Interaction Design

GOMS + Fitts' Law + Hick's Law

This Lecture

Rule Based Evaluation

- GOMS and KLM
- Fitts' law
- Hicks law

GOMS and KLM

GOMS

- Describe the user behaviour in terms of
  - Goals
    - Edit manuscript, locate line
  - Operators
    - Elementary perceptual, motor or cognitive acts
  - Methods
    - Procedure for using operators to accomplish goals
  - Selection rules
    - Used if several methods are available for a given goal

- Family of methods
  - KLM, CMN-GOMS, NGOMSL, CPM-GOMS

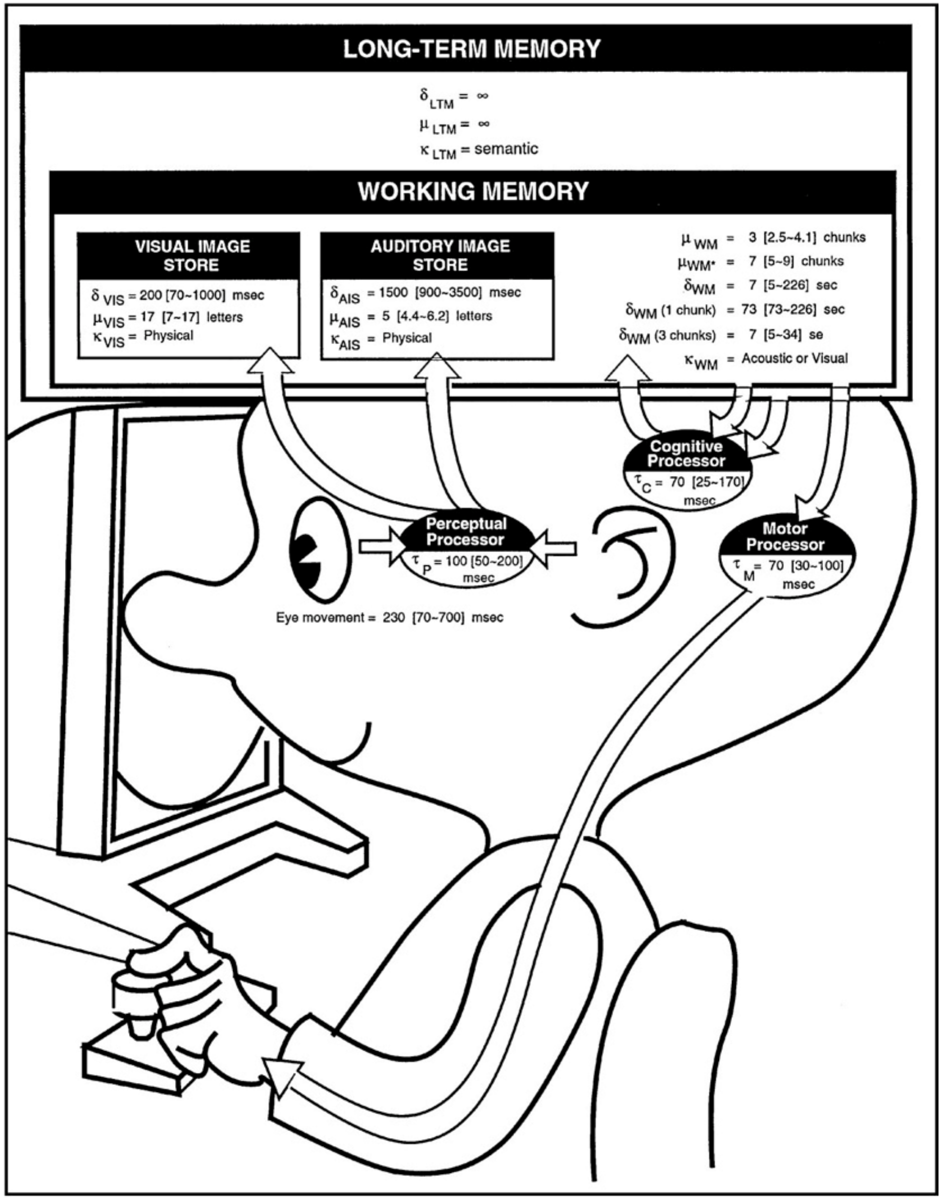
Sources:

- Card, S.K. et al. (1980). The keystroke-level model for user performance time with interactive systems. Communications of the ACM , 23(7), 396-410.
- http://web.eecs.umich.edu/~kieras/docs/TA_Modeling/GOMSforTA.pdf

Using GOMS Analysis

- Check that frequent goals can be achieved quickly
- Making operator hierarchy is often the value
  - Functionality coverage & consistency
    - Does UI contain needed functions?
    - Consistency: are similar tasks performed similarly?
  - Operator sequence
    - In what order are individual operations done?

