Keystroke Level model

Olivier Georgeon & Frank Ritter, march 2012

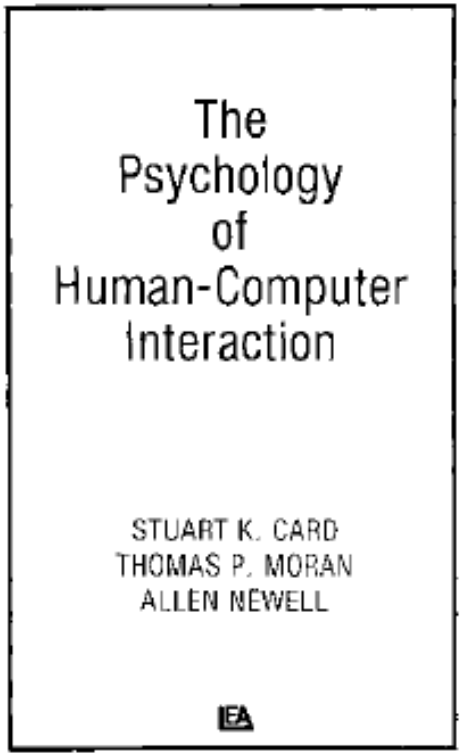
The KLM: HCI's First Predictive Model

The Keystroke-Level Model for User Performance Time with Interactive Systems

Stuart K. Card and Thomas P. Moran
Xerox Palo Alto Research Center

Allen Newell
Carnegie-Mellon University

Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*, 396-410.



The Psychology of Human-Computer Interaction

STUART K. CARD
THOMAS P. MORAN
ALLEN NEWELL

LEA

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Outline

- Problem statement
- Model
- Empirical validation
- Applications

Problem Statement

- Goal
  - Develop simple model to describe time to do task with a given method on an interactive system
- $T_{task} = T_{acquire} + T_{execute}$
  where
  T_{task} = total time to complete task
  $T_{acquire}$ = time to select method to complete task
  $T_{execute}$ = time to perform method
- Model predicts $T_{execute}$
- Assume expert users and no errors

Model

- $T_{execute} = \sum$(time to execute primitive op)
- Primitive operations
  - **K** key/button press
  - **P** point to target with mouse
  - **H** home hands to keyboard or mouse
  - **D** draw line with mouse
  - **M** mental preparation (pause)
  - **R** system response time

Model (cont.)

- Times for primitive operations are predicated from experiments
  - Time to press key/button ranges between
    - 0.08 sec/char fast typist
    - 0.28 sec/char average typist
    - 1.2 sec/char slow typist

Original parameters

Operator	Description	Time (s)
K	PRESS A KEY OR BUTTON Pressing a modifier key (e.g., shift) counts as a separate operation, Time varies with typing skill:	
	Best typist (135 wpm)	.08
	Good typist (90 wpm)	.12
	Average skilled typist (55 wpm)	.20
	Average non-secretary typist (40 wpm)	.28
	Typing random letters	.50
	Typing complex codes	.75
	Worst typist (unfamiliar with keyboard)	1.20
P	POINT WITH A MOUSE Empirical value based on Fitts' law. Range from .8 to 1.5 seconds. Operator does *not* include the button click at the end of a pointing operation	1.10
H	HOME HAND(S) ON KEYBOARD OR OTHER DEVICE	.40
D(n_D.l_D)	DRAW n_D STRAIGHT-LINE SEGMENTS OF TOTAL LENGTH l_D. Drawing with the mouse constrained to a grid.	.9 n_D + .16 l_D
M	MENTALLY PREPARE	1.35
R(t)	RESPONSE BY SYSTEM Different commands require different response times. Counted only if the user must wait.	t

Encoding Method

- Code method as sequence of primitive operations, sum up times
- Example: replace 5 letter word with another
- Editor 1: keyboard based
  - next line — **MK**[(line feed)]
  - substitute command — **MK**[s]
  - enter new word — **5K**[xxxxx]
  - terminate arg — **MK**[(return)]
  - enter old word — **5K**[zzzzz]
  - terminate arg — **MK**[(return)]
  - terminate command — **K**[(return)]
- $T_{execute} = 4t_M + 15t_K = 8.4$ sec

