3   Basic HCI Principles and Models

Principles for User Interface (UI) design

- Implementation and technology independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)

- Ben Shneiderman's list of principles
  - **Principle 1 : Recognize User Diversity**
  - **Principle 2 : Follow the 8 Golden Rules**
  - **Principle 3 : Prevent Errors**

http://media.pearsoncmg.com/aw/aw_shneiderma_dtui_4/chapter2.pdf

Principles for User Interface (UI) design

- Implementation and technology independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)

- Ben Shneiderman's list of principles
  - **Principle 1 : Recognize User Diversity**
  - Principle 2 : Follow the 8 Golden Rules
  - Principle 3 : Prevent Errors

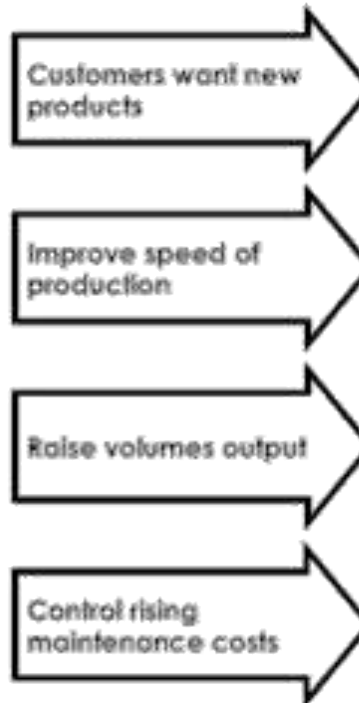http://media.pearsoncmg.com/aw/aw_shneiderma_dtui_4/chapter2.pdf

Principle 1: Recognize User Diversity

- Obvious and simple (?) - nevertheless in reality extremely difficult

- To be done *before* the design

- Basic concepts to structure the problem

    - Usage profiles

        » Different types of users

        » Different types of usage scenarios

        » Dependent on the situation of the user

    - Task profiles

        » What is the goal of the user?

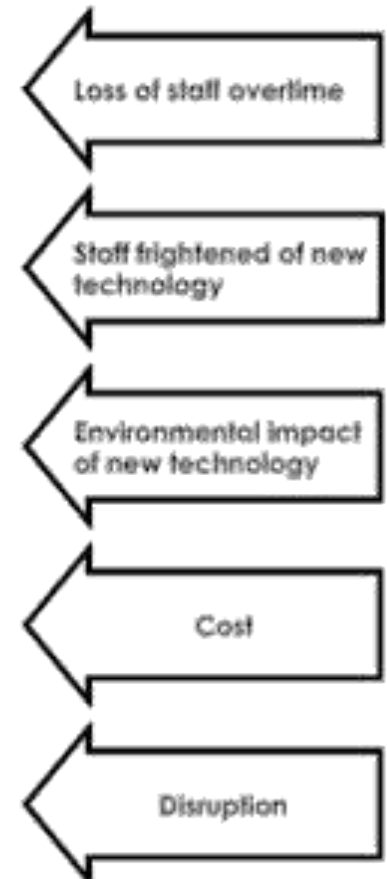   » How does the user want to achieve the goal?

Usage Profiles – Stakeholders

- Possible stakeholders
  - Shareholders
  - Government
  - Senior executives
  - Your coworkers
  - Suppliers
  - The press
  - Interest groups
  - Customers
  - Analysts
  - The public
  - The community
  - Your family

