IE 403/476

Human-Computer Interaction

Week 6-Lec 1

Example

Suppose a user is writing some text using a text editor program. At some instant, the user notices a single character error (i.e., a wrong character is typed) in the text. In order to rectify it, the user moves the cursor to the location of the character (using mouse), deletes the character and retypes the correct character. Afterwards, the user returns to the original typing position (by repositioning the cursor using mouse).

Calculate the time taken to perform this task (error rectification) following a KLM analysis.

Something Missing!!

- What about M (mental operator) – where to place them in the list?
- It is usually difficult to identify the correct position of M
  - However, we can use some guidelines and heuristics

But what it is this M Operator?

- How was 1.35 arrived at?
  - Modeling variety of methods
  - $t_M = t_{execute} - (rest)$
  - t_M / N_{chunks}
  - *Number of Ms crucial in modeling*

M Placement Heuristics

- First, use one heuristic rule (Rule 0) to determine
  - Placement of candidate M operators
- Then, apply two heuristic rules (Rules 1–2) to determine
  - whether to delete each M

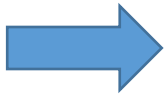
M placement Heuristics

General Rule:

- Initial insertion of candidate Ms
  - Insert Ms in front of all keystrokes (K)
  - Insert Ms in front of all acts of pointing (P) that select commands

– Mouse-operated widgets (like buttons, check boxes, radio buttons, and links) are considered commands

- Do not insert Ms before any P that points to an argument



– Text entry is considered as argument

Rule 0

92.5 C

M Placement Heuristics

- Rule 1: deletion of anticipated Ms
  - If an operator following an M is *fully anticipated* in an operator immediately preceding that M, then delete the M
  - Example : if user clicks the mouse with the intention of typing at the location, then delete the M as a consequence of rule 0 – YOU already know that you are going to type

With respect to prev slide example

M Placement Heuristics

- Rule 2: deletion of Ms within cognitive units
  - If a string of MKs belongs to a *cognitive unit* then delete all Ms except the first
  - A cognitive unit refers to a chunk of cognitive activities which is predetermined
  - Example - if a user is typing “100”, MKMKMK becomes MKKK (since the user decided to type 100 it constitutes a single cognitive unit)

Another Example: deleting a word

Using Mouse to select word

- M
- P [start of word]
- BB [click]
- M
- P [end of word]
- K [shift]
- BB [click]
- H [to keyboard]
- M
- K [Del]
- Total: 3M + 2P + 4B + 1K = 6.93 sec

Pressing Del key *n* times

- M
- P [start of word]
- BB [click]
- H
- M
- K [Del]
- × *n* [where *n* = length of word]
- Total: 2M+P+2B+H+*n*K
- = 4.36 + 0.28*n* sec

KLM Limitations

- Although KLM provides an easy-to-understand-and-apply predictive tool for interactive system design, it has few significant constraints and limitations
  - It can model only “expert” user behavior
  - User errors can not be modeled

;

GOMS MODEL

GOMS,

- a hierarchical cognitive (thought) process is assumed

KLM

- Elementary (cognitive) steps or operators required to carry out a complex interaction task
  - The listing of operators implies a linear and sequential cognitive behaviour

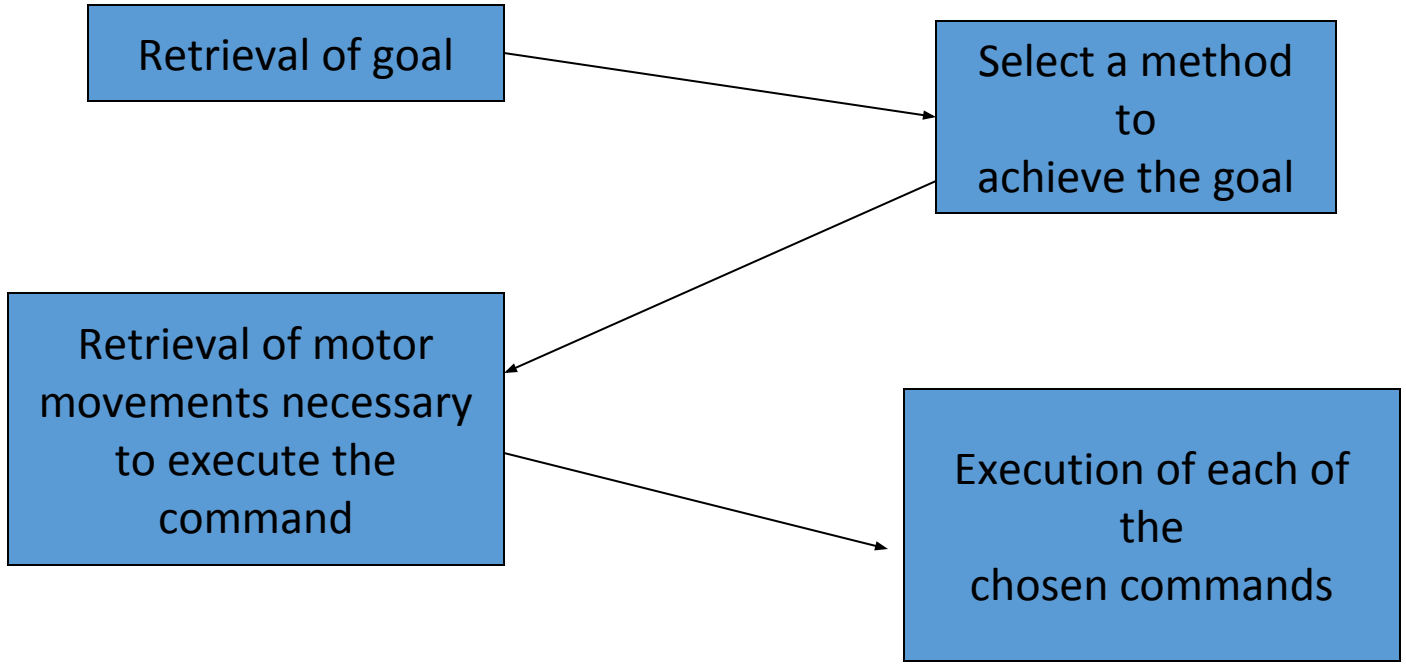
- Both assume error-free and 'logical' behavior
  - A logical behavior implies that we think logically, rather than driven by emotions

