PV182
Human Computer Interaction

Lecture 10

Cognitive Models

Fotis Liarokapis
liarokap@fi.muni.cz

14th October 2019

Cognitive Models

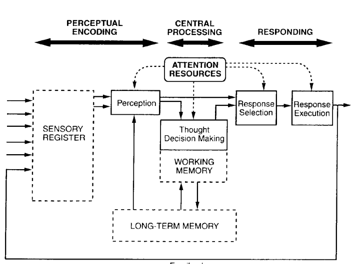
Cognitive Models (Low Level)

- Sources:
  - Marti Hearst (SIMS, UC Berkeley)
  - Robert Stevens (www.cs.man.ac.uk)
  - Susan E. Brennan (www.psychology.stonybrook.edu)
  - Rebecca W. Boren (Arizona State University)

Cognitive Modeling Based Evaluation

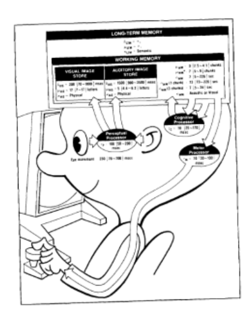
- Fitts' Law
  - Used to predict time needed to select a target
- Keystroke-Level Model
  - Low-level description of what users must do to perform a task
- GOMS
  - Structured, multi-level description of what users must do to perform a task

Model of Human Processing



The Model Human Processor

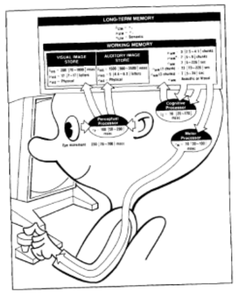
- Perceptual system
  - Sensors
- Cognitive system
  - Processors
- Motor system
  - Effectors



(Card, Moran, & Newell, 1983)

Important Parameters

- Memory capacity
- Decay
- Representation
- Processing cycle time



Sample Times

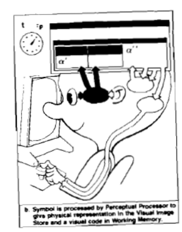
- Eye-movement = 230 [70~700] ms
  - Typical time = 230 ms
  - "Fastman" = 70 ms
  - "Slowman" = 700 ms
- Perceptual processor: 100 [50~200]
- Cognitive processor: 70 [25~170]
- Motor processor: 70 [30~100]

Model of Simple RT Problem

- Task: Press button
  - When symbol appears

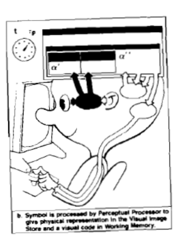
Model of Simple RT Problem .

- Task: Press button
  - When symbol appears
- 1. Perceptual processor captures it in the visual image store & represents it in working memory
  - 100 [50~200]



Model of Simple RT Problem ..

- Task: Press button
  - When symbol appears
- 2. Cognitive processor recognizes the presence of a symbol
  - 70 [25~170]



Model of Simple RT Problem ...

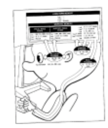
- Task: Press button
  - When symbol appears
- 3. Motor processor pushes the button
  - 70 [30~100]

Model of Simple RT Problem ....

- Task: Press button when symbol appears
- 1. The perceptual processor captures it in the visual image store and represents it in working memory
  - 100 [50~200]
- 2. The cognitive processor recognizes the presence of a symbol
  - 70 [25~170]
- 3. The motor processor pushes the button
  - 70 [30~100]
- Total time?

Model of Simple RT Problem .....

- Each of these action primitives takes some small amount of time (in msec.)
- The Model Human Processor provides a range of parameters you can use to predict precisely how long something will take, or to compare the time needed for alternative actions



Hick's Principle of Uncertainty

- Predicts how long a response will take in a given situation, based on how likely (or uncertain) the different possibilities are

Decision Complexity

- The speed with which an action can be selected is strongly influenced by the number of possible alternative actions that could be selected
- Hick-Hyman Law of reaction time shows a logarithmic increase in reaction time (RT) as the number of possible stimulus-response alternatives (N) increases
  - Humans process information at a constant rate

RT = a + bLog2N

Hick-Hyman Law

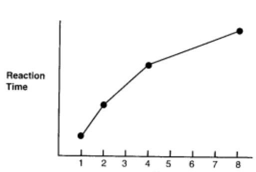


FIGURE 9.1
The Hick-Hyman Law of reaction time. The figure shows the logarithmic increase in RT as the number of possible stimulus-response alternatives (*N*) increases. This can sometimes be expressed by the formula: $RT = a + b\text{Log}2\,N$.

