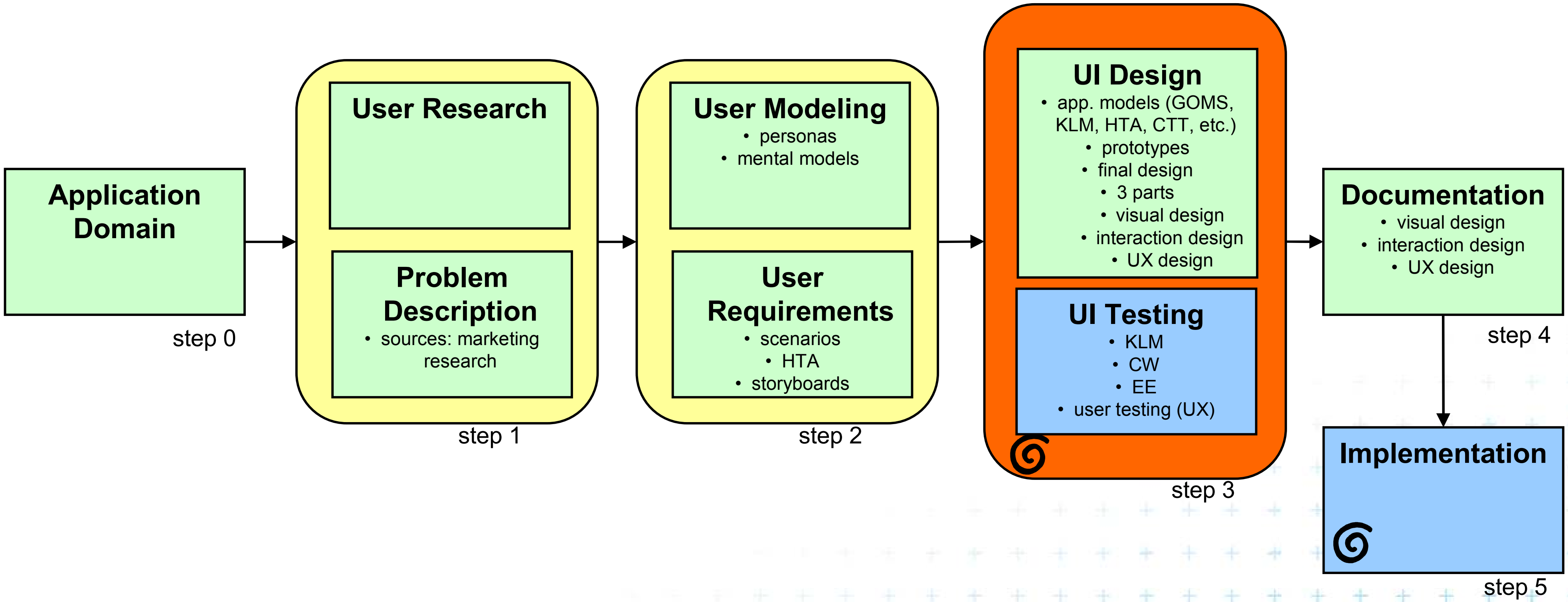
DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

NUR - Psychological aspects, MHP

User interface design - big picture



What is a model?

- A model is…
  – a simplification of reality
- A model is…
  – useful only if it helps in designing, evaluating, or otherwise providing a basis for understanding the behavior of a complex artifact such as a computer system
- To be useful, a model must be…
  – simpler than the behavior it models (i.e., extremely complex models are of questionable value)

Cognitive Modeling: Definition

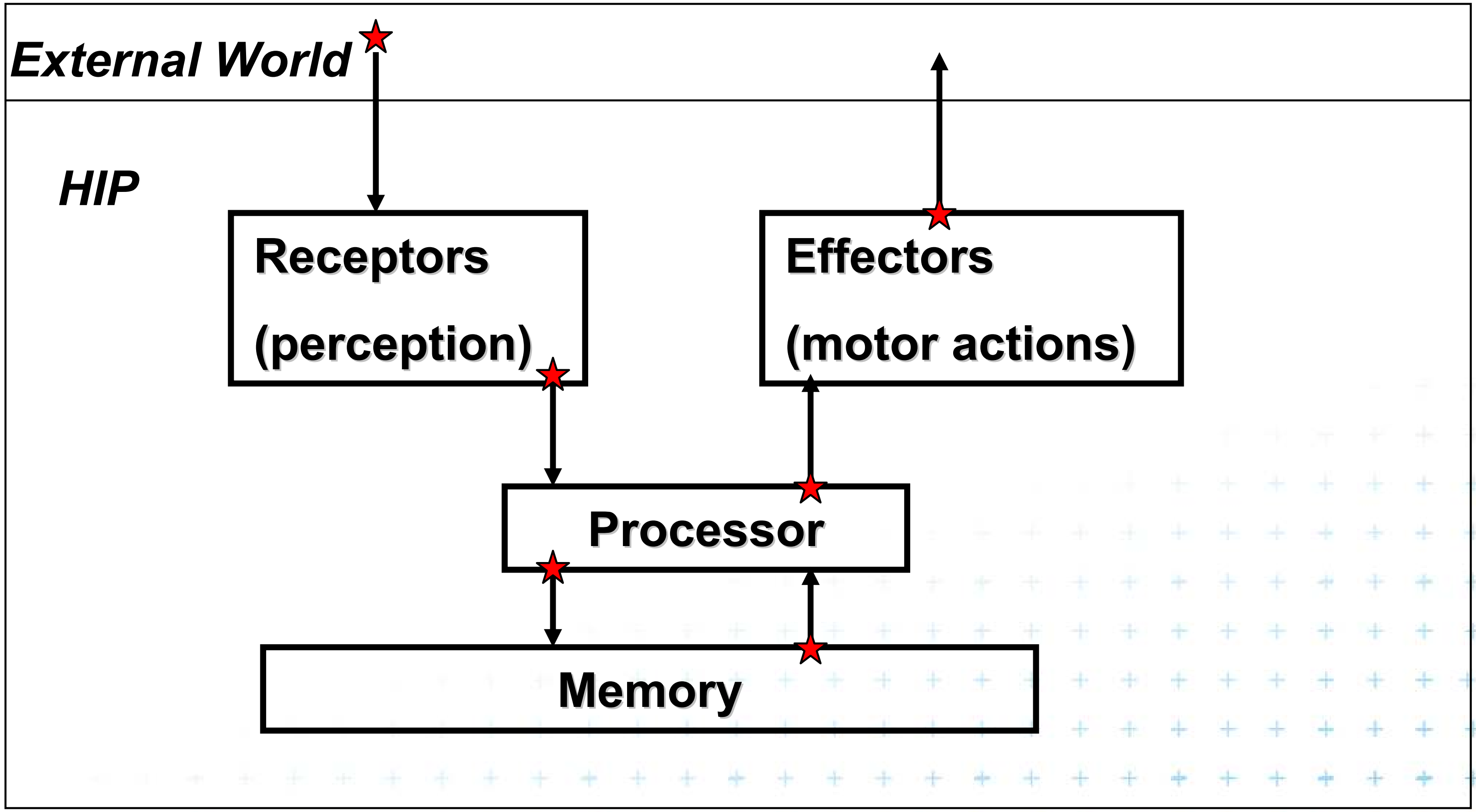
- A theory that produces a computational model of how people perform tasks and solve problems by using psychological principles and empirical studies.

Cognitive Modeling: Role

- Limits the design space
- Answers specific design decisions
- Estimates total task time
- Estimates training time
- Identifies complex, error-prone stages of the design
- A means of testing current psychological theories

DCGI

Cognitive Modeling:
Human Information Processor (HIP)



DCGI

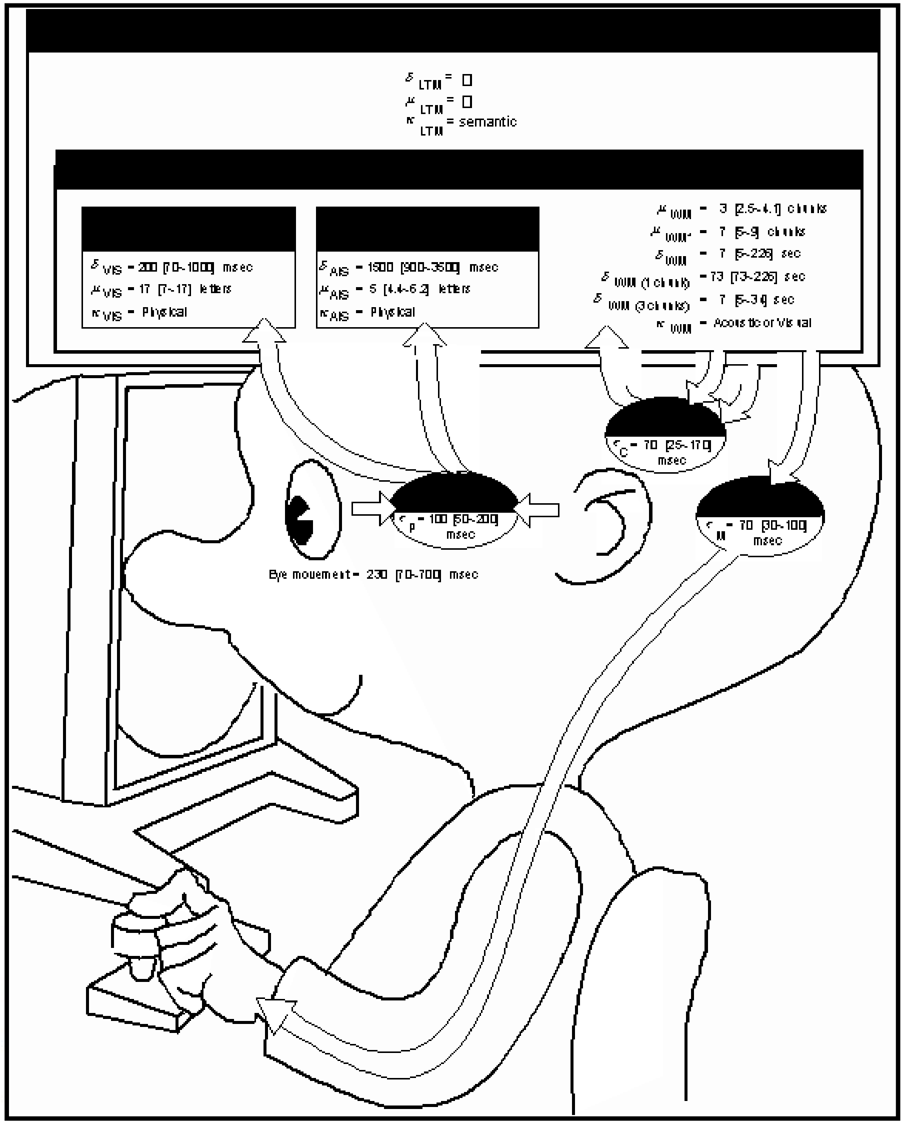
HOW TO MODEL HUMANS

Model Human Processor (MHP)

