User Modeling 1

Predicting thoughts and actions
…and how long they take

User Modeling

- Build a model of how a user works, then predict how she will interact with the interface.
- Goals (Salvendy, 1997):
  1. Predict performance of design alternatives
  2. Evaluate suitability of designs to support and enhance human abilities and limitations
  3. Generate design guidelines that enhance performance and overcome human limitations

Note: No mockup is required
…just a design spec.

Three Views of Users (again)

- Human as a biomechanical machine
- Human as Information Processing machine
  - "Procedural" models
  - Many subfamilies and related models
- Human as a social actor
  - Situated action
  - Activity theory
  - Distributed cognition

} *Next lecture*

Agenda

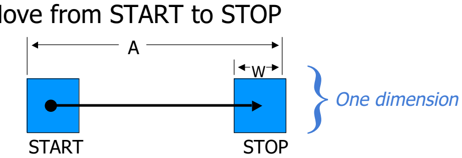
- Physical models
  - Fitt's Law
  - Simulations
- Cognitive models
  - Model Human Processor (MHP)
  - Goals, operators, methods & selection rules (GOMS)
    - Basic GOMS
    - Keystroke-level model (KLM) GOMS
    - Variations on GOMS
  - Rule-based models
    - Production systems
    - Grammar-based models

Fitts' Law

- Fitts' Law
  - Models movement times for selection tasks
  - Paul Fitts: WWII human factors pioneer
- Basic idea: Movement time for a *well-rehearsed* selection task
  - Increases as the distance to the target increases
  - Decreases as the size of the target increases

Moving

- Move from START to STOP



Index of Difficulty:

ID = $\log_2$ (2A/W) (in unitless bits)

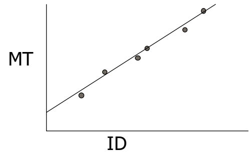
distance — width of target

Movement Time

MT = a + b*ID

or

$MT = a + b \log_2 (A/W + 1)$



Different devices/sizes have different movement times--use this in the design

Where can this be applied in interface design?

What about movement in 2D?

Extending to 2D, 3D

- What is W in 2D? In 3D?
- Larger movements?
  - Short, straight movements are replaced by biomechanical arcs
  - In that case, what is *A*?

Simulations

- Higher-level, view humans as components of a human-machine system
  - Analytical solution of model becomes too difficult, so is replaced by simulation
- E.g., MicroAnalysis and Design (maad.com)
  - Micro Saint - any type of models!
  - WinCrew - workload models
  - Supply Solver - supply chain

Agenda

- Physical models
  - Fitt's Law
  - Simulations
- **Cognitive models**
  - **Model Human Processor (MHP)**
  - Goals, operators, methods & selection rules (GOMS)
    - Basic GOMS
    - Keystroke-level model (KLM) GOMS
    - Variations on GOMS
  - Rule-based models
    - Production systems
    - Grammar-based models

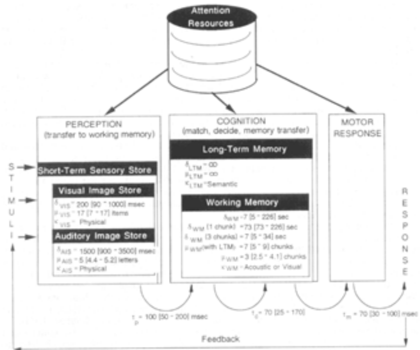
Model Human Processor

- Card, Moran, & Newell (1983, 1986)
- "Procedural" models:
  - People learn to use products by generating rules for their use and "running" their <u>mental model</u> while interacting with system
- Components
  - A cognitive *system architecture*
    - Set of memories and processors
    - Set of "principles of operation"
  - Discrete, sequential model
    - Each stage has timing characteristics (add the stage times to get overall performance times)

Three Systems in MHP Architecture

- Perceptual, cognitive, motor systems
  - Timing parameters for each stage/system
  - Cycle times (τ):
    - $\tau_p \approx 100$ ms ("middle man" values)
    - $\tau_c \approx 70$ ms
    - $\tau_m \approx 70$ ms
  - Perception & Cognition have memories
  - Memory parameters
    - Code, decay time, capacity

MHP: Model and Parameters



MHP: Principles of Operation

- Set up rules for how the components respond.
- Can be based on experimental findings about humans.
  - Recognize-act cycle
  - variable perceptual processor rate
  - encoding specificity
  - Discrimination
  - variable cognitive processor rate
  - Fitts' law
  - Power law of practice
  - Uncertainty
  - Rationality
  - problem space

Applying the MHP

- Example: Designing menu displays
  - 16 menu items in total
  - Breadth (1x16) vs. Depth (4x4) ?

