

Questions?

- Reading Assignment on KLM
 - chapter 4 “The Humane Interface” by Raskin

High Level Model of Human Behavior

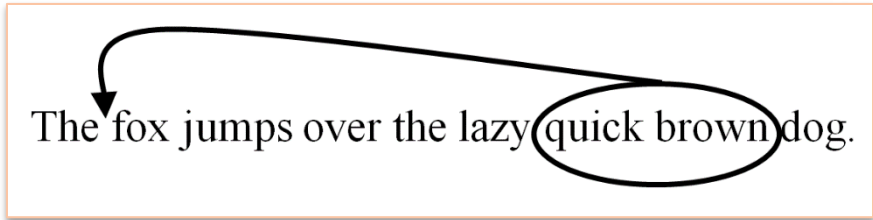
- Model Human Processor (or HIP) for short (isolated) tasks
 - Matching a symbol to memory
 - Determining the fastest speed to type on two different keyboards
- People will become **skilled** and want efficient methods for **routine** tasks
- GOMS are a family of methods used to model the user behavior for complex tasks
- GOMS for quantitative analysis of interface design
 - Predicts how long an **experienced worker** will take to perform an operation with an interface design
 - Models **cognitive skills**, not problem-solving
 - Family of methods (KLM, CMN-GOMS, CPM-GOMS, NGOMSL)

GOMS (Card et al.)

- 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 (KLM, CMN-GOMS)
(models cognitive tasks, not problem-solving)
- CPM-GOMS or NGOMSL takes into account nonexpert behavior such as learning times → more accurate and complete

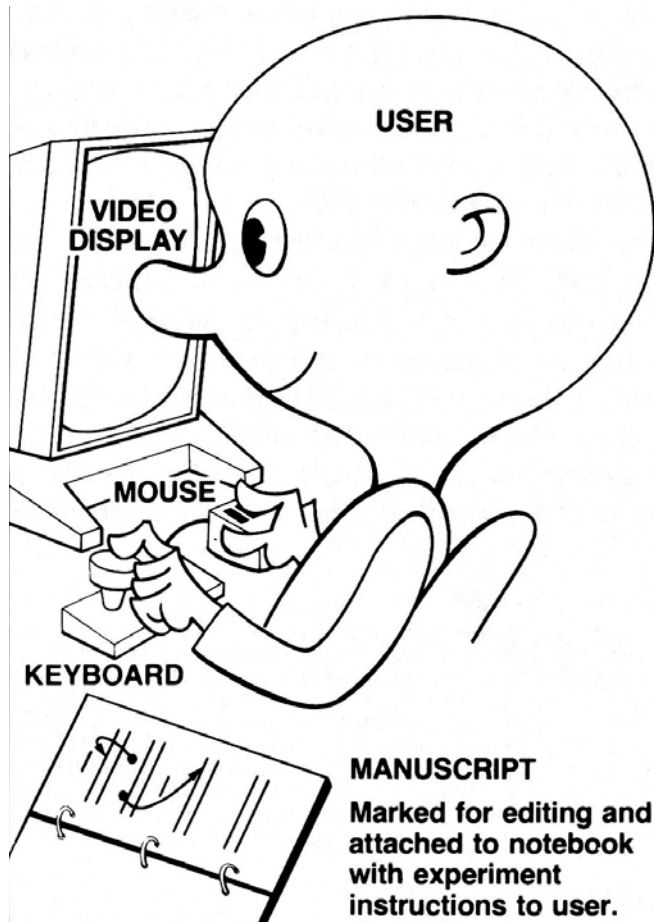
Overview of GOMS

Top-level goal	Edit manuscript, or, more specifically, move “quick brown” to before “fox”
Subgoal	Highlight text
Operators	Move-mouse Click mouse button Type characters (keyboard shortcuts)
Methods	For the editing goal: <ul style="list-style-type: none">1. Delete-word-and-retype (<i>retype</i> method)2. Cut-and-paste-using-keyboard-shortcuts (<i>shortcuts</i> method)3. Cut-and-paste-using-menus (<i>menus</i> method) For the highlighting subgoal: <ul style="list-style-type: none">1. Drag-across text (<i>dragging</i> method)2. Double-click first; shift-click last (<i>all-clicking</i> method)
Selection rules	For the editing goal: <ul style="list-style-type: none">If the text to be moved is one or two characters long, use <i>retype</i> methodElse, if remember shortcuts, use <i>shortcuts</i> methodElse, use <i>menus</i> method For the highlighting subgoal: <ul style="list-style-type: none">If the text to be moved is not whole words, use <i>dragging</i> methodElse, use <i>all-clicking</i> method



