

Lecture 6 – Usability

Terry Winograd

CS147 - Introduction to Human-Computer
Interaction Design

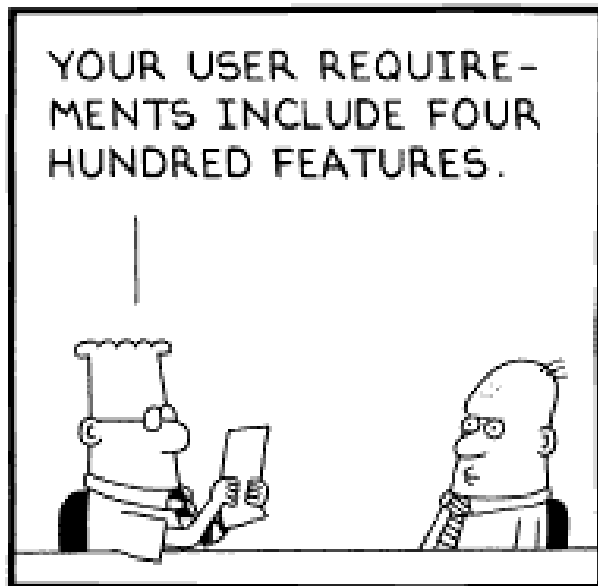
Computer Science Department

Stanford University

Autumn 2006

Learning Goals

- Understand when and how design tradeoffs are made
- Have a first-level knowledge of the major concepts in usability and their associated guidelines
- Be able to effectively apply principles, design rules, usability guidelines, heuristics, patterns, etc.



www.dilbert.com scottadams@aol.com



4/17/01 © 2001 United Feature Syndicate, Inc.



Copyright © 2001 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

Usability Goals (ID 1.5)

- Effective
- Efficient
- Safe
- Useful
- Learnable
- Memorable

These can be in conflict with one another

User Experience Goals (ID 1.5)

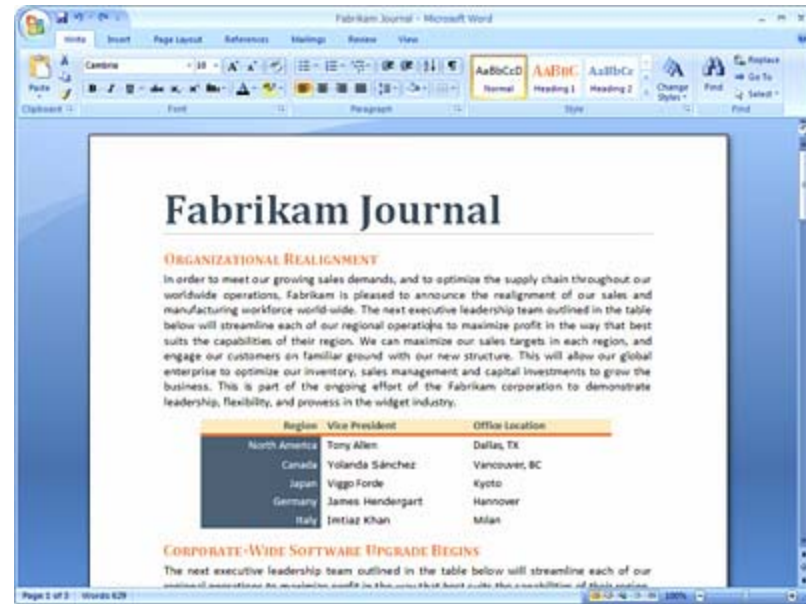
- Satisfying
- Enjoyable
- Fun
- Entertaining
- Helpful
- Motivating
- Aesthetically pleasing
- Supportive of creativity
- Rewarding
- Emotionally fulfilling

These can be hard to evaluate

Components of Usability (adapted from Nielsen and *Interaction Design* Section 5.1)

- Guessability
- Learnability
- Retention
- Efficiency
- Error protection
- Experienced User Performance
- Supportability in an environment
- Transfer of skills
- Satisfaction

Example: Microsoft Office 2007



Components of Usability (adapted from Nielsen and *Interaction Design* Section 5.1)

