Designing Interactive Systems I
Lecture 10: GOMS, Interface Efficiency, Golden Rules

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

Winter term 2015/2016

http://hci.rwth-aachen.de/dis
In 1995, now-famous web guru Jakob Nielsen had less than 24 hours to recommend if adding three new buttons to Sun’s home page was a good idea.

- Check out his “Alertbox” online column for good (and often fun) web design advice.
- He found that each new, but unused button costs visitors .5 million $ per year.
- 2 of the 3 new buttons were taken back out.
- The method he used for his estimate: GOMS.
GOMS

• Goals, Operators, Methods, Selection rules

• Card, Moran, Newell: The Psychology of HCl, 1983

• To estimate execution and learning times before a system is built
GOMS: Components

- **Goals** describe user's end goals
  - Routine tasks, not too creative/problem-solving
    - E.g., “copyedit manuscript”
  - Leads to hierarchy of subgoals

- **Operators** are elementary user actions
  - Key presses, menu selection, drag & drop, reading messages, gestures, speech commands, …
  - Assign context-independent duration (in ms)

- **Methods** are “procedures” to reach a goal
  - Consist of subgoals and/or operators

- **Selection rules**
  - Which method to use for a (sub)goal
    - E.g., to delete some text (individual preferences apply!)
Sample Method and Operators

GOAL: HIGHLIGHT-ARBITRARY-TEXT

1. MOVE-CURSOR-TO-BEGINNING  1.10s
2. CLICK-MOUSE-BUTTON        0.20s
3. MOVE-CURSOR-TO-END        1.10s
4. SHIFT-CLICK-MOUSE-BUTTON  0.48s
5. VERIFY-HIGHLIGHT          1.35s
Variants Of Goms

• **GOMS** (Card, Moran, and Newell 1983)
  - Model of goals, operators, methods, selection rules
  - Predict time an experienced worker needs to perform a task in a given interface design

• **Keystroke-level GOMS model** (simplified version)
  - Comparative analyses of tasks that use mouse (GID) and keyboard
  - Correct ranking of performance times using different interface designs

• **CPM-GOMS** (critical path method)
  - Computes accurate absolute times
  - Considers overlapping time dependencies

• **NGOMSL** (natural GOMS language)
  - Considers non-expert behavior (e.g., learning times)
Keystroke-Level Model

• Execution time for a task = sum of times required to perform the serial elementary gestures of the task

• Typical gesture timings
  • Keying $K = 0.2$ sec (tap key on keyboard, includes immediate corrections)
  • Pointing $P = 1.1$ sec (point to a position on display)
  • Homing $H = 0.4$ sec (move hand from keyboard to mouse or v.v.)
  • Mentally preparing $M = 1.35$ sec (prepare for next step, routine thinking)
  • Responding $R$ (time a user waits for the system to respond to input)

• Responding time $R$ effects user actions
  • Causality breakdown after 100 ms
  • User will try again after 250 ms $\Rightarrow R$

• Give feedback that input received & recognized
Keystroke-Level Calculation

• List required gestures
  • E.g., HK = move hand from mouse to keyboard and type a letter

• Compute mental preparation times Ms
  • Difficult: user stops to perform unconscious mental operations
  • Placing of Ms described by rules

• Add gesture timings
  • E.g., HMPK = H + M + P + K = 0.4 + 1.35 + 1.1 + 0.2 = 3.05 sec

• Rule terminology
  • **String**: sequence of characters
  • **Delimiter**: character marking beginning (end) of meaningful unit
  • **Operators**: K, P, and H
  • **Argument**: information supplied to a command
Rules for Placing Ms

• Rule 0, initial insertion for candidate Ms
  • Insert Ms in front of all Ks
  • Place Ms in front of Ps that select commands, but not Ps that select arguments for the commands

• Rule 1, deletion of anticipated Ms
  • Delete M between two operators if the second operator is fully anticipated in the previous one
    • E.g., PMK ⇒ PK

• Rule 2, deletion of Ms within cognitive units (contiguous sequence of typed characters that form a name)
  • In a string of MKs that form a cognitive unit, delete all Ms except the first
    • E.g., “ls” ⇒ MK MK MK ⇒ MK K MK
Rules for Placing Ms