MHP



LONG-TERM MEMORY
δ LTM = ∞
μ LTM = ∞
κ LTM = semantic
WORKING MEMORY
VISUAL IMAGE STORE
δ VIS = 200 [70~1000] msec
μ VIS = 17 [7~17] letters
κ VIS = Physical
AUDITORY IMAGE STORE
δ AIS = 1500 [900~3500] msec
μ AIS = 5 [4.4~6.2] letters
κ AIS = Physical
μ WM = 3 [2.5~4.1] chunks
μ WM* = 7 [5~9] chunks
δ WM = 7 [5~226] sec
δ WM (1 chunk) = 73 [73~226] sec
δ WM (3 chunks) = 7 [5~34] se
κ WM = Acoustic or Visual
Cognitive Processor
τ C = 70 [25~170] msec
Perceptual Processor
τ P = 100 [50~200] msec
Motor Processor
τ M = 70 [30~100] msec
Eye movement = 230 [70~700] msec

Interacting with a GUI

In many ways, interacting with a GUI is similar to solving a problem …

It involves:

- establishing a goal (e.g. editing a text)
- forming an intention (or sub-goal) (e.g. deleting a phrase)
- specifying a sequence of actions (method)
- executing the actions
- perceiving the state of the system
- interpreting the state of the system
- evaluating the state of the system with respect to the goals and intentions

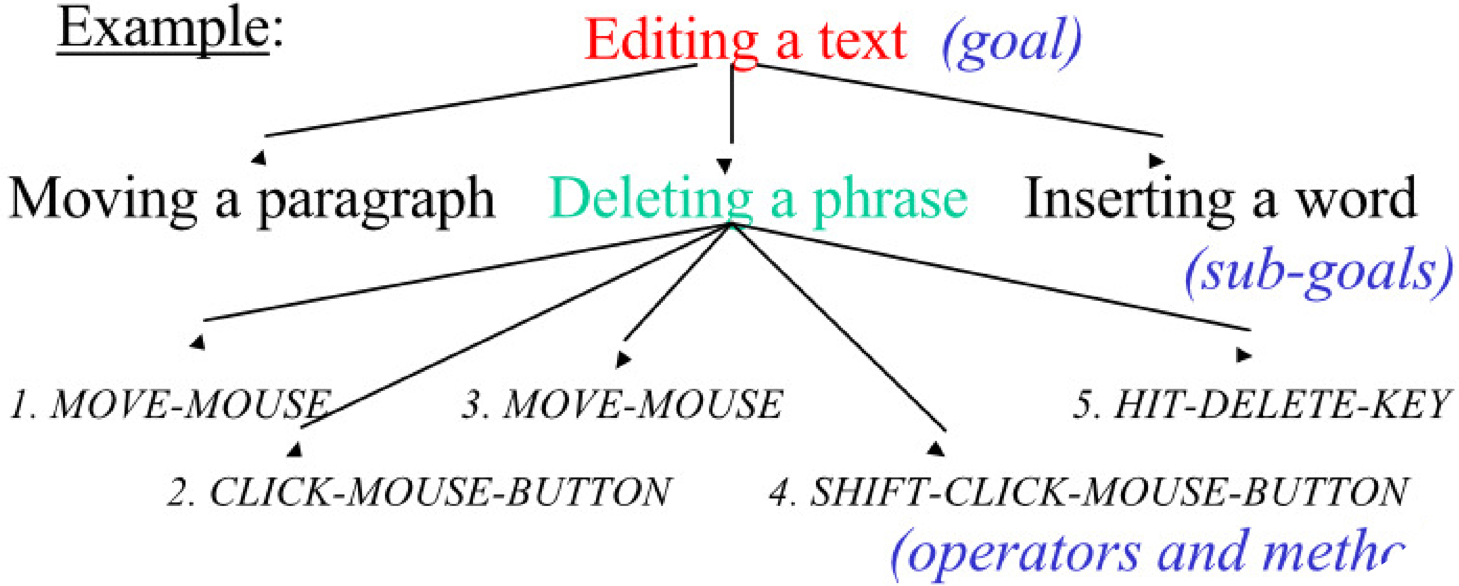
How to do GOMS Analysis

- 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 user interface (UI)
- Iterate!

Analysis of Interaction

Usually, the means of accomplishing the goal are reduced to a certain set of methods and operators (actions).

Example:



The GOMS models – principle

(Goals Operators Methods and Selections rules)

The lowest-level operators in an analysis should present important properties (e.g. execution time) that are constant.

The time it takes the user-computer system to perform a task can then be calculated by summing the times it takes for the system to perform the serial operators that the task comprises.

⟶ GOMS allows you to predict how long a user will take to perform a particular task when using a given GUI design.

The GOMS Keystroke-Level Model (KLM)

The operators are termed keystroke level if they are at the level of actions like pressing keys, moving the mouse, pressing buttons, etc.

There is a standard set of operators for use in GOMS, whose execution times have been estimated from experimental data.

Keying (K), Pointing (P), Homing (H), Mentally preparing (M) and Responding (R)

Keying

Keying (K = 0.2 sec) : the time it takes to tap a key on the keyboard or click with a mouse button.

In practice, typing speed varies widely …

K = 0.08 sec for highly skilled typist (135 wpm)
K = 0.2 sec for a typical skilled typist (55 wpm)
K = 0.88 sec for an average unskilled typist (40 wpm)
K = 1.2 sec for a novice

In addition, typing speed is not independent of what is being typed …

Pointing

Pointing (P = 1.1 sec) : the time it takes a user to point to a position on a display.

The actual time required can be determined from Fitt's law (next week's lecture).

For typical situations, it ranges from 0.8 to 1.5 sec, with an average of 1.1 sec.

If great accuracy is not required, or the movement distances or target sizes are not unusual, this average can be used.

Homing

Homing **(H = 0.4 sec)** : the time it takes a user's hand to move from the keyboard to the mouse or from the mouse to the keyboard.

Since the target are pretty large, and the movement well practiced, moving the hand between keyboard and mouse, and vice versa, is relatively fast.

Mentally preparing

Mentally preparing (M = 1.35 sec) : the time it takes a user to prepare mentally for the next step.

