Modeling

Chaklam Silpasuwanchai

Asian Institute of Technology

chaklam@ait.asia

January 2, 2020

Overview

Model

- Model is a **simplification of reality**, e.g., physicist's equation for the trajectory of a tossed ball
- Model is useful because they allow us to explore the phenomena, think about them, make changes, and so on, **without actually constructing the building or throwing the ball**.
- Model is either descriptive or predictive
  - **Descriptive**: models using verbal analogy and metaphor to describe phenomena
  - **Predictive**: models using closed-form mathematical equations to predict phenomena

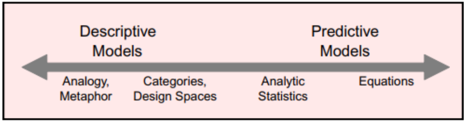


Figure: Source: Figure 7.42 (Mackenzie)

Key-action model

- With KAM, keyboard keys are categorized as either symbol keys, executive keys, or modifier keys.
- **Symbol** keys deliver graphic symbols to an application such as a text editor. These are typically letters, numbers, or punctuation symbols.
- **Executive** keys invoke actions in the application or at the system level. Examples include enter, F1, and ESC
- **Modifier** keys do not generate symbols or invoke actions. Instead, they set up a condition that modifies the effect of a subsequently pressed key. Examples include shift and alt

Key-action model

- As you will see, the usage of descriptive model is simply allow us to delineates a problem space, and to identify any possible problem
- For example, consider the usage of left and right hands. On the left, there are TAP, CAPS, ESC, and WINDOWS that are not mirrored. On the right, there are at least 18 keys that are not mirrored, e.g., BACKSPACE, INSERT, DELETE, etc.
- With a 4-18 left-right ratio of executive keys, the current desktop is clearly right-side bias!
- Given that the right hand is already occupied with the mouse, isn't the right hand too overloaded?
- In fact, research shows that common GUI tasks favor left-hand mouse usage! (Mackenzie, 2003)

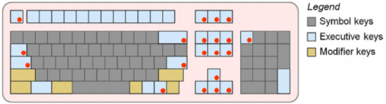


Figure: Source: Figure 7.3 (Mackenzie)

Key-ambiguity model

- The key-action model only captures one aspect of keyboards, namely the actions associated with each key.
- Another way to think about keyboards is in the **ambiguity** of key presses. If pressing a symbol key always produces the same symbol, then the situation is simple. But if a key can produce two or more symbols, then this is worth thinking about.
- For this, a "key-ambiguity" descriptive model might be useful (MacKenzie and Soukoreff, 2002)
- Optimal ambiguous keyboard would depend on the arrangement which is also dependent on ngrams

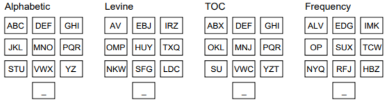


Figure: Source: Lesher et al. 1998

Model of bi-manual control

- Describes how we use our two hands in interaction
- Guiard proposes a simple model, describing the logic of division of labor between the two hands
- For example, a right-handed graphic artist is sketching, the left hand is used to manipulate the frame of workspace using coarse movement. At the same time, the right hand follows the frame of workspace and sketching takes place using fine movements
- Unsurprisingly, this model was widely adopted for HCI research in two-handed interaction

Hand	Role and Action
Non-preferred	• Leads the preferred hand • Sets the spatial frame of reference for the preferred hand • Performs coarse movements
Preferred	• Follows the non-preferred hand • Works within established frame of reference set by the non-preferred hand • Performs fine movements

Figure: Source: Figure 7.4 (Mackenzie)

Model of bi-manual control

- Back then, **scrolling** is a time-consuming process where users have to acquire the scrollbar on the right, which is a target acquisition task that takes up **two seconds** per trial
- Microsoft's **IntelliMouse** introduces new affordance in 1996, introducing a scrolling wheel
- Guiard model informs us that indeed, it was a bad idea as left hand would be a much better idea. Why? Imagine scrolling while selecting - it's impossible with this design

Figure: Source: Figure 7.7 (Mackenzie)

