Human Performance Modeling

Irene Rae
Computer Sciences

CS-570 Introduction to Human-computer Interaction

User Performance Modeling

Quantitatively predict what people will do before you build a system.

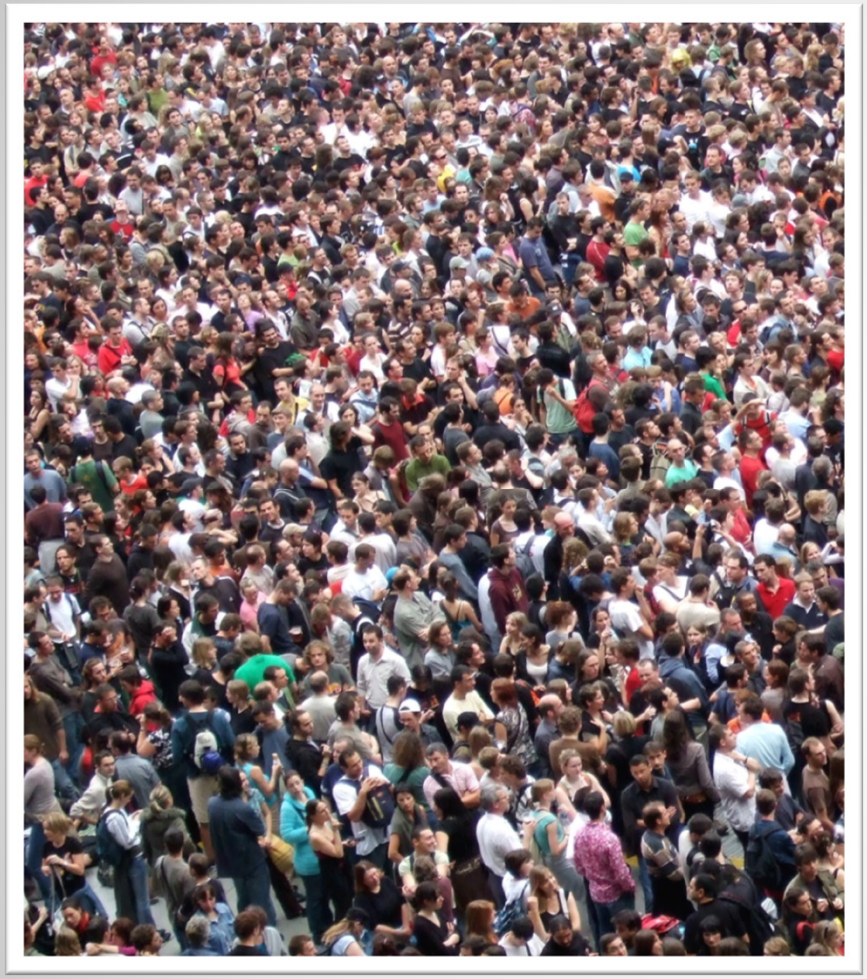
Also called:

- Human Behavior Modeling (military)
- Cognitive Modeling
- Computational Cognitive Modeling
- Cognitive Performance Modeling

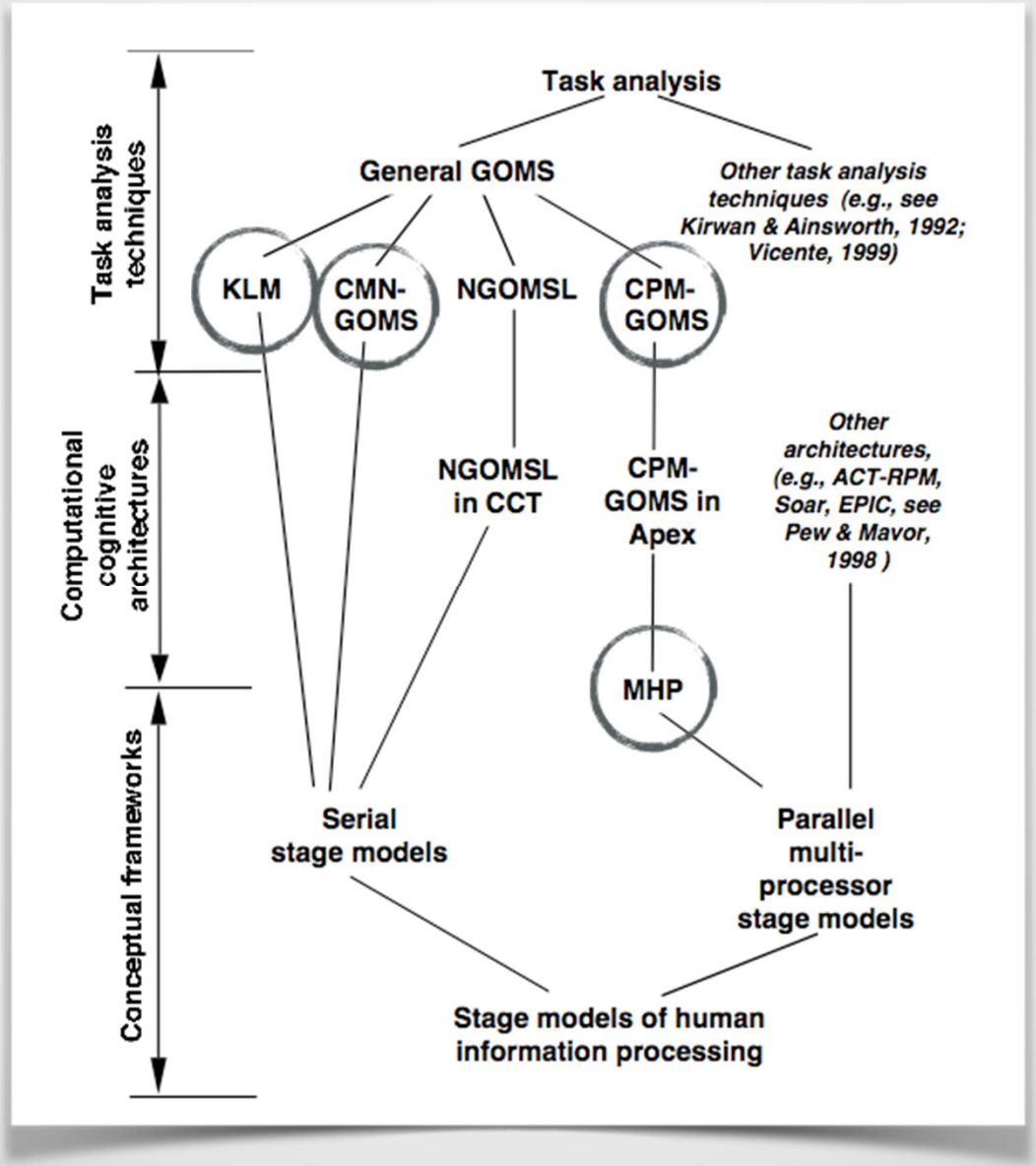
Why Model?

