GOMS model


- Goals, Operators, Methods, Selection rules
- Top level: Goals (hierarchical) that users wish to accomplish
- Bottom level: Operators
  - elementary acts (perceptual, motor, cognitive) that change mental state or task environment
GOMS model

- **Goals** express what needs to be done
- **Operators** are elementary cognitive, perceptual, motor acts
- **Methods** accomplish a **Goal** by invoking **Operators**
- **Selection rules** choose an appropriate **Method**
- **Variants:**
  - KLM [simplest form of GOMS]
  - NGOMSL (Natural GOMS Language)

NGOMSL  
D. Kieras, 1988

- Editing a document online based on marked-up paper manuscript (or PDF)
- Goal "edit the document" accomplished by method that decomposes it further into subgoals

Method for goal: edit the document
Step 1. Get next unit task information from marked-up manuscript.
Step 2. Decide: If no more unit tasks, then return with goal accomplished.
Step 3. Accomplish goal: move to the unit task location.
Step 4. Accomplish goal: perform the unit task.
Step 5. Go to 1.

Selection rule rule set for goal: perform the unit task
If the task is moving text, then
accomplish goal: move text.
If the task is deletion, then
accomplish goal: delete text.
If the task is copying, then
accomplish goal: copy text.
...etc....
Return with goal accomplished.

Method for goal: move to the unit task location
Step 1. Get location of unit task from manuscript.
Step 2. Decide: If unit task location on screen, return with goal accomplished.
Step 3. Use scroll bar to advance text.
Step 4. Go to 2.

http://www.eecs.umich.edu/~kieras/goms.html
(But, all documents linked are actually in http://web.eecs.umich.edu/~kieras/docs/GOMS/)

Feiner, COMS W4170, Fall 2015
GOMS model

- Variants:
  - CPM-GOMS (Cognitive, Perceptual, Motor or Critical Path Method) handles parallel actions

CPM-GOMS
B. John and W. Gray, GOMS Analysis for Parallel Activities, *CHI 94 Conference Companion*

Diagram details are not intended to be legible on slide or in course notes!

**Bold line is critical path**
CPM-GOMS

- Project Ernestine: GOMS analysis of two terminals used by TAOs (Toll and Assistance Operators)
- Original
  - 300 baud
  - Character display
- New
  - 1200 baud
  - Well designed GUI
    - Keyboard minimized hand travel
    - UI reduced keystrokes
    - Screen displayed faster

Back-of-envelope calculations predicted 20% increase in performance
- Each second saved per average call = $3M/year

But
- CPM-GOMS analysis showed .63 seconds slower (weighted for call types and frequency)
- Field trial showed .65 seconds slower
- Predicted loss = $2M/year
Reasons for performance decrease

- Eliminated keystrokes not on CP (Critical Path)—ones that didn’t affect overall timing, and
- When reducing keystrokes, some were moved from off CP to on CP, introducing delay

Section of CPM-GOMS analysis near beginning of call

- Proposed workstation (right) removes 2 keystrokes (7 motor & 3 cognitive ops), but none are on CP (in bold).
**CPM-GOMS**


- Section of CPM-GOMS analysis at end of call
  - Proposed workstation (right) adds 1 keystroke (3 motor & 1 cognitive op) **directly on CP** (in bold)

**Reasons for performance decrease**

- Decreased parallelism in use of hands
  - Old: LH pressed a key, moving while RH still keying, so was ready when RH done
  - New: That key was moved closer to other keys, so RH would press it in sequence, on CP

- Added wait to see crucial info
  - Old: Displayed first line faster (info in CP)
  - New: Whole screen displayed faster, but first line was delayed by > .5 seconds
Model Human Processor

- Model (inspired by computers) of how humans perceive, process, and act on information
- Processors
  - Perceptual processor
  - Cognitive processor
  - Motor processor
- Memory
  - Visual image store
  - Auditory image store
  - Working memory
  - Long-term memory

Processor
- $\tau$ Cycle time
- $\delta$ Decay
- $\mu$ Capacity
- $\kappa$ Coding

Cognitive Processor has Recognize–Act cycle: Contents of WM initiate actions in LTM ("recognize"), which modify contents of WM ("act").
**Model Human Processor**