How long it takes to perform a mental act depends on the cognitive processes involved …

This operator is based on the fact that when reasonably experienced users are engaged in routine operation of a computer, there are pauses in the stream of actions (about 1 sec long) that are associated with routine acts such as remembering a filename or finding something on the screen.

Responding

Responding (R) : the time a user must wait for a computer to respond to input.

If a user operates a control and nothing appears on the display for more than approximately 250 msec, s/he is likely to become uneasy, to try again, or to begin to wonder whether the system is failing.

→ It is important that interfaces provide feedback if delays are unavoidable (e.g. a progress bar that accurately reflects the time remaining).

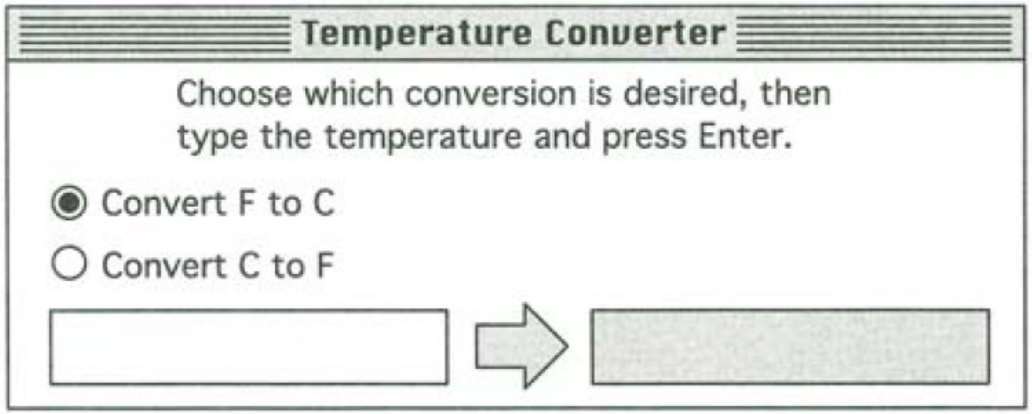
Keystroke Level Model (KLM): Summary

Describe the task using the following operators:

- K: pressing a key or a pressing (or releasing) a button
  - tK = 0.08 - 1.2s (0.2 good rule of thumb)
- P: pointing
  - tP = 1.1s (without button press)
- H: Homing (switching device)
  - tH = 0.4s
- M: Mentally prepare
  - tM = 1.35s
- R(*t*): system response time
  - tR = t

Example: KLM Calculations
Converting Temperature (1)

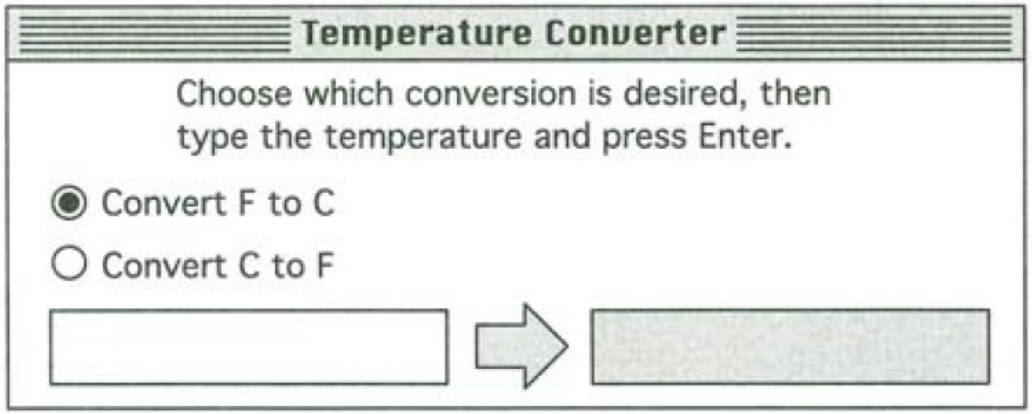
- Convert **92.5**



Assume the focus is on the dialog box, so typing on the keyboard will enter text in the text field directly

Example: KLM Calculations
Converting Temperature (1)

- Convert **92.5**

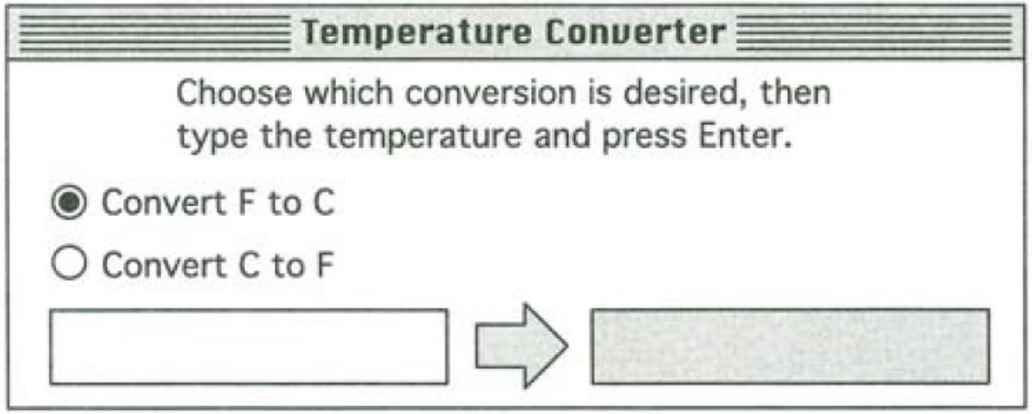


Assume the focus is on the dialog box, so typing on the keyboard will enter text in the text field directly

