Designing Interactive Systems I

Ten Golden Rules, Responsiveness

Prof. Dr. Jan Borchers
Media Computing Group
RWTH Aachen University

Winter Semester ’23/’24

https://hci.rwth-aachen.de/dis
Review

• GOMS and KLM
  • Execution times for routine tasks
• Information efficiency
• Character efficiency
Ten Golden Rules of Interface Design
Ten Golden Rules of Interface Design

1. Keep the interface simple!
2. Speak the user’s language!
3. Be consistent and predictable!
4. Provide feedback & be responsive!
5. Minimize memory load!
6. Avoid errors, help to recover, offer undo!
7. Design clear exits and closed dialogs!
8. Include help and documentation!
9. Address diverse user needs!
10. Hire a graphic designer!
1. Keep the Interface Simple!

• Most important rule
• First design is often too complex & awkward
• Avoid feature creep
  • Some consumers will ask for more and more features
  • But usability must not suffer
  • Experience: 80% of users use only 20% of features (e.g., Word)
  • Honorable goal would be: Next version will have no new features, just be easier to use
  • If pressed, move feature sets out to sub-dialogs
Feature Creep Example: Blu-Ray Players

Example: Simple Alarm Clock
Modul: Grundlagen der Rechnergeschichte und Maschinentheorie III - Seminar

<table>
<thead>
<tr>
<th>Nummer</th>
<th>DMK-SNM-0303</th>
<th>OE: DMK-SNM Neue Medien</th>
<th>Benutzerdefinierte Funktion abrufen</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typ</td>
<td>Modul</td>
<td></td>
<td>Status inaktiv</td>
<td></td>
</tr>
<tr>
<td>Kategorie</td>
<td>Lehrveranstaltung</td>
<td></td>
<td>Veranstalter: DMK, DMK Medien &amp; Kunst</td>
<td></td>
</tr>
<tr>
<td>Bezeichnung</td>
<td>Grundlagen der Rechnergeschichte und Maschinentheorie III - Seminar</td>
<td></td>
<td>Anmeldebedingung: Teilnehmende</td>
<td></td>
</tr>
<tr>
<td>Anmeldebedingung</td>
<td>Modul (2)</td>
<td>Texte Englisch, Texte</td>
<td>Anmeldungen, Codes, Gruppenzugehörigkeiten, Anmeldeinfo</td>
<td></td>
</tr>
<tr>
<td>Thema</td>
<td>Besprechender Text</td>
<td>Zoom mit &lt;F2&gt;</td>
<td>Untertitel/Kurzinfo</td>
<td></td>
</tr>
<tr>
<td>Voraussetzungen</td>
<td>abgeschlossenes 1. + 2. Semester</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lehrform/Ablauf</td>
<td>Seminar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lernziele</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lerninhalte</td>
<td>Gemeinsames Erarbeiten der Rechner- und Maschinengeschichte des 19. Jahrhunderts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bibliographie/Literatur</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECTS Credits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminen</td>
<td>Mittwoch Nachmittag: 23.11., 30.11., 14.12., 11.1.06., 24.1.06. (Di, am+pm)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ort</td>
<td>Studienbereich Neue Medien, Sihlquai 131, 8005 Zürich</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bemerkungen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Anlass-Nr. like "dmk_snmu%"
2. Speak the User’s Language!

• Take words and concepts from the application domain, not technology

• Determine terminology during initial user interviews and task analysis

• Example:
  “File” means less to an architect who is new to computers than “drawing”

• Applies to words for objects, but also work processes and tasks (e.g., “order”)
Example: Samsung Tablet
Example: Zoom Captions

Meeting-Einstellungen

- Synchronisierte Kalender: Deaktiviert
- Meeting-Hinweis: Aktiviert
- Geschlossene Untertitel: Aktiviert

Zoom erinnert Sie daran bevorstehenden Meetings beizutreten.
Example: Apple Music

• Talks about “music”, “songs”, “video”, “movies”, “playlists”, not “files”
  • In menus, dialogs, and online help (⇒Rule 3: Consistency)

• Exceptions: E.g., “File” menu
  • Conflict with cross-application consistency
3. Be Consistent and Predictable!

