Designing Interactive Systems I

The Ten Golden Rules of Interface Design, Responsiveness

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

Winter Semester ’20/’21

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

- Key objective of controlled experiments?
  - What effects may occur?
  - What can you do to minimize those effects?
  - Types of validity?

- What is the Keystroke-Level GOMS model?
  - Novice vs. expert users?

- 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
Example for Feature Creep: Blu-Ray Player

Example: Simple Alarm Clock
<table>
<thead>
<tr>
<th>Säule 1</th>
<th>Säule 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thema</td>
<td>Beschreibender Text</td>
</tr>
<tr>
<td>Untertitel/Kurzinfo</td>
<td></td>
</tr>
<tr>
<td>Voraussetzungen</td>
<td>abgeschlossenes 1. + 2. Semester</td>
</tr>
<tr>
<td>Lehreinheit/Ablauf</td>
<td>Seminar</td>
</tr>
<tr>
<td>Lernziele</td>
<td></td>
</tr>
<tr>
<td>Lerninhalte</td>
<td>Gemeinsames Erarbeiten der Rechner- und Maschinenentwicklung des 19. Jahrhunderts</td>
</tr>
<tr>
<td>Bibliographie/Literatur</td>
<td></td>
</tr>
<tr>
<td>ECTS Credits</td>
<td></td>
</tr>
<tr>
<td>Termin</td>
<td>Mittwoch Nachmittag 22.11. / 30.11. / 14.12. / 11.1.06 / 24.1.06 (Di, am+pm)</td>
</tr>
<tr>
<td>Ort</td>
<td>Studienbereich Neue Medien, StiHg 131, 8005 Zürich</td>
</tr>
<tr>
<td>Bemerkungen</td>
<td></td>
</tr>
</tbody>
</table>
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: 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!
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 Viewer
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.

<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
  - Lower 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
Sign-In
Hello, Jan Borchers
Please enter your password to continue
Password	Forgot your password?
Sign-In
Choose a delivery address

Most recently used

- Jan Borchers, Ahornstraße 55, Aachen, 52074 Germany

Other addresses

- 

Add a new address

Your pickup locations

- 

Order Summary

- Items: €55.88
- Postage & Packing: €0
- Order Total: €55.88

Use this address

Choose an address to continue checking out. You'll still have a chance to review your order before it's final.

How are delivery costs calculated?
1 Delivery address
Jan Borchers
Ahornstraße 55,
Aachen, 52074
Add delivery instructions

2 Choose a payment method
Mehr erfahren
Jan, jetzt 40 € Startgutschrift sichern & bis zu 3% bei Amazon.de zurückbekommen mit der Amazon.de Prime VISA Karte.

Order Summary
Items: €55.68
Postage & Packing: €8.79
Order Total: €64.67

Other payment methods
1. Delivery address
Jan Borchers
Ahornstraße 55,
Aachen, 52074
Add delivery instructions

2. Payment method
Bank account ending in
Invoice address: Same as delivery address
~ Add a gift card or promotional code
Enter Code Apply

3. Review items and delivery
● Save time next time
  Use this shipping address and payment method for future purchases.

Estimated delivery: 28 Jan 2021 - 1 Feb 2021
Items dispatched from BrightSparks:
Dulux Milk Frother
€55.88
Qty: 1
Sold by: BrightSparks
Not eligible for Amazon Prime (Learn more)
Gift options not available.

Choose a delivery option:
☐ €17.00 Expedited Shipping: get it by Saturday, Jan 23 - Tuesday, Jan 26
☐ €8.79 Standard Delivery: get it by Thursday, Jan 28 - Monday, Feb 1

Order Total: €64.67

A copy of your direct debit mandate and mandate ID can be accessed in your account under "Edit or Delete Payment"

Need help? Contact us via our Help pages or contact us

When ordering via a marketplace seller please accept their terms and conditions. For further information, company details, terms and conditions, and cancellation rights, please click on the seller's name.

You can return items dispatched from and/or sold by Amazon.de within 30 days of the date of purchase. This guarantee applies in addition to your statutory right to cancel your order within two weeks from the date of receipt. Please note that we only accept returns of shrink-wrapped or sealed data carriers such as CDs, cassettes, VHS-videos, DVDs, PC- and video games, as well as software in their original, sealed or wrapped.
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 users don’t read manuals!

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

[F. Guimbretière, Stanford, UIST 2000]
To change a shortcut, select it, click the key combination, and then type the new keys.

### All Applications
- **Show Help menu**: ⌘ H
- **Sleep**: ⌘ S
- **Keynote.app**
  - **Title**: ⌘ T
- **Pages.app**
  - **Undo**: ⌘ Z
- **Mail.app**
  - **As Normal Priority**: ⌃ L
  - **As High Priority**: ⌃ Shift 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: 🍃 95%
10. Hire a Graphic Designer!
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