- Card, Moran & Newell (1983)
  - most influential model of user interaction
    - used in GOMS analysis
  - 3 interacting subsystems
    - cognitive, perceptual & motor
    - each with processor & memory

      described by parameters

      e.g., capacity, cycle time
    - serial & parallel processing



Adapted from slide by Dan Glaser

MHP

- Input/output
- Processing
  - serial action
    - pressing key in response to light
  - parallel perception
    - driving, reading signs & hearing

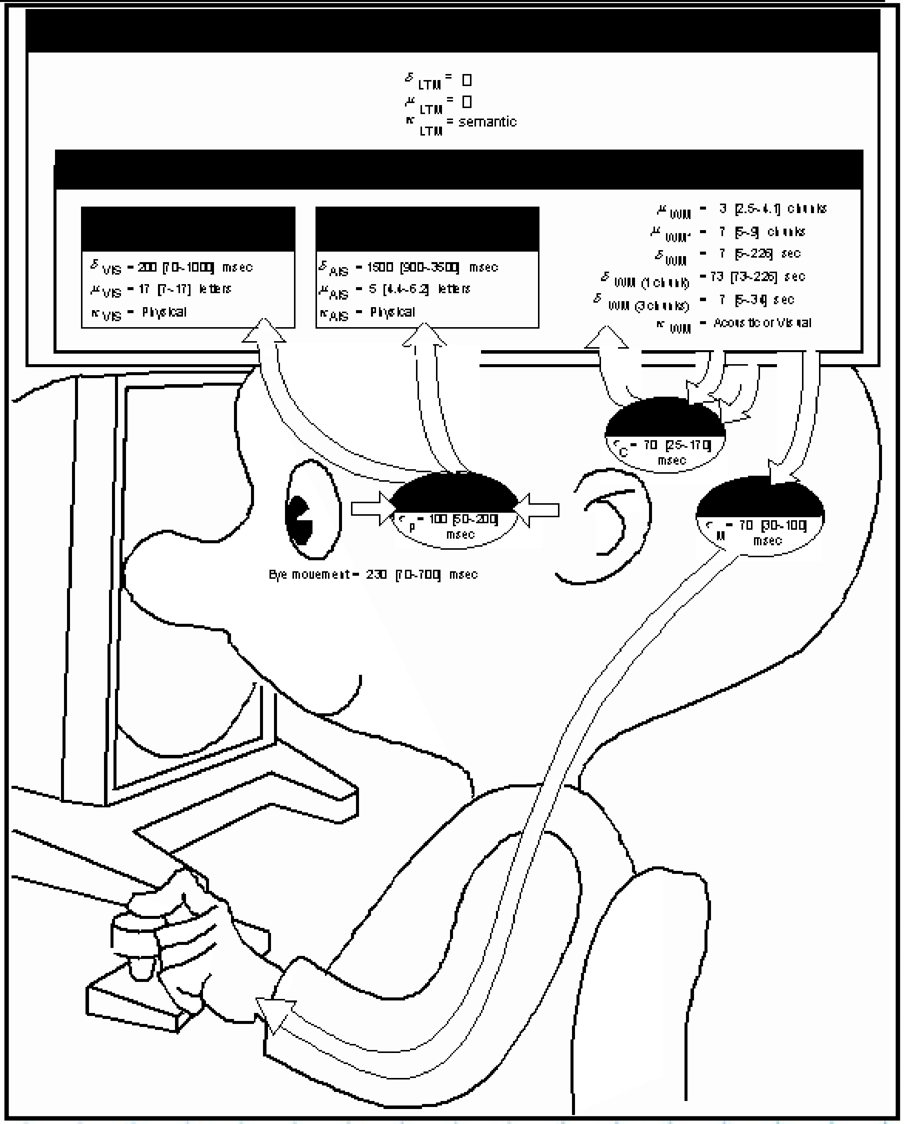
MHP data

- Based on empirical data
  - word processing in the ‘70s
- Processors have
  - cycle time (τ)
- Memories have
  - storage capacity (μ)
  - decay time of an item (δ)
  - info code type (κ)
    - physical, acoustic, visual & semantic

Perceptual Subsystem Parameters

- Processor
  - cycle time (τ) = 100 msec
- Visual Image Store
  - storage capacity (μ) = 17 letters
  - decay time of an item (δ) = 200 msec
  - info code type (κ) = physical
    - physical properties of visual stimulus

      e.g., intensity, color, curvature, length
- Auditory Image Store
  - similar parameters



Memory

There are three types of memory function:

Sensory memories

Attention

Short-term memory or working memory

Rehearsal

Long-term memory

Selection of stimuli governed by level of arousal.

sensory memory

- Buffers for stimuli received through senses
  - iconic memory: visual stimuli
  - echoic memory: aural stimuli
  - haptic memory: tactile stimuli
- Examples
  - “sparkler” trail
  - stereo sound
- Continuously overwritten

Short-term memory (STM)

- Scratch-pad for temporary recall
  - rapid access ~ 70ms
  - rapid decay ~ 200ms
  - limited capacity - 7± 2 chunks

Examples

212348278493202

0121 414 2626

HEC ATR ANU PTH ETR EET

DCGI

Brown-Peterson task (about forgetting)

- Subjects presented with trigram (XQJ)
- Experimenter presents number (257)
- Subject counts backwards by 3's (2/sec)
- After x seconds, subjects recall trigram

Other memory test

- Shepard & Tehgtsoonian (1961)

  – Presented 200 3-digit numbers in a row.

  – E.g. … 492, 865, 931, 758… 865, …

  – Task: report when you hear a repeated number

Memory processes

Say the following list of words once to yourself, and then, immediately thereafter, try to recall all the words, in any order, without looking back at them:

Table, cloud, book, tree, shirt, cat, light, bench, chalk, flower, watch, bat, rug, soap, pillow

Long-term memory (LTM)

- Repository for all our knowledge
  - slow access ~ 1/10 second
  - slow decay, if any
  - huge or unlimited capacity

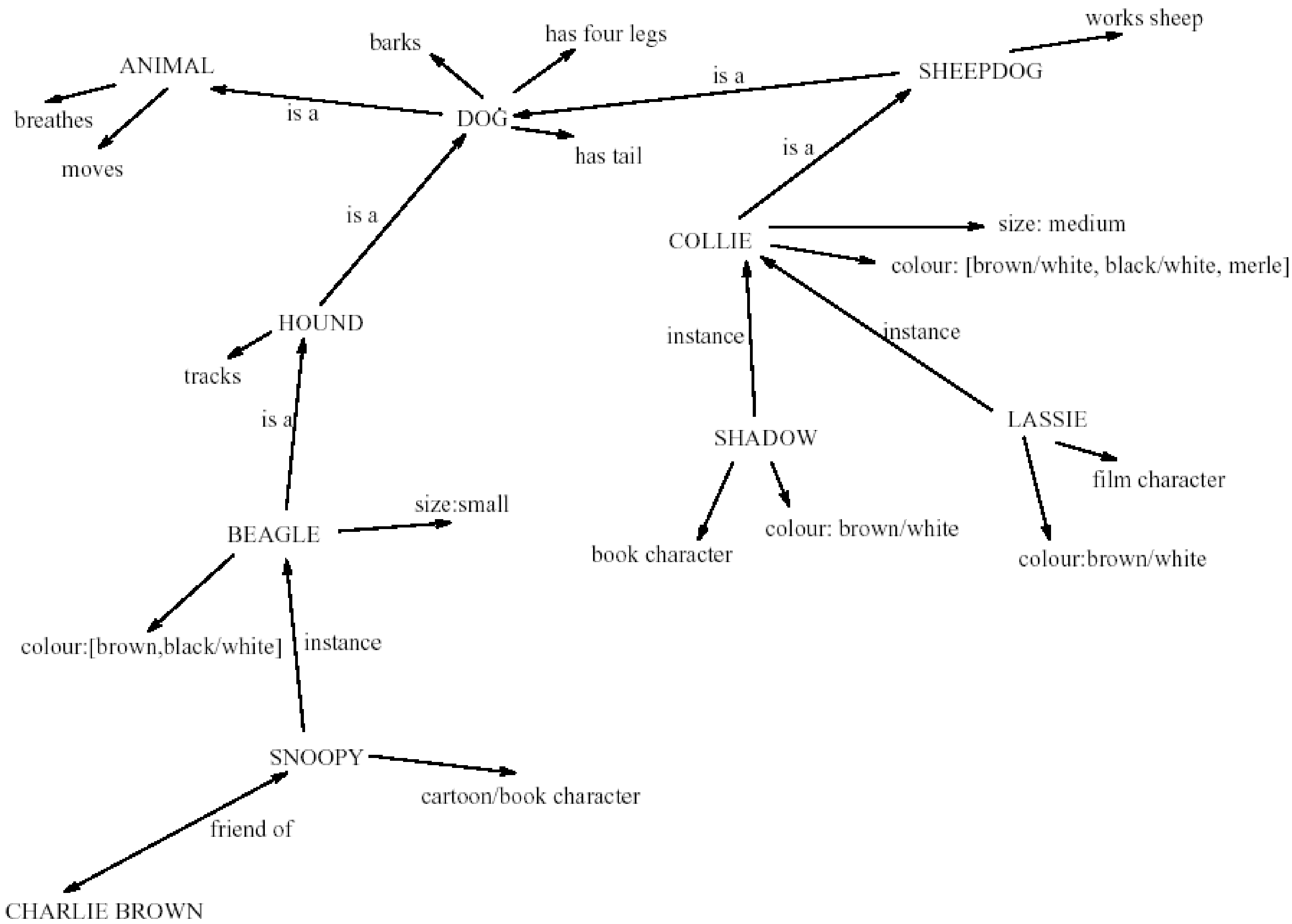
- Two types
  - episodic – serial memory of events
  - semantic – structured memory of facts,concepts, skills

semantic LTM derived from episodic LTM

Long-term memory (cont.)