- Consistency is needed across many levels:
  - Similar commands for similar situations
  - Consistent terminology in menus, dialogs, help pages, etc.
  - Consistent fonts, layout, color coding, upper/lower cases, etc. throughout the system
  - Only few obvious exceptions
    - No clear-text echo when entering passwords
    - Extra security check before erasing files, etc.
Example: Xerox Star Command Buttons

- Same (physical) buttons to copy a file, a word in a text editor, an object in a graphics program, etc.
- Still true today (Cut/Copy/Paste)
Consistency through Vertical Design
Predictability

• Follow the “Principle of Least Surprise”
  • System should always react so that it minimizes the user’s surprise (and therefore, confusion and irritation)

• Don’t do unexpected things
  • …and don’t make actions unexpectedly difficult (“…how do I print this in duplex?”)

• Users (especially experts) like to be “in control”
  • They initiate actions, the system responds
Principle of Least Surprise

Hi! I am Clippy, your office assistant. Would you like some assistance today?

Yes  No

Your battery is fully charged!
PowerPoint Office Assistant

Object on the master
The object you are trying to select is on the slide master, not on the current slide.
- Take me to the slide master
- Tell me about the slide master
- Thanks for the tip.

Office Assistant
Sorry, you must click an option before you can close the Assistant. Please click OK now, and then click an option.

OK
Timeouts are Evil!
Emergency Exit
Press on bar for 3 seconds
Door lock will release in 15 seconds

Sortie de secours
Appuyer sur la barre pendant 3 secondes
Le dispositif de verrouillage se déclenchera dans 15 secondes
4. Provide Feedback & Be Responsive!

- Recall the Seven Stages of Action
  - Complete & continuous feedback bridges Gulf of Evaluation
- Each user action requires some feedback
  - Subtle for small/short/frequent actions (e.g., key press, menu selection)
  - More noticeable for main/long/infrequent actions (e.g., saving or deleting files)
  - Icons in GUIs simplify visualizing object state and actions: direct manipulation
- Nothing is more frustrating for the user than “Where am I?” or “What is it doing now?”
Example: Windows 2000 Progress Dialog for Copying Files

• What’s wrong with this picture?
Example: Menu Selection

• What happens when you select a menu item?
macOS Catalina Menu
Haptic Feedback
5. Minimize Memory Load!

- Short-term memory: limited capacity (ca. 4 ± 1 chunks)

- Avoid situations where prior dialog information has to be reproduced from memory
  - E.g., user should not have to type anything in twice.

- Display information so it’s easy to parse (Gestalt laws)

- Provide obvious access to help pages for codes, abbreviations, etc.

- It’s easier to minimize memory load with GUIs than command line interfaces
  - “Read & Select” instead of “Remember & Type”
Keyboard Shortcut List

You can use keyboard shortcuts—combinations of keys you press at the same time—to quickly accomplish many common tasks, such as selecting text and manipulating objects.

Many menu options include their keyboard shortcuts, such as `⌘N`, where the symbol represents a “modifier key” on your keyboard.

Modifier key symbols are listed in the table below. For a complete list of all keyboard shortcuts in Keynote, including many not shown in the menus, see Keyboard shortcuts.

### Symbols for modifier keys

<table>
<thead>
<tr>
<th>Modifier key</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>⌘</td>
</tr>
<tr>
<td>Shift</td>
<td>⌘</td>
</tr>
<tr>
<td>Option</td>
<td>⌥</td>
</tr>
<tr>
<td>Control</td>
<td>⌘</td>
</tr>
<tr>
<td>Return</td>
<td>⌘</td>
</tr>
</tbody>
</table>
6. Avoid Errors, Help to Recover!

• Errors lead to stress
  • So offer simple, constructive, concrete, helpful, and comfortable instructions to recover
  • System state should not change through wrong input, or should be easy to restore

• Best: Design system so mistakes cannot be made in the first place. Examples:
  • Selection instead of (mis)typing
  • Cannot type letters in numerical data fields
  • Arcade game machines have virtually no error messages!
  • Automatic correction of illegal characters in file names
The name “a:b” can't be used.
Try using a name with fewer characters, or with no punctuation marks.
6. Avoid Errors, Help to Recover!