Hick's Principle of Uncertainty

- RT = a + b log2N
  - RT = reaction time
  - a, b = constants
  - N = number of possible responses,
  - assuming all are equally probable

- +1 is due to uncertainty whether to respond

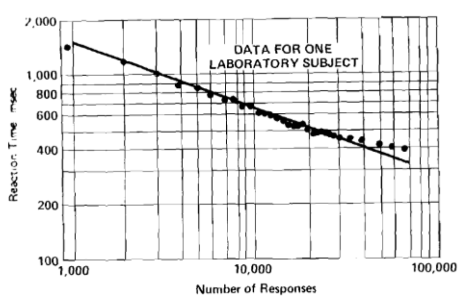
Decision Making Process

- The most efficient way to deliver a given amount of information is by a smaller number of complex decisions rather than a large number of simple decisions
- An example is this decision making process:
  - Would you like to have a big long-hair dog or a small nervous dog or a black cat or a small no-hair cat ?
    - vs.
    - Dog or a cat ? … dog
    - Big or small ? … small
    - Quiet or nervous ? … quiet

Power Law of Practice

- When something is done again and again, performance follows a power law:
  - You keep improving with practice, but as you become an expert, you improve less and less

Power Law of Practice



Note:

- The power law of practice describes *quantitative* changes in skilled behavior (both cognitive and motor), but not *qualitative* changes (changes in strategies)

Fitt's Law

- *Moving a mouse to a target:*
- *What can vary?*
- how long it takes
- how far you have to move
- how big the target is

Fitt's 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

Fitt's Law

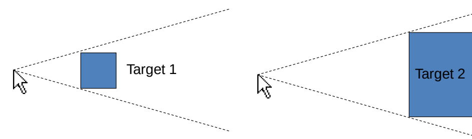
Time (in msec) = $a + b \log_2(D/S+1)$

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

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

Fitt's Law

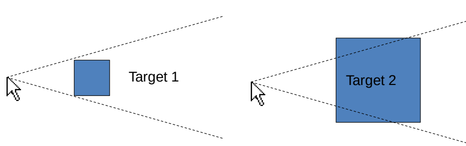
Time = $a + b \log_2(D/S+1)$



Same ID → Same Difficulty

Fitt's Law

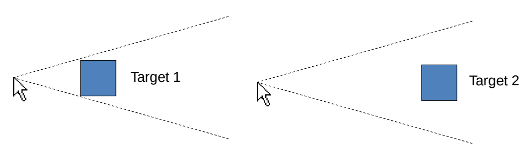
Time = $a + b \log_2(D/S+1)$



Smaller ID → Easier

Fitt's Law

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



Larger ID → Harder

Keystroke-Level Model

- Another "discount" usability method
- Main idea:
  - Walk through the interface, counting how many operations it would take an expert user to perform
  - Look for ways to optimize
  - Look for potential sources of error
- KLM is very low-level (tiny operations)

Keystroke-Level Model

- How to make a KLM
  - List specific actions user does to perform task
    - **K**eystrokes and **B**utton presses
    - Mouse movements / **P**ointing
    - Hand movements between keyboard & mouse / **H**oming
    - **D**rawing
    - System **R**esponse time (if it makes user wait)
  - Add **M**ental operators
  - Assign execution times to steps
  - Sum execution times
- Only provides execution time and operator sequence

KLM times

operator	remarks	time(s)
K	Press key	
	good typist (90wpm)	0.12
	poor typist (40wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitt's law	$0.1 \log_2(D/S+0.5)$
	average movement	1.10
H	home hands to/from kbd	0.40
D	drawing / domain dependent	-
M	mentally prepare	1.35
R	response from system – measure	-

KLM Example

- Replace all instances of a 4-letter word.
  - (example from Hochstein)

Description	Operation	Time (sec)
Reach for mouse	H[mouse]	0.40
Move pointer to "Replace" button	P[menu item]	1.10
Click on "Replace" command	K[mouse]	0.20
Home on keyboard	H[keyboard]	0.40
Specify word to be replaced	M4K[word]	2.15
Reach for mouse	H[mouse]	0.40
Point to correct field	P[field]	1.10
Click on field	K[mouse]	0.20
Home on keyboard	H[keyboard]	0.40
Type new word	M4K[word]	2.15
Reach for mouse	H[mouse]	0.40
Move pointer on Replace-all	P[replace-all]	1.10
Click on field	K[mouse]	0.20
Total		**10.2**

According to this KLM model, it takes 10.2 seconds to accomplish this task.