Forces for Change

Customers want new products

Improve speed of production

Raise volumes output

Control rising maintenance costs

Plan: Upgrade factory with new manufacturing equipment

Forces against Change

Loss of staff overtime

Staff frightened of new technology

Environmental impact of new technology
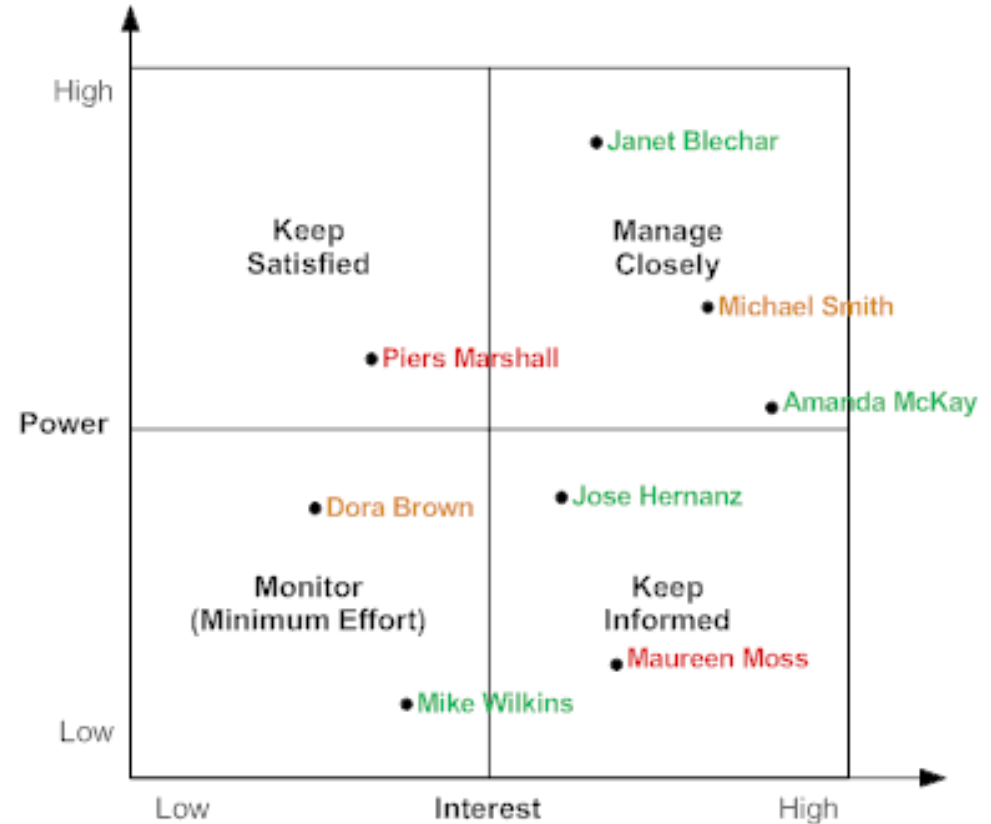
Cost

Disruption

Force-field Analysis

http://www.mindtools.com

Usage Profiles – Approach

- Identify stakeholders
  - Brainstorming
  - Review past projects
  - Interviews

- Categorise stakeholders
  - Amount of interest
  - Amount of influence
  - Positive / negative attitude
  - Reasons for attitude

- Draw a force-field analysis and keep it in mind throughout the project

Power vs. interest grid

http://www.mindtools.com

Usage Profiles – More than the People

- "Know thy user"
  (Wilfred J. Hansen, User Engineering Principles for Interactive Systems, 1971)

- Starting point for design: what is the background of the user?
  - Different people have different requirements for their interaction

- Complex multi-dimensional classification problem!

- Issues to be taken into account
  - Goals, motivation, personality
  - Education, cultural background, training
  - Age, gender, physical abilities, …
  - Multiple user communities, various combinations of background

- Well-known and frequently used classification
  - Novice users
  - Knowledgeable intermittent users
  - Expert frequent users

Task Profiles

- The goal: find out what the user is trying to do!
  - Needs of users, goals and resulting tasks

- Supported tasks should be determined before the design starts
  - Determine granularity of atomic tasks: flexibility vs. ease of use

- Functionality should *only* be added if identified to help solving tasks
  - Temptation: ~~add unneeded functionality only because it is "cheap" to achieve~~!