- Semantic memory structure
  - provides access to information
  - represents relationships between bits of information
  - supports inference
- Model: semantic network
  - inheritance – child nodes inherit properties of parent nodes
  - relationships between bits of information explicit
  - supports inference through inheritance

LTM - semantic network



DCGI

Models of LTM - Frames

- Information organized in data structures
- Slots in structure instantiated with values for instance of data
- Type–subtype relationships

DOG

Fixed
    legs: 4

Default
    diet: carniverous
    sound: bark

Variable
    size:
    colour

COLLIE

Fixed
    breed of: DOG
    type: sheepdog

Default
    size: 65 cm

Variable
    colour

Models of LTM - Scripts

Model of stereotypical information required to interpret situation

Script has elements that can be instantiated with values for context

Script for a visit to the vet

Entry conditions:	*dog ill* *vet open* *owner has money*
Result:	*dog better* *owner poorer* *vet richer*
Props:	*examination table* *medicine* *instruments*
Roles:	*vet examines* *diagnoses* *treats* *owner brings dog in* *pays* *takes dog out*
Scenes:	*arriving at reception* *waiting in room* *examination* *paying*
Tracks:	*dog needs medicine* *dog needs operation*

DCGI

PROCEDURAL KNOWLEDGE

Models of LTM - Production rules

Representation of procedural knowledge.

Condition/action rules

if condition is matched

then use rule to determine action.

IF dog is wagging tail
THEN pat dog

IF dog is growling
THEN run away

LTM - Storage of information

- rehearsal
  - information moves from STM to LTM
- total time hypothesis
  - amount retained proportional to rehearsal time
- distribution of practice effect
  - optimized by spreading learning over time
- structure, meaning and familiarity
  - information easier to remember

LTM - Forgetting

decay

- information is lost gradually but very slowly

interference

- new information replaces old: retroactive interference
- old may interfere with new: proactive inhibition

so may not forget at all memory is selective …

… affected by emotion – can subconsciously `choose' to forget

LTM - retrieval

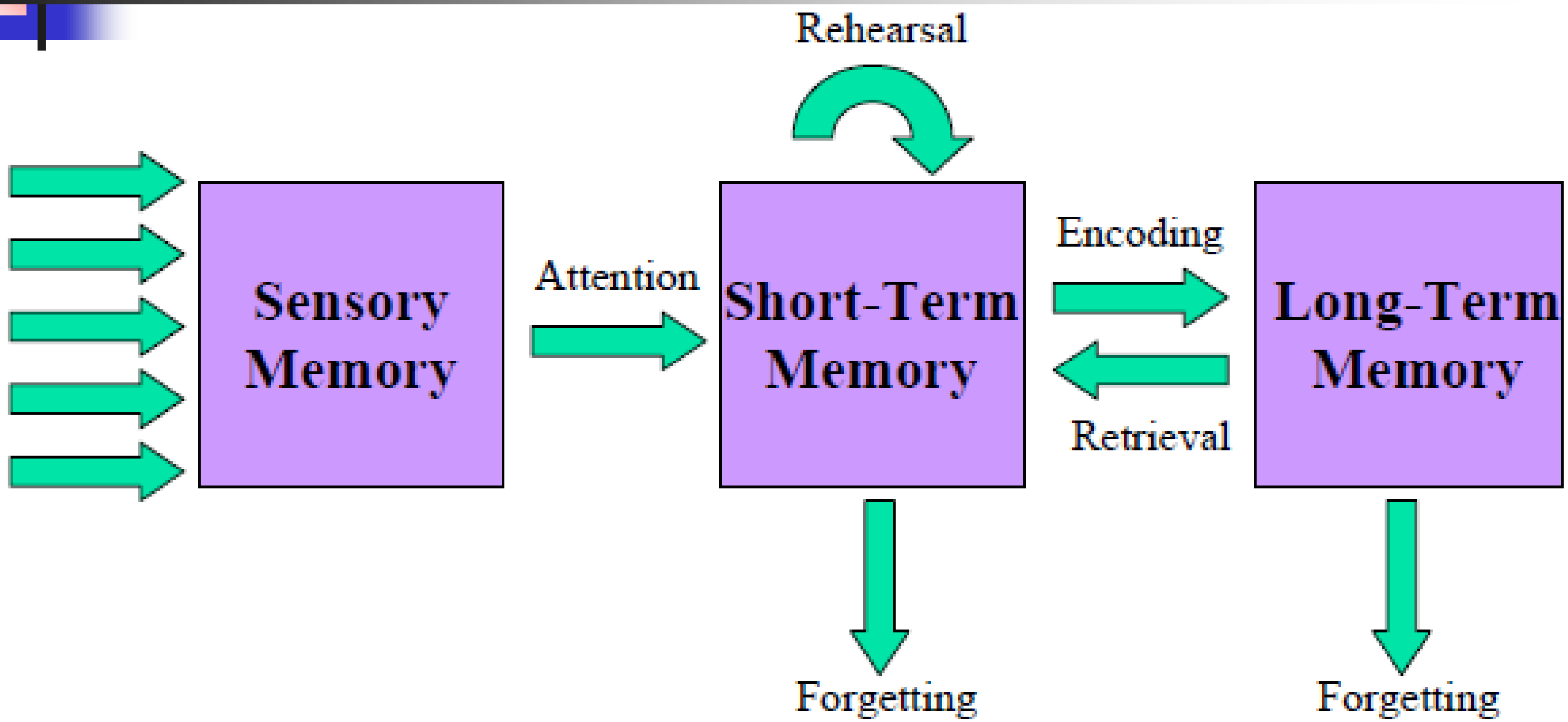
recall

- information reproduced from memory can be assisted by cues, e.g. categories, imagery

recognition

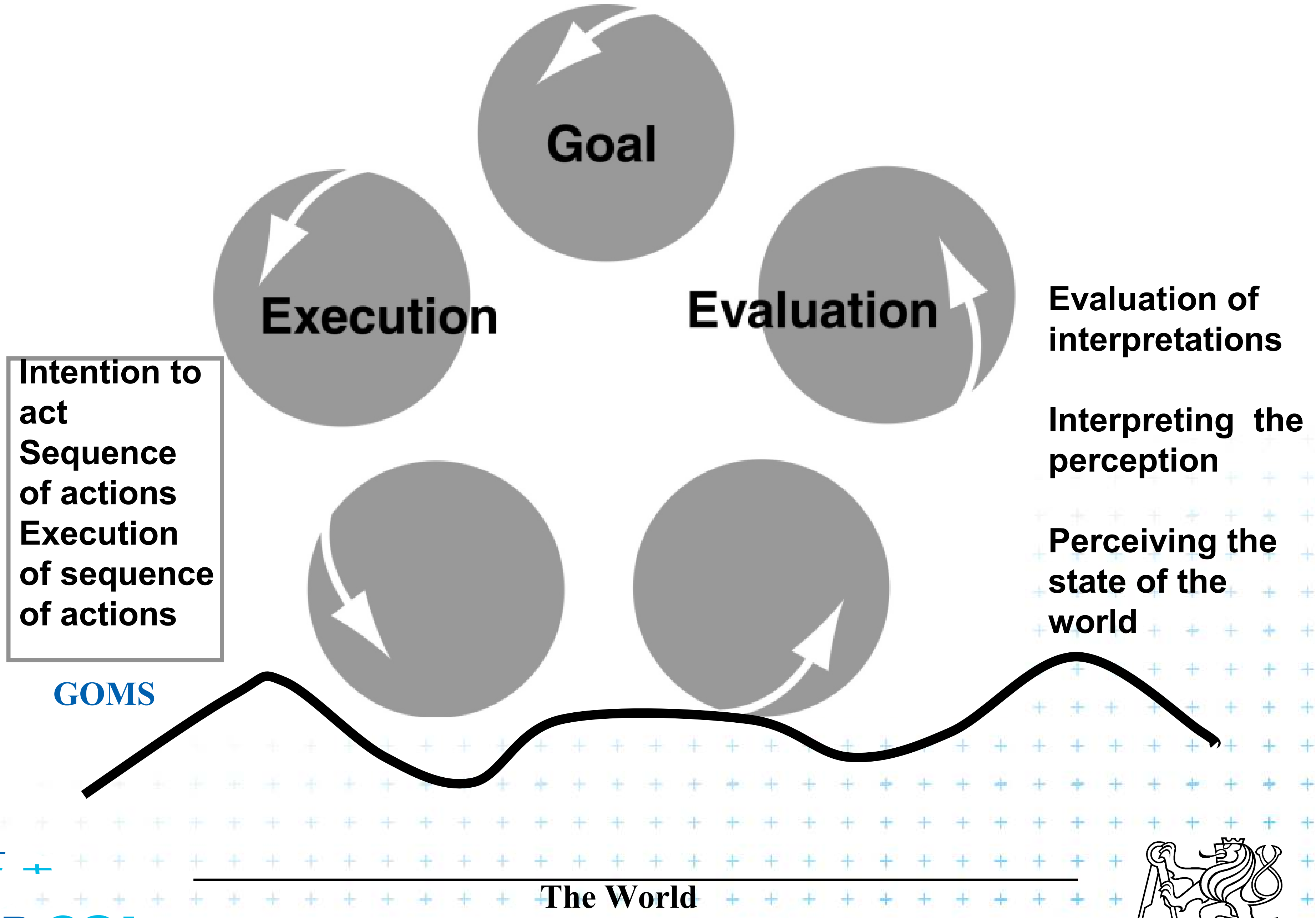
- information gives knowledge that it has been seen before
- less complex than recall - information is cue

Memory structure



MODELS OF HUMAN BEHAVIOR

Norman's Human Action Cycle (1988)



DCGI

GOMS - Card, Moran & Newell (1983)

- Engineering model of user interaction

  – task analysis (“how to” knowledge)
    - Goals - user’s intentions (tasks)

      e.g., delete a file, edit text, assist a customer
    - Operators - actions to complete task

      cognitive, perceptual & motor (MHP)

      low-level (e.g., move the mouse to menu)

Relation to HTA?