Step 1: K K K K K
Step 2: MK MK MK MK MK
Step 3: MKKKK MK **= 3.7sec**

Example: KLM Calculations Converting Temperature (2)

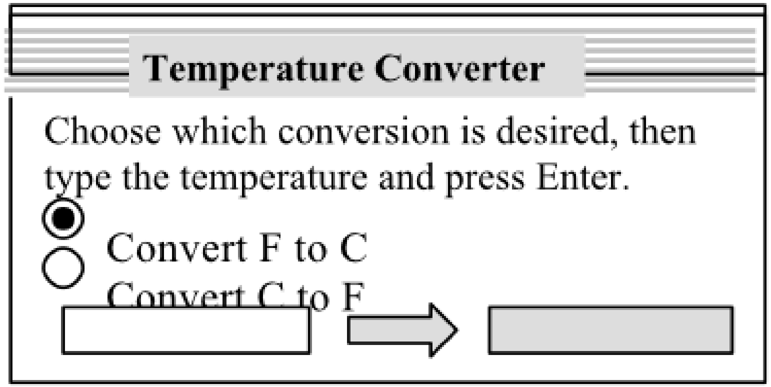
- The hand is on keyboard
- The wrong conversion is selected - convert **25.5C to F**



Assume the focus is on the dialog box, so typing on the keyboard will enter text in the text field directly

KLM calculations (1)

Example



- move hand to the mouse: H
- point to the desired radio button: P
- click on the radio button: K
- move hands back to the keyboard: H
- type four characters: KKKK
- tap ENTER: K

KLM calculations (2)

Listing the gestures (K, P and H) is the easy part … The more difficult part is figuring out at what points the user will stop to perform an unconscious mental operation (the mental preparation M times).

<u>Examples of activities that take an M</u>:

- initiating a task
- making a strategy decision
- retrieving a chunk from memory
- finding something on the screen
- thinking of a task parameter
- verifying that an action is correct

Heuristics for placing mental operations (0)

Rule 0: Initial insertion of candidates Ms

Insert Ms in front of all Ks.
Place Ms in front of all Ps that select commands

(but do not place Ms in front of any Ps that point to arguments of those commands.)

Example:

H PK H KKKK K

H MPMK H MKMKMKMK MK

Heuristics for placing mental operations (1)

Rule 1: Deletion of anticipated Ms

If an operator following an M is fully anticipated in an operator just previous to that M, then delete that M.

For example if you move the mouse with the intent of tapping a mouse button when you reach your target, then you delete the M before the K.

Example: **H MPMK H MKMKMKMK MK**

Heuristics for placing mental operations (2)

Rule 2: Deletion of Ms within cognitive units

If a string of MKs belongs to a cognitive unit,
Then delete all the Ms but the first.

A cognitive unit is a contiguous sequence of typed characters that form a command name or that is required as an argument to a command.

Example: **H** **MPK** **H** **MKMKMKMK** **MK**

Heuristics for placing mental operations (3)

Rule 3: Deletion of Ms before consecutive terminators

If a K is a redundant delimiter at the end of a cognitive unit, such as the delimiter of a command Immediately following the delimiter of its argument, then delete the M in front of it.

<u>Example</u>: *if a command is followed by ESC ENTER, do not place M between ESC and ENTER*

Heuristics for placing mental operations (4)

Rule 4: Deletion of Ms that are terminators of command

If a K is a delimiter that follows a constant string (e.g. a command name) then delete the M in front of it. But if the K is a delimiter for an argument string or any string that can vary, then keep the M in front of it.

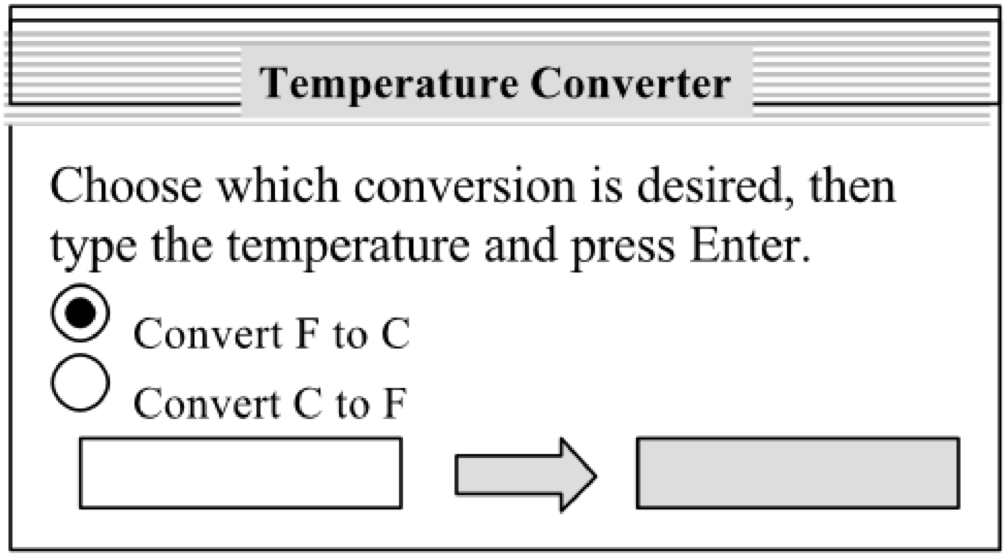
Example: *if an ENTER is needed after a command DIR, do not place a M before ENTER*

