CS160: User Interface Design

Human Information Processing (KLM, GOMS, Fitts' law) 01/03/10

Most heavily used features directly mapped (volume, play/pause)
Circular movements mapped to linear operations

http://www.youtube.com/watch?v=WhsO449Fzq4

Individual Programming Assignment 4 (due Mar 3)
Assignment: Low Fidelity Prototype

Due Mar 15
Identify project mission statement
Create low-fidelity prototype that supports 3 tasks
- 1 easy, 1 moderate, 1 difficult task
Create a video prototype showing (cameras next class)
How it supports the 3 tasks
Context in which it will be used (back story)
Must include narration
Test the prototype with target users
No one from this class
Not your friends

Review: Human Info Processor

5 Parts
Perceptual
Cognitive
Motor (will discuss today)
Working memory
Long-term memory

Unified model
Probably inaccurate
Predicts perf. well
Very influential

Review: Pop-Out and Causality

Motor Processor

Receive input from the cognitive processor
Execute motor programs
Pianist: up to 16 finger movements per second
Point of no-return for muscle action
Recognize-Act Cycle

Are Q and X letters?

Cognitive Processor

Page 70 of Card Moran and Newell

Clocks starts when 2nd letter is flashed
Move letter x into visual store WM

Recognize the symbol as codes
Classify the codes as letters
Match the fact that they are both letters
Initiate motor response
Process motor command
Approx 450 (180-980) ms

Human Interaction Loops

(Newell)

System loop: 10's (years)
Design loop: 10's (days-10's months)
Task loop: 10's-100's (weeks)
Unit task loop: 10's
Operator loop: 10 s

Motor loop: 5 ms
Perception

Principles of Operation

Interface should respect limits of human performance
Pre-attentive features pop-out
Events within cycle time fuse together
Causality

Recognize-Act Cycle of the cognitive processor
On each cycle contents in WM initiate cognitive actions
Cognitive actions modify the contents of WM

Discrimination Principle
Retrieval is determined by candidates that exist in memory relative to retrieval cues
Interference by strongly activated chunks
Two strong cues in working memory
Link to different chunks in long term memory

Topics

Memory
Decision Making and Learning
Fitts’ Law
GOMS and KLM

Simple Experiment

Volunteer

Start saying colors you see in list of words
When slide comes up
As fast as you can

Say “done” when finished
Everyone else time it…
Simple Experiment

Do it again

Say “done” when finished

Stage Theory

- Sensory Image Store
- Working Memory
- Long Term Memory

- maintenance
  - rehearsal
- decay
- decay, displacement
- decay?
- interference?
- chunking / elaboration
### Stage Theory

**Working memory is small**
- Temporary storage
- decay
- displacement

**Maintenance rehearsal**
- Rote repetition
- Not enough to learn information well

### LTM and Elaboration

**Recodes information**

**Organize (chunking)**

**Relate new material to already learned material**

**Link to existing knowledge, categories**

**Attach meaning**
- Make a story

### LTM Forgetting

**Causes for not remembering an item?**
1) Never stored: encoding failure
2) Gone from storage: storage failure
3) Can’t get out of storage: retrieval failure

**Interference model of forgetting**
- One item reduces ability to retrieve another
- Proactive interference (3)
  - Earlier learning reduces ability to retrieve later info.
- Retroactive interference (3 & 2)
  - Later learning reduces the ability to retrieve earlier info.

### Recognition over Recall

**Recall**
- Info reproduced from memory

**Recognition**
- Presentation of info helps retrieve info (helps remember it was seen before)
- Easier because of cues to retrieval

**We want to design UIs that rely on recognition!**
Recall

Recognition

Facilitating Retrieval: Cues

Any stimulus that improves retrieval
Example: giving hints
Other examples in software?
icons, labels, menu names, etc.