Task	Greenphone	iPhone	Neo1973
Add a contact	21.022 sec 22 keystrokes	20.150 sec 20 keystrokes	40.018 sec 26 keystrokes
Send an SMS	14.912 sec 14 keystrokes	16.100 sec 10 keystrokes	18.025 sec 11 keystrokes
Add a number from a call	22.240 sec 12 keystrokes	16.600 sec 8 keystrokes	18.868 sec 13 keystrokes
Call a contact	4.921 sec 4 keystrokes	10.600 sec 4 keystrokes	7.988 sec 5 keystrokes
Change a contact name	22.326 sec 28 keystrokes	24.100 sec 16 keystrokes	41.688 sec 16 keystrokes

Modeling Humans



Taxonomy of Modeling



Model Human Processor (MHP)

Goals

- Provide approximate predictions of gross human behavior
- Help remember theory & facts about HCI
- Provide "good enough" answers to guide design
- Provide explanations for & support empirical results

Questions the MHP **can** answer

How long will it take to do a task on the computer?

Prediction

Is design A likely to be "better" (faster) than design B?

Explanation

Why is design A "better" (faster) than design B?

Questions MHP **can't** answer

User preferences, likeability, desirability

Complex problem-solving & learning

Questions dealing with the subtleties of human behavior, creative thoughts, humor, etc.

Example:

When I knock on the table,

you knock on the table.

Observation:

Knock

… short pause …

knock

Why the pause?

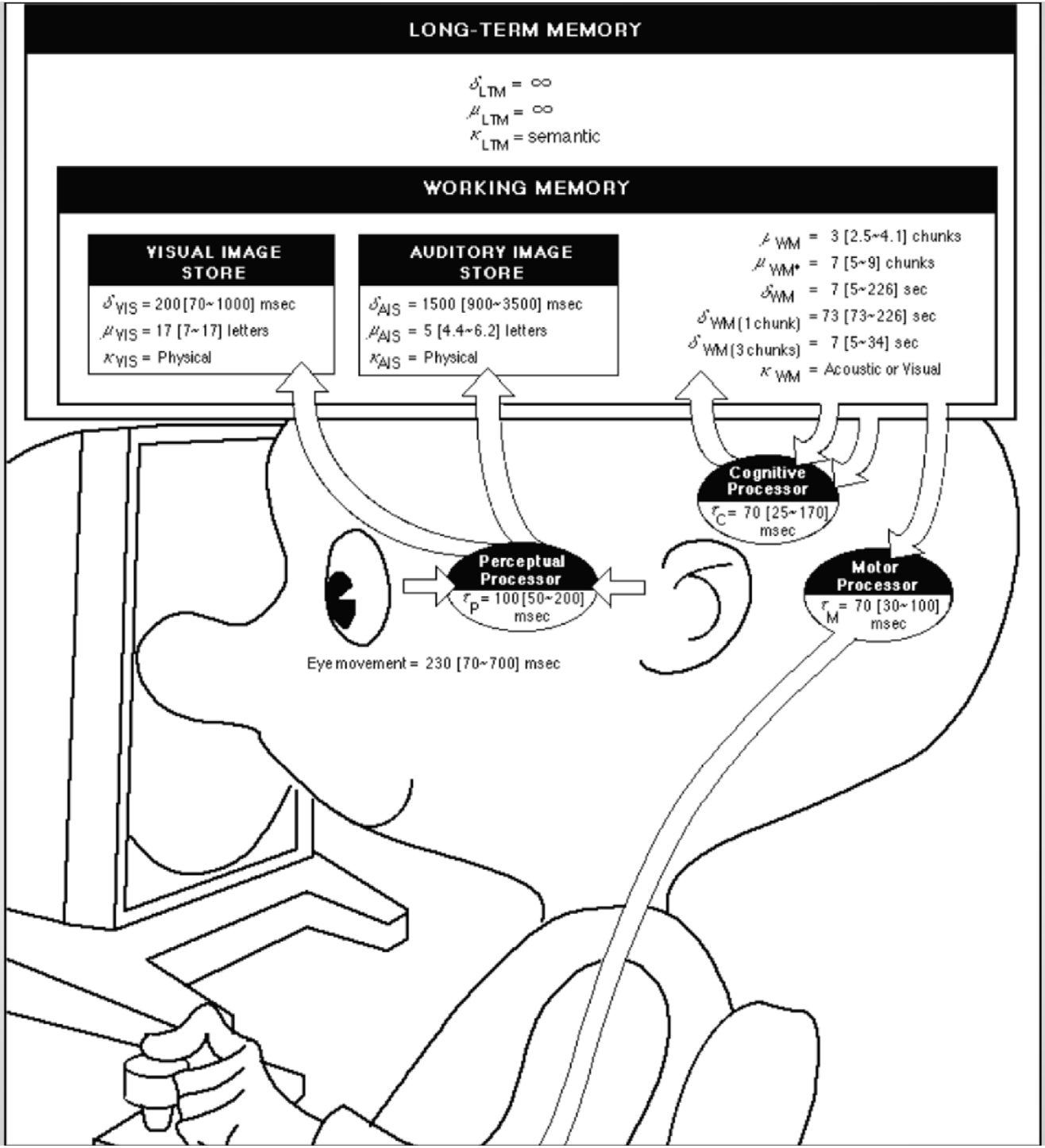
Model Human Processor Subsystems

Three interacting systems

- Perceptual system — hear knock
- Cognitive system — process knock
- Motor system — respond with knock

Each subsystem has:

- A processor
- Memories it interacts with


LONG-TERM MEMORY
$\delta_{LTM} = \infty$
$\mu_{LTM} = \infty$
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
$\delta_{WM(1\,chunk)}$ = 73 [73~226] sec
$\delta_{WM(3\,chunks)}$ = 7 [5~34] sec
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