GOMS example I

- Setting



- Analysis

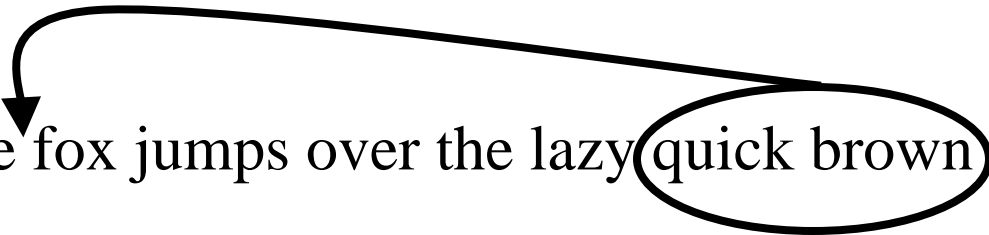
GOAL: EDIT-MANUSCRIPT

- . **GOAL: EDIT-UNIT-TASK** *repeat until no more unit tasks*
- . . **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.**

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-COMP0SED-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	0.20
•	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)

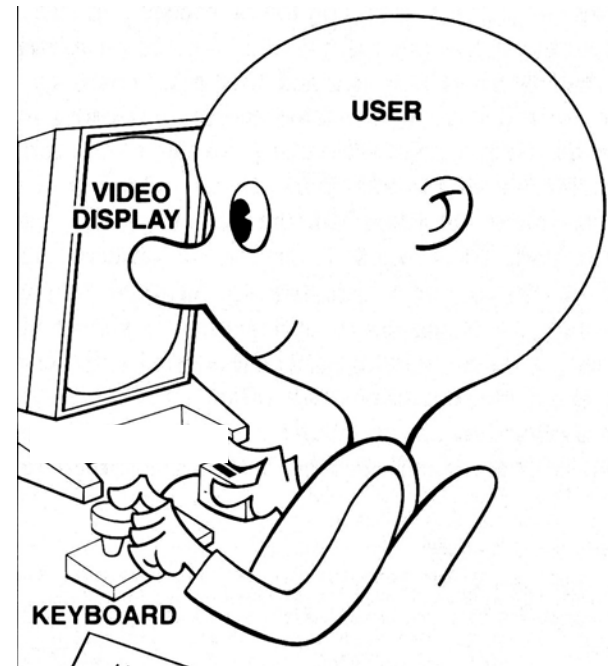
16.25

*Is all this
feedback in
order?*

*Issuing commands
will be used a lot!
can we shorten this
procedure? Consider
keyboard shortcuts.*

Keystroke Level Model (KLM)

- Focused on expert user
 - no selection rules (no decision making)
- 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 operators (K, P, H, R(t))
- Apply KLM rules [0-4] to add M's
- Transform R followed by an M
(computer and user can work at the same time)
 - 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 for Placing M

0: Insert M

- *In front of all K*
- *In front of all P's selecting a command (not arguments)*