GOMS

- Explicit task structure
  - hierarchy of goals & sub-goals
- Methods - sequences of actions (operators)

  based on <u>error-free expert</u>

  may be multiple methods for accomplishing same goal

  e.g., shortcut key or menu selection

GOMS

- Selections - rules for choosing appropriate method
    - method predicted based on context

- Example: when more methods for accomplishing some subtask – we have to use some kind of strategy to choose appropriate method – e.g. deleting one or more characters

GOMS

- Analysis of explicit task structure
  - add parameters for operators
    - approximations (MHP) or empirical data
    - single value or parameterized estimate
  - predict user performance
    - execution time (count statements in task structure)
    - short-term memory requirements (stacking depth of task structure)
  - benefits
    - apply before implementation (comparing alternative designs)

2. GOMS

- **G**oals, **O**perators, **M**ethods, **S**election Rules
  – Developed by Card, Moran and Newell
- Probably the most widely known and used technique in this family

Quick Example

- Goal (the big picture)
  - go from hotel to the airport
- Methods (or subgoals)?
  - walk, take bus, take taxi, rent car, take train
- Operators (or specific actions)
  - 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

Goals

- Something the user wants to achieve
- Examples?
  - go to airport
  - delete file
  - create directory
- Hierarchical structure
  - may require many subgoals

Goal

- End state trying to achieve
- Then decompose into subgoals

- Moved sentence
  - Select sentence
  - Cut sentence
  - Paste sentence
    - Move to new spot
    - Place it

Methods

- Sequence of steps to accomplish a goal
  - goal decomposition
  - can include other goals
- Assumes method is learned & routine
- Examples
  - drag file to trash
  - retrieve from long-term memory command

Methods

- Sequence of operators (procedures) for accomplishing a goal (may be multiple)

- Example: Select sentence
  - Move mouse pointer to first word
  - Depress button
  - Drag to last word
  - Release

Operators

- Specific actions (small scale or atomic)
- Lowest level of analysis
  - can associate with times
- Examples
  - Locate icon for item on screen
  - Move cursor to item
  - Hold mouse button down
  - Locate destination icon
  - User reads the dialog box

Operators

- Basic actions available for performing a task (lowest level actions)

- Examples: move mouse pointer, drag, press key, read dialog box, …

Selection Rules

- If > 1 method to accomplish a goal, Selection rules pick method to use
- Examples
  - IF <*condition*> THEN accomplish <*GOAL*>
  - IF <*car has automatic transmission*> THEN <*select drive*>
  - IF <*car has manual transmission*> THEN <*find car with automatic transmission*>

Selection Rules

- Invoked when there is a choice of a method
- GOMS attempts to predict which methods will be used

- Example: Could cut sentence either by menu pulldown or by ctrl-x

GOMS Output

- Execution time
  - add up times from operators
  - assumes ?
    - experts
  - very good rank ordering
  - absolute accuracy ~10-20%

Assumptions

- “Expert” is performing UI operations
- Interacting with system is problem solving
- Decompose into subproblems
- Determine goals to attack problem
- Know sequence of operations used to achieve the goals
- Timing values for each operation

How to do GOMS Analysis

- Generate task description
  - Pick high-level user Goal
  - Write Method for accomplishing 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

Operators vs. Methods

- Operator: the most primitive action
- Method: requires several Operators or subgoal invocations to accomplish
- Level of detail determined by
  - KLM level - keypress, mouse press
  - Higher level - select-Close-from-File-menu
  - Different parts of model can be at different levels of detail

GOMS Procedure

- Walk through sequence of steps
- Assign each an approximate time duration

- -> Know overall performance time

- (Can be tedious)

GOMS Example: PDA Text Entry

- goal: enter-text-PDA
  - move-pen-to-text-start
  - goal: enter-word-PDA
  - ...repeat until no more words
    - write-letter ...repeat until no more letters
    - [select: goal: correct-misrecognized-word] ...if incorrect
- expansion of correct-misrecognized-word goal:
  - move-pen-to-incorrect-letter
  - write-letter

GOMS Example

- Retrieve the article entitled “Why Goms?”
  - – written by Bonnie John, 1995, in ACM DL

GOMS: Goal Structure

- Goal: Retrieve article from ACM DL
  - Goal: Go to ACM
    - Goal: Enter ACM URL
    - Goal: Submit URL
  - Goal: Go to DL
    - Goal: Locate DL link
    - Goal: Select the link
  - Goal: Select method
    - [Method: Search method
    - Goal: Search for article
      - Goal: Enter search parameters
      - Goal: Submit search
      - Goal: Identify article from results
    - Goal: Select the article]
    - [Method: Browse method - <take home exercise>]
  - Goal: Save article to disk
    - Goal: Initiate save action
    - Goal: Select location
    - Goal: save article to that location

GOMS example: Delete a word

- Goal: delete a word in a sentence.
- Method #1: use the menu
  - Recall that the word has to be highlighted.
  - Recall that the command is “cut”.
  - Recall that “cut” is in the Edit Menu.
  - Accomplish goal of selecting and executing “cut”.
  - Return: goal accomplished.

DCGI

GOMS example (cont.)

- Method #2: use the delete key
  - Recall where to position cursor in relation to word to be deleted.
  - Recall which key is delete key.
  - Press “delete” key to delete each letter.
  - Return: goal accomplished.
- Operators used in these methods
  - Click mouse, Drag cursor over text, Select menu, Move cursor, Press KB key, Think, ...

GOMS example (cont.)

- Selection rules:
  – Use mouse/menu method (#1) if there’s a lot of text to delete.
  – Else use “delete” key (method #2).

KLM (a low-level variant of GOMS)

- Keystroke Level Model.
- Simple, but accurate. Widely used.
- Scope:
  - *skilled* users
  - doing a task *error-free*.
  - using a *specific method* in a UI.
- CogTool has this built-in.

KLM Operators

- User Operators:
  - K (keystroke), P (point), H (homing), D (drawing), M (mental: think).
  - Times for each are provided to you
    - based on extensive research/empirical data.
- System Operator:
  - R (respond).

Limitations

- GOMS is not for
  - Tasks where steps are not well understood
  - Inexperienced users
- Why?

GOMS Variants

- GOMS is often combined with a keystroke level analysis
  - KLM - Keystroke level model
  - Analyze only observable behaviors such as keypresses, mouse movements
  - Low-level GOMS where method is given
- Tasks split into two phases
  - Acquisition of task - user builds mental rep.
  - Execution of task - using system facilities

Procedure

- How KLM works
  - Assigns times to different operators
  - Plus: Rules for adding M's (mental preparations) in certain spots

KLM = subset of GOMS

- Six keystroke-level primitive operators

  K - press a key or button

  P - point with a mouse

  H - home hands

  D - draw a line segment

  M - mentaly prepare to do an action

  R - system response time

- No selections

DCGI

Example

Move Sentence

1. Select sentence

Reach for mouse	H	0.40
Point to first word	P	1.10
Click button down	K	0.60
Drag to last word	P	1.20
Release	K	0.60
		3.90 secs

2. Cut sentence

Press, hold ^		Point to menu
Press and release 'x'	or	Press and hold mouse
Release ^		Move to "cut"
		Release

3. ...

Current Design: Delete a file by dragging it to the trash icon

1. Point to file icon (P)
2. Press & hold mouse button (B)
3. Drag file to trash icon (P)
4. Release mouse button (B)
5. Point to original window (P)

$$3P + 2B = 3.5 \text{ sec.}$$

New Design: Adding a command to menu

1. Point to file icon (P)
2. Click button (BB)
3. Point to file menu (P)
4. Press and hold button (B)
5. Point to delete command (P)
6. Release mouse button (B)
7. Point to original window (P)

4P + 4B = 4.8 sec.

Assumptions

- These previous scenarios work only work if the user is currently able to view all the needed windows and icons.
- If the trash icon for example is buried under other windows the first procedure is slowed down quite a bit.

Inserting Mental Operators: Where does the user stop and think?

1. Initiating a process.
2. Making strategic decisions.
3. Retrieving a chunk from user's short term memory
4. Finding something on the screen.
5. Verifying intended action is complete.

Mental Operators - New vs Experienced Users

- New users stop and check feedback after every step
- New users have small chunks
- Experienced users have elaborate chunks
- Experienced users may overlap mental operators with physical operators

Delete a file by dragging icon to trash

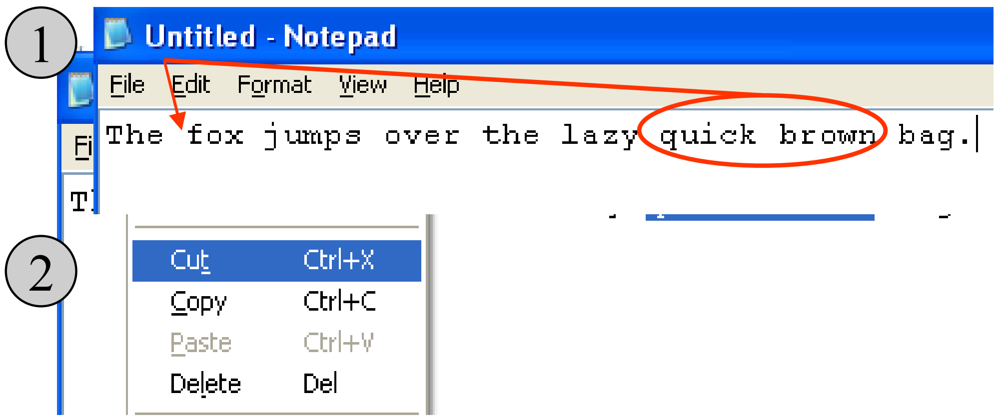
1. Initiate delete. (M)
2. Find file icon. (M)
3. Point to file icon. (P)
4. Press & hold button. (B)
5. Verify icon reverse video. (M)
6. Find trash icon. (M)
7. Drag file to trash icon. (P)
8. Verify trash reverse video. (M)
9. Release button. (B)
10. Verify bulging trash icon. (M)
11. Find original window. (M)
12. Point to window. (P)