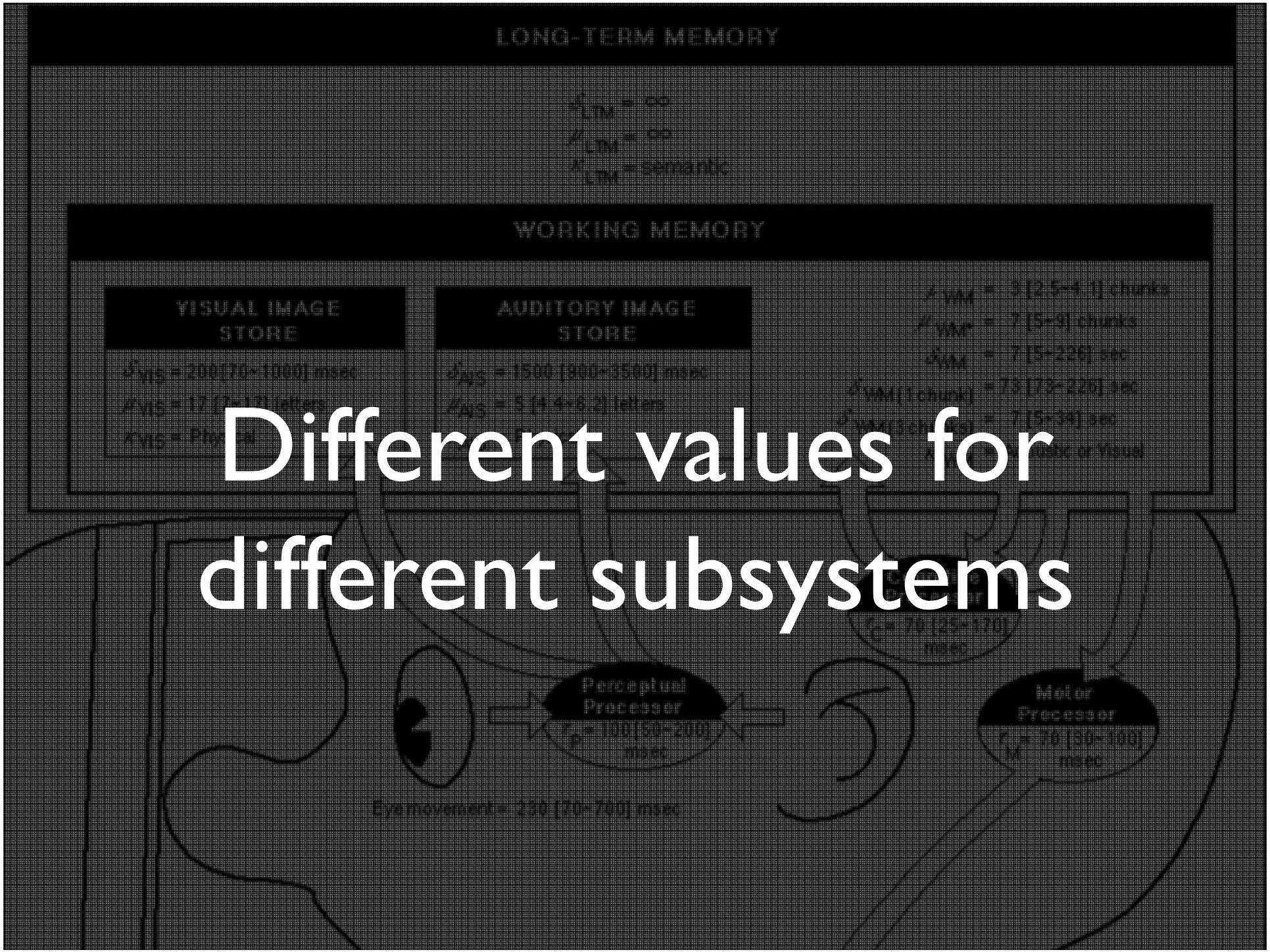
Parameters

Memory parameters

- ∂ = Decay time of an item
- μ = Storage capacity in items
- κ = Code type (physical, acoustic, visual, semantic)

