Designing Interactive Systems I

GOMS, Interface Efficiency

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

Winter Semester ’23/’24

https://hci.rwth-aachen.de/dis
GOMS
A Story

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 articles at the Nielsen Norman Group for good (and often fun) web design advice.

- He found that each new, but unused button costs visitors 0.5 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
- In Card, Moran, Newell: The Psychology of HCI, 1983
- To estimate execution and learning times before a system is built
GOMS: Components

- **Goals** describe users’ 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 in Copyediting

GOAL: HIGHLIGHT-ARBITRARY-TEXT

A. MOVE-CURSOR-TO-BEGINNING 1.10s
B. CLICK-MOUSE-BUTTON 0.20s
C. MOVE-CURSOR-TO-END 1.10s
D. SHIFT-CLICK-MOUSE-BUTTON 0.48s
E. VERIFY-HIGHLIGHT 1.35s
GOMS Variants

- **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 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)
KLM: 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** $K = 0.2 \text{ s}$ (tap key on keyboard, includes immediate corrections)
  - **Pointing** $P = 1.1 \text{ s}$ (point to a position on display)
  - **Homing** $H = 0.4 \text{ sec}$ (move hand from keyboard to mouse or v.v.)
  - **Mentally preparing** $M = 1.35 \text{ sec}$ (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 $250 \text{ ms} \Rightarrow R$

- 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., “dir” ⇒ MK MK MK ⇒ MK K K
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., “bla⏎⏎” ⇒ M 3K MK MK ⇒ M 3K MK
- 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., “clear⏎” ⇒ M K K K K K MK ⇒ M 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!
  • **Keying** $K = 0.2$ s, **Pointing** $P = 1.1$ s, **Homing** $H = 0.4$ s, **Mentally preparing** $M = 1.35$ s
The Dialog Box Solution with Radio Buttons...

Choose which conversion is desired, then type the temperature and press Enter

Convert F to C
Convert C to F

23.7 → 74.66
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 HMKMKMKMKMKMK)
  - Rule 1 (deletion of anticipated M’s): (HMP_K HMKMKMKMKMKMKMK)
  - Rule 2 (deletion of M’s within cog. units): (HMP_K HMK_K_K_K_K MK)
  - Result: HMPK HMKKKKK 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
Information Efficiency
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 $= \frac{\text{Minimal number of characters required for the task}}{\text{Number of characters entered in the UI}}$

[Prof. Dr. Jan Borchers: Designing Interactive Systems I • WS 2023/24]

[Jef Raskin: The Humane Interface, 2000]
How to Measure Information Required

- 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: \( \frac{1}{n} \log_2(n) \)
- \( n \) alternatives with different probabilities \( p(i) \)
  - Information per alternative:
    \[
    p(i) \cdot \log_2\left(\frac{1}{p(i)}\right)
    \]
  - Total amount = sum over all alternatives
- Consider situation as a whole
  - Probability of messages required
  - Information measures freedom of choice (information ≠ meaning)
How Much Info Does Temperature Converter Need?

• 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
Character Efficiency Of Temperature Converters

Type C or F, value, enter

M K K K K M K

3.9s  char. eff. 67%

Type value, then C or F

M K K K M K

3.7s  char. eff. 80%

Bifurcated

M K K K K

2.15s  char. eff. 100%
Minimal info required for the task = 11.4 bits/message

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

<table>
<thead>
<tr>
<th>Numbers</th>
<th>Prob.</th>
<th>Values</th>
<th>$p(i)$</th>
<th>Information in bits</th>
<th>Overall (values × 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>
Temperature Converter With Diff. Keyboards

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

  - 128 keys standard keyboard (~5 bits/key in practice): $E = \frac{11.4}{4 \cdot 5} \approx 55\%$

  - 16 keys numeric keypad: $E = \frac{11.4}{4 \cdot 4} \approx 70\%$

  - 12 keys dedicated keypad: $E = \frac{11.4}{4 \cdot 3.6} \approx 80\%$