Placement of M's

rule 0: place **M** before all **K**'s not part of arg strings and before all **P**'s that select commands

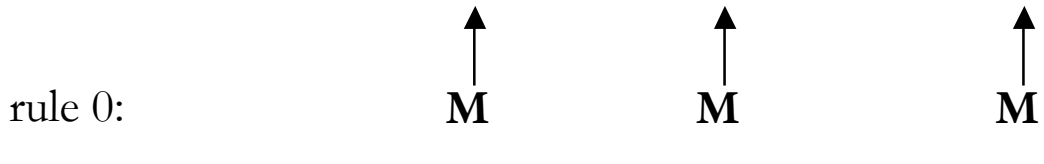
for each **M** do

- rule 1: if operator fully anticipated, delete M
  (e.g. **PMK → PK**)
- rule 2: if string of **MK**'s belongs to cognitive unit, delete all **M**'s but first
- rule 3: if **K** is redundant terminator, delete **M**
- rule 4: if **K** terminates constant string (*i.e.* command), delete it
  if **K** terminates variable string (*i.e.* arg), keep it

Placement of M's

Example: Mouse-based Editor 2 sequence

H[m]**P**[word]**K**[y]**H**[k]**K**[R]**5K**[…]**K**[(esc)]

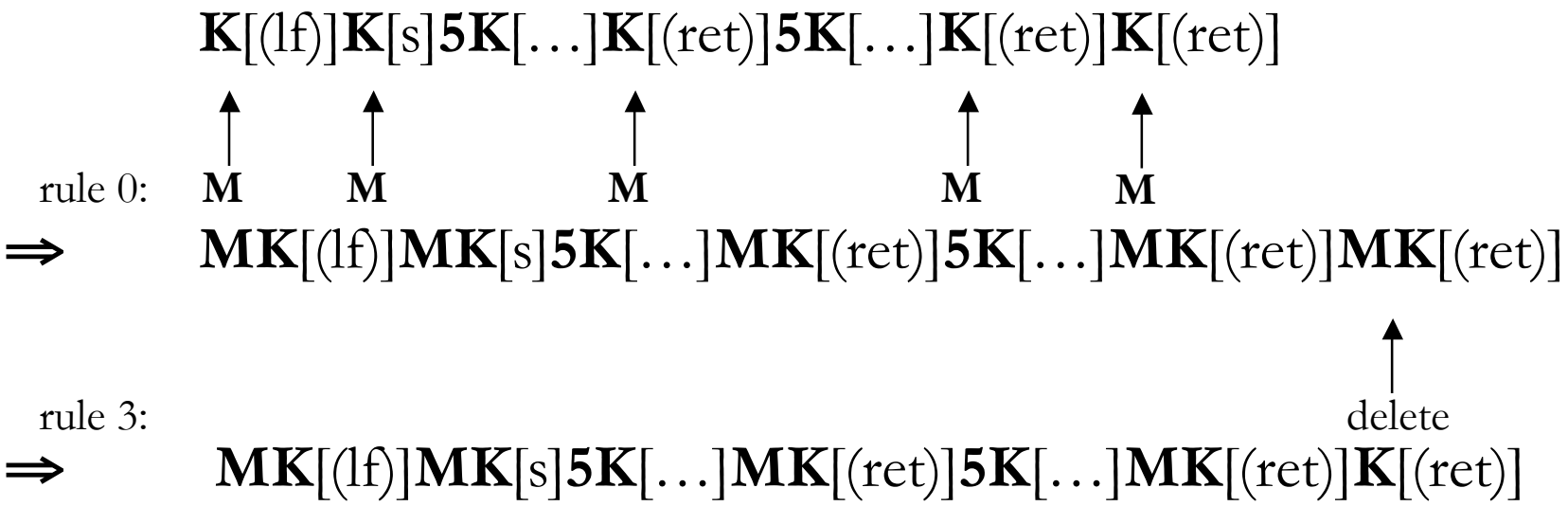


⇒ **H**[m]**P**[word]**MK**[y]**H**[k]**MK**[R]**5K**[…]**MK**[(esc)]