- Guessability
- Learnability
- Retention
- Efficiency
- Error protection
- Experienced User Performance
- Supportability in an environment
- Transfer of skills
- Satisfaction

Principles Affecting Learnability [Dix]

- Predictability
 - User-centered – Perceived affordance
- Synthesizability
 - Visibility
- Familiarity
 - Metaphors
- Generalizability
 - Uniform model
- Consistency

Heuristics

"Eight Golden Rules of Interface Design" [Shneiderman]

1. Strive for consistency.
2. Enable frequent users to use shortcuts.
3. Offer informative feedback.
4. Design dialog to yield closure.
5. Offer simple error handling.
6. Permit easy reversal of actions.
7. Support internal locus of control.
8. Reduce short-term memory load.

10 Usability Heuristics [Nielsen]

- Simple and Natural Dialogue
- Speak the Users' Language
- Minimize User Memory Load
- Consistency
- Feedback
- Clearly Marked Exits
- Shortcuts
- Good Error Messages
- Prevent Errors
- Help and Documentation

Principles for Transforming Difficult Tasks into Simple Ones [Norman]

1. Use both knowledge in the world and knowledge in the head
2. Simplify the structure of tasks
3. Make things visible
4. Get the mappings right
5. Exploit the power of constraints
6. Design for error
7. When all else fails, standardize

So Many Lists, So Little Time.....

- How do you use design rules?
 - Restrict the space of design options
 - Anticipate what will work well
 - Avoid and/or repair usability problems
 - Understand users' difficulties
 - Recognize underlying reasons

Design Rules can often be most useful when they give you questions rather than answers

Kinds of Rules

- Principles
 - Largely independent of technology
- Guidelines
 - More technology oriented but still general
- Standards
 - Explicit and specific
- Heuristics
 - Broad-brush design rules
- Patterns
 - Best practices

The differences among these is blurry

Guidelines

- Distilled experience/wisdom
 - Often based on past mistakes
- Can be at different levels of detail
 - Some are very specific to technologies

Smith and Mosier Guidelines (1986)

•36

▶ **Blinking Marker Symbols**

When a user must read a displayed item that is blink coded, consider adding an extra symbol such as an asterisk to mark the item, and then blinking that marker symbol rather than blinking the item itself.

COMMENT: This practice will draw attention to an item without detracting from its legibility.

REFERENCE: BB 1.10.3; Smith and Goodwin, 1971b.

•37

▶ **Optimal Blink Rate**

When blink coding is used, select a blink rate in the range from 2 to 5 Hz, with a minimum duty cycle (ON interval) of 50 percent.

COMMENT: Although equal ON and OFF intervals are often specified, an effective code can probably be provided even when the OFF interval is considerably shorter than the ON (a “wink” rather than a blink), as in occulting lights used for Navy signaling.

REFERENCE: BB 1.10.4; MS 5.15.3.3.2.

Standards

- Ergonomic standards
 - Physical properties of humans
 - May be legislated
- Interaction standards
 - Can be enforced and are important for products that are part of a larger family of products
 - Examples
 - Noun verb operation
 - Consistent menus

File Menu

The **File menu** provides commands that pertain to housekeeping tasks for documents. It also contains the Quit command. All of the standard operations are described here. If you add additional commands to the File menu, be sure that they fit the category of taking care of documents. Figure 4-61 shows a sample File menu.

Figure 4-61 A File menu

| File | |
|---------------|----|
| New | ⌘N |
| Open | ⌘O |
| Close | ⌘W |
| Save | ⌘S |
| Save As... | |
| Revert | |
| Page Setup... | |
| Print... | ⌘P |
| Quit | ⌘Q |

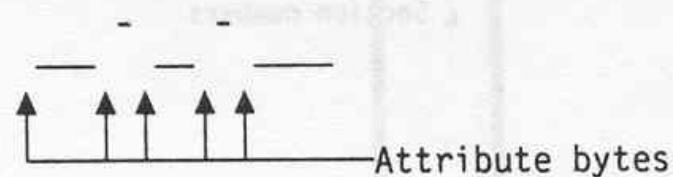
IBM Common User Access

Nonprogrammable Terminals

You may use either of two methods to create multi-part entry fields:

- Use a protected field, consisting of two attribute bytes and a separator character, to separate individual parts of a multi-part entry field. For example:

Employee number



- Use only attribute bytes, not separator characters, to separate individual entry fields in a multi-part entry field. The attribute bytes appear blank.
- The attribute byte for each part should cause automatic tabbing at the end of that part to the next part.