- Frequency of actions (relative to user profiles) leads to design choices
  - The more frequent an action, the easier its invocation
  - Example:
    » very frequent actions invoked by special keys (e.g. DEL)
    » intermediately frequent actions invoked by keyboard shortcut, special button, …
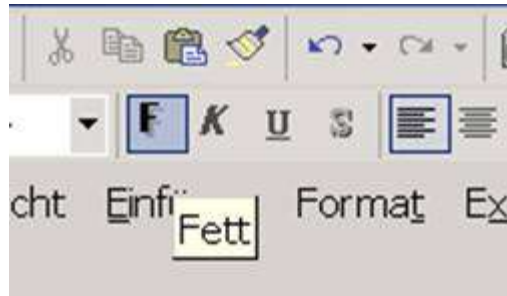   » infrequent actions invoked through menu selections, form fillings, …

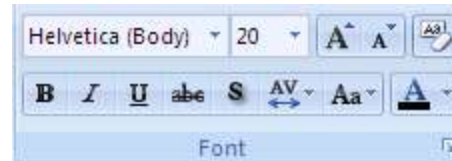# Hypothetical Frequency of Tasks
(Example: a travel booking system)

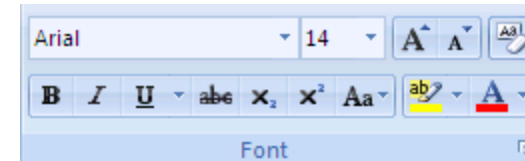| *Task* *Position*	Group reservation	Change of itinerary	Booking child care	Comparing sales agent performance
Sales agent	0.2	0.1	0.1	0
Manager	0	0	0	0.3
Family	0.05	0.05	0.3	0
Business traveler	0.01	0.2	0.01	0

Task Frequency – Examples

Word 2003 Powerpoint 2007 Word 2007

- *Bold* format is available in the toolbar
- *Subscript* requires menu and dialog
- Assumption for the standard UI is that user needs more often bold than subscript
- For users with different needs customization is possible

Task Frequency: Trade-off between Quick Access and Over-crowed Interface

- Example toolbar
  - More tasks directly available in the toolbar make it quicker to do these tasks
  - Increasing the number of options in the toolbar increase the time needed to locate them
  - Screen area that is used

Principles for User Interface (UI) design

- Implementation and technology independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)

- Ben Shneiderman's list of principles
  - **Principle 1 : Recognize User Diversity**
  - **Principle 2 : Follow the 8 Golden Rules**
  - **Principle 3 : Prevent Errors**

8 Golden Rules - Rule 1: Consistency

- Many forms of consistency:
  - Consistent sequences of actions in similar situations
  - Identical terminology used in prompts, menus, help screens
  - Consistent color, capitalization, layout, fonts etc.

- Bad example: WWW
  - No real guidelines and no authority
    » How are links represented?
    » Where is the navigation?
  - Styles and "fashion" change quickly...

Consistency: Levels

- Consistency levels
  - lexical
  - syntactic
  - semantic

- **Inconsistent** – variant 1
  - Delete/insert character
  - Delete/insert word
  - Remove/insert line
  - Delete/insert paragraph

- **Inconsistent** - variant 2
  - Take-away/insert character
  - Delete/add word
  - remove/put-in line
  - eliminate/create paragraph

- **Consistent**
  - Delete/insert character
  - Delete/insert word
  - Delete/insert line
  - Delete/insert paragraph

- **Inconsistent** - variant 3
  - Character deletion/insertion
  - Delete/insert word
  - Line deletion/insertion
  - Delete/insert paragraph

Lexical / Syntactic Consistency

- Lexical Consistency
  - Coding consistent with common usage, e.g.
    - » red = bad, green = good
    - » left = less, right = more
  - Consistent abbreviation rules
  - Equal length or first set of unambiguous chars
  - Devices used same way in all phrases
  - Character delete key is always the same

- Syntactic Consistency
  - Error messages placed at same (logical) place
  - Always give command first - or last
  - Apply selection consistently, e.g. select text then apply tool or select tool and then apply to a text
  - Menu items always at same place in menu (muscle memory)

Semantic Consistency

- Global commands that are always available
  - Help
  - Abort (command underway)
  - Undo (completed command)

