Questions?

- Midterm results
- Project step 2 available
GOMS (Card et al.)

- **Family of methods** (KLM, CMN-GOMS, NGOMSL, CPM-GOMS)
- **Describe the user behavior in term of**
  - **Goals**
    - A *thing to do*
    - *Like a function name – independent of application (i.e., “print page” works in any word processor)*
  - **Operators**
    - *Elementary perceptual, motor or cognitive actions.*
  - **Methods**
    - *Sequences of sub-goals and operators that can accomplish a goal.*
    - *Like the body of a function – implements a goal.*
  - **Selection rules**
    - *Used if several methods are available for a given goal*
    - *Not subjective – GOMS should run like a program*
- **For skilled users only**
  (models cognitive tasks, not problem-solving)
GOMS example I

- Setting

![Diagram of user interacting with video display, mouse, and keyboard]

MANUSCRIPT
Marked for editing and attached to notebook with experiment instructions to user.

- Analysis

GOAL: EDIT-MANUSCRIPT
  - GOAL: EDIT-UNIT-TASK
    - GOAL: ACQUIRE-UNIT-TASK
    - GET-NEXT-PAGE if at end of manuscript page
    - GET-NEXT-TASK
    - GOAL: EXECUTE-UNIT-TASK
    - GOAL: LOCATE-LINE
    - [select: USE-QS-METHOD
      USE-LF-METHOD]
    - GOAL: MODIFY-TEXT
    - [select: USE-S-COMMAND
      USE-M-COMMAND]
    - VERIFY-EDIT.

repeat until no more unit tasks
GOMS example II
(From HCI Models, Theories and Frameworks, J. Carroll)

- Using a text editor to edit the following text as shown

  The fox jumps over the lazy quick brown dog.

- Goals and sub-goals?
- Operators?
- Methods?
- Selection rules?
* Expansion of MOVE-TEXT goal
GOAL: MOVE-TEXT
  - GOAL: CUT-TEXT
    - GOAL: HIGHLIGHT-TEXT
      [select**: GOAL: HIGHLIGHT-PHRASE-COMPOSED-OF-WORDS
      - MOVE-CURSOR-TO-FIRST-WORD 1.10
      - DOUBLE-CLICK-MOUSE-BUTTON 0.40
      - MOVE-CURSOR-TO-LAST-WORD 1.10
      - SHIFT-CLICK-MOUSE-BUTTON 0.40
      - VERIFY-HIGHLIGHT 1.35
      GOAL: HIGHLIGHT-ARBITRARY-TEXT
        - MOVE-CURSOR-TO-BEGINNING-OF-TEXT
        - PRESS-MOUSE-BUTTON
        - MOVE-CURSOR-TO-END-OF-TEXT
        - RELEASE-CLICK-MOUSE-BUTTON
        - VERIFY-HIGHLIGHT]
  - GOAL: ISSUE-CUT-COMMAND
    - MOVE-CURSOR-TO-EDIT-MENU 1.10
    - CLICK-MOUSE-BUTTON 0.20
    - MOVE-CURSOR-TO-CUT-ITEM 1.10
    - VERIFY-HIGHLIGHT 1.35
    - CLICK-MOUSE-BUTTON
  GOAL: PASTE-TEXT
  - GOAL: POSITION-CURSOR-AT-INSERTION-POINT
    - MOVE-CURSOR-TO-INSERTION-POINT 1.10
    - CLICK-MOUSE-BUTTON 0.20
    - VERIFY-POSITION 1.35
  - GOAL: ISSUE-PASTE-COMMAND
    - MOVE-CURSOR-TO-EDIT-MENU 1.10
    - CLICK-MOUSE-BUTTON 0.20
    - MOVE-CURSOR-TO-PASTE-ITEM 1.10
    - VERIFY-HIGHLIGHT 1.35
    - CLICK-MOUSE-BUTTON 0.20

TOTAL TIME PREDICTED (SEC) 16.25
Keystroke Level Model (KLM)

• Focused on expert user – no selection rules
• Describe the task using the following operators:
  – K: pressing a key or pressing (or releasing) a button
    \[ t_K = 0.2 \text{ sec} \]
  – P: pointing
    \[ t_P = 1.1 \text{ sec (without button press)} \]
  – H: Homing (switching device)
    \[ t_H = 0.4 \text{ sec} \]
  – M: Mentally prepare
    \[ t_M = 1.35 \text{ sec} \]
  – R(t): system response time
    \[ t_R = t \]
How to use KLM

- Encode using all physical operator (K, P, H, D(n,l), R(t))
- Apply KLM rules [0-4] to add M’s
- Transform R followed by an M
  - If $t \leq t_M$: $R(t) \rightarrow R(0)$
  - If $t_M < t$: $R(t) \rightarrow R(t - t_M)$
- Compute the total time by adding all times
  - Will describe expert user behavior
KLM Heuristics (Raskin p. 77)

0: Insert M
   - In front of all K
   - In front of all P’s selecting a command

1: Remove M between fully anticipated operators
   - PMK → PK

2: if a string of MKs belong to cognitive unit delete all M but first
   - 4564.23: MKM Kim KmKMKM KMKMK → MKKKKKKK

3: if K is a redundant terminator then delete M in front of it
   - ←↓: MKMK → MKK

4a: if K terminates a constant string (command name) delete the M in front of it
   - cd ←↓: MKMK → MKK

4b: if K terminates a variable string (parameter) keep the M in front of it
   - cd class ←↓: MKKKMKKKKKKKM → MKKKMKKKKKKKM
Converting Temperature
(“Humane Interface”, Raskin)

- Convert 92.5F to Celsius

Assume:
- the focus is on the dialog box
- 4 characters for temperature
Converting Temperature
(“Humane Interface”, Raskin)

• Convert 92.5F to Celsius

Assume:
- the focus is on the dialog box
- 4 characters for temperature

• MKKKKMKK (3.7s)
• HMPKHMKMK (7.15s)

=> Average: 5.4s
Converting Temperature

• Your design. Can you do better?
• Design interface in pairs
• Perform KLM analysis and report

Try Google:
• 92.5f in °C
• 37°c in °F
Converting temperature: Raskin’s Redesign
(“Humane Interface”, Raskin)

Temperature Converter

Type in the temperature to be converted. The converted temperature will appear on the right as you type.

MKKKK => 2.15s
Pros and Cons

What are other factors?

• Learnability
• Accuracy
• Cognitive load
GOMS: Application and limitations

• **Applications**
  - CAD system
  - Telephone operator (CPM-GOMS)
  - Text editing with keyboard and mouse (KLM)

• **Limitations**
  - Skilled users
  - Does not deal with error
  - Does not deal with skill acquisition
  - Does not deal with high level issues (Functionality, workload, Fatigue)
  - Better for relative than absolute timing
NYNEX Example

Worked well, but:

• Compared to marketing data
• Compared to very expensive field trial
• Individual Goal estimates very inaccurate (~50% off)
• Core analysis obvious in retrospect
  (savings not in bottleneck don’t help)

Was GOMS necessary, or could an expert analysis have revealed this?
Value of KLM/GOMS?

• Possibly good for high value decisions
• Possibly good for making strong argument
• Definitely good for helping designers develop an intuition about works and doesn’t and the impact of design decisions on speed

• Look at www.hcibib.org for work on GOMS