Processor parameter

- τ = Cycle time

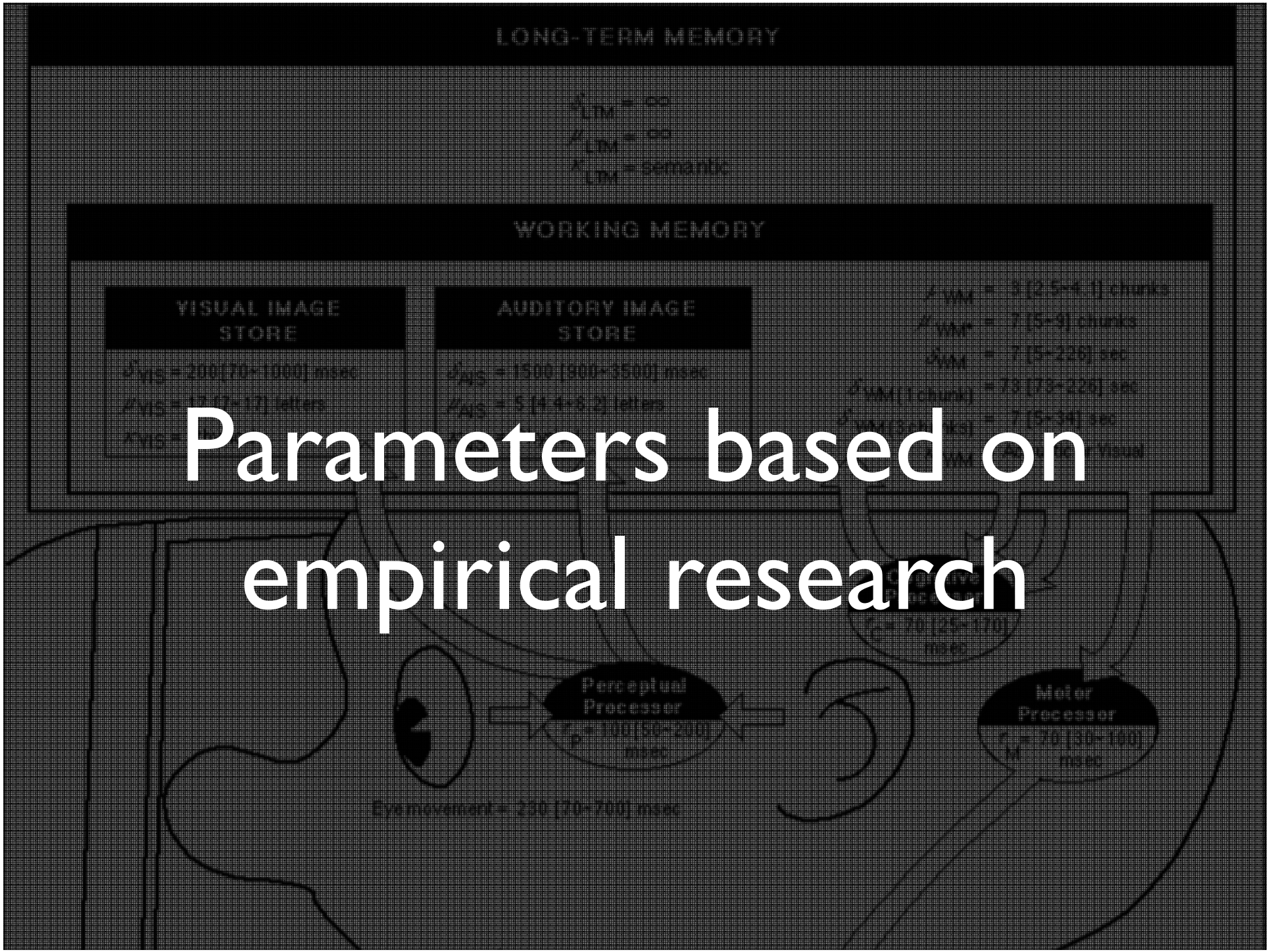


Different values for different subsystems

Model Assumptions

Slow users, middle speed users, fast users

Quantitative values are shown as middle speed



Parameters based on empirical research

10 Principles of Operation

P0. Recognize-act cycle of cognitive processor
P1. Variable perceptual processor rate principle
P2. Encoding specificity principle
P3. Discrimination principle
P4. Variable cognitive processor rate principle
P5. Fitt's law
P6. Power law of practice
P7. Uncertainty principle
P8. Rationality principle
P9. Problem space principle

What do we do with this stuff?

Example 1

Problem:

How long will it take someone to tell you the solution to a simple arithmetic problem on the screen?

$$3 \times 7 = ?$$

Steps to solving

1. Read numbers [perceptual processor]
2. Identify numbers [long-term memory]
3. Understand problem [working memory + cognitive processor]
4. Recall solution [long-term memory]
5. Map solution to keys [perceptual processor + long-term memory]
6. Type in solution [motor processor + perceptual processor]

Table 45.2. Cognitive Processing Rates

Rate at which an item can be matched against working memory:			
Digits	33 [27~39]	msec/item	Cavanaugh (1972)
Colors	38	msec/item	Cavanaugh (1972)
Letters	40 [24~65]	msec/item	Cavanaugh (1972)
Words	47 [36~52]	msec/item	Cavanaugh (1972)
Geometric shapes	50	msec/item	Cavanaugh (1972)
Random forms	68 [42~93]	msec/item	Cavanaugh (1972)
Nonsense syllables	73	msec/item	Cavanaugh (1972)
Range =	27~93 msec/item		
Rate at which four or fewer objects can be counted:			
Dot patterns	46	msec/item	Chi & Klahr (1975)
3-D shapes	94 [40~172]	msec/item	Akin & Chase (1978)
Range =	40~172 msec/item		
Perceptual judgment:	106 [85~169]	msec/inspection	Welford (1976)
Choice reaction time:	92 msec/inspection		Welford (1973)
	153 msec/bit		Hyman (1953)
Silent counting rate:	167 msec/digit		Landauer (1962)

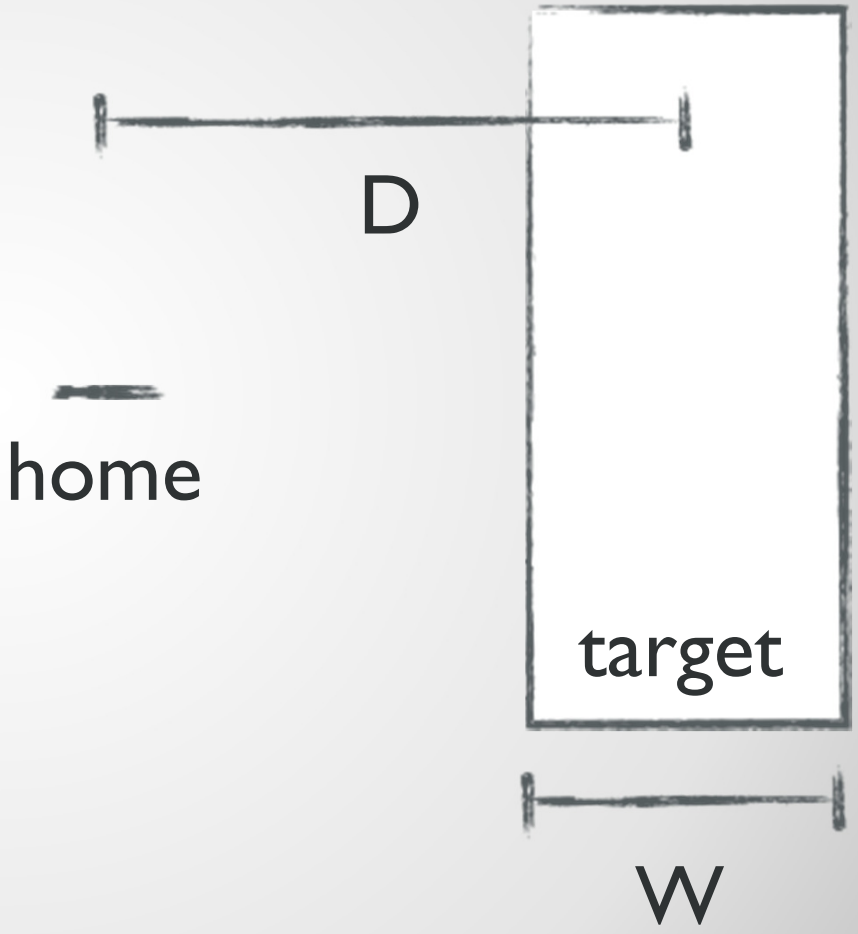
This table gives cycle times (msec per cycle) from various experimental paradigms that might be identified with the cognitive processor cycle time. (1) The rates at which an item can be matched against working memory are measured using Sternberg's (1975) paradigm in which the time is recorded for a subject to say whether a new item is in a previously presented list. The times per item given here are the slopes of the response times as a function of the number of items in the previously presented list. (2) The rate at which four or fewer objects can be counted are from subitizing experiments measuring the time for

P5. Fitts' Law

Fitts' Law predicts pointing movements

Modeling pointing movements

Predicts that time required to rapidly move to a target is a function of the distance to and size of the target.



