2 Basic HCI Principles

2.1 Motivation: Users and Developers

2.2 Principle 1: Recognize User Diversity

2.3 Principle 2: Follow the 8 Golden Rules

2.4 Principle 3: Prevent Errors

2.5 Background: The Psychology of Everyday Action

2.6 GOMS: Goals, Operators, Methods, Selection Rules

Corresponding extension topic:
E1 Fitt’s Law
This is a first (and not yet very deeply discussed) approach to design for usability.
What the User Sees

• Users see only what is openly visible!
What the Developer Knows

- Users have little idea about:
  - architecture,
  - state transitions,
  - dependencies
  - application context
  - system restrictions
  - …

- And users often do not want to know about it.
A Computer Screen and its Interpretation

• What do we see?

• What is shown?

• What is the meaning?
Answers from Skilled Computer Users

- Win2000 desktop
- Text and figures
- Icons and toolbars
- Overlapping windows
- Scroll bars and menus
- Task bar and status information
- Representations of documents
Basic (Naive) Technical Answers

- 2-D surface
- Controllable pixels

- Image with a resolution of 1400x1050 pixels
- For each pixel the colour can be set
- The change of colour can be controlled rapidly
Perfect User’s Answers

- My work environment
- Meeting notes
- Budget for next year
- Request to write a technical article
- Background information on a psychological phenomenon
Metaphor Example 1 – Overlaying Windows

• What is the meaning of the fact that a window is behind another window?

• What is real? What is illusion?

• What does iconizing do?

• Models? Conceptual... Implementation... Represented...
Metaphor Example 2 – Scrollbar vs. Hand

- Moving *up* the scroll bar moves *down* the document

- What happens in reality? What do we imagine? What is the metaphor?
Metaphor Example 2 – Scrollbar vs. Hand (contd.)

• Moving **up** the hand
  Moves **up** the document

• What happens in reality?
  What do we imagine?
  What is the metaphor?
Metaphor Example 2 - Scrollbar vs. Hand (contd.)

- Adequacy of interaction mechanism depends on content displayed
2 Basic HCI Principles

2.1 Motivation: Users and Developers

2.2 Principle 1: Recognize User Diversity

2.3 Principle 2: Follow the 8 Golden Rules

2.4 Principle 3: Prevent Errors

2.5 Background: The Psychology of Everyday Action

2.6 GOMS: Goals, Operators, Methods, Selection Rules

Corresponding extension topic:
E1 Fitt’s Law
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:
  (see http://media.pearsoncmg.com/aw/aw_shneiderma_dtui_4/chapter2.pdf)
  – Principle 1 : Recognize User Diversity
  – Principle 2 : Follow the Eight Golden Rules
  – Principle 3 : Prevent Errors
• Similar lists exist in several variants
Principle 1: Recognize User Diversity

- Simple and obvious - nevertheless in reality extremely difficult
- *Example*: consider an online travel agent
  - Travel agent booking many flights a day – everyday
  - A teacher organizing a field trip (once a year) and making bookings for a large group
  - A business person changing bookings while travelling
  - A family looking for a package holiday
- Basic concepts to structure the problem:
  - Usage profiles
  - Task profiles
Usage Profiles

• “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 with computers.
• 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: Unneeded functionality should never be added just 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 fillins, …
Hypothetical Frequency of Tasks
(Example of a booking system for travel)

<table>
<thead>
<tr>
<th>Position</th>
<th>Task</th>
<th>Group reservation</th>
<th>Change of itinerary</th>
<th>Booking child care</th>
<th>Comparing sales agent performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales agent</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Manager</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td>0.05</td>
<td>0.05</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Business traveler</td>
<td>0.01</td>
<td>0.2</td>
<td>0.01</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Task Frequency - Examples

- 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 the customization is available
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 increases the time needed to locate them
  - Screen area that is used
2 Basic HCI Principles

2.1 Motivation: Users and Developers
2.2 Principle 1: Recognize User Diversity
2.3 Principle 2: Follow the 8 Golden Rules
2.4 Principle 3: Prevent Errors
2.5 Background: The Psychology of Everyday Action
2.6 GOMS: Goals, Operators, Methods, Selection Rules

Corresponding extension topic:
E1 Fitt’s Law
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…
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
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
8 Golden Rules – Rule 5: Prevent Errors

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

• Detect errors or possibles 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, keep all files that have been created by the user)
8 Golden Rules –
Rule 6: Permit 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. “let’s 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 tradeoff 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 rule with care
  - Sometimes complex menu structures are unavoidable
  - With sufficient training and support, also cryptic mnemonics are acceptable for frequent users
2 Basic HCl Principles

2.1 Motivation: Users and Developers
2.2 Principle 1: Recognize User Diversity
2.3 Principle 2: Follow the 8 Golden Rules

More about rule 1: Consistency...

2.4 Principle 3: Prevent Errors
2.5 Background: The Psychology of Everyday Action
2.6 GOMS: Goals, Operators, Methods, Selection Rules

Corresponding extension topic:
E1   Fitt’s Law
Consistency (1)

• Consistency levels
  – lexical
  – syntactic
  – semantic

• Consistent
  – Delete/insert character
  – Delete/insert word
  – Delete/insert line
  – Delete/insert paragraph

• 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

• Inconsistent - variant 3
  – Character deletion/insertion
  – Delete/insert word
  – Line deletion/insertion
  – Delete/insert paragraph
Consistency (2)

• 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)
Consistency (3)

- Semantic Consistency
- Global commands 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

• 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?