- Offer undo
  - As many actions as possible should be reversible
  - Lowers anxiety because users know errors are correctable
  - Encourages users to try out new functions
  - Ideal: multiple undo, and at multiple levels
7. Design Clear Exits & Closed Dialogs!

- Three most common questions of users during a dialog:
  - Where am I?
  - What can I do here?
  - How do I get back to where I was?
- Clear exits (“Back”, “Quit”) help with Question 3
- Closed dialogs:
  - Provide feeling of having completed a step
  - Allows user to relax, “take a breath”, frees the mind for the next step
Thank you, your order has been placed.
An e-mail confirmation has been sent to you.

Order Number: **104-1969352-5141057**
- 1 item will be shipped to Chatchavan Wacharamanotham by Amazon.com. Estimated delivery January 18, 2011 - February 7, 2011

> Review or edit your order

Next time use Express Checkout with PayPhrase
Buy on Amazon and across the web with a simple phrase.

Choose your PayPhrase: "C's Surprising Doors"
(Use this suggestion, see others, or enter your own)

Orders will ship to: Chatchavan Wacharamanotham, Lehrstuhl In...
Orders will be paid using: VISA ****

Create your PayPhrase

A Payphrase is an easy-to-remember shortcut to shipping and payment information in your Amazon.com account. Use it on Amazon.com and across the web. (Learn more)
Connection was lost
Would you want to reconnect?

Yes  OK
8. Include Help and Documentation!

- Hierarchy of help systems, with increasing breadth and decreasing ease-of-access:
  - Dynamic Descriptors, such as Tooltips (but let users disable them!)
  - Online tutorials and references
  - Printed documentation (but…)

- More active help can be useful:
  - Assistants and Wizards
  - But danger: system takes over initiative, which breaks Rule 3 (predictability)

Users don’t read manuals!
9. Address Diverse User Needs!

- Novices want more explanations

- Frequent users want less fussy and faster interaction
  - They value (configurable) keyboard shortcuts, macro recording, programmability, and quick responses without unnecessary feedback (for them)

- Different age ranges have different interface expectations

- Technology affinity ("enjoying to play with gadgets") varies widely among people

- But conflict: If in doubt, Rule 1 ("Keep the interface simple") is more important! May have to focus on a user group
Example: PostBrainstorm

• New users get popup menu

• Experienced users remember the gestures to select frequent commands from the menu

• The menu does not even pop up when the gesture is done rapidly

• But: If you ever forget the gesture, just wait for a fraction of a second, and you can revert to using the popup menu

• The result: Fluid and reversible transition from menu selection to gesture commands

Figure 1: To zoom, the user moves the pen from the rest area into the Items... octant (a). Submenus (Highlight, Move, Zoom) appear and the first level menu items are selected as greyed out (b). Entering the Zoom octant submenu, then moving back to the rest area dismisses the root level menu and brings up the zoom menu with the current zoom value (75%) displayed in the center (c). A new zoom value of 100% is selected by moving into the octant for the desired value and back to the center at which point the zoom is applied (d). Several zoom values can be tried out during the same interaction since the zoom menu stays in place until the pen is lifted. The dashed circles added to the illustration (a) and (b) show the transition boundaries for leaving and entering the rest area (see text). For explanatory purposes, the figures in this paper explicitly show the pen track and the underlying selected object is shown only in Figure 3. In normal use, the pen track is not displayed and the selected object is visible behind the transparent menu.

Figure 2: After selecting Item... → Zoom from the root menu (a), the user selects Numeric to enter the new zoom value as a sequence of digits (b). The zoom menu is dismissed and the Quikwriting system is brought up (c) so that the zoom value (d).

Multiple items to a menu system as well. Cirrin [9] is a soft key-board in which letters are arranged at the circumference of a circle. Like Quikwriting it provides a way to enter successive letters of a word in a continuous stroke without having to lift the pen. After an initial training period, words can be remembered as a kind of shorthand. The initial layout of 26 primary entries without hierarchy makes it less convenient to extend to a menu system.