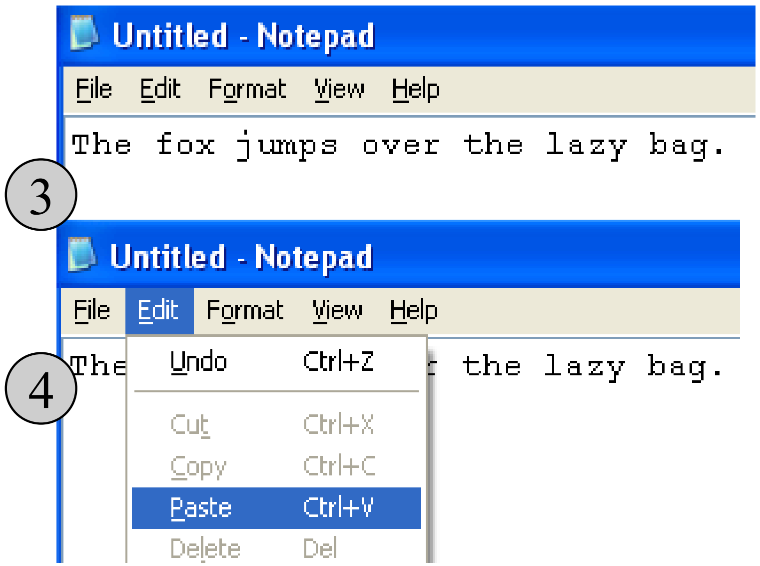
$$3P + 2B + 7M = 12.6 \text{ sec.}$$

DCGI

Method Used

Cut-and-paste-using-menus





M=1.35
P=1.10
K=0.20

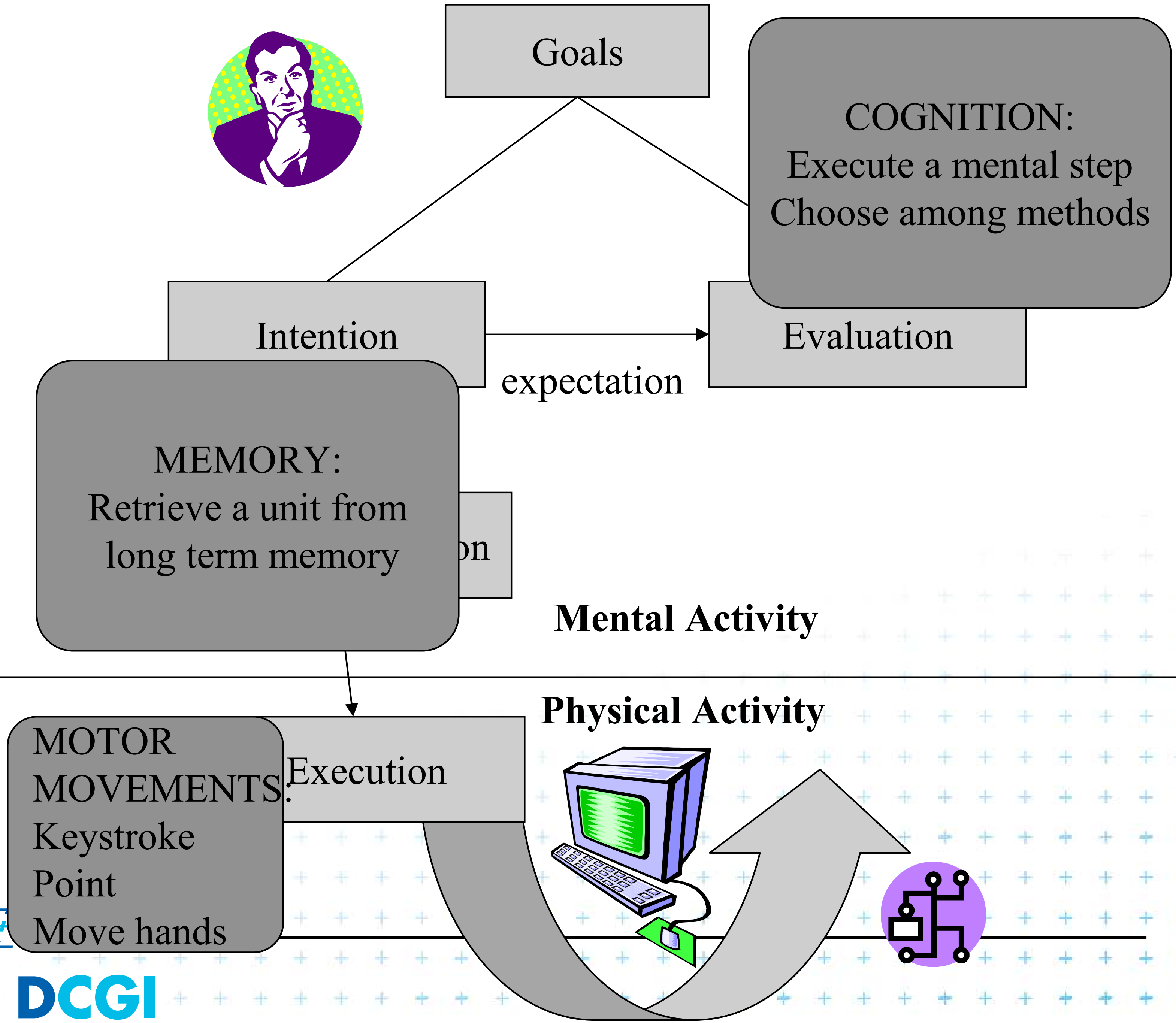
Description	Operator	Duration (sec)
Mentally Prepare	M	1.35
Move cursor to “quick”	P	1.10
Double-click mouse button	K	0.40
Move cursor to “brown”	P	1.10
Shift-click mouse button	K	0.40
Mentally Prepare	M	1.35
Move cursor to Edit Menu	P	1.10
Click mouse button	K	0.20
Move cursor to Cut menu item	P	1.10
Click mouse button	K	0.20
Mentally Prepare	M	1.35
Move cursor to before “fox”	P	1.10
Click mouse button	K	0.20
Mentally Prepare	M	1.35
Move cursor to Edit menu	P	1.10
Click mouse button	K	0.20
Move cursor to Paste menu item	P	1.10
Click mouse button	K	0.20
TOTAL PREDICTED TIME		**14.90**

Production Systems

- Cognitive Complexity Theory
  - Uses goal decomposition from GOMS and provides more predictive power
  - Goal-like hierarchy expressed using production rules
    - if condition, then action
    - Makes a generalized transition network

Other human features

- Besides time “constants” we have to take into account also other features
- E.g. when we perform a task repeatedly – we get better and better (time necessary shrinks)
- Besides MHP we have to use additional “rule”



Goals
COGNITION:
Execute a mental step
Choose among methods
Intention
expectation
Evaluation
MEMORY:
Retrieve a unit from
long term memory
on
Mental Activity
Physical Activity
MOTOR
MOVEMENTS:
Keystroke
Point
Move hands
Execution

DCGI

Power Law of Practice

Task time on the nth trial follows a power law

- $T_n = T_1 n^{-a}$, where a = 0.4
- i.e., you get faster the more times you do it!
- applies to skilled behavior (perceptual & motor)
- does not apply to knowledge acquisition or quality

Hick's Law

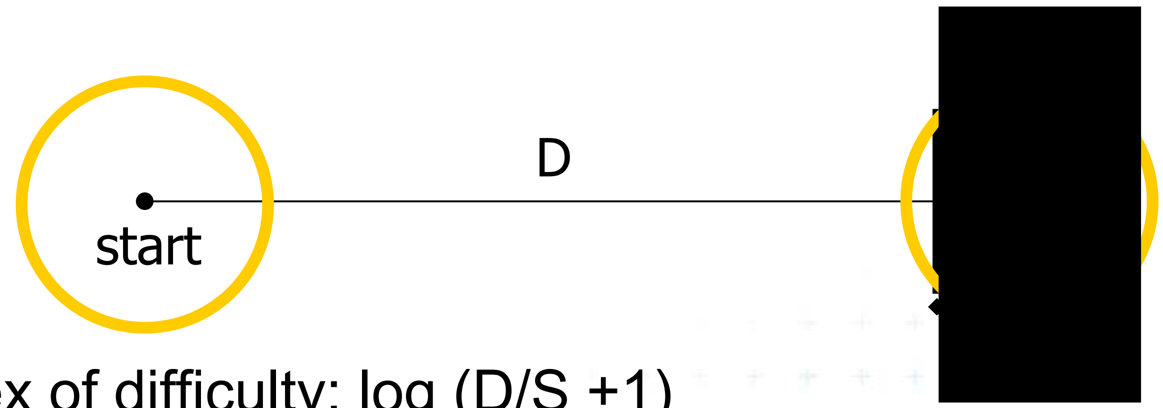
- Time it takes for a user to make a decision.
- Given n equally probable choices, the average reaction time T required to choose among them:

  $T = b \log_2(n + 1)$

Fitts's Law

- Time T to move your hand to a target of size S at distance D away is

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



  - Index of difficulty: $\log (D/S + 1)$

    S is in direction of motion ("length" arbitrary)

    Note that distance is between center points

Fitts' Law

Models movement time for selection tasks

The movement time for a well-rehearsed selection task:

- increases as the distance to the target increases
- decreases as the size of the target increases

Fitts' Law

Time (in msec) = a + b $\log_2(D/S+1)$

where

a, b = constants (empirically derived)