- Operations valid on all reasonable objects
  - if object of class "X" can be deleted, so can object of class "Y"

- Applicability
  - to command line user interfaces
  - to keyboard short cuts
  - to speech interfaces
  - to tool bars
  - to menus
  - to selection operation
  - to gestures
  - …

Consistency: Capture through Grammars

- Task-Action-Grammar (TAG), Reisner 1981
  - Task[direction,unit] → symbol[direction] + letter[unit]
  - Symbol[direction=forward] → "CTRL"
  - Symbol[direction=backward] → "ALT"
  - Letter[unit=word] → "W"
  - Letter[unit=paragraph] → "P"

- Example - Commands
  - Move cursor on word forward: CTRL-W
  - Move cursor on word backward: ALT-W
  - Move cursor on paragraph forward: CTRL-P
  - Move cursor on paragraph forward: ALT-P

Inconsistencies

- Dragging file operations?
  - folder on same disk vs. folder on different disk
  - file to trash can vs. disk to trash can

- Fitts' Law suggests bigger buttons for more often used operations

- Sometimes inconsistency is wanted
  - E.g. Getting attention for a dangerous operation
  - Consistency on semantic level may cause inconsistency on syntactic level
  - Example:
    - » Confirmation of operation is default option
    - » Confirmation of reformat command?

Format Preload (C:)

⚠ WARNING: Formatting will erase ALL data on this disk.
To format the disk, click OK. To quit, click CANCEL.

[OK] [Cancel]
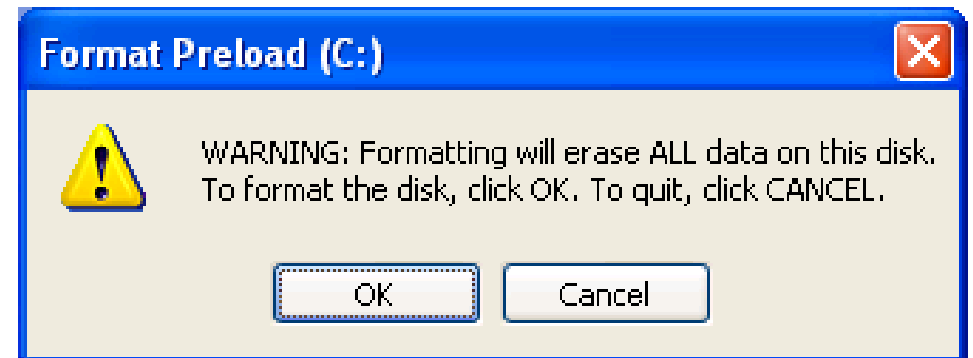
8 Golden Rules - Rule 2: Shortcuts

- Enable shortcuts: Improves speed for experienced users

- Shortcuts on different levels
  – Access to single commands, e.g. keyboard shortcuts (CTRL+S) or toolbar
  – Customizing of commands and environments, e.g. printer preset (duplex, A4, …)
  – Reusing actions performed, e.g. history in command lines, macro functionality

- Shortcuts to single commands are related to consistency
  – CTRL+X, CTRL+C, CTRL+V in Microsoft & Apple applications for cut, copy and paste
  – However CTRL+S (saving a document) is only implemented in some applications…
  – Apple applications are more consistent in shortcuts (e.g. CTRL-S) due to early guidelines/toolkits for developers

8 Golden Rules - Rule 3: Feedback

- For **any** action performed the user should have appropriate and informative feedback

- For frequent actions it should be modest, peripheral

- For infrequent actions it should be more substantial

PowerPoint is saving w:\My Documents\work\lmu\lehrauftrag2009\lectur...

Stop a Hardware device

Confirm devices to be stopped, Choose OK to continue.

Windows will attempt to stop the following devices. After the devices are stopped they may be removed safely.

Microsoft ACPI-Compliant Control Method Battery

OK   Cancel

8 Golden Rules - Rule 4: Closure

- Sequences of actions should have a beginning, middle, and end.
  - Satisfaction of accomplishment = relief

- On different levels –
  - E.g. in the large: Web shop - it should be clear when I am in the shop, and when I have successfully check-out
  - E.g. in the small: a progress bar

5% of mozilla-win32-1.6a-installer.ex...

