptive
    - System does all prompts, user responds
    - Sometimes necessary



## 2.2 Multithreading

- Allowing user to perform more than one task at a time
- Two types
  - Concurrent
    - Input to multiple tasks simultaneously
  - Interleaved
    - Many tasks, but input to one at a time



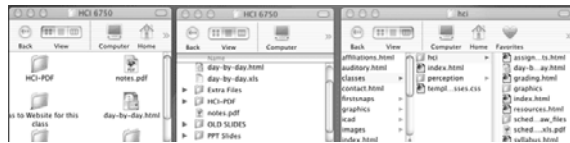
## 2.3 Task Migratability

- Ability to move performance of task to entity (user or system) who can do it better
  - Auto-pilot in planes
  - Spell-checking
  - Safety controls in plant
- For what kinds of tasks should the user be in control?



## 2.4 Substitutivity

- Flexibility in details of operations
  - Allow user to choose suitable interaction methods
  - Allow different ways to
    - perform actions, specify data, configure
  - Allow different ways of presenting output
    - to suit task & user



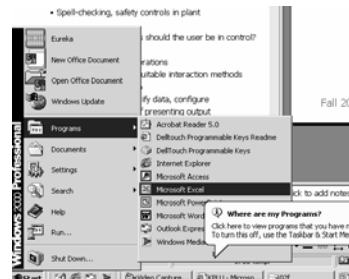
6750-Spr '07



21

## 2.5 Customizability

- Ability of user to modify interface
  - By user - adaptability
    - Is this a good thing?
  - By system - adaptivity
    - Is this a good thing?



6750-Spr '07



22

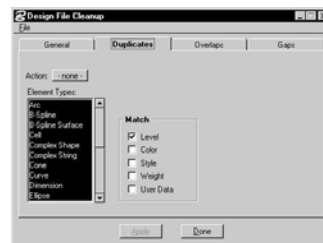
### 3. Robustness Principles

- Supporting user in determining successful achievement and assessment of goals
  - Observability
  - Recoverability
  - Responsiveness
  - Task Conformance



### 3.1 Observability

- Can user determine internal state of system from what she perceives?
  - Browsability
    - Explore current state (without changing it)
  - Reachability
    - Navigate through observable states
  - Persistence
    - How long does observable state persist?



## Observability - Role of Feedback

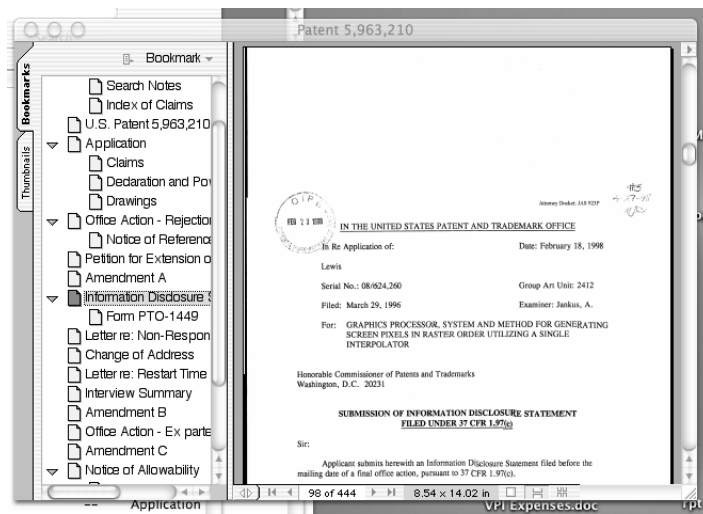
- Feedback helps create observability
- Feedback taxonomy (generally don't need all of these)
  - “I understand what you have asked me to do”
  - “I am doing what you have asked me to do”
    - “And it will take me this much longer”
    - Song and dance routine to distract user (busy interval as opposed to idle interval)
    - “And here are some intermediate results to keep you happy until I am done
  - “All done, what’s next?”