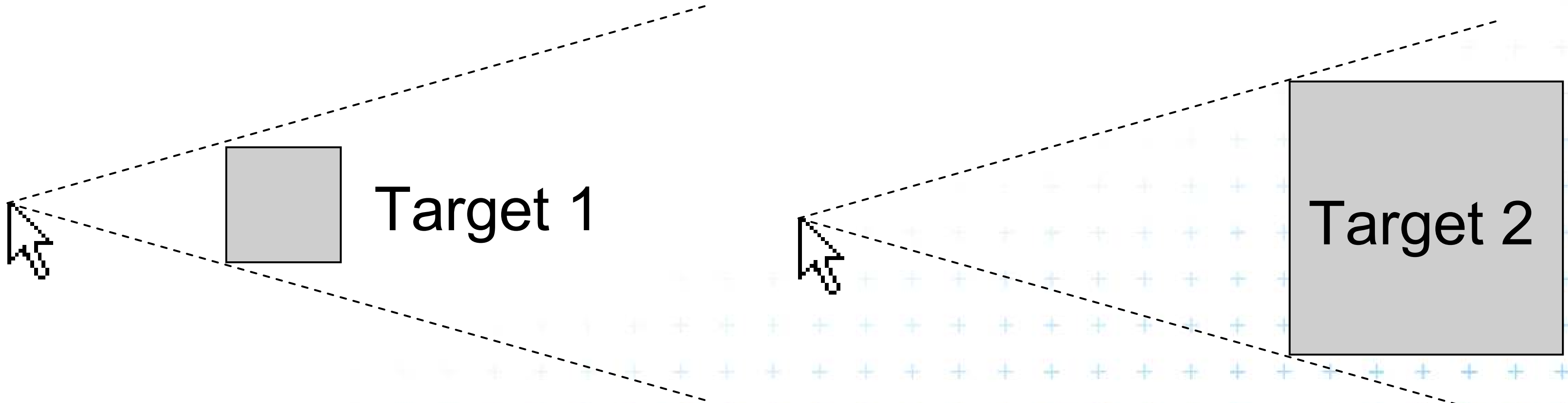
D = distance

S = size

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

Fitts’ Law

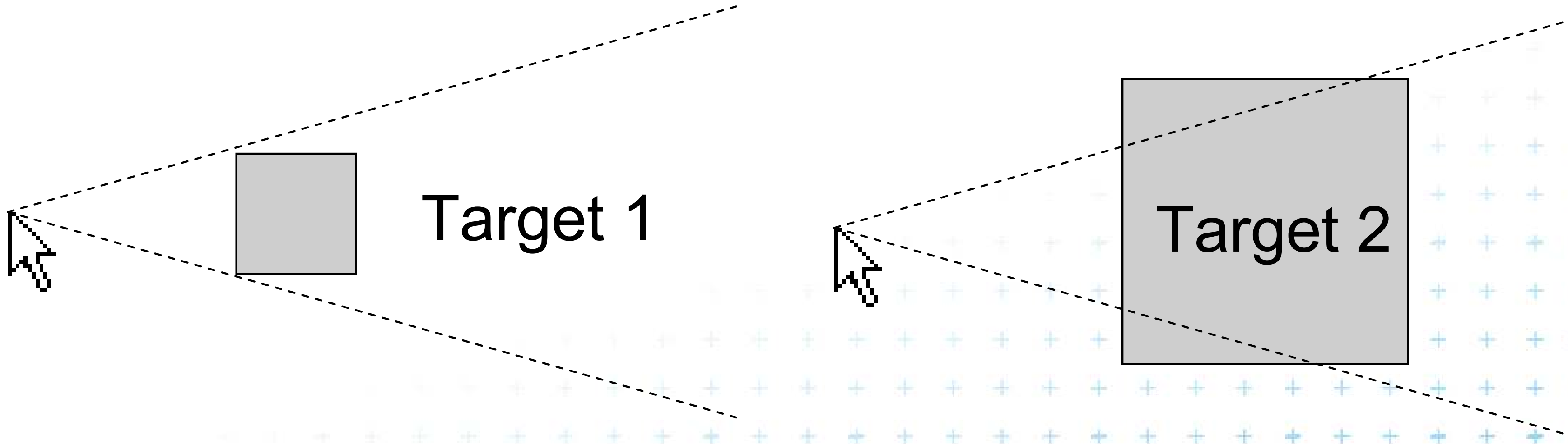
$$\text{Time} = a + b \log_2(D/S+1)$$



Same ID → Same Difficulty

Fitts' Law

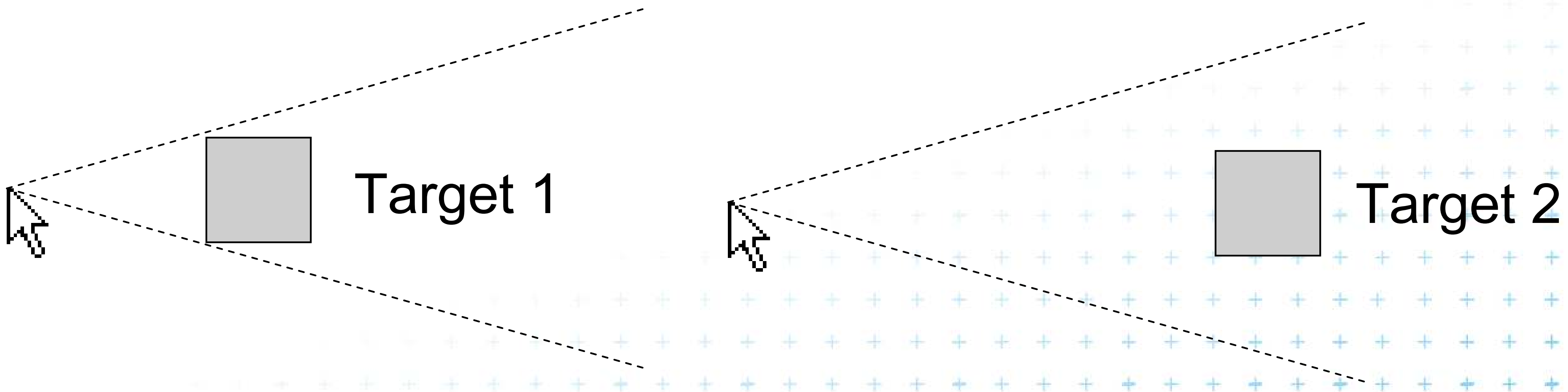
$$Time = a + b \log_2(D/S+1)$$



Smaller ID → Easier

Fitts' Law

$$\text{Time} = a + b \log_2(D/S+1)$$



Larger ID → Harder

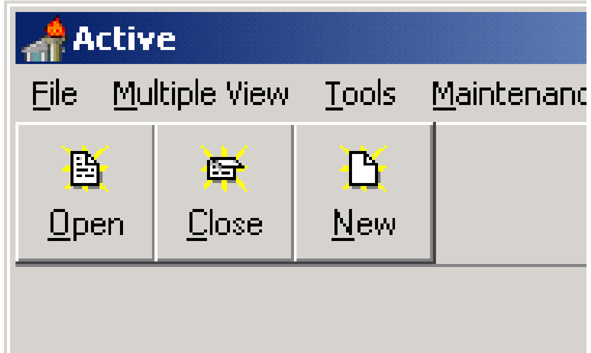
Determining Constants for Fitts' Law

- To determine a and b
  - design a set of tasks with varying values for D and S (conditions)
- For each task condition
  - multiple trials conducted and the time to execute each is recorded and stored electronically for statistical analysis
- Accuracy is also recorded
  - either through the x-y coordinates of selection or
  - through the error rate — the percentage of trials selected with the cursor outside the target

A Quiz Designed to Give You Fitts

http://www.asktog.com/columns/022DesignedToGiveFitts.html

Microsoft Toolbars offer the user the option of displaying a label below each tool. Name at least one reason why labeled tools can be accessed faster. (Assume, for this, that the user knows the tool.)



A Quiz Designed to Give You Fitts

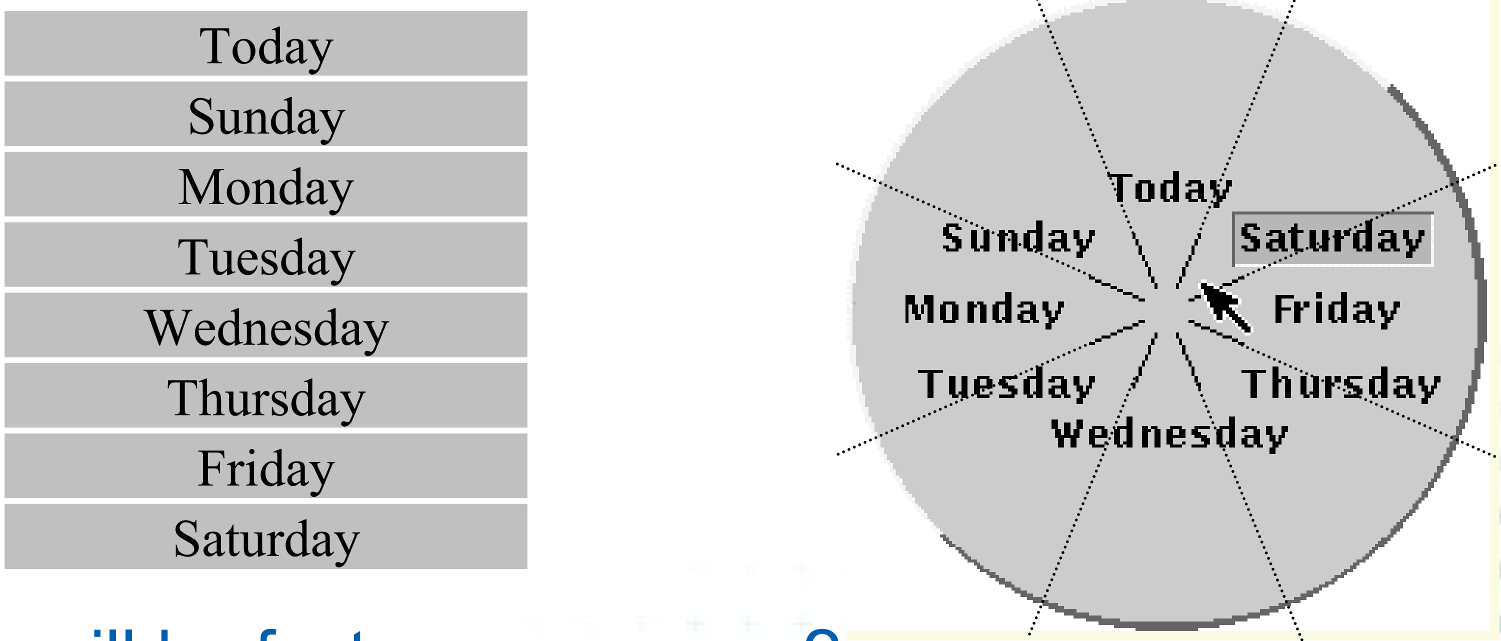
1. The label becomes part of the target. The target is therefore bigger. Bigger targets, all else being equal, can always be acccessed faster, by Fitt's Law.