Anything related to
Item or situation where it was learned
Decision Making and Learning

Hick's Law

Cost of taking a decision:  \( T = a + b \log_2(n + 1) \)

Power Law of Practice

Task time on the nth trial follows a power law

\[ T_n = T_1 n^{-a} + c \]

where \( a = .4 \), \( c = \) limiting constant

You get faster the more times you do it!
Power Law of Practice

Task time on the nth trial follows a power law

\[ T_n = T_1 n^{-a} + c \]

where \( a = 0.4, c \) = limiting constant
You get faster the more times you do it!

Applies to skilled behavior (sensory & motor)

Does not apply to
Knowledge acquisition
Improving quality

Stages of skill acquisition

Example: Using a manual transmission

- **Cognitive**
  - Verbal representation of knowledge
- **Associative**
  - Proceduralization
  - Form of chunking
- **Autonomous**
  - More and more automated
  - Faster and faster
  - No cognitive involvement
  - Difficult to describe what to do

Fitts’ Law
Motor Processor

- Receive input from the cognitive processor
- Execute motor programs
- Pianist: up to 16 finger movements per second
- Point of no-return for muscle action

Fitts’ Law

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

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

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

- Models well-rehearsed selection task
- \( T \) increases as the distance to the target increases
- \( T \) decreases as the size of the target increases

Hand movement based on series of microcorrections

- \( X_i \) = remaining distance after \( i \)th move
- Relative movement accuracy remains constant \( \frac{X_i}{X_i+1} = \epsilon \)
Considers Distance and Target Size

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

- Same ID → Same Difficulty
- Smaller ID → Easier

Experimental Data
Extend Fitts' Law to 2D Targets?

To apply Fitts' Law to 2D targets, consider the following:

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

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

---

**Toolbar Matrix Example**

You have a palette of tools in a graphics application that consists of a matrix of 16x16-pixel icons laid out as a 2x8 array that lies along the left-hand edge of the screen. Without moving the array from the left-hand side of the screen or changing the size of the icons, what steps can you take to decrease the time necessary to access the average tool?
Tool Matrix Example

1. Change the array to 1x16, so all the tools lie along the edge of the screen.

2. Ensure that the user can click on the very first row of pixels along the edge of the screen to select a tool. There should be no buffer zone.

GOMS and KLM

GOMS (Card et al.)

Describe the user behavior in term 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

Quick Example

Goal (the big picture)
Go from hotel to the airport

Operators (or subgoals)?
Walk, take bus, take taxi, rent car, take train

Methods (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
GOMS Output

Execution time
Add up times from operators/methods
Assumes experts (mastered the tasks)

Error free behavior
Very good rank ordering
Absolute accuracy ~10-20%

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?

How to do GOMS Analysis

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

Detailed Task Description
Using a text editor edit the following text as shown

The fox jumps over the lazy quick brown dog.

Goals and sub-goals?
Operators?
Methods?
Selection rules?

Keystroke Level Model (KLM)

Describe the task using the following operators:

K: pressing a key or a pressing (or releasing) a button
t_k = 0.08 - 1.2s (0.2 good rule of thumb)
P: pointing
t_p = 1.1s (without button press)
H: Homing (switching device)
t_H = 0.45
D(n/l): Drawing segmented lines
t_D = 0.9n + .16l
M: Mentally prepare
t_M = 1.35s
R(t): system response time

GOMS Example II

KLM Heuristic Rules (Raskin's)

0: Insert M in front of all K
In front of all P's selecting a command (not in front of P's ending command)
1: Remove M between fully anticipated operators
PK → P
2: if a string of MKs belong to cognitive unit delete all M but first
MKKKKKKKKKK → MKKKKKK
3: if K is a redundant terminator then delete M in front of it
MKMK → MK
4a: if K terminates a constant length string (command name) delete the M in front of it
MKMK → MK
4b: if K terminates a variable length string (parameter) keep the M in front of it
MKMK → MK
Using KLM

Encode using all physical operator (K, P, H, D(n,l), R(t))

Apply Raskin’s KLM rules [0-4]

Transform R followed by an M
If $t \leq t_M$: $R(t) \rightarrow R(0)$
If $t_M < t$: $R(t) \rightarrow R(t - t_M)$

Compute the total time by simply adding all time