Fitts' Law Parameters

T – the average time taken to complete the movement.

a – the start/stop time of the device (constant)

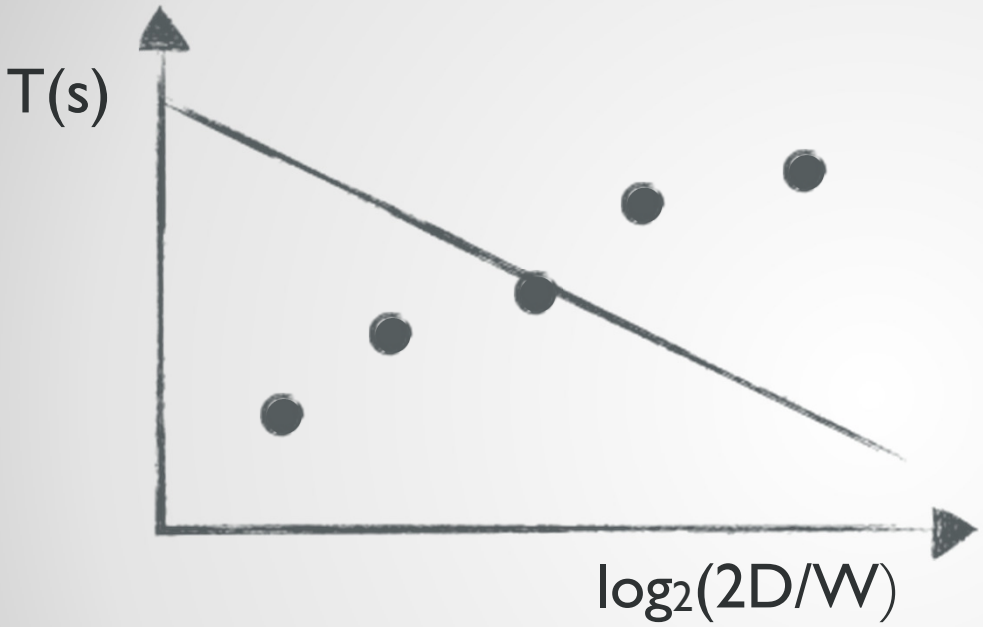
b – the inherent speed of the device (constant)

D – the distance from the starting point to the center of the target

W – the width of the target measured along the axis of motion

$$T = a + b * ID \qquad ID = \log_2 * (2*D/W)$$

Fitts' Law



$$T = a + b * \log_2(2 * D/W)$$

T – the average time taken to complete the movement.

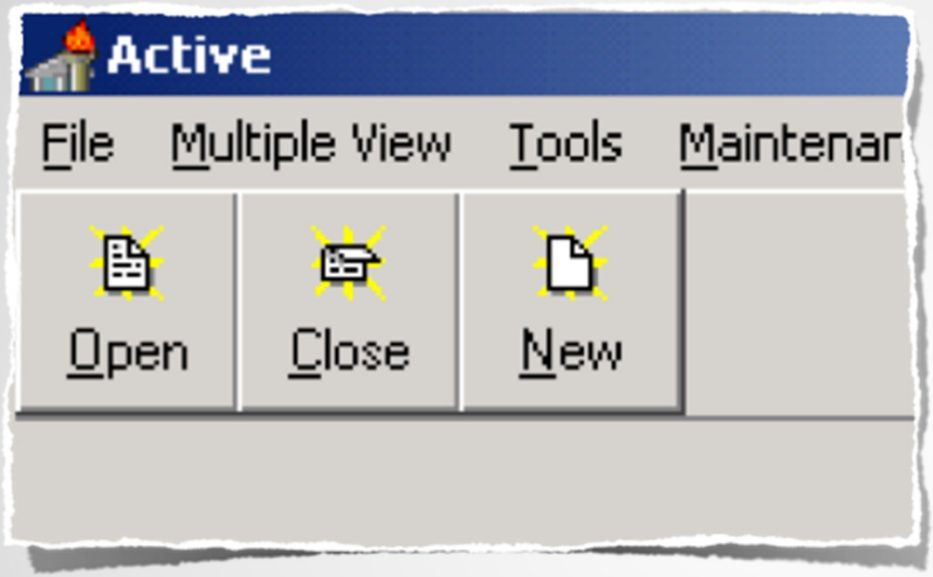
a – the start/stop time of the device (constant)

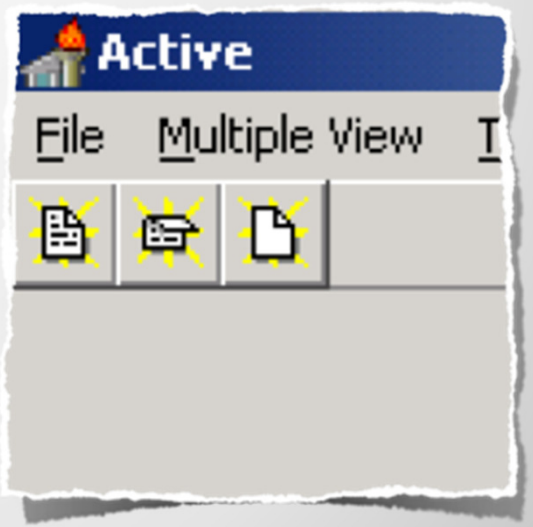
b – the inherent speed of the device (constant)

D – the distance from the starting point to the center of the target

W – the width of the target measured along the axis of motion

Why does Fitts' Law matter?





Summary

Goals of the Model Human Processor

Helps us make approximate predictions of gross human behavior

Helps us remember facts & predict human-computer interaction behavior

Informs design decisions when data not available

Helps explain data when it is

Model Human Processor consists of:

3 interconnected memories & processors

10 principles of operation

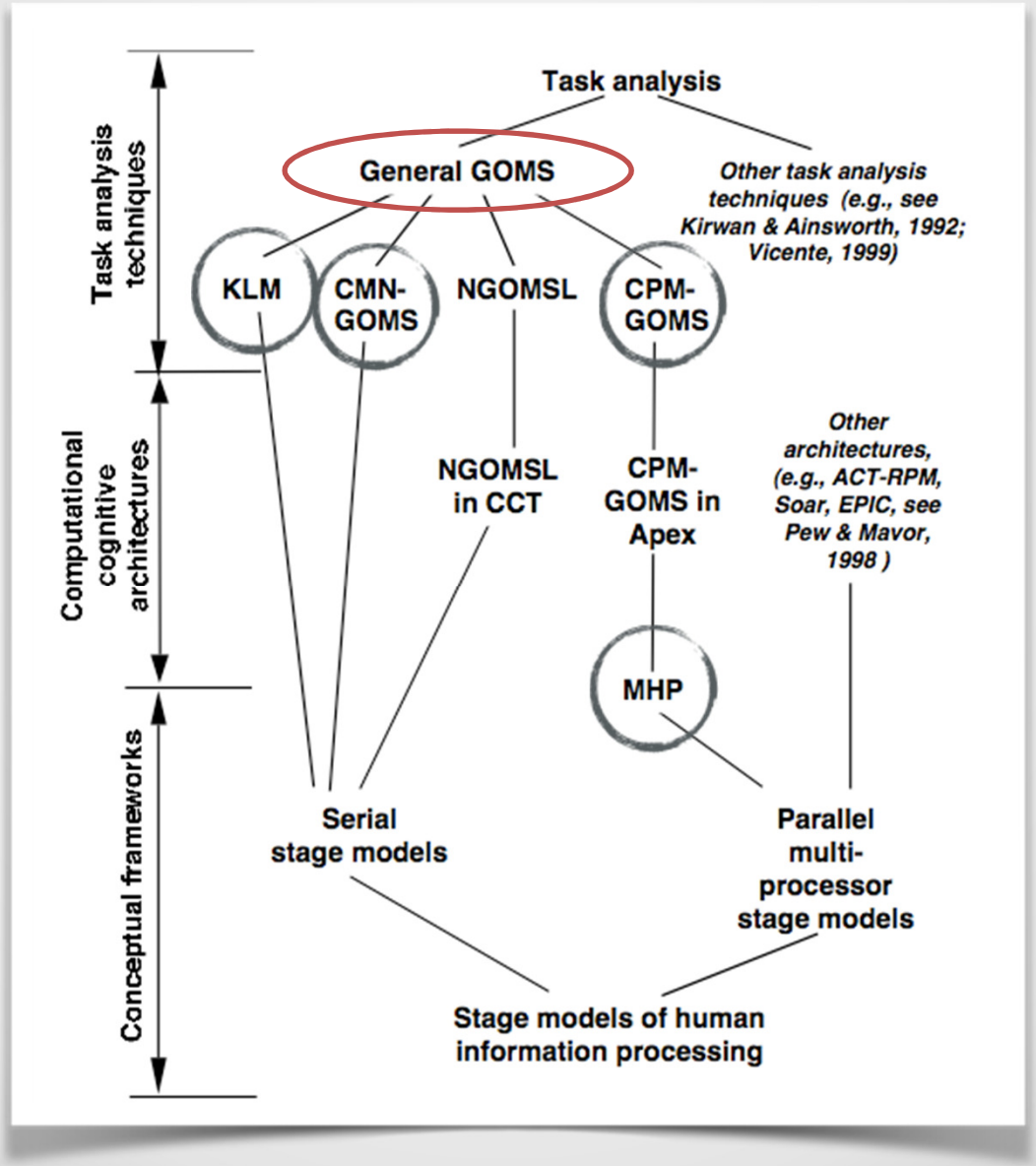
Summarized by 4 parameter values:

- μ = Storage capacity in items
- ∂ = Decay time of an item
- κ = Code type (physical, acoustic , visual, semantic)
- τ = Cycle time

Questions?

Can we model actual interactions with an interface?

Taxonomy of Modeling



GOMS

Goals, **O**perators, **M**ethods, and **S**election

Goals – what the user wants to accomplish

Operators – The means that leads to a goal at a detailed level

Methods – Sequences of operators

Selection rules – Rules (general or personal) for choosing a certain method

What is a GOMS model?

A description of the knowledge that a user must have in order to carry out tasks on a device or a system

Composed of methods that are used to achieve specific **goals**

Methods are composed of operators at the lowest level

Operators are specific steps (gestures) that a user performs and are assigned a specific execution time

If a goal can be achieved by more than one method, then **selection rules** are used to determine the proper method.

GOMS Characteristics

Combines cognitive aspects with analysis of task

Quantitatively

- Results in predictions of **time**

Qualitatively

- GOMS can explain the predictions
- Focus on methods to accomplish goals

When is GOMS analysis used?

Situations where users will be expected to perform tasks they have already mastered

- **Expert** users
- **Routine** work
- When time is crucial

GOMS Example

Goal – edit an article

Operators – arrow keys, mouse, 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.