## Observability – Acrobat Reader

Acrobat Reader  
with ToC to  
give context

Forest is the  
bookmarks,  
tree is the  
single page



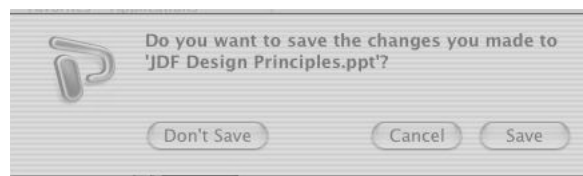
## 3.2 Recoverability

- Ability to take corrective action upon recognizing error
  - Difficulty of recovery procedure should relate to difficulty of original task
  - Forward recovery
    - Ability to fix when we can't undo
  - Backward recovery
    - Undo previous error(s)



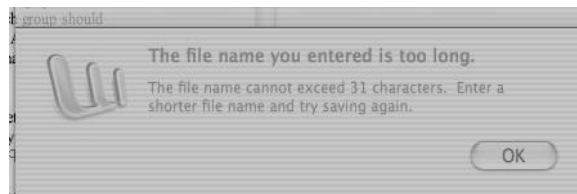
## Do Not Set the User Up

- Make it hard for the user to make errors
  - Instead of allowing them to make error and then saying "tsk, tsk"
- Gray out disabled menu items
- Ask for confirmation of major actions



## Do Not Set the User Up

- Don't let the user do something that will lead to an error message



## 3.3 Responsiveness

- Users perception of rate of communication with system
  - Response time
    - Time for system to respond in some way to user action(s)
  - Users perceptions not always right
  - Response OK if matches user expectations
  - Once user enjoys fast response, is hard to go back to slower one
    - Dial-up versus DSL/Cable modem



## Responsiveness

- Response to motor actions
  - Keyboarding, mouse movement – less than 100 msec
  - Rich human factors literature on this
- Consistency is important – experimental results
  - Users preferred longer but more consistent response time
  - Times that differed 10%-20% were seen as same
- Sometimes argued that too fast is not good
  - Makes user feel like they need to do something quickly to keep up with computer



## 3.4 Task Conformance

- Does system support all tasks user wishes to perform in expected ways?
  - Task completeness
    - Can system do all tasks of interest?
  - Task adequacy
    - Can user understand how to do tasks?
- Does it allow user to define new tasks?





## Application

- In doing design and implementation of your project, revisit this list
- Assess your design against these usability principles



## Styleguides

- Codify many of these principles for a particular look and feel
  - Mac OS, Windows, Motif, Palm, Blackberry
- Developed in concert with toolkit, but go beyond toolkit



# Typical TOC - MAC OS X

<u>Introduction to the Apple Human Interface Guidelines</u> What Are the Mac OS X Human Interface Guidelines? Who Should Read This Document? Organization of This Document Conventions Used in This Document See Also	Using Existing Technologies Providing User Assistance Internationalizing Your Application Storing Passwords Printing Choosing Colors Setting Fonts and Typography Characteristics Selecting Attributes Associated With People Speech Technologies
<b>Part I: Fundamentals</b> <b>Human Interface Design</b> Human Interface Design Principles Keep Your Users in Mind <b>The Development Process</b> Design Decisions Managing Complexity Extending the Interface Involving Users in the Design Process <b>Part II: The Macintosh Experience</b> <b>First Impressions</b> Packaging Installation General Installer Guidelines Setup Assistants <b>Mac OS X Environment</b> The Finder The Dock The File System Multiple Users Remote Log In Assistive Technologies Networking Application Services Displays The Always-On Environment	<b>Part III: The Aqua Interface</b> <b>User Input</b> The Mouse and Other Pointing Devices The Keyboard Selecting Editing Text <b>Drag and Drop</b> Drag and Drop Overview Drag and Drop Semantics Selection Feedback Drag Feedback Destination Feedback Drop Feedback Clippings <b>Text</b> Fonts Style <b>Icons</b> Icon Genres and Families Icon Perspectives and Materials Conveying an Emotional Quality in Icons Suggested Process for Creating Aqua Icons Tips for Designing Aqua Icons <b>Cursors</b> Standard Cursors Designing Your Own Cursors



# More TOC

<b>Menus</b> Menu Behavior Designing the Elements of Menus The Menu Bar and Its Menus Contextual Menus Dock Menus <b>Windows</b> Types of Windows Window Appearance Window Behavior Utility Windows The About Window Preferences Windows Inspectors and Info Windows Find Window Fonts Window and Colors Window <b>Dialogs</b> Types of Dialogs and When to Use Them Dialog Behavior The Open Dialog Dialogs for Saving, Closing, and Quitting The Choose Dialog The Printing Dialogs <b>Controls</b> Buttons Selection Controls Adjustment Controls Indicators Text Controls View Controls Grouping Controls	<b>Layout Examples</b> Positioning Controls Sample Layouts Grouping Controls Using Small and Mini Versions of Controls <b>Keyboard Shortcuts Quick Reference</b> <b>Tab View Differences Between Mac OS X Versions</b> <b>Document Revision History</b>
---	--



# Excerpt from OS X Styleguide

## Drag and Drop Overview

Ideally, users should be able to drag any content from any window to any other window that accepts the content's type. If the source and destination are not visible at the same time, the user can create a **clipping** by dragging data to a Finder window; the clipping can then be dragged into another application window at another time.