GOMS model of a system usage

- A family of user interface modeling techniques
- Goals, Operators, Methods, and Selection rules
  - Higher-level than KLM
  - Input: detailed description of UI and task(s)
  - Output: various qualitative and quantitative measures

GOMS (Card, Moran, & Newell)

- **Goal** - what the user wants to achieve
- **Operator** - elementary perceptual, motor, or cognitive act
- **Method** - a **series** of operators that forms a procedure for doing something
- **Selection rule** - how the user decides between methods *(if...then...)*.

GOMS (continued)

- Examples:
- **Goal** - editing a paper *(high level)*
- cutting and pasting text *(low level)*
- **Operator** - typing a keystroke
- **Method** - set of operators for cutting
- **Selection rule** - how the user chooses a method

Applications of GOMS analysis

- Compare UI designs
- Profiling
- Building a help system
  - GOMS modeling makes user tasks and goals explicit
  - Can suggest questions users will ask and the answers

HCI LAB

What GOMS can model

- Task must be goal-directed
  - Some activities are more goal-directed than others
  - Even creative activities contain goal-directed tasks
- Task must a routine cognitive skill
- Task can include serial and parallel tasks

HCI LAB

GOMS Output

- Functionality coverage and consistency
  - Does UI contain needed functions?
  - Are similar tasks performed similarly?
- Operator sequence
  - In what order are individual operations done?
  - Abstraction of operations may vary among models

HCI LAB

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

HCI LAB

Operators vs. Methods

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

HCI LAB

GOMS Example

- Move text in a word processor
  - (example from Hochstein)

```
*Expansion of MOVE-TEXT goal
GOAL: MOVE-TEXT
.     GOAL: CUT-TEXT
.     .     GOAL: HIGHLIGHT-TEXT
.     .         .       [select**: GOAL: HIGHLIGHT-WORD
.     .         .                  .   MOVE-CURSOR-TO-WORD
.     .         .                  .   DOUBLE-CLICK-MOUSE-BUTTON
.     .         .                  .   VERIFY-HIGHLIGHT
.     .         .                  GOAL: HIGHLIGHT-ARBITRARY-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                          1.10
.     .         .     PRESS-MOUSE-BUTTON                                0.10
.     .         .     MOVE-CURSOR-TO-CUT-ITEM                           1.10
.     .         .     VERIFY-HIGHLIGHT                                  1.35
.     .         .     RELEASE-MOUSE-BUTTON                              0.10
```

HCI LAB

GOMS Example

- Move text in a word processor
  - (example from Hochstein)

```
.     GOAL: PASTE-TEXT
.     .     GOAL: POSITION-CURSOR-AT-INSERTION-POINT
.     .         MOVE-CURSOR-TO-INSERTION-POIONT                     1.10
.     .         CLICK-MOUSE-BUTTON                                  0.20
.     .         VERIFY-POSITION                                     1.35
.     .         GOAL: ISSUE-PASTE-COMMAND
.     .         .     MOVE-CURSOR-TO-EDIT-MENU                      1.10
.     .         .     PRESS-MOUSE-BUTTON                            0.10
.     .         .     MOVE-MOUSE-TO-PASTE-ITEM                      1.10
.     .         .     VERIFY-HIGHLIGHT                              1.35
.     .         .     RELEASE-MOUSE-BUTTON                          0.10
                                  TOTAL TIME PREDICTED (SEC)       14.38
```

Based on the above GOMS analysis, it should take 14.38 seconds to move text.

HCI LAB

Advantages of GOMS

- Very general purpose
- Allows for individual differences
- Much predictive power about timing
- Good at predicting "ideal" performance
- Gives several qualitative and quantitative measures
- Model explains **why** the results are what they are
- Less work than user study
- Easy to modify when interface is revised
- Research ongoing for tools to aid modeling process

HCI LAB

Disadvantages of GOMS

- Not so good at predicting errors
- Takes a long time to conduct analysis
- Whole may not be the sum of the parts
- Ignores the nature of internal symbolic representations - focus is very low-level
- **Not as easy** as heuristic analysis, guidelines, or cognitive walkthrough

HCI LAB

Disadvantages of GOMS

- Only works for goal-directed tasks
- Assumes tasks are performed by expert users
- Evaluator must pick users' tasks/goals
- Does not address several important UI issues, such as
  - readability of text
  - memorability of icons, commands
- Does not address social or organizational impact

HCI LAB

Summary

- We can use Cognitive Modeling to make predictions about interface usability
- Complementary to Usability Studies
- In practice:
  - GOMS not used often
  - Fitt's law often used for determining best case for new kinds of input methods

HCI LAB

Questions

HCI LAB

Acknowledgements

- Prof. Ing. Jiří Sochor