Figure 22. Multiple-choice listfield field interaction. All of the choices in this multiple-

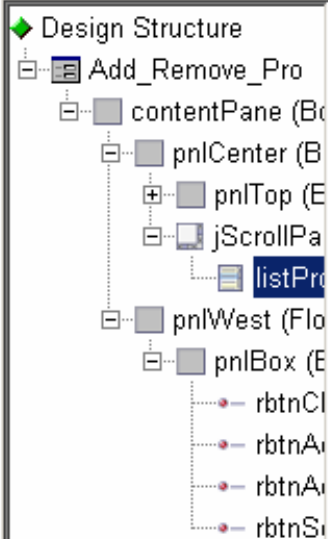
Standards and Guidelines Embedded in Toolkits

- Macintosh Toolbox
- Open Look, Windows, ...
- Java Swing, AWT, ...
- TCL/TK, Prefuse,.....

*Toolkits provide an API
(Applications Programming Interface) that
gives the programmer a wide range of
presentation and control at a high level of
abstraction*

Java Swing Interface

Swing | Swing Containers



Add or Remove Programs

Currently installed programs: Sort by: Name

- Adobe Acrobat 5.0
- Adobe Photoshop 7.0
- Advanced Networking Pack**
- Advanced Control Suite
- Borland JBuilder 9.0
- DirectX 9
- Java 2 Runtime Environment, SE v 1.4.2
- Java 2 Runtime Environment, SE v 1.4.2_03
- Java 2 SDK, SE v 1.4.2
- Java 2 SDK, SE v 1.4.2_03
- JFrameBuilder 3.0.0
- Macromedia Dreamweaver MX
- Macromedia Extension Manager
- Macromedia Fireworks MX
- Macromedia Flash MX
- Media Manager
- Microsoft .NET Framework 1.1
- Microsoft Office XP

Buttons: Change, Add New, Add/Remove, Set Program

Design

| | |
|-----------|---------------------|
| BaseClass | JList |
| (Name) | listPrograms |

Appearance

| | |
|------------------|-----------|
| Background | (Default) |
| Foreground | (Default) |
| Opaque | true |
| SelectionBack... | (Default) |
| SelectionFore... | (Default) |
| ToolTipText | |

Behavior

| | |
|-----------------|-------|
| DragEnabled | false |
| Enabled | true |
| FixedCellHeight | -1 |

Items Editor

SelectionBackground
(java.awt.Color) The background color of selected cells.