Menu:
1. Menu item a
2. Menu item b
3. Menu item c
4. Menu item d
5. Menu item e
6. Menu item f

…

Main Menu:
1. Menu item a
2. Menu item b
3. Menu item c
4. Menu item d

Submenu:
1. Menu item a
2. Menu item b
3. Menu item c
4. Menu item d

Submenu:
1. Menu item a
2. Menu item b
3. Menu item c
4. Menu item d

MHP: Calculations

Breadth (1x16):

τ_p perceive item, transfer to WM

τ_c retrieve meaning of item, transfer to WM

τ_c Match code from displayed to needed item

τ_c Decide on match

τ_m Execute eye mvmt to (a) menu item number (go to step 6) or (b) to next item (go to step 1)

τ_p Perceive menu item number, transfer to WM

τ_c Decide on key

τ_m Execute key response

Time = $[((16+1)/2)\ (\tau_p + 3\tau_c + \tau_m)] + \tau_p+\tau_c+\tau_m$

Time = 3470 msec

Serial terminating search over 16 items

Depth (4x4):

Same as for breadth, but with 4 choices, and done up to four times (twice, on average):

Time = $2 \times [((4+1)/2)\ (\tau_p + 3\tau_c + \tau_m)] + \tau_p+\tau_c+\tau_m$

Time = 2380 msec

Therefore, in this case, 4x4 menu is predicted to be faster than 1x16.

Related Modeling Techniques

- Many modeling techniques fall within this "human as information processor" model
- Common thread - hierarchical decomposition
  - Divide behaviors into smaller chunks
  - Questions:
    - What is unit chunk?
    - When to start/stop?

Agenda

- Physical models
  - Fitt's Law
  - Simulations
- Cognitive models
  - Model Human Processor (MHP)
  - **Goals, operators, methods & selection rules (GOMS)**
    - **Basic GOMS**
    - **Keystroke-level model (KLM) GOMS**
    - **Variations on GOMS**
  - Rule-based models
    - Production systems
    - Grammar-based models

GOMS

- **G**oals, **O**perators, **M**ethods, **S**election Rules
  - Card, Moran, & Newell (1983)
- Assumptions
  - 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

GOMS: Components

- Goals
  - State to be achieved
- Operators
  - Elementary perceptual, cognitive, motor acts
    - Not so fine-gained as Model Human Processor
- Methods
  - Procedures for accomplishing a (sub)goal
    - e.g., move cursor via mouse or keys
- Selection Rules
  - if-then rules that determine which method to use
- Note: All have times associated with them

GOMS: Procedure

- Build the model by walking through sequence of steps
- Determine a branching tree of operators, methods, and selection rules
- Count the operations to generate performance prediction
  - Can convert into a timing prediction by using an extension of GOMS

GOMS: Example

- Menu structure (breadth vs. depth, again)
- Breadth (1x16):

Goal: perform command sequence
- **Goal**: perform unit task of the command (Loops)
  - **Goal**: determine which unit task to do
    - **Operator**: Look at screen, determine next command
  - **Goal**: Execute unit task
    - **Select**: Which method to enter number of command
    - e.g. IF item # between 1 & 9 THEN use 1-KEY METHOD
    - **Operator**: Use 1-Key Method
    - **Operator**: Verify Entry… etc.