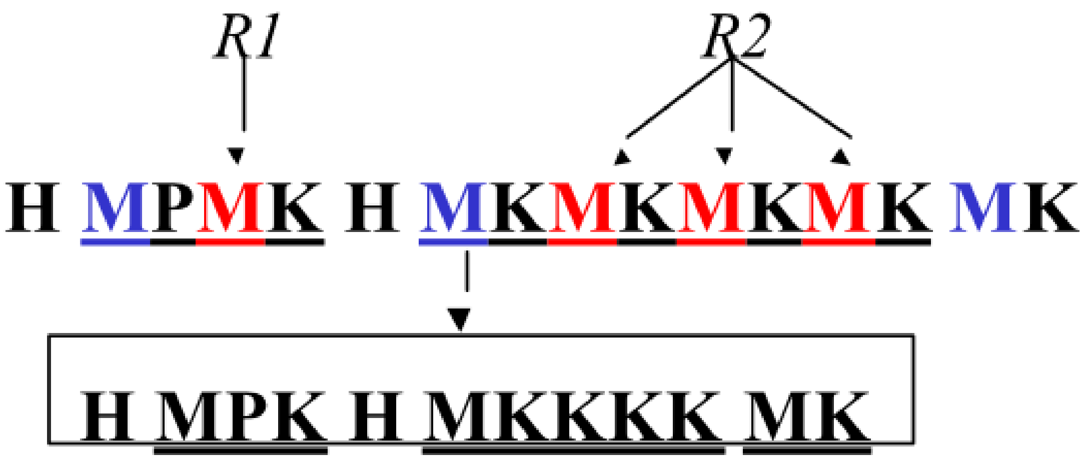
Heuristics for placing mental operations (5)

Rule 5: Deletion of overlapped Ms

Do not count any portion of an M that overlaps an R (i.e. a delay with the user waiting for a response from the computer).

Example


Temperature Converter
Choose which conversion is desired, then type the temperature and press Enter.
Convert F to C
Convert C to F


R1
R2
H MPMK H MKMKMKMK MK
H MPK H MKKKK MK

Task Time Calculation for the Temperature Converter

K = 0.2   P = 1.1   H = 0.4   M = 1.35

H <u>MPK</u> H <u>MKKKK</u> <u>MK</u> = 7.15 sec

When the correct conversion is already selected:

<u>MKKKK</u> <u>MK</u> = 3.7 sec

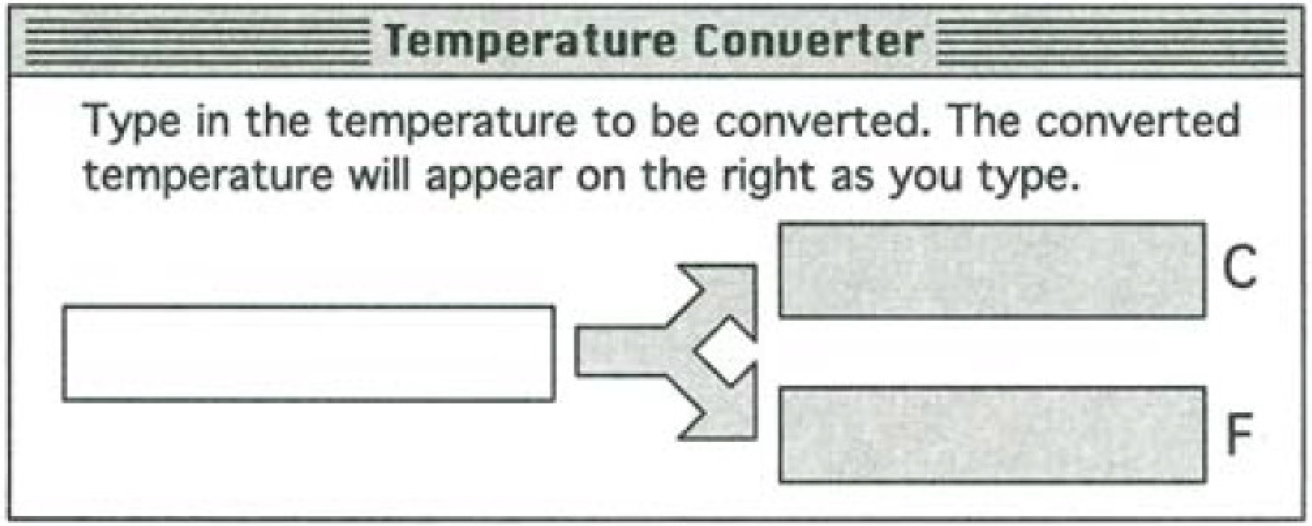
Example: KLM Calculations
Converting Temperature – Design 2

- Simple text interface with the following prompt:

"To convert temperatures, type the numeric temperature, followed by C if it is in degrees Celsius or F it is in degrees Fahrenheit. Then press enter key. The converted temperature will be displayed"

- Convert **92.5**

Example: KLM Calculations
Converting Temperature – Design 3



- Convert **92.5**

Implications of Operators Times for Interaction Style Choices

Keystrokes are fast

Ms, such as memory retrievals and visual searches are very slow

Mouse moves are very slow

Switching hand between mouse and keyboard is moderately slow

What GOMS Can Model

- GOMS helps with comparing different UI designs
- Task must be goal-directed
  - Some activities are more goal-directed
    - Creative activities may not be as goal-directed
- Task must be a routine cognitive skill
  - As opposed to problem solving
  - Good for things like machine operators

Advantages of GOMS

- Gives qualitative & quantitative measures
- Model explains the results
- Less work than user study – no users!
- Easy to modify when UI is revised

- Research: Need tools to aid modelling process since it can still be tedious

Disadvantages of GOMS

- Not as easy as other evaluation methods
  - E.g. Heuristic evaluation
- Takes lots of time and effort
- Only works for goal-directed tasks
- Assumes tasks **expert** performance without **error**
- Does not address several UI issues,
  - readability, memorisability of icons, commands

Fitts' Law + Hick's Law

Fitts' law and Hick's law

Fitts: Estimate movement time to select a target on a computer display.