• Rule 3, deletion of Ms before consecutive terminators
  • If K is redundant delimiter at end of a cognitive unit, delete the M in front of it
  • E.g., “blaₘₘ” ⇒ M 3K MK MK ⇒ M 3K MK K

• Rule 4, deletion of Ms that are terminators of commands
  • If K is a delimiter that follows a constant string then delete the M in front of it (not for arguments or varying strings)
  • E.g., “clearₘ” ⇒ M K K K K K MK ⇒ M K K K K K K
  
  Note that the ‘clear’ command does not take any arguments, and is therefore a constant string. ‘ls,’ on the other hand, can take arguments and Rule 4 cannot be applied there.

• Rule 5, deletion of overlapped Ms
  • Do not count any M that overlaps an R
  • E.g., user waiting for computer response
Exercise: Temperature Converter

• Convert from degrees Fahrenheit (F) to Celsius (C) or vice versa, requests equally distributed

• Use keyboard or mouse to enter temperature

• Assume active window awaiting input, an average of four typed characters (including point and sign), and no typing errors

• Task: create and analyze your own interface!
The Dialog Box Solution with Radio Buttons...
...And Its Keystroke-Level Model

- Case 1: select conversion direction
  - Move hand to mouse, point to desired button, click on radio button (HPK)
  - Move hands back to keyboard, type four characters, tap enter (HPK HKKKK K)
  - Rule 0 (insert M's): (HMPMK HMKMKMKMK MK)
  - Rule 1 (deletion of anticipated M's): (HMP_K HMKMKMKMKMK MK)
  - Rule 2 (deletion of M's within cog. units): (HMP_K HMK_KK_K K MK)
  - Result: HMPK HMKKKK MK
  - Estimated time = 7.15 sec
- Case 2: correct conversion direction already selected
  - MKKKKMK = 3.7 sec
  - Average time = (7.15 + 3.7) / 2 = 5.4 sec
GOMS Results

• Execution (& learning) times of trained, routine users for repetitive tasks (goals), leading to cost of training, daily use, errors
  • Can be linked to other costs (purchase, change, update system), resulting in $$$ answers
  • Use to model alternative system offers
    • E.g., “new NYNEX computers cost $2M/year more” [Gray93]

• Estimate effects of redesign
  • Training cost vs. long-term work time savings

• Starting point for task-oriented documentation
  • Online help, tutorials, …

• Don’t use for casual users or new UI techniques
  • Operator times not well defined
Measuring Interface Efficiency

• How fast can you expect an interface to be?

• Information as quantification of amount of data conveyed by a communication (Information theory)
  • E.g., speech, messages sent upon click…

• Lower bound on amount of information required for task is independent of interface design

• Information-theoretic efficiency \( E = \frac{\text{Minimal info required for the task}}{\text{Info supplied by user}} \)
  • \( E \in [0, 1] \) (e.g., \( E = 0 \) for providing unnecessary information)

• Character efficiency = \( \frac{\text{Minimal number of characters required for the task}}{\text{Number of characters entered in the UI}} \)

[Jeff Raskin: The Humane Interface, 2000]
Quantify Amount of Data

• Information is measured in bits
  • 1 bit represents choice between 2 alternatives

• n equally likely alternatives
  • Total information amount: $\log_2(n)$
  • Information per alternative: $(1/n)\log_2(n)$

• n alternatives with different probabilities $p(i)$
  • Information per alternative: $p(i)\log_2(1/p(i))$
  • Total amount = sum over all alternatives

• Consider situation as a whole
  • Probability of messages required
  • Information measures freedom of choice (information ≠ meaning)
Example: Temperature Converter

- Input assumptions (given)
  - 50% Fahrenheit, 50% Degree Celsius
  - 75% positive, 25% negative
  - only decimal input (no integer numbers)
  - All digits are equally likely
  - Only four characters input
Example: Temperature Converter

• Keystroke efficiency

  • Type C or F, value, enter: M K K K K M K ⇒ 3.9 sec (char. eff. 67 %)

  • Type value, then C or F: M K K K K M K ⇒ 3.7 sec (char. eff. 80%)

  • Bifurcated: M K K K = 2.15 sec (char. eff. 100 %)
### Example: Temperature Converter