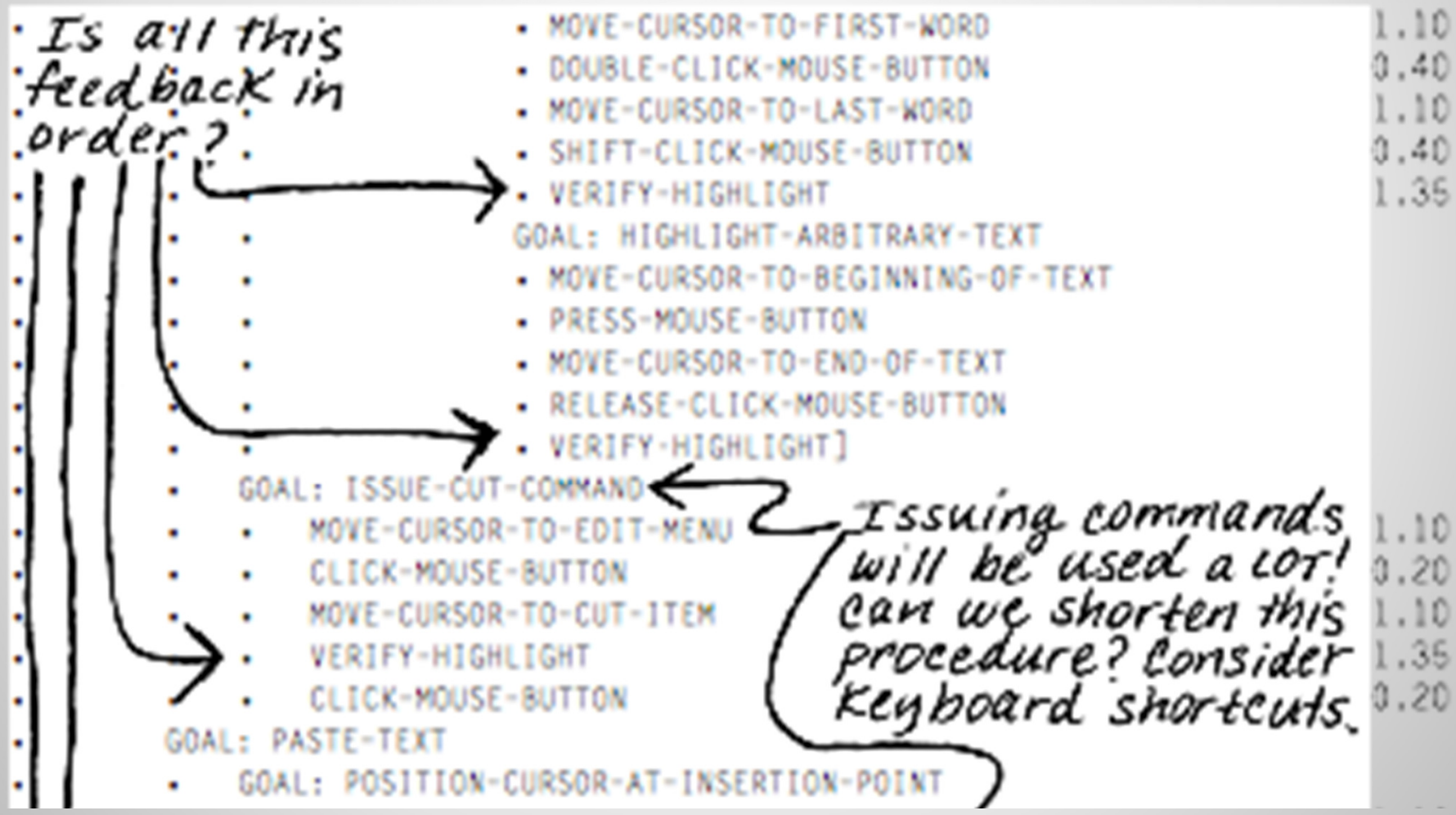
CMN-GOMS

CMN = Card, Moran, & Newell who introduced GOMS to HCI

Operators are strictly sequential

Breadth-first until relevant level of detail, could be at a keystroke level

CMN-GOMS Example



CPM-GOMS

CPM = Cognitive Perceptual Motor or
Critical Path Method

Based on Model Human Processor, involving parallel processing

Uses operators as in CMN-GOMS

Questions?

Keystroke Level Model (KLM)

Simples of GOMS techniques, serial model

Uses duration estimates for keystroke-level operators

Quantitatively – predicts time for skilled users

Qualitatively – highlights new ideas

Keystroke Operators

The following is from Card, Moran and Newell (1983), The Psychology of Human-Computer Interaction, Lawrence Erlbaum, p. 264

Operator	Description and Remarks	Time (sec)
K	PRESS KEY OR BUTTON Pressing the SHIFT or CONTROL key counts as a separate K operation. Time varies with the typing skill of the user; the following shows the range of typical values: Best typist (135 wpm) Good typist (90 wpm) Average skilled typist (55 wpm) Average non-secretary typist (40 wpm) Typing random letters Typing complex codes Worst typist (unfamiliar with keyboard)	 **.08** **.12** **.20** **.28** **.50** **.75** **1.20**
P	POINT WITH MOUSE TO TARGET ON DISPLAY The time to point varies with distance and target size according to Fitts's Law, ranging from .8 ro 1.5 sec, with 1.1 being average. This operator does not include the (.2 sec) button press that often follows. Mouse pointing time is also a good estimate for other efficient analogue pointing devices, such as joysticks (see Chapter 7).	**1.10**
H	HOME HAND(S) ON KEYBOARD OT OTHER DEVICE.	**.40**
D(n_D,l_D)	DRAW n_D STRAIGHT-LINE SEGMENTS OF TOTAL LENGTH l_D CM. This is a very restricted operator; it assumes that drawing is done with the mouse on a system that constrains all lines to fall on a square .56 cm grid. Users vary in their drawing skill; the time given is an average value.	**.9 n_D +.16 l_D**
M	MENTALLY PREPARE.	**1.35**
R(t)	RESPONSE BY SYSTEM.	t

Figure 8.1. The operators of the Keystroke-Level Model.
The **K** times are from Figure 2.14, except the .28 sec, which is the average typing rate of the non-secretarial users in Experiment 8A. The **P** time is from Chapter 7. The **H** time is from Chapter 5 and Chapter 7. The **D** time function and the coefficients were derived from least-squares fits on the drawing test data from the four MARHUP users. The time for **M** was estimated from the data in Experiment 8A.

K = key press

P = pointing

H = home hands

D = drawing a line

M = mental thinking

R_t = system response time

T_{total} = K+P+H+D+M+R

Operator Timings

Keying = 0.2 sec

The time it takes to tap a key on the keyboard

Pointing = 1.1 sec

The time it takes a user to point to a position on a display

Homing = 0.4 sec

The time it takes a user's hand to move from the keyboard to the GID or from the GID to the keyboard

Mentally Preparing = 1.35 sec

The time it takes a user to prepare mentally for the next step

Responding = based on task and system

The time a user must wait for a computer to respond to input

Building a KLM

List the overt actions necessary to do the task

Keystrokes and buttons (K)

Mouse movements (P)