Hick: Estimate time to make a selection decision.

Fitts' law

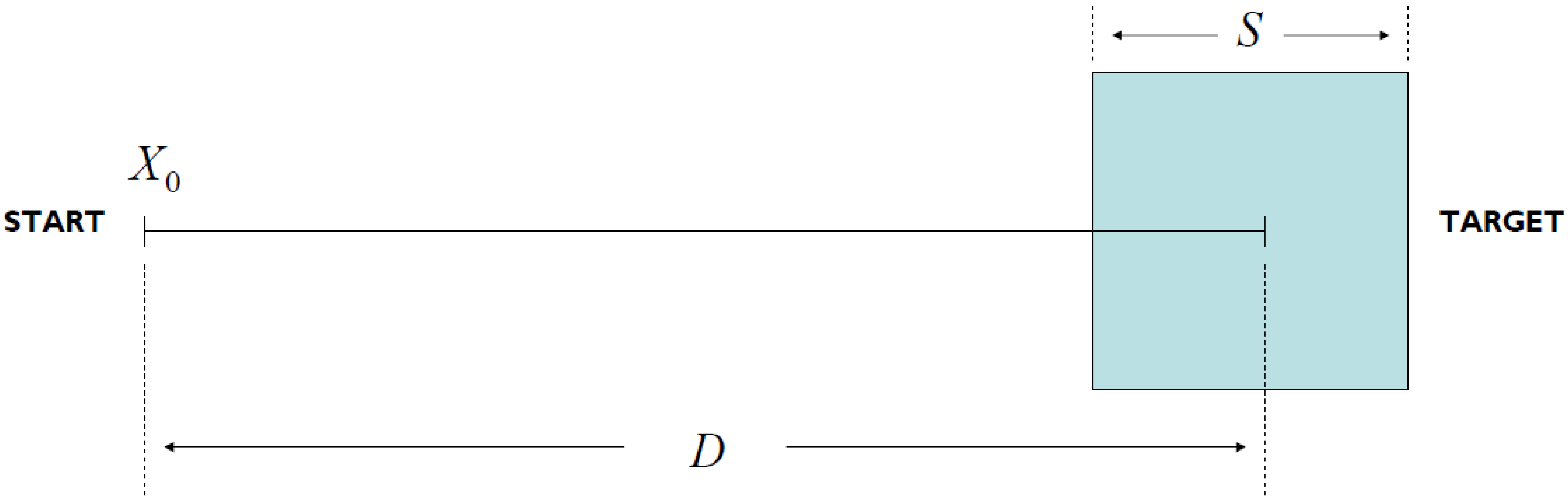
Fitts' law is a robust model of human psychomotor behaviour developed in 1954.

It enables the prediction of human movement and human motion based on *rapid, aimed* movement (not drawing or writing)

Movement time is affected by the *distance* moved and the *precision* demanded by the size of the target …

The precision is measured by the target's "index of difficulty".

Fitts' Law



Time　　Distance

$$T = a + b \; \mathbf{log\_2} \; (\; 2 \; D/S \;)$$

Coefficients　　Size/ Width

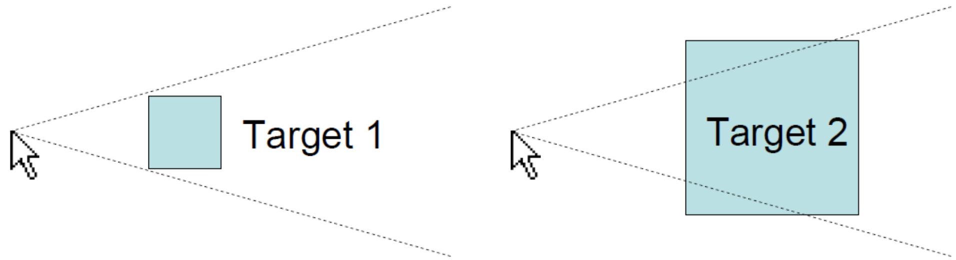
a, b = constants (empirically derived)
D = distance
S = size

ID is Index of Difficulty = $\log_2(D/S+1)$

Fitts' Law

- Models well-rehearsed selection task
  - T increases as the ***distance*** to the target increases
  - T decreases as the ***size*** of the target increases
- Considers distance and target size