2. When labels are not used, the tool icons crowd together.

Fitts' Law Example

Pop-up Linear Menu

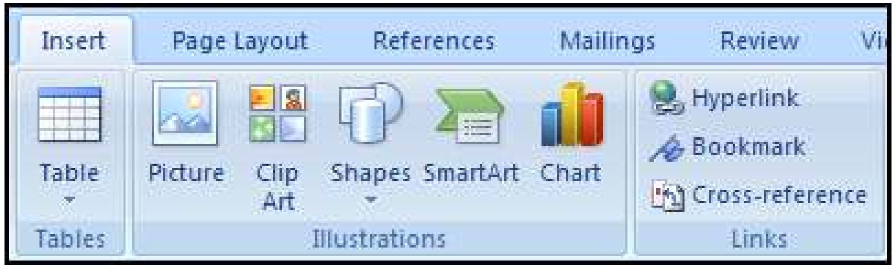
Pop-up Pie Menu



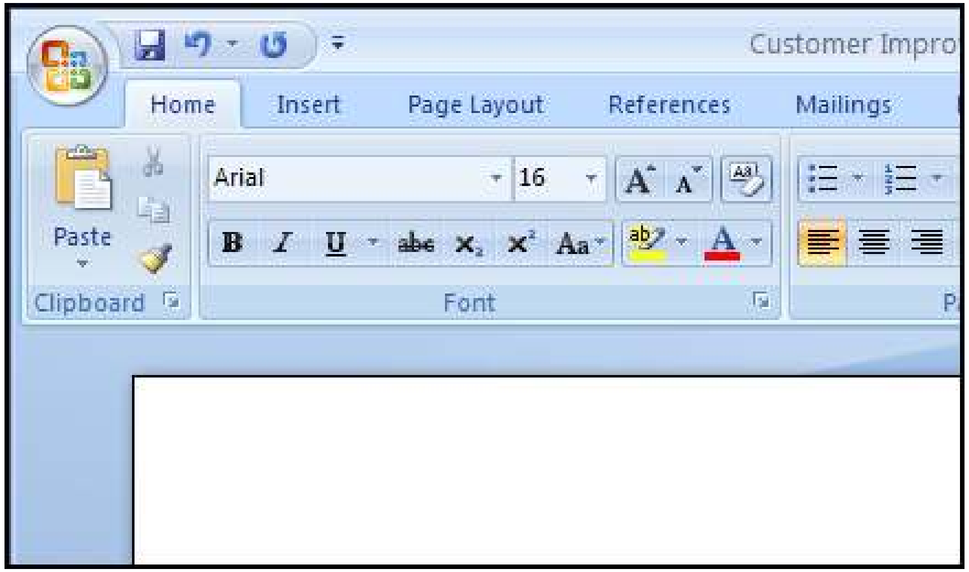
- Which will be faster on average?
  - pie menu (bigger targets & less distance)

Source: Landay, James. "Human Abilities". CS160 UC Berkeley.

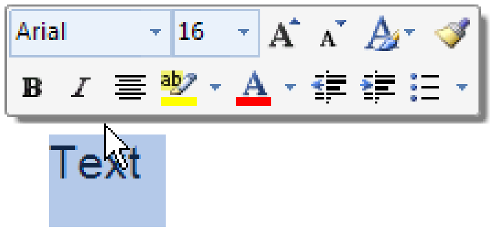
Fitt's Law in Microsoft Office 2007



Larger, labeled controls can be clicked more quickly



Magic Corner: Office Button in the upper-left corner



Mini Toolbar: Close to the cursor

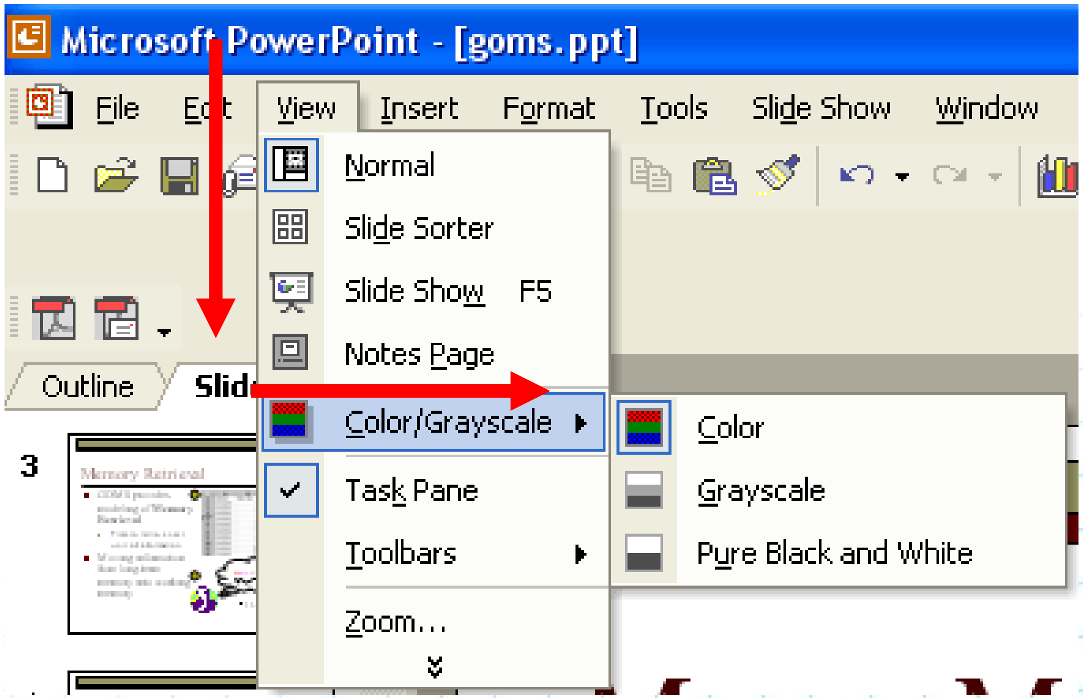
Source: Jensen Harris, An Office User Interface Blog : Giving You Fitts. Microsoft, 2007.

Motor: Key Input

- Parameters of keyboard input based on
  - Skill of the typist
    - Best Typist (120 wpm): 80 msec
    - Worst Typist: 1200 msec
  - Predictability & continuity of the text to be typed
    - Typing random letters: 500 msec

Motor: Mouse Movement

- Fitts's Law is a robust predictor of mouse movement
- Sometimes distance metric is not clear-cut
  - Nested menus

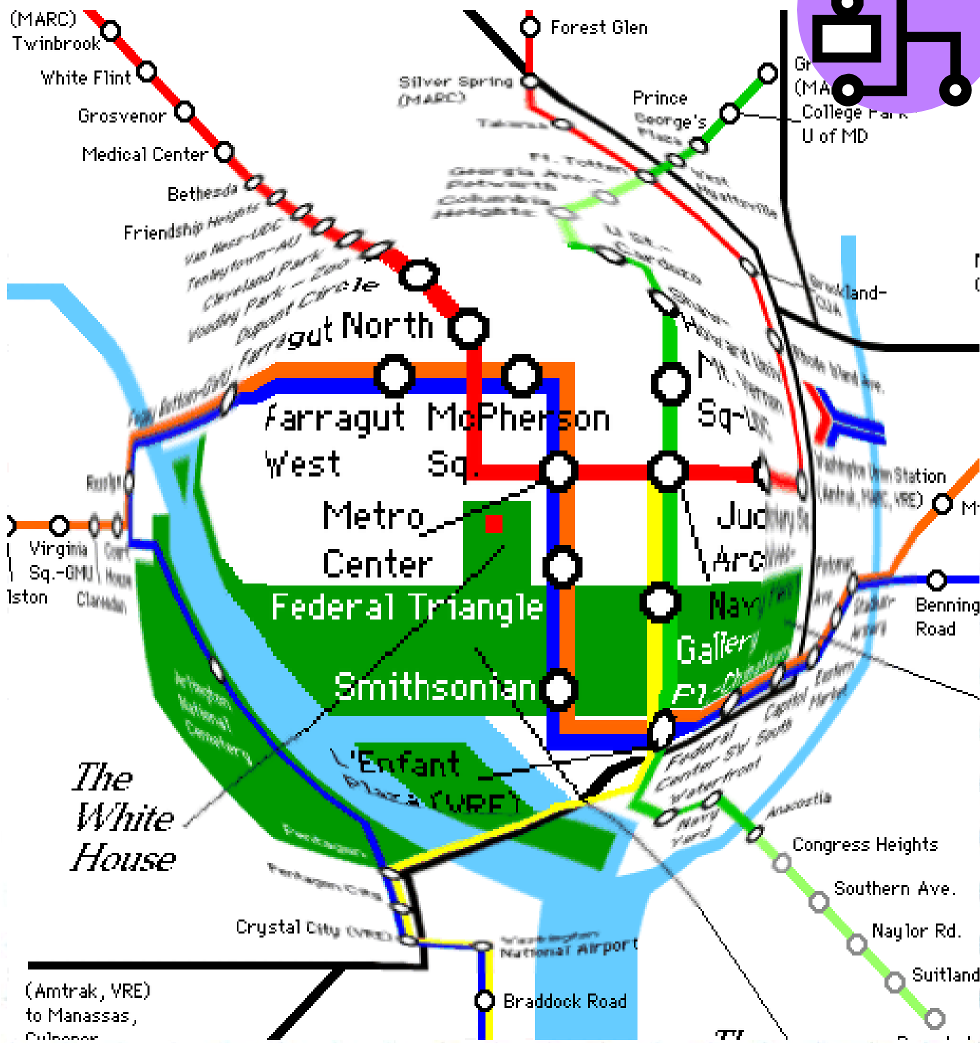


Motor: Applying Fitts's Law

- Fitts's law recommends
  - **Larger target sizes**
  - **Smaller distances to targets**
  - **Usage of corners and edges (they have "infinite" height and width)**
    - Macintosh menus are faster than Windows/Unix style menus because they lie on the screen edge