GOMS: Models user's behavior in terms of:

- Goals
  - What the user wants to do.
- Operators
  - Specific steps a user is able to take and assigned a specific execution time.
  - the notion of operators is not restricted to those seven
    - The modeler has the freedom to define any "elementary" cognitive operation and use that as operator
- Methods
  - Well-learned sequences of subgoals and operators that can accomplish a goal.
- Selection Rules
  - Guidelines for deciding between multiple methods.

Execution of a Mental Step

- GOMS allows explicit representation of mental steps of a task (the “Cognitive Processor”):



General Example

- Goal (the big picture)
  - Go from hotel to the airport
- Operators:
  - Walk, take bus, take taxi, rent car, take train
- Methods
  - Locate bus stop; wait for bus; get on the bus
- Selection rules (choosing among methods)?
  - Example: Walking is cheaper, but tiring and slow
  - Example: Taking a bus is complicated abroad

How to do GOMS analysis

- Generate task description
  - Pick high-level user **Goal**
  - Write **Methods for reaching Goal - may invoke subgoals**
  - Write **Methods for subgoals**
    - This is recursive
    - Stops when **Operators are reached**
- Evaluate description of task
- Apply results to UI
- Iterate!

GOMS: A Family of Models

- Keystroke-Level Model (KLM)
- **Card, Moran, and Newell (CMN-GOMS)**
- Natural GOMS Language (NGOMSL)
- Cognitive-Perceptual-Motor GOMS (CPM-GOMS)

General Example

- Goal: Edit text
- Operators
  - Use arrow keys
  - Use mouse
  - Use other keys
- Method: Delete text (sub-goal)
  - Positioning: 1) arrow key 2) mouse
  - Marking: 1) double click 2) use mouse
  - Delete (and add text): 1) start writing 2) press delete, then write new text
- Selection rules: if close, use arrow key etc.

- Goal: delete text (*n* chars long)
- Select:
  - method 1 if $n > 10$
  - method 2 if $n < 10$
- Method 1: Goal: highlight text & delete
  - Goal: highlight text
    - Point
    - Click
    - Point
    - Shift
    - Click
- Method 2: Goal: delete n char

Description	Duration (sec)
GOAL: MOVE-TEXT	
…….**GOAL: CUT-TEXT**	
……**GOAL: HIGHLIGHT-TEXT**	
……..MOVE-CURSOR-TO-BEGINNING	1.10
……..CLICK-MOUSE-BUTTON	0.20
……..MOVE-CURSOR-TO-END	1.10
……..SHIFT-CLICK-MOUSE-BUTTON	0.48
……..VERIFY-HIGHLIGHT	1.35
…….....**GOAL: ISSUE-CUT-COMMAND**	
……MOVE-CURSOR-TO-EDIT-MENU	
……PRESS-MOUSE-BUTTON	0.10
……MOVE-MOUSE-TO-CUT-ITEM	1.10
……VERIFY-HIGHLIGHT	1.35
……RELEASE-MOUSE-BUTTON	0.10
……**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	
……PRESS-MOUSE-BUTTON	0.10
……MOVE-MOUSE-TO-PASTE-ITEM	
……VERIFY-HIGHLIGHT	1.35
……RELEASE-MOUSE-BUTTON	0.10
TOTAL PREDICTED TIME	**14.38**

Example

- Suppose we want to find out the definition of a word from an online dictionary. How can we model this task with (CMN)GOMS?

Example

- We shall list the goals (high level tasks) first
  - Goal: Access online dictionary (first, we need to access the dictionary)
  - Goal: Lookup definition (then, we have to find out the definition)

Example

- Next, we have to determine the methods (operator or goal-operator sequence) to achieve each of these goals
  - Goal: Access online dictionary
    - Operator: Type URL sequence
    - Operator: Press Enter

Example

- Next, we have to determine the methods (operator or goal-operator sequence) to achieve each of these goals
  - Goal: Lookup definition
    - Operator: Type word in entry field
    - Goal: Submit the word
      - Operator: Move cursor from field to Lookup button
      - Operator: Select Lookup
    - Operator: Read output

Example

- Thus, the complete model for the task is
  - Goal: Access online dictionary
    - Operator: Type URL sequence
    - Operator: Press Enter
  - Goal: Lookup definition
    - Operator: Type word in entry field
    - Goal: Submit the word
      - Operator: Move cursor from field to Lookup button
      - Operator: Select Lookup button
    - Operator: Read output

Example

- Notice the hierarchical nature of the model
- Note the use of operators
  - The operator “type URL sequence” is a high-level operator defined by the modeler
  - “Press Enter” is a keystroke level operator
  - Note how both the low-level and high-level operators co-exist in the same model

Example

- Note the use of methods
  - For the first goal, the method consisted of two operators
  - For the second goal, the method consisted of two operators and a sub-goal (which has a two-operators method for itself)