Saving:
...a-win32-1.6a-installer.exe from ftp.mozilla.org

Estimated time left: 25 min 50 sec (557 KB of 11.1 MB copied)
Download to: C:\d...\mozilla-win32-1.6a-installer.exe
Transfer rate: 7.83 KB/Sec

☐ Close this dialog box when download completes

Open   Open Folder   Cancel
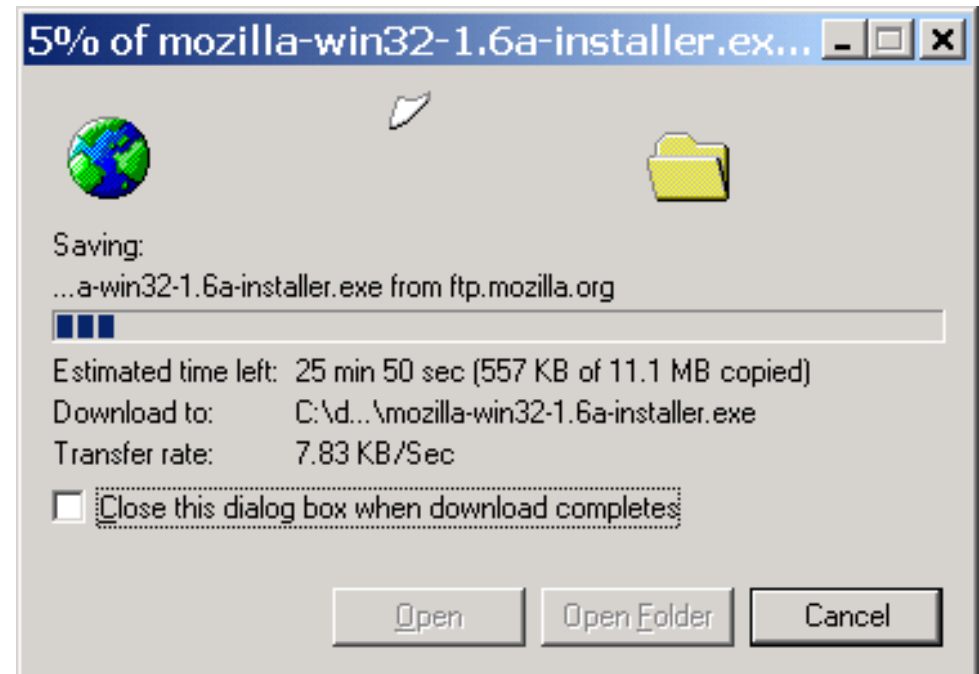
8 Golden Rules - Rule 5: Prevent Errors

- Create UIs that make it hard to make errors
  - Examples:
    - » Menus instead of commands
    - » Options instead of alphanumeric field (only certain values allowed)

- Detect errors or possible errors
  - Examples
    - » Leaving an editor without saving
    - » Writing to a file that already exists
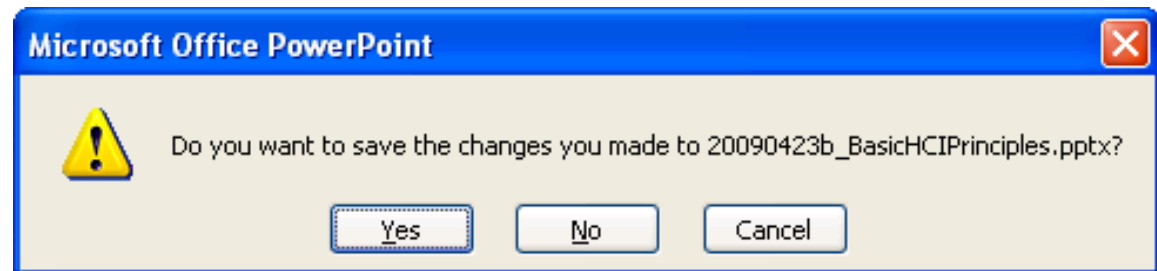
- Provides safety for the user