$$T = a + b\log_2(D/S+1)$$

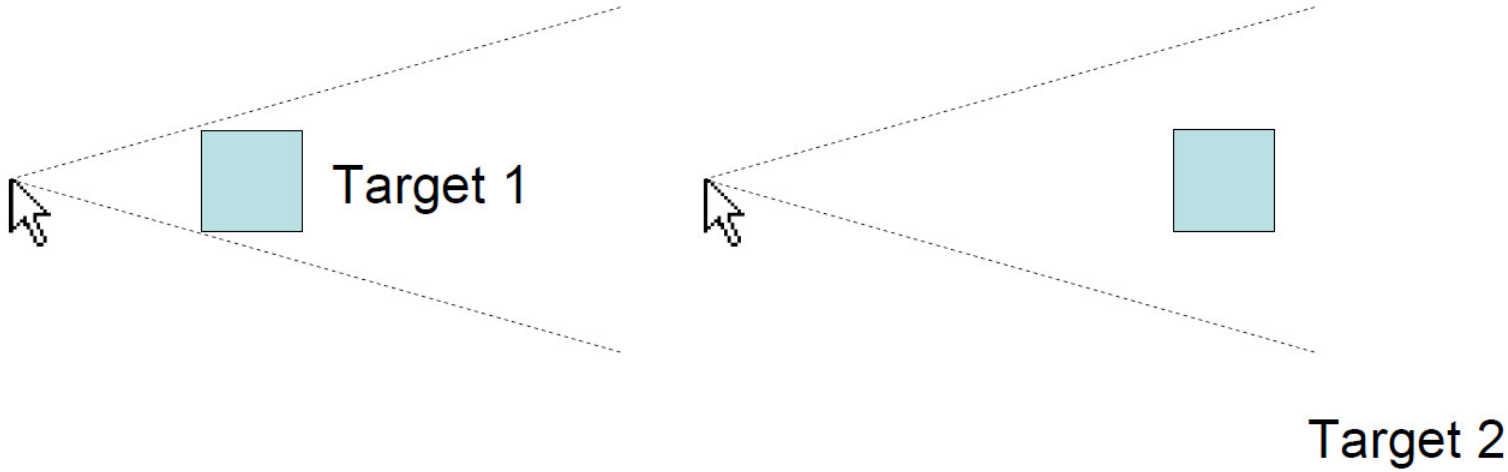


Smaller ID → Easier

Fitts' Law

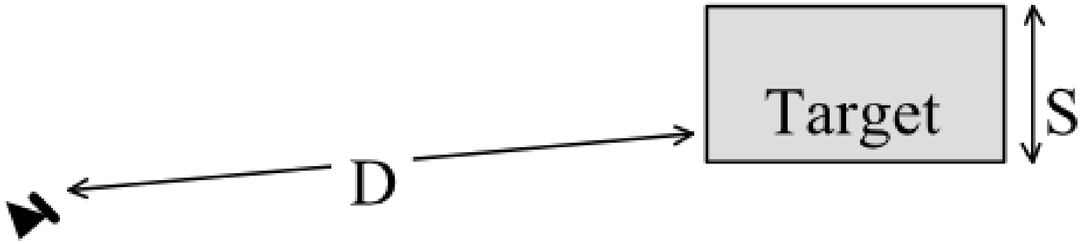
- Considers distance and target size

$$T = a + b\log_2(D/S + 1)$$



Larger ID → Harder

Fitts' law : Index of Difficulty



Index of Difficulty : **ID = $\log_2$ (D / S+1)** bit

D = distance or amplitude to move (length of a straight line from the position at which the cursor started to the closest point on the target)

S = size or tolerance of the region within which the move terminates

Fitts' law : Movement Time (MT)

MT = a + b ID = a + b log2 (D / S+1) ms

… where the coefficients a and b are determined experimentally and are mainly dependent on the pointing device.

Fitts' law : Applicability

- Fitts' law applies only to the kind of motions we make when we are using most GUIs: **motions that are small** relative to human body size and that are **uninterrupted** (i.e. movements that can be made in one continuous motion).

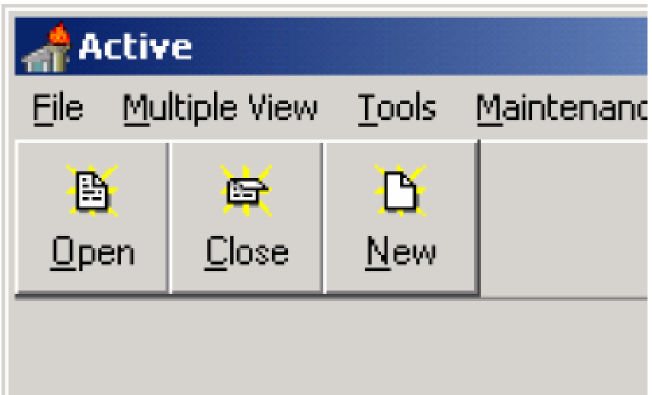
- The law can be used to assist in GUI design (estimation of task execution time, choice of objects' sizes)

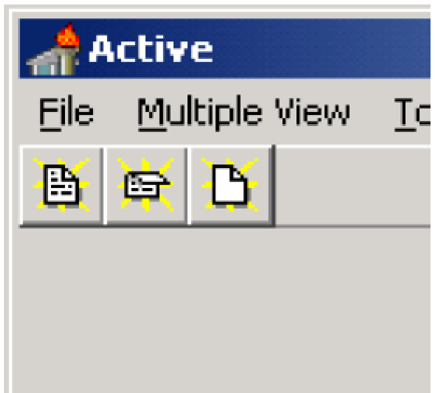
- Commonly, Fitts' law is used to evaluate input devices.

Activity: MS Toolbar

- Microsoft Toolbars offer the user the option of displaying a label below each tool. Name at least one reason why labelled tools can be accessed faster.

  (Assume, for this, that the user knows the tool.)