THE FLOWMENU

The FlowMenu is presented as a radial menu with 8 octants and a central rest area (figure 1). Starting from the rest area, the user selects a top-level menu item by entering the corresponding octant. As she does, sub-menus for this menu appear laid out further away from the center while non-selected top-level items are grayed out. Moving the pen to the submenu octant and reentering the rest area from this octant will trigger menu selection. The user can abort the interaction by removing the pen from the surface before reentering the rest area. With a simple FlowMenu, the user can access 8 top-level menu items, each with 8 submenu items. However since each selection of a menu ends with the cursor at the center of the menu, successive menu interactions can be merged together to build deeper hierarchies and arbitrarily long sequences of interactions. Figure 1 shows an example where after selecting the zoom submenu from the system menu, the system menu is removed and the zoom menu is brought up to let the user adjust the zoom.

Merging menu selection and parameter entry is easy because commands are segmented by the return of the cursor to the rest area. To let the user enter an alphanumerical value after a menu selection we remove the menu from the screen and present in its place a Quikwriting pad. Figure 2 shows such an interaction. The selection Item... → Zoom → Numeric brings up the Quikwriting system to let the user enter a numeric zoom value. The user can learn a composite sequence of commands and text as the superposition of simple loop gestures such as shown in figure 2d. The system can also be used in a way similar to control menus by letting the user...
To change a shortcut, select it, click the key combination, and then type the new keys.

<table>
<thead>
<tr>
<th>Launchpad &amp; Dock</th>
<th>Mission Control</th>
<th>Keyboard</th>
<th>Input Sources</th>
<th>Screen Shots</th>
<th>Services</th>
<th>Spotlight</th>
<th>Accessibility</th>
<th>App Shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### All Applications

- **Show Help menu**: \(&%/\)
- **Sleep**: ^\&\S
- **Keynote.app**
  - **Title**: ^\&\T
  - **Undo**: ^\&\Z
- **Pages.app**
  - **As Normal Priority**: ^\&\L
  - **As High Priority**: \&\L

### Full Keyboard Access

In windows and dialogs, press Tab to move keyboard focus between:

- Text boxes and lists only
- All controls

Press Control+F7 to change this setting.

Keyboard battery level: ![Battery Icon] 95%
10. Hire a Graphic Designer!
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bez</td>
<td>ASI</td>
<td>ASI</td>
<td>ASI</td>
<td>Drie</td>
<td>Einf</td>
<td>Grau</td>
<td>Har</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Zei</td>
</tr>
<tr>
<td>Status</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leitung</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thema</td>
<td>Untertitel/Kurzinfo</td>
<td>Voraussetzungen</td>
<td>Lehrlform/Ablauf</td>
<td>Lernziele</td>
<td>Lerninhalte</td>
<td>Bibliographie/Literatur</td>
<td>ECTS Credits</td>
<td>Termine</td>
<td>Ort</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HOHE QUALITÄT ZUM NIEDRIGSTEN PREIS!

Seit 11.10.

PREISSTURZ!
Ten Golden Rules of Interface Design

1. Keep the interface simple!
2. Speak the user’s language!
3. Be consistent and predictable!
4. Provide feedback & be responsive!
5. Minimize memory load!
6. Avoid errors, help to recover, offer undo!
7. Design clear exits and closed dialogs!
8. Include help and documentation!
9. Address diverse user needs!
10. Hire a graphic designer!
Responsiveness and Performance
Responsiveness

• See also: Jeff Johnson, *GUI Bloopers 2.0*

• Key usability problem of interactive systems
  • Bad responsiveness opens Gulf of Evaluation

• Examples for bad responsiveness:
  • A screen pointer that doesn’t keep up
  • Delayed response to button-clicks
  • Sliders and scrollbars that lag
  • Applications that go “dead” during disk operations
  • Multiple screen repaints
Reasons for Poor Responsiveness

- Importance not widely known
  - UI designers think of other things first
- UI designers rarely specify responsiveness
- Programmers tend to equate it with performance
- This kind of tuning is always difficult
  - “We’ll get it in the next release,” and so on

- Developers treat human input like machine input
- Simple, naïve implementations
- GUI tools and platforms are inadequate
  - Limitations of online apps (which everybody knows about)
Example: Scrollbar

• Does text move as you scroll (good) or after you let go (bad)?