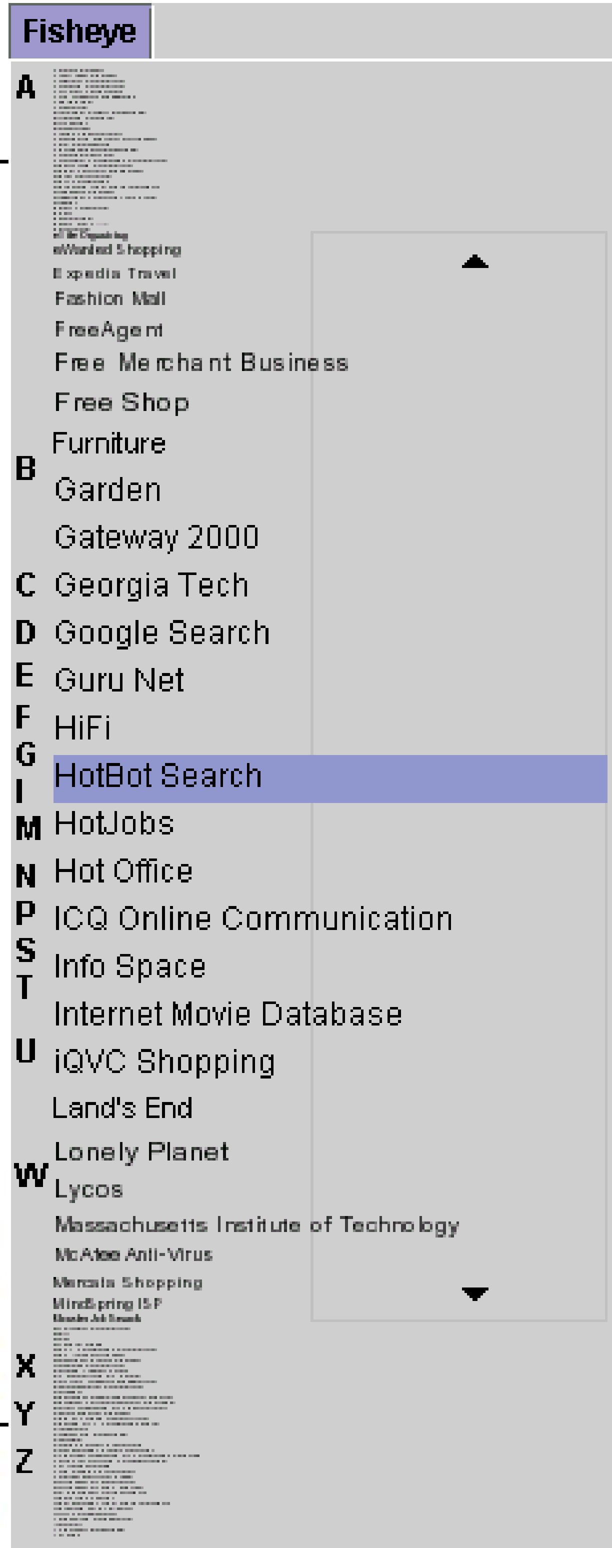
Motor: Fisheye Model

- Provide local context against a global context
- Focuses on screen space versus user's attention
- 3 properties
  - Focal point
  - Distance from focus, ***D***
  - Level of detail, ***LOD***
- Degree of Interest
  - Function to determine whether to display an item or not and its size



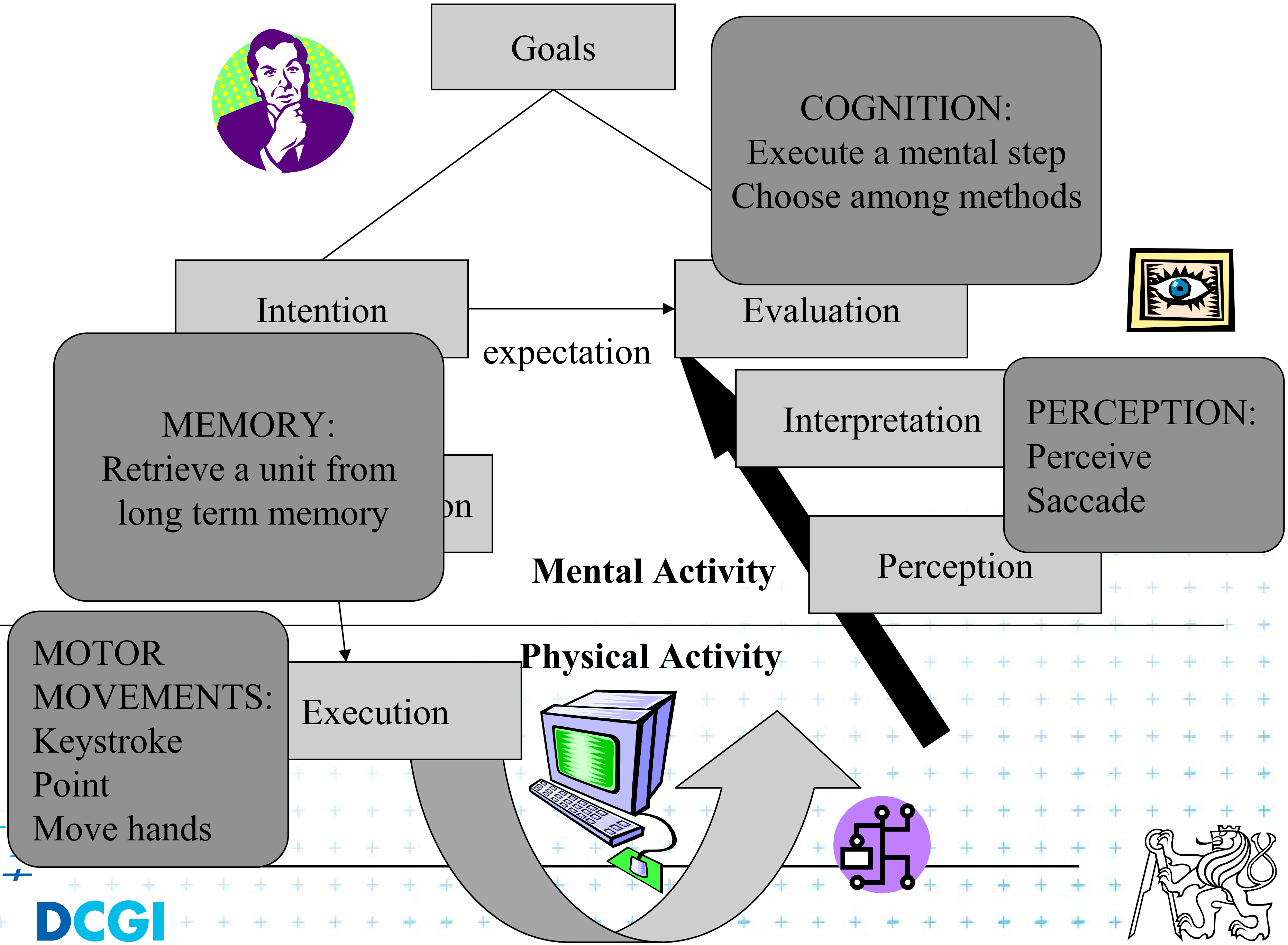
Motor: Fisheye Menu

- Good for browsing tasks
- Allows one to present entire menu without having to use hierarchies or scrolling
- Longer learning curve
- http://www.cs.umd.edu/hcil/fisheyemenu/fisheyemenu-demo.shtml



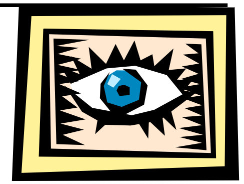
Motor: Hand Movements

- Switching between keyboard and mouse
  - ≈ 360 msec
- Differences in times due to distance from home position on keyboard and the size of the targets
  - Joystick ≈ 260 msec
  - Arrow keys ≈ 210 msec


Goals
COGNITION:
Execute a mental step
Choose among methods
Intention
expectation
Evaluation
MEMORY:
Retrieve a unit from
long term memory
Interpretation
PERCEPTION:
Perceive
Saccade
Mental Activity
Perception
MOTOR
MOVEMENTS:
Keystroke
Point
Move hands
Execution
Physical Activity

DCGI

Perception

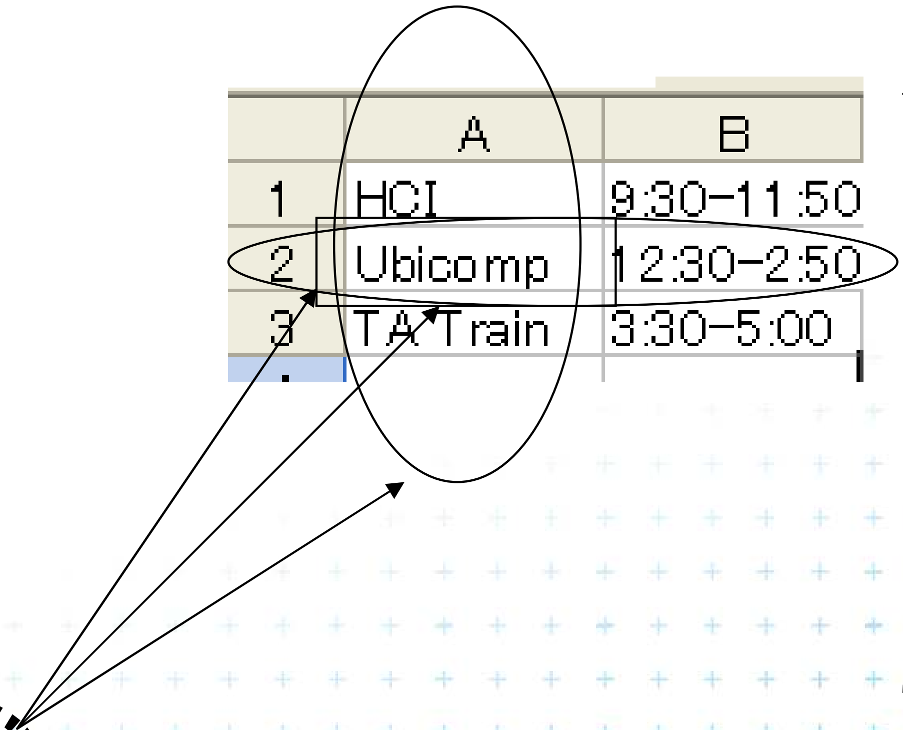


- Recognition or perception
  - Measure the time to respond to stimuli
    - Responding to lights
    - Recognizing words
- Saccade: fast movement of eye, head, etc.
  - Measure the time to move and take in information in each jump
    - Eye jerking around, scanning or moving to the next location

Perception

An example: spreadsheet perception

- Looking for cell addresses and retrieving data



230 msec
130 msec, store row label

230 msec
130 msec, store col label

230 msec
1350 msec, retrieve row & col label

Total: 2300 msec

Summary of Cognitive Parameters

Retrieve from memory	1200 msec
Execute a mental step	70 msec
Choose among methods	1250 msec
Enter a keystroke	230 msec
Point with a mouse	1500 msec
Move hands to mouse	360 msec
Perceive	100 msec
Make a saccade	230 msec

DCGI

Thank you for your attention