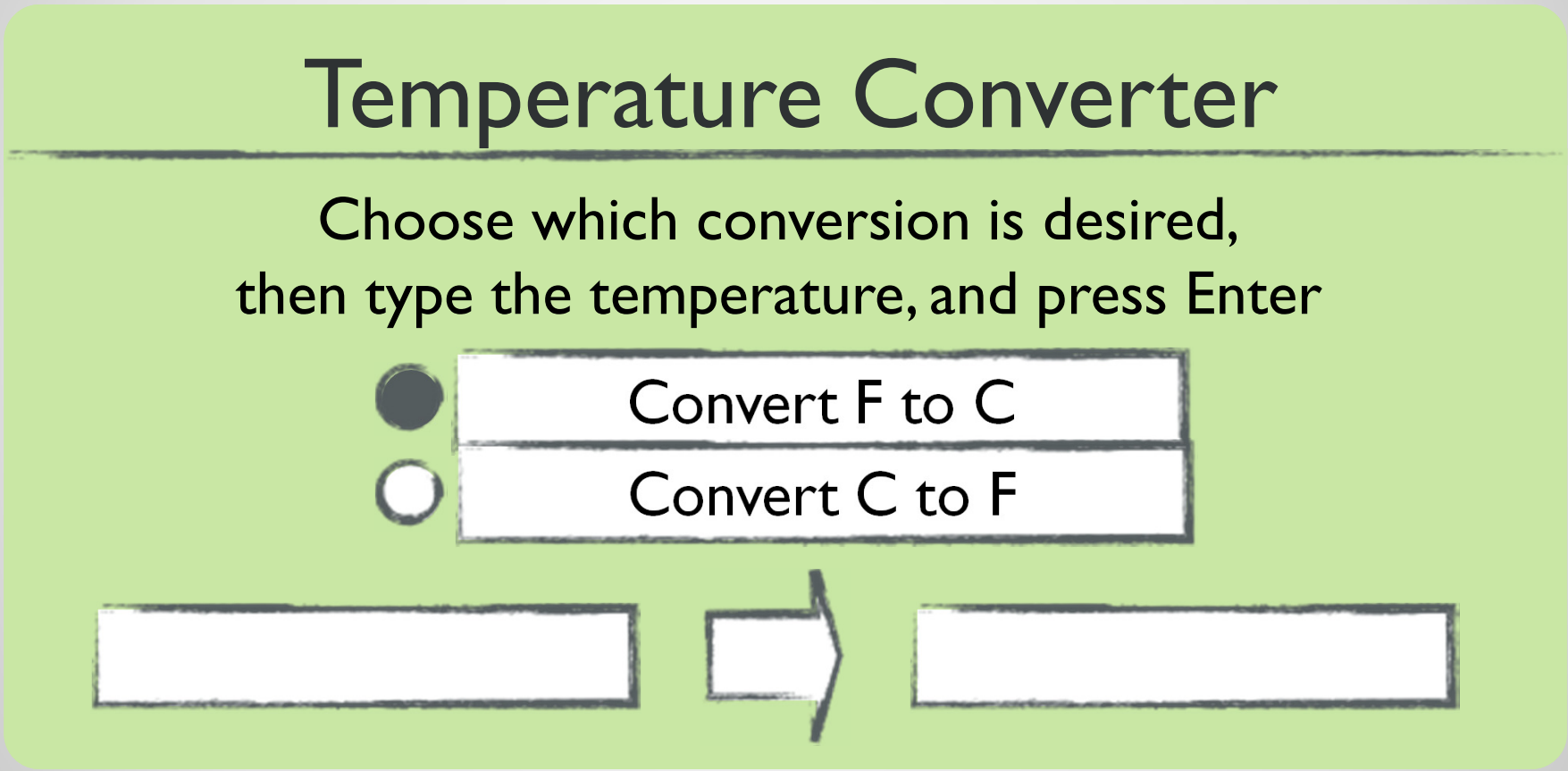
Hand movements from keyboard to mouse (H)

System response time (R)

Mental operators (M) by CMN heuristics

Add up execution times

Let's try it out!



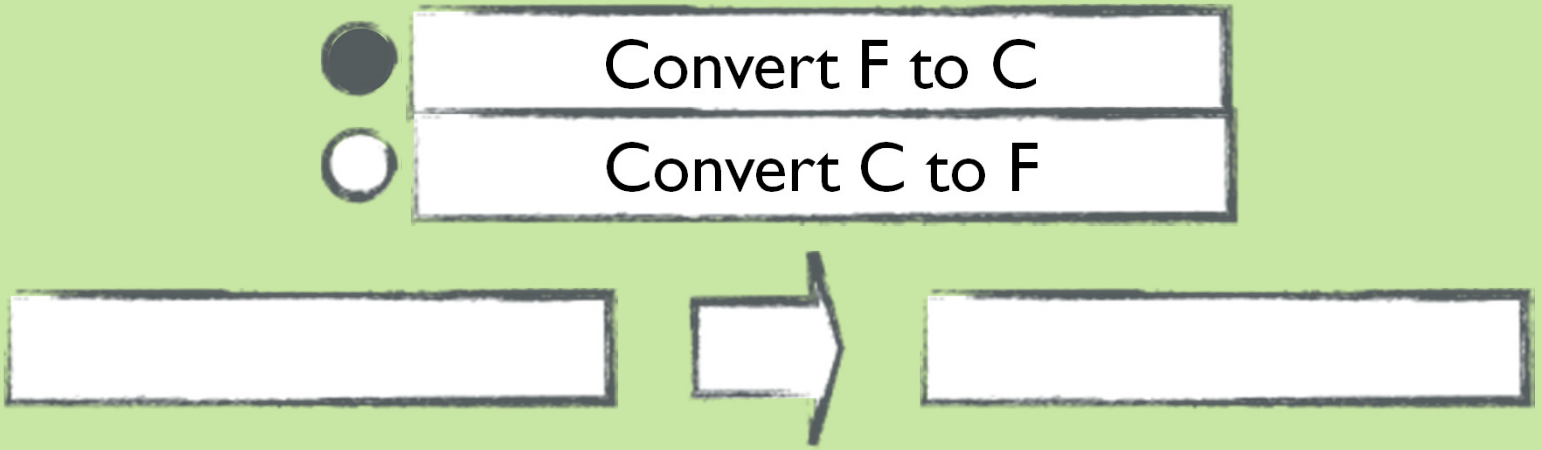
Using KLM

Assume an average of four typed characters in an entered temperature, including any decimal point and sign

Assume (for simplicity) that typing is perfect; error detection and notification aren't needed

Temperature Converter

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



K = key press

P = pointing

H = home hands

D = drawing a line

M = mental thinking

R_t = system response time

T_{total} = K+P+H+D+M+R

Inserting Mental Operators: Rule 0

Initial insertion of candidate Ms

Insert Ms in front of all keystrokes (K).

Insert Ms in front of all acts of pointing (P).

Do not insert Ms before any Ps that point to an argument.

- In GUIs, mouse-operated widgets (buttons, check boxes, etc) are considered commands.
- Text entry considered as an argument.

Arguments?

Pointing to a cell on a spreadsheet is pointing to an argument – no M

Pointing to a word in a manuscript is pointing to an argument – no M

Pointing to an icon on a toolbar is pointing to a command – M

Pointing to the label of a dropdown menu is pointing to a command -- M

Inserting Mental Operators: Rule 1

Deletion of anticipated Ms

If an operator following an M is fully anticipated in an operator immediately previous to that M, delete the M.

Inserting Mental Operators: Rule 2

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

Example: If typing "100", MKMKMK becomes MKKK

Inserting Mental Operators: 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.

Example: If coding in Java, ending a line with a ";" followed by a carriage return. The semi-colon is a terminator, and the carriage return is a redundant terminator, since both serve to end the line of code, thus, MKMK = MKK.

Inserting Mental Operators: Rule 4

Deletion of Ms that are terminators of commands

If K is a delimiter that follows a constant string, a command name (like "print"), or something that is the same every time you use it, then delete the M in front of it.

If a K terminates a variable string (e.g., name of file to be printed which is different each time), leave it in.

Inserting Mental Operators: Rule 5

Deletion of overlapped Ms

Do not count any portion of an M that overlaps an R – a delay, with the user waiting for a response from the computer

Questions?

CogTool

Assignment 3

http://cogtool.com/

Assignment posted on the website

Thursday: Using CogTool, bring a laptop if possible