• If designer doesn’t specify, developer will make a decision

• That will usually be the technically simplest
  • Since developers are not trained in user interface theory and concepts
  • Just as UI designers are generally not trained in implementing large software products in C++
Some Eternal Facts

• Responsiveness ≠ performance!

• Processing resources will always be limited
  • We still look at hourglass as much as 15 years ago
  • UIs are real-time systems with deadlines based on human cognition
  • Software does not need to do everything instantly, or in a given order, or even at all
Three Human Deadlines

• **0.1 seconds**
  - Perception of cause and effect (recall CMN model)
  - E.g., delay between moving mouse and pointer following, or between mouse click and inverting button

• **1 second**
  - Turn-taking in conversation, minimum reaction time for unexpected events
  - E.g., you have 1s max to show progress indicator, open window, or finish system-initiated operations (like auto-save)

• **10 seconds**
  - Typical human attention span
  - Max. time for one step of a task
    - E.g., entering a check into a banking program, or completing one step of a wizard
  - Max. time to finish giving input for a task
    - E.g., from selecting “Print” menu entry to sending off the print job
Design Techniques for Responsiveness

• Meet human-time deadlines
  • Rely on the three deadlines and recognize the differences
  • Acknowledge user input immediately, and display busy and progress indicators
  • Use them as frequently as you can, you never know when it will take longer
• Example “Progress bar”:
  • Make it real, show total items remaining, overall progress, and estimated total time remaining
  • Only useful if it advances roughly linearly! (no hanging at 99% please)
  • Estimated time should always go down, never up
  • “Less than a minute” is better than “47 seconds” (why?)
Design Techniques for Responsiveness

- Display important information first
  - Example: How to draw a clock

- Work in parallel
  - Delegate work that isn’t time-critical to background processes
  - Work ahead by preparing likely requests

- Optimize Queueing
  - Create a logical order by looking at all pieces first, then prioritize
Design Techniques for Responsiveness

- Manage time dynamically
- Adjust the strategy if not keeping up
- Decrease quality or quantity to keep up

Example: WordStar (1978)
- Ran on a 1 MHz computer, killed by IBM PC
- Written by an amateur, but he accommodated by making the system responsive
- WordStar never dropped characters typed
- Characters typed were always on screen instantly
- Instead stopped updating other areas of the screen
Design Techniques for Responsiveness

- Test under different conditions
  - Test under heavy loads
  - Test on slower systems, like your customers have
  - Test over slower net connections
Latency

- Latency is crucial in every interactive system
- Stay well below 100 ms from input to display

Throughput ≠ Latency

- Experimenters, watch your end-to-end latency (test with oscilloscope and LDR)
  - LCDs add dozens of ms of constant lag, USB can add 0..125 ms of lag *with jitter*(!)
- CRTs have near-zero lag, embedded systems and FPGA emulators can read inputs with zero lag
- See [http://hci.rwth-aachen.de/latency](http://hci.rwth-aachen.de/latency)
Summary

- **10 Golden Rules of Interface Design**
  - Keep the interface simple!
  - Speak the user’s language!
  - Be consistent and predictable!
  - Provide feedback & be responsive!
  - Minimize memory load!
  - Avoid errors, help to recover, offer undo!
  - Design clear exits and closed dialogs!
  - Include help and documentation!
  - Address diverse user needs!
  - Hire a graphic designer!

- **Responsiveness and Performance**
Roadmap

Human

- Performance
- Models of interaction
  - Affordances
  - Mappings
  - Constraints
  - Types of knowledge
  - Errors
- Visual Design

Case Studies

- History of HCI
- Visions
- Technology Phases

Development Process

- Iterative design
- User observation
- Ideation
- Prototyping
- User studies and evaluation
- Interaction design notation

Diagram:

- Goal
- Plan
- Compare
- Specify
- Interpret
- Perceive
- Perform
- World

Diagram arrows:

- Compare → Specify
- Interpre → Perceive
- Perform → World
- World → Perform
- Goal → Plan
- Plan → World