Drag and drop should be considered an ease-of-use technique. Except in cases where drag and drop is so intrinsic to an application that no suitable alternative methods exist—dragging icons in the Finder, for example—there should always be another method for accomplishing a drag-and-drop task.

The basic steps of the drag-and-drop interaction model parallel a copy-and-paste sequence in which you select an item, choose Copy from the Edit menu, specify a destination, and then choose Paste. However, drag and drop is a distinct technique in itself and does not use the Clipboard. Users can take advantage of both the Clipboard and drag and drop without side effects from each other.

A drag-and-drop operation should provide immediate feedback at the significant points: when the data is selected, during the drag, when an appropriate destination is reached, and when the data is dropped. The data that is pasted should be target-specific. For example, if a user drags an Address Book entry to the "To" text field in Mail, only the email address is pasted, not all of the person's address information.

You should implement Undo for any drag-and-drop operation you enable in your application. If you implement a drag-and-drop operation that is not undoable, display a confirmation dialog before implementing the drop. A confirmation dialog appears, for example, when the user attempts to drop an icon into a write-only drop box on a shared volume, because the user does not have privileges to open the drop box and undo the action.

(Color added for emphasis.)



# Styleguides

- **General User Interface Design Style Guides**
  - Apple Human Interface Guidelines (Mac OS X) [Design Guidelines](#)
  - [Microsoft User Interface Guidelines](#) (Click in the left tree on User Interface Design...)
  - [Windows XP Guidelines](#)
  - Yale [Web Style Guide \(2nd Edition\)](#)
  - Java Look and Feel [Guidelines](#) (version 1)
  - Java Look and Feel [Guidelines version 2](#)
  - Java Look and Feel [Guidelines: Advanced Topics](#)
  - IBM 3D design [Guidelines](#)
  - Silicon Graphics Indigo Magic [User Interface Guidelines](#)
- **Open Source Usability Guidelines**
  - [Motif Style Guide](#)
  - KDE [User Interface Guidelines](#)
  - [Gnome Human Interface Guidelines 1.0](#)
- **Corporate User Interface Standards and Guidelines (samples)**
  - Telstra [Online Standards](#)
  - Taligent [Human Interface Guidelines](#)
  - Ameritech Graphical [User Interface Standards and Design Guidelines](#)

- [http://www.experiencedynamics.com/science\\_of\\_usability/ui\\_style\\_guides/](http://www.experiencedynamics.com/science_of_usability/ui_style_guides/)



## And More Styleguides ....

- **Government funded Usability Guidelines**
  - MITRE Guidelines for [Designing User Interface Software](#) (US Airforce)
  - Research based [Web Design and Usability Guidelines](#) (Dept. of Health and Human Services)
  - Cancer Institute [Usability Guidelines](#)
  - NASA [User Interface Guidelines](#)
  - Canadian Command Decision Aiding Technology (COMDAT) Operator-Machine Interface (OMI) [Style Guide: Version 1.0](#)
- **Gaming Devices (J2ME games)**
  - [Games Usability Guidelines](#) (from Nokia)
- **Wireless and Mobile Usability Guidelines**
  - Palm OS [Design Guidelines](#)
  - Openwave GSM [Guidelines](#)
  - Openwave [Top 10 Usability Guidelines for WAP Applications](#)
  - Blackberry and RIM [wireless handheld UI Developers Guide](#) (PDF)
  - Sprint [Usability Requirements for XHTML](#) (Application Developers Program)
  - NTT DoCoMo [imode service guideline](#) (user interfaces)
- **Accessibility Guidelines**
  - Techniques for [Web content Accessibility Guidelines 1.0](#)



## Project

- Anyone without a team yet?
  - You need to find one!!!
- Interesting topics?



## Upcoming

- Human Capabilities
  - Physical
  - Cognitive
- Project team & topic due Thursday