1: Remove M between *fully anticipated* operators

- $PMK \rightarrow PK$

2: if a string of MKs belong to a *cognitive unit* delete all M but first

- $4564.23: MKMKMKMKMKMKMK \rightarrow MKKKKKKK$

3: if K is a *redundant terminator* then delete M in front of it

- $\swarrow\swarrow: MKMK \rightarrow MKK$

4a: if K terminates a constant string (command name) delete the M in front of it

- $cd\swarrow: MKKMK \rightarrow KKMK$

4b: if K terminates a variable string (parameter) keep the M in front of it

- $cd\ class\swarrow: MKKKMKKKKKMK \rightarrow MKKKMKKKKKMK$

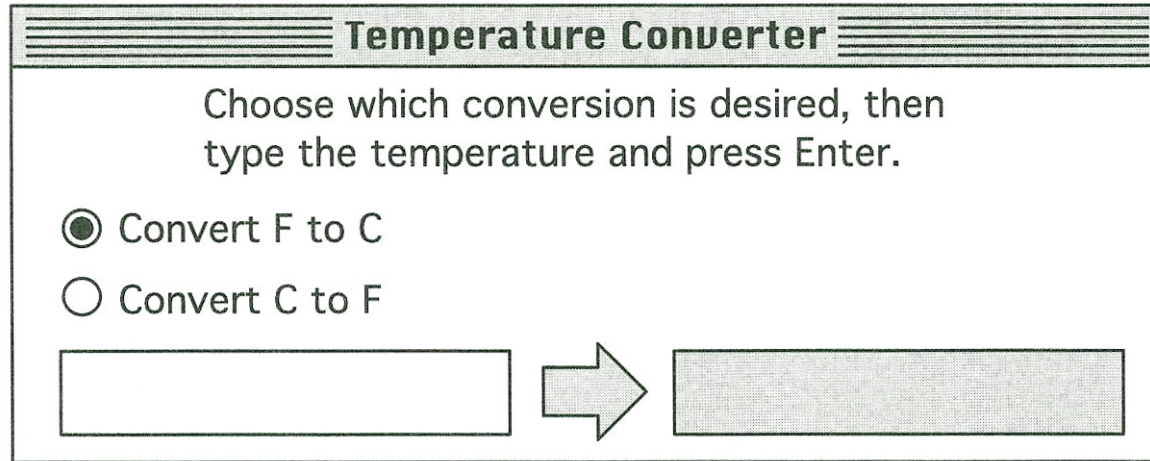
5: Delete overlapped Ms

- *do not count any portion of an M that overlaps an R*

Converting Temperature

(“Humane Interface”, Raskin)

- Convert 92.5F to Celsius



Temperature Converter

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

☒ Convert F to C
☐ Convert C to F

→

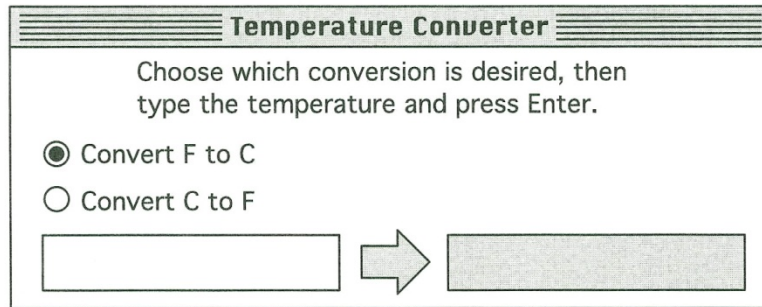
Assume :

- 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

- F to C: MKKKKMK (3.7s)
- C to F: HMPKHMKKKKMK (7.15s)

=> Average: 5.4s

- Move hand to the GID
H
- Point to the desired radio button
HP
- Click on the radio button
HPK

- Move hands back to the keyboard
HPKH
- Type the four characters
HPKHKKKK
- Tap Enter
HPKHKKKKK

→HMPMKHMKMKMKMKMK

→ HMPKHMKKKKMK

Converting Temperature

- Your design. Can you do better?
- Design an interface yourself
- Perform KLM analysis and report

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.

C

F

MKKKK => 2.15s

Pros and Cons

What are other factors?

- Learnability
- Accuracy
- Cognitive load

Temperature Converter

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

☒ Convert F to C
☐ Convert C to F

→

Temperature Converter

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

⇄ C
 F

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 GOMS?

- Possibly good for high value decisions
- Possibly good for making strong argument
- Definitely good for helping designers develop an intuition about what works and what doesn't and the impact of design decisions on speed
- Look at www.hcibib.org for work on GOMS