Buttons: Properties, Events

Macintosh Carbon

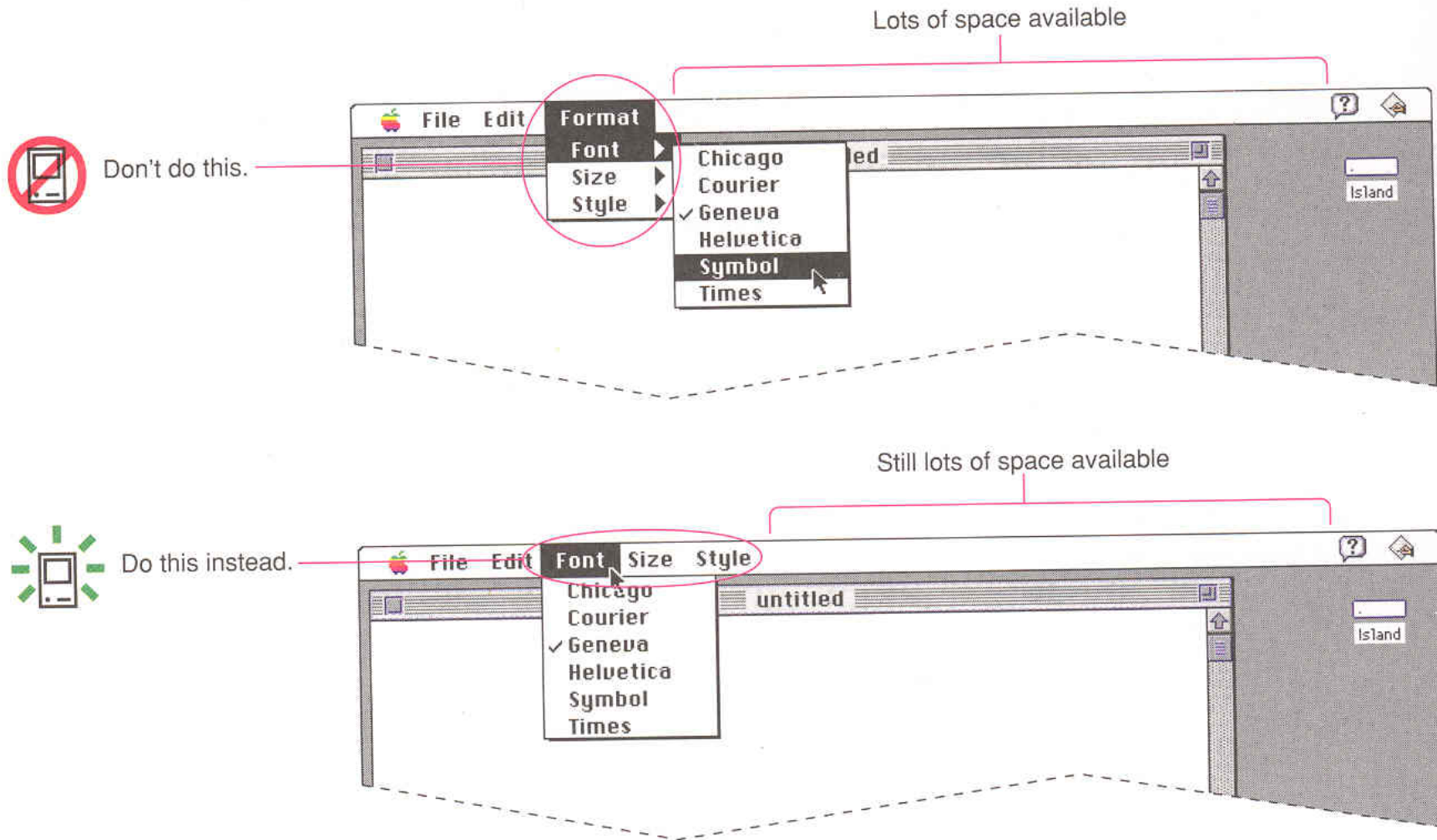


10 Usability Heuristics [Nielsen]

- Simple and Natural Dialogue
- Speak the Users' Language
- Minimize User Memory Load
- Consistency
- Feedback
- Clearly Marked Exits
- Shortcuts
- Good Error Messages
- Prevent Errors
- Help and Documentation

Apple Macintosh Guidelines

Figure 4-37 Don't use submenus unnecessarily



Patterns

- Inspired by architect Christopher Alexander's *Pattern Language*
- Distill common best practices
- Apply best at early design stages to inspire designs

Alexander's Pattern Language

180 WINDOW PLACE

... this pattern helps complete the arrangement of the windows given by ENTRANCE ROOM (130), ZEN VIEW (134), LIGHT ON TWO SIDES OF EVERY ROOM (159), STREET WINDOWS (164)...
...



Everybody loves window seats, bay windows, and big windows with low sills and comfortable chairs drawn up them.

... These kinds of windows which create "places" next to them are not simply luxuries; they are *necessary*. A room which does not have a place like this seldom allows you to feel fully comfortable or perfectly at ease. Indeed, a room without a window place may keep you in a state of perpetual unresolved conflict and tension -- slight, perhaps, but definite.

When can you use design rules?