- Values derived from human studies
- $\tau_P$ Perceptual processor cycle time
  - $< 100$ msecs visual stimulus for $n$ msecs is perceived same as double intensity stimulus for $n/2$ msecs
  - $100 \ [50\sim 200]$ msecs
- $\tau_C$ Cognitive processor cycle time
  - Time to count mentally
  - $70 \ [25\sim 170]$ msecs
- $\tau_M$ Motor processor cycle time
  - Tapping
  - $70 \ [30\sim 100]$ msecs

**Motor Processor Cycle Time: Anecdotal Evidence**

- $1208 / 60 = 20.13$ bps (two hands)
- $20.13 / 2 = 10.07$ bps (one hand)
- Implies $\tau_M \leq 99.3$ msec
  - Includes fatigue 😊

http://www.extremesportdrumming.com
http://worldsfastestdrummer.com
Model Human Processor

- $\delta_{\text{VIS}}$ Visual store decay
  - Show letters for 50 msecs
  - Blank screen for specified time $n$
  - Show pointer @ random letter location for 50 msecs
  - Can user identify letter @ pointer location?
  - Can do 50% of the time for $n \leq 200$ msecs

- $\mu_{\text{WM}}$ Working memory capacity
  - Present set of letters briefly, then ask users to report ones they see. Always limited, even though they say they see all. ~ 3 chunks
  - $\mu_{\text{WM}} = 7 \pm 2$ chunks (with rehearsal)
  - George Miller, 1956

- $\mu_{\text{VIS}}$ Visual store capacity
  - Present $q$ rows of $n$ letters each, followed by a pointer to one row. Then ask what was in the row. If subject gives $m$ of $n$ letters, $\mu_{\text{vis}} = (m/n) \times q \times 17$ [7~17] letters
Memory Capacity and Decay

- Model Human Processor v. Model Chimpanzee Processor

- Try it yourself!
Seven Stages of Action
D. Norman, 1988

"The basic idea is simple. To get something done, you have to start with some notion of what is wanted—the goal that is to be achieved. Then, you have to do something to the world, that is, take action to move yourself or manipulate someone or something. Finally, you check to see that your goal was made. So there are four different things to consider: the goal, what is done to the world, the world itself, and the check of the world. The action itself has two major aspects: doing something and checking. Call these execution and evaluation."
— D. Norman,
The Psychology of Everyday Things, 1988
Seven Stages of Action
D. Norman, 1988

- Gulf of execution
  - Mismatch between what you want to do and what you can do
- Gulf of evaluation
  - Mismatch between what world state tells you and what you want to know

Feiner, COMS W4170, Fall 2015
**Widget-Level Models**

- Apply KLM approach not just to low-level actions (key presses and mouse motion), but to interactions with high-level “widgets” (e.g., CogTool)
- Evaluate individual widgets, then make predictions about UIs composed of those widgets
- Develop standard design patterns
  - E.g., www.designofsites.com

**Transition Networks**

- Show UI states and actions that cause transitions between states
- Can be used to analyze consistency / simplicity

---

*Fig. 9.9 State diagram of a user interface with an inconsistent syntax.*

Foley et al. 90
Grammars

- Express structure of a UI
- Can be used to analyze consistency/simplicity

Evaluating UIs through Formal Grammars
P. Reisner, 81

- Evaluated “action language” of two drawing system UIs: ROBART 1 and ROBART 2
  - ROBART 2 was designed to be easier (and tests showed it was)
- Described each language as a formal grammar (modified BNF)
- Showed ROBART 2 had simpler grammar. For example,
  - ROBART 1 had two different ways to select type of object to create (text different—just type at keyboard) and differing numbers of actions for other objects
  - ROBART 2 had only one way to select type of object to create
- Used the grammars to predict how users would perform
  - Predicted that Robart 2 would be easier to learn/remember
  - Predicted users would take varying amounts of time to learn/remember how to select objects in Robart 1 and not in Robart 2
  - Predicted users would try to treat text the same way as other objects after first learning how to select other object types to create
- Predictions confirmed by analysis of time to learn with documentation, observations of use without documentation, error rate, and questionnaires
Evaluating UIs through Formal Grammars
P. Reisner, 81

- Prior “conventional wisdom”: In order of decreasing priority:
  - Minimize # lowest level action primitives
  - Minimize length of action sequences
  - Minimize # rules
- Reisner showed *this order should be reversed* to ease learning for naïve users doing nonroutine tasks