Result: Average Number of Steps = 8.5+1+1+1=11.5

GOMS: Example, cont'd

- Depth (4x4):
- Similar steps, in slightly different order and looping conditions
  - Result: Average Number of Steps = 2*(2.5+1+1)=9
- Comparison: Depth is ~25% faster in this case
  - Card et al. did not specify step length (in time)
  - Similar to Model Human Processor results

GOMS: Limitations

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

GOMS: Application

- NYNEX telephone operation system
  - GOMS analysis used to determine critical path, time to complete typical task
  - Determined that new system would actually be slower
  - Abandoned, saving millions of dollars

GOMS: Variants

- Keystroke Level Model
  - Analyze only observable behaviors
    - Keystrokes, mouse movements
  - Assume error-free performance
  - Operators:
    - K: keystroke, mouse button push
    - P: point with pointing device
    - D: move mouse to draw line
    - H: move hands to keyboard or mouse
    - M: mental preparation for an operation
    - R: system response time

Example of KLM

- Breadth menu (1x16)
  - M: Search 16 items
  - 1 or 2 K: Enter 1 or 2-digit number
  - K: Press return key

Time=M + K(first digit) + 0.44K(second digit) + K(Enter)
(Look up values, and when to apply "M" operator)
Time=1.35 + 0.20 + 0.44(0.20) + 0.20 = 1.84 seconds

Note: Many assumptions about typing proficiency, M's, etc.
Also ignores most of the time spent determining which task to perform, and how to perform it.

Example of KLM, cont'd

- Depth menu (4x4)
  - M: Search 4 items
  - K: Enter 1-digit number (no M, since expert user)
  - K: Press return key

Time=M + K(Digit) + K(Enter)
Time=1.35 + 0.20 + 0.20 = 1.75 seconds

- Compare the various models in terms of times and predictions:
  - Vary in times, but not in performance predictions

Other GOMS Variants

- NGOMSL (Kieras, 1988)
  - Very similar to GOMS
  - Goals expressed as noun-action pair
    - e.g., "delete word"
  - Same predictions as other methods
  - More sophisticated, incorporates learning, consistency (relevant to usability and design)
  - Handles expert-novice differences, etc.

Agenda

- Physical models
  - Fitt's Law
  - Simulations
- Cognitive models
  - Model Human Processor (MHP)
  - Goals, operators, methods & selection rules (GOMS)
    - Basic GOMS
    - Keystroke-level model (KLM) GOMS
    - Variations on GOMS
  - **Rule-based models**
    - **Production systems**
    - **Grammar-based models**

Production Systems

- IF-THEN decision trees (Kieras & Polson, 1985)
  - e.g. Cognitive Complexity Theory
  - Uses goal decomposition from GOMS and provides more predictive power
  - Goal-like hierarchy expressed using if-then production rules
- Very long series of decisions
  - Note: In practice, very similar to NGOMSL
    - Bovair et al (1990) claim they are identical
  - NGOMSL model easier to develop
  - Production systems easier to program

Grammars

- To describe the interaction, a formalized set of productions rules (a language) can be assembled.
- "Grammar" defines what is a valid or correct sequence in the language.
- Reisner (1981) "Cognitive grammar" describes human-computer interaction in Backus-Naur Form (BNF) like linguistics
- Used to determine the consistency of a system design

Task Action Grammars (TAG)

- Payne & Green (1986, 1989)
- Concentrates on overall structure of language rather than separate rules
- Designed to predict relative complexity of designs
- Not for quantitative measures of performance or reaction times.
- Consistency & learnability determined by similarity of rules

Agenda

- Physical models
  - Fitt's Law
  - Simulations
- Cognitive models
  - Model Human Processor (MHP)
  - Goals, operators, methods & selection rules (GOMS)
    - Basic GOMS
    - Keystroke-level model (KLM) GOMS
    - Variations on GOMS
  - Rule-based models
    - Production systems
    - Grammar-based models

What are the problems with predictive models?

Modeling Problems

- Terminology - example
  - High frequency use experts -> command language
  - Infrequent novices -> menus
  - What do "frequent", "novice" really mean?
- Dependent on "grain of analysis" used
  - Can break down pumping gas into 7, 20, or 50 tasks
  - That affects number of rules and their types

Modeling Problems (contd.)

- Does not involve user per se
  - Doesn't inform designer of what user wants
- Time-consuming and lengthy
- One user, one computer issue (lack of social context)
  - i.e., non-situated
  - Can use Socially-Centered Design

Next lecture