- **Patterns** are useful at early design stages
- More specific **guidelines** and **standards** are applied at detailed design and implementation stages
- **Principles** are useful to analyze breakdowns and suggest changes during prototyping and testing, and to decide tradeoffs among heuristics and guidelines

Design Tradeoffs

- How can dimensions be evaluated together?
 - Only defined in context of users and tasks
 - Require clear consensus on priorities
- Can require violating one to satisfy another
 - e.g., Consistency/efficiency [Grudin]
- Need to understand the principles behind them
 - Level of “authority”

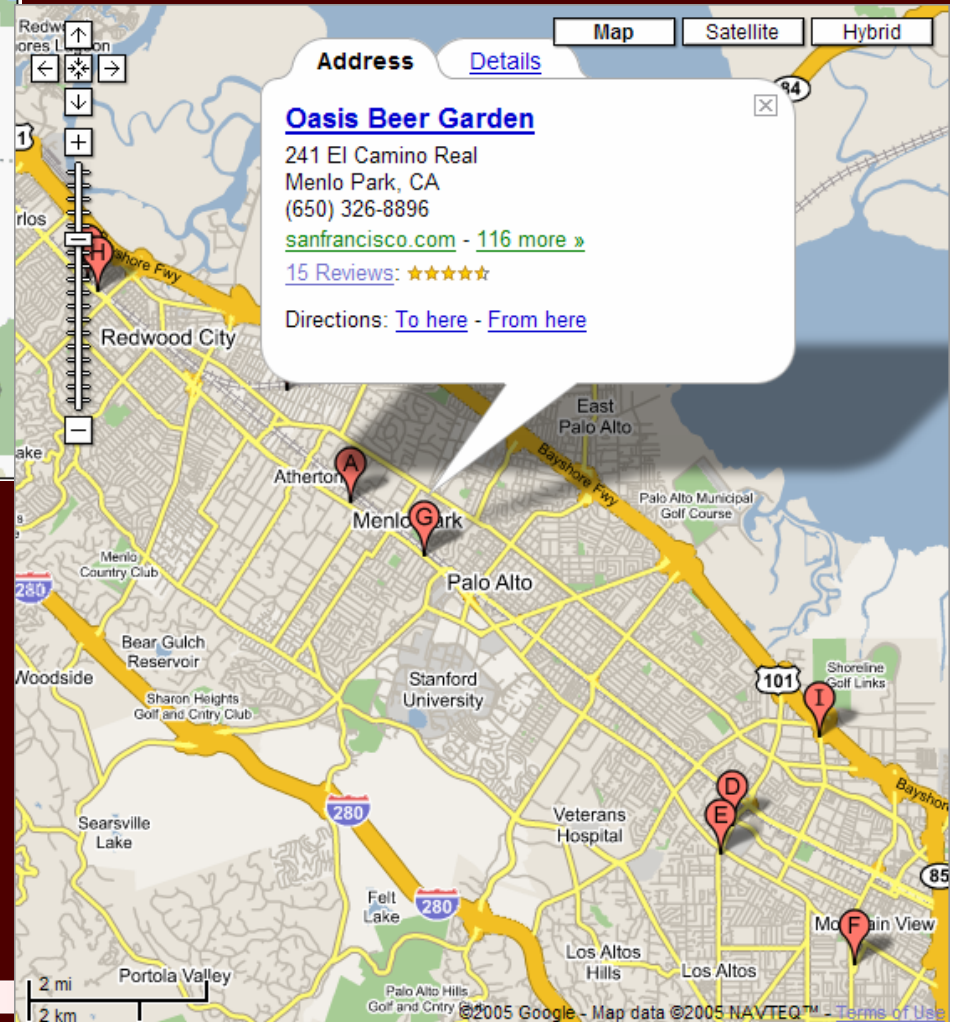
The Whole User Experience

- Context/ecology of this design within the larger space
- Style
 - e.g., Apple
 - e.g., Google
- Larger user context

Example: Apple Visual Design



Example: Google Maps



User Context: Out of the Box Experience



Good design meets the Real World

- Design organizations and their cultures
- Different assumptions about users and goals
- History and legacy
- Conflicting priorities
- External constraints and regulations

Don't be so quick to assume that "bad" designs are the result of ignorance or stupidity