⇒ **H**[m]**P**[word]**K**[y]**H**[k]**MK**[R]**5K**[…]**MK**[(esc)]

Placement of M's

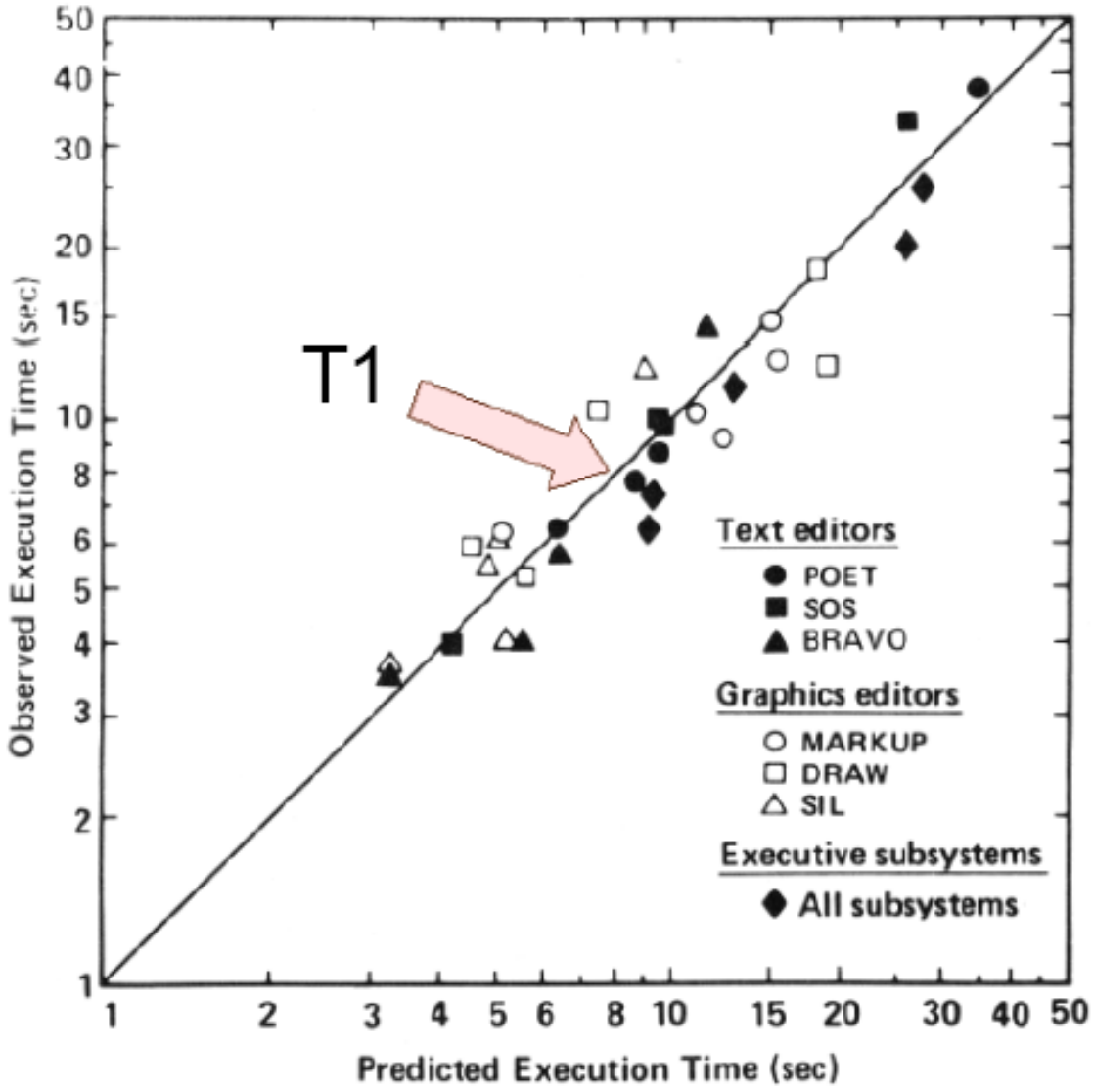
Example: Keyboard-based Editor 1 sequence