- Different options for handling:
  - Involve the user (current practice)
  - Prevent the error or its consequences on system level (e.g. create backups/versions when a file is overwritten)

8 Golden Rules - Rule 6: Easy Reversal of Actions

- As a basic rule – all actions should be reversible
  - Relieves anxiety of users, encourages exploration of unfamiliar options

- Providing UNDO functions (possibly with infinite depth)

- Allow undo of groups of actions

- Undo is not trivial if user is not working sequentially
  - E.g. write a text, copy it into the clipboard, undo the writing
    → the text is still in the clipboard!

- Reversal of action becomes a usage concept
  - Browser back-button is used for navigation (for the user a conceptual reversal of action)
  - Formatting of documents – e.g. "lets see how this looks, … don't like it, … go back to the old state"

8 Golden Rules - Rule 7: Feeling in Control

- Users (in particular experienced) like to feel to be in control of the system

- Gaines, 1981:
  - User should initiate actions (initiator instead of responder)
  - Avoid non-causality

- The system should be predictable
  - No surprising system actions, no tedious but unavoidable sequences of data entries, no unexpected silence or waiting state
  - Otherwise anxiety and dissatisfaction rise

- Note: some current developments are in contrast, e.g.:
  - Proactive computing
  - Intelligent agents

- General trade-off between transparency and intelligence of system

8 Golden Rules -
Rule 8: Reduce Short-term Memory Load

- The system should remember, not the user
  - George A. Miller, 1956: The magical number Seven, Plus or Minus Two
  - Humans can recall 7 +/- 2 chunks of information for a short time

- Interface designs have to be simple to comply with human memory

- Examples that create problems
  - Multi-page forms where the user has to know at form N what she filled in in form N-1
  - Abbreviations introduced in one step and used in the following (e.g. user selects a destination – as the name of a city – and the system does the following steps by showing the airport code)

- Helpful:
  - Keep dialogues compact (avoid splitting of pages)
  - Use memory aids (visual or audio) for mnemonics

- Apply the rule with care!
  - Sometimes complex menu structures are unavoidable
  - With sufficient training and support, also cryptic mnemonics are acceptable for frequent users

Summary – 8 Golden Rules

- Consistency

- Shortcuts

- Feedback

- Closure

- Prevent Errors

- Reversal

- Control

- Memory Load

Add New E-mail Account

Auto Account Setup
Clicking Next will contact your e-mail server and configure your Internet service provider or Microsoft Exchange server account settings.

Your Name: Barbara Sankovic
Example: Barbara Sankovic

E-mail Address: a@bcom
Example: barbara@contoso.com

Password: ****
Retype Password: ****
Type the password your Internet service provider has given you.

☐ Manually configure server settings or additional server types

< Back   Next >   Cancel

1) Does not show there is a (potential) error in the email address – just greys out the 'Next' button.
2) When passwords do not match, it allows 'Next' but gives a detailed error message.

Principles for User Interface (UI) design

- Implementation and technology independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)

- Ben Shneiderman's list of principles
  - **Principle 1 : Recognize User Diversity**
  - **Principle 2 : Follow the 8 Golden Rules**
  - **Principle 3 : Prevent Errors**

Principle 3: Prevent Errors - Classical Techniques

(Note: golden rule number 5 discusses the same topic on higher level…)

A few classical "tricks" to prevent errors (Source: Shneiderman)

- Correct matching pairs
  - Examples: { } in program text, <B>**bold**</B> in HTML
  - Prevention: insert both brackets in one action; or remind of missing bracket

- Complete sequences
  - Assistance to complete a sequence of actions to perform a task
    » For advanced users: planning and editing the sequence
  - Examples: log-on sequences, wizards, scripts

What is an "error" after all?

- Command correction
  - Aim: Trying to prevent users entering incorrect commands
    » Examples: file completion on Unix / helpful error messages / menus instead of commands