Hick's law

The more choices of a given kind you have, the longer it takes you to come to a decision.

$$DT = a + b \log_2 (n+1)$$

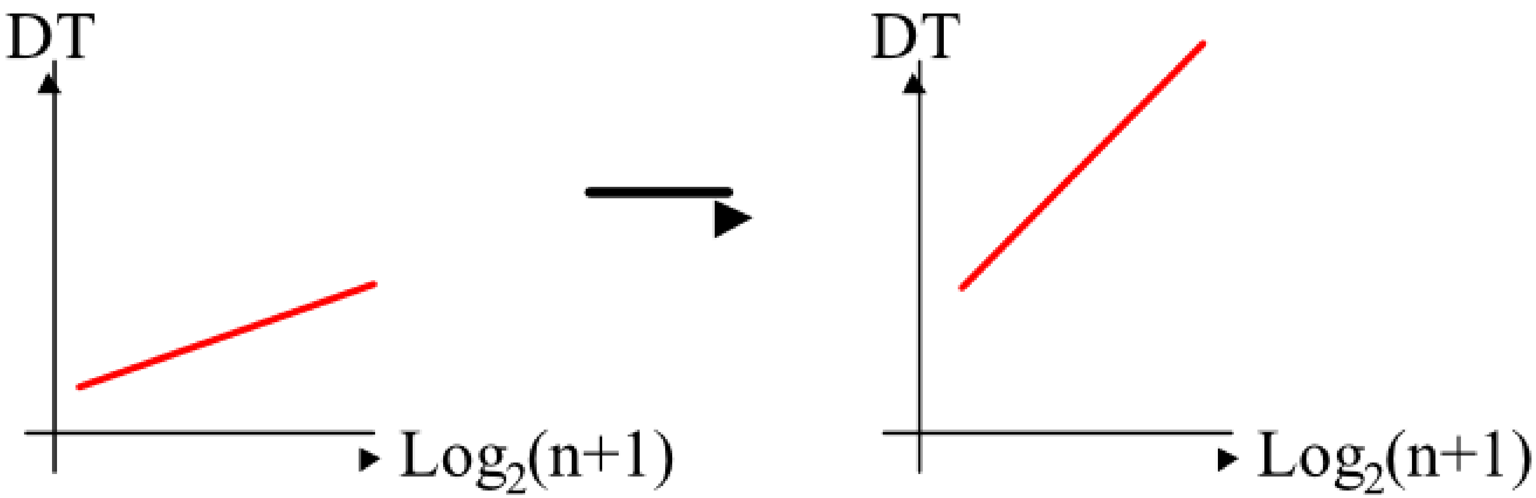
n → number of choices

… where the coefficients a and b are dependent on many conditions, including how the choices are presented and how habituated to the system the user has become.

Hick's law : Effect of Layout

$$DT = a + b \log_2 (n+1)$$

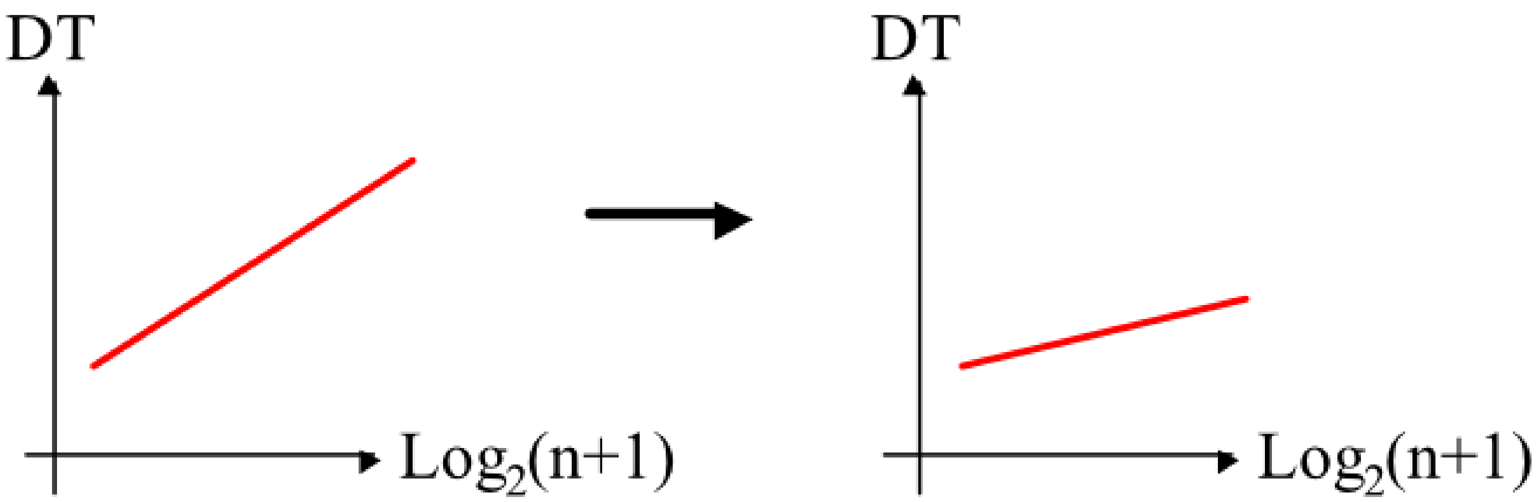
If the choices are presented in a confusing manner, both a and b can increase.



Hick's law : Effect of Habituation

$$DT = a + b \log_2 (n+1)$$

Habituation decreases b.



Summary

- GOMS
  - Predicts time to complete tasks
- Hick’s and Fitts’
  - Finer graining time prediction

- Based on cognitive psychology research
- No users involved in studies
- Very low-level analysis