<table>
<thead>
<tr>
<th>Numbers</th>
<th>Prob.</th>
<th>Values</th>
<th>p(i)</th>
<th>Information in bits</th>
<th>Overall (values x information in bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.dd</td>
<td>12.5 %</td>
<td>100</td>
<td>0.00125</td>
<td>0.012</td>
<td>1.2</td>
</tr>
<tr>
<td>-d.d</td>
<td>12.5 %</td>
<td>100</td>
<td>0.00125</td>
<td>0.012</td>
<td>1.2</td>
</tr>
<tr>
<td>.ddd</td>
<td>25 %</td>
<td>1000</td>
<td>0.00025</td>
<td>0.003</td>
<td>3</td>
</tr>
<tr>
<td>d.dd</td>
<td>25 %</td>
<td>1000</td>
<td>0.00025</td>
<td>0.003</td>
<td>3</td>
</tr>
<tr>
<td>dd.d</td>
<td>25 %</td>
<td>1000</td>
<td>0.00025</td>
<td>0.003</td>
<td>3</td>
</tr>
</tbody>
</table>

⇒ Minimal info required for the task = 11.4 bits/message

⇒ Simple approach: $4 \log_2(12) \approx 14$ bits
Example: Temperature Converter

- Information efficiency: $E = \frac{11.4 \text{ bits}}{\text{Info supplied by user}}$

- 128 keys standard keyboard (5 bits/key): $E = \frac{11.4}{4 \times 5} \approx 55\%$
- 16 keys numeric keypad: $E = \frac{11.4}{4 \times 4} \approx 70\%$
- 12 keys dedicated keypad: $E = \frac{11.4}{4 \times 3.5} \approx 80\%$
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: VCR
Example: Simple Alarm Clock
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: Telephone Book Menu
Example: iTunes

• 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

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 GUls 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?
Windows 7 Menu
Mac OS X Menu
Haptic Feedback
5. Minimize Memory Load!

- Short-term memory: limited capacity (ca. $7 \pm 2$ 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.

### 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

Fail.m.th
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
To change a shortcut, select it, click the key combination, and then type the new keys.

**Launchpad & Dock**
- Show Help menu: ⇧⌘H
- Sleep: ⇧⌘S

**Mission Control**
- Title: ⇧⌘T

**Keyboard**
- Undo: ⇧⌘Z

**Input Sources**
- As Normal Priority: ⇧⌘L
- As High Priority: ⇧⌘L

**Screen Shots**

**Spotlight**

**Accessibility**

**App Shortcuts**

---

**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!
<table>
<thead>
<tr>
<th>Modul: Grundlagen der Rechnergeschichte und Maschinentheorie III - Seminar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nummer</strong>: DMK-SNM-0303</td>
</tr>
<tr>
<td><strong>Typ</strong>: Modul</td>
</tr>
<tr>
<td><strong>Kategorie</strong>: Lehrveranstaltung</td>
</tr>
<tr>
<td><strong>Bezeichnung</strong>: Grundlagen der Rechnergeschichte und Maschinentheorie III - Sen</td>
</tr>
<tr>
<td><strong>Anmeldebedingung</strong>: abgeschlossenes 1. + 2. Semester</td>
</tr>
<tr>
<td><strong>Lehrform/Ablauf</strong>: Seminar</td>
</tr>
<tr>
<td><strong>Lernziele</strong>: Gemeinsames Erarbeiten der Rechner- und Maschinentheorie des 19. Jahrhunderts</td>
</tr>
<tr>
<td><strong>ECTS Credits</strong>: Bitte geben Sie die ECTS-Credits an.</td>
</tr>
<tr>
<td><strong>Termine</strong>: Mittwoch Nachmittag: 23.11. / 30.11. / 14.12. / 11.1.06 / 24.1.06 (Di-am+pm)</td>
</tr>
<tr>
<td><strong>Ort</strong>: Studienbereich Neue Medien, Sihlquai 131, 8005 Zürich</td>
</tr>
</tbody>
</table>

**Beschreibender Text** (Zoom mit <F2>)
HOHE QUALITÄT ZUM NIEDRIGSTEN PREIS!

PREISSTURZ!

Seit 11.10.
## 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!