Empirical Validation

- Conducted experiment on real systems to compare predictions to observations
- 14 tasks on 3 types of systems
  - text editors
  - graphics systems
  - command executers
- Users
  - practice using system
  - were tested to determine typing and drawing speed

Observation vs. prediction



Some Recent Applications of the KLM

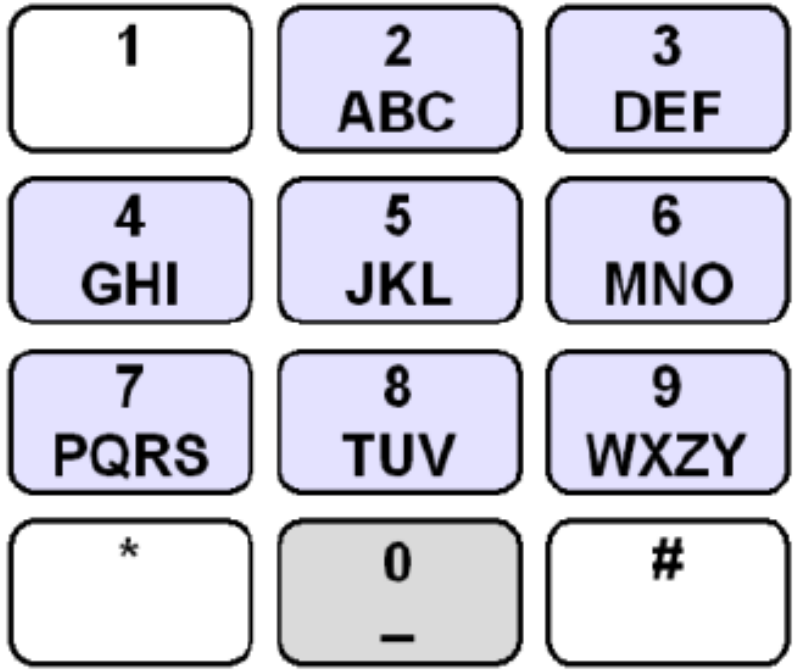
- Balter, O. (2000). Keystroke level analysis of email message organization. *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2000* 105-112. New York: ACM
- Dunlop, M. & Crossan, A. (2000). Predictive text entry methods for mobile phones. *Personal and Ubiquitous Computing* 4, 134-143.
- Holleis, P., Otto, F., Hussmann, H., & Schmidt, A. (2007). Keystroke-level model for advanced mobile phone interaction. *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2007*, 1505-1514. New York: ACM.
- Pettitt, M., Brunett, G. & Stevens, A. (2007). An extended keystroke level model (KLM) for predicting visual demand on in-vehicle information systems. *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2007*, 1515-1524. New York: ACM.

The KLM and Text Entry

- The KLM and Text Entry
  - The original experiment did not include a task such as:
    - *Enter a 43-character phrase of text*
- Why?
  - It's too simple for the KLM
  - Prediction reduces to 43 x *tK*
  - *tK determined for each user in a pre-test*
  - The KLM task just confirms the pre-test

T9 Entry of “beep”

2	a
23	be
233	bed
2337	beer
2337n	beds
2337nn	adds
2337nnn	bees
2337nnnn	beep
2337nnnn0	beep_



Model for T9 Entry of “beep”

MP – the time for performing a physical match between a stimulus (the presented word) and a code stored in the user's short-term memory (the desired word).

Summary

- $T_{task} = T_{acquire} + T_{execute}$
- $T_{execute}$ = Key + Point + Home + Draw + Mental + System
- Very low level
- Simplistic
- Famous reference
- Still useful 25 years later