Model of bi-manual control

Two-handed interaction research is ongoing

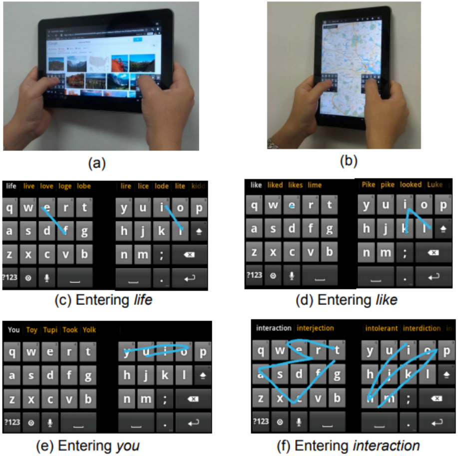


Figure: Bi et al., **Bimanual Gesture Keyboard**, CHI 2012

Model of bi-manual control

Two-handed interaction research is ongoing

BiTouch and BiPad:
Designing Bimanual Interaction for Hand-held Tablets

Julie Wagner[1,2,3]
wagner@lri.fr

Stéphane Huot[2,1,3]
huot@lri.fr

Wendy E. Mackay[1,2,3]
mackay@lri.fr

[1] INRIA
F-91405 Orsay, France

[2] Univ Paris-Sud (LRI)
F-91405 Orsay, France

[3] CNRS (LRI)
F-91405 Orsay, France

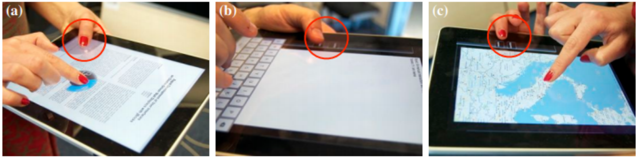


Figure: Wagner et al., **BiTouch and BiPad: Designing Bimanual Interaction for Hand-held Tablets**, CHI 2012

Three-state model of graphical input

- Another descriptive model is **Buxton's three-state model of graphical input**
- The model is a simple characterization of the operation of computer pointing devices in terms of **state transitions.**
- Such simple model may seem obvious but its simplicity affords extension to multi-button interaction, stylus, or finger input.

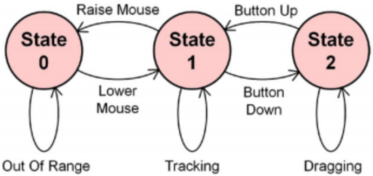


Figure: Source: Figure 7.9 (Mackenzie)

Three-state model of graphical input

- Apple in 1994 commercialized a new pointing device in its PowerBook 500 notebook computer: the **Trackpad touchpad**
- One of the interaction techniques supported by touchpads is "**lift-and-tap**" where primitive operations like clicking, double-clicking, and dragging are implemented without a button
- Two observations follow: (1) lift-and-tap necessitates **extra state transitions** over mouse, and (2) the use of state 1-0-1 transitions for lift-and-tap is confounded with clutching (lifting the mouse to reposition) which uses the same state transitions, thus possibly System enters the Tracking state instead of the Dragging state
- To fix this, additional sensing capability was used to implement state 1-2 transitions by **pressing harder** or use **gestures**

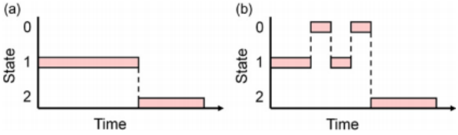


Figure: Source: Figure 7.10 (Mackenzie)

Predictive models

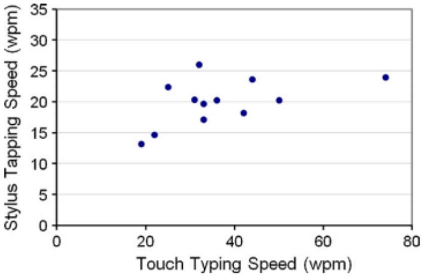
- A predictive model is an equation, where predictors (x) predicts the outcome (y).
- Since it is an equation, both x and y use ratio-scale data. For example, y cannot be nominal (categorical) as in the case of experimental design
- Models in HCI has primary two purpose: (1) provides a mathematical explanation of the relationships between predictors and targets, (2) serve as a function for analyses of design alternatives
- Simplest model in HCI comes in the form of a linear regression model

Linear regression model

(a)

Participant	Stylus Tapping Speed (wpm)	Touch Typing Speed (wpm)
P1	18.2	42
P2	23.6	44
P3	26.0	32
P4	20.3	50
P5	20.3	36
P6	17.1	33
P7	24.0	74
P8	14.7	22
P9	20.3	31
P10	19.7	33
P11	22.4	25
P12	13.1	19

(b)



Linear regression model

- R^2 is a common metric for linear regression model which measures the amount of variation that is explained by the model
- Given 60wpm, the model predict stylus speed of 23.1wpm
- SE = 3.39wpm. Values within -1.96 SE and +1.96 SE of a prediction are within a 95% confidence interval. So for the model developed here, there is 95% confidence that a user whose touch-typing speed is 60wpm will have a stylus-tapping speed between 23.1 - (1.96 * 3.39) = 16.4wpm and 23.1 + (1.96 * 3.39) = 29.7wpm

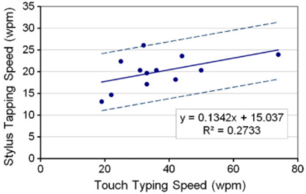


Figure: Source: Figure 7.13 (Mackenzie)

More than one predictor

- Prediction model can include **multiple predictors** and come in the form of multiple regression as:
- **Stepwise linear regression** is commonly used

$$y = a + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots$$

Participant	Time To Reach Criterion (hours)	Age (years)	Daily Computer Usage (hours)
P1	2.3	16	8
P2	2.1	17	7
P3	2.5	18	8
P4	3.6	23	6
P5	2.6	25	7
P6	5.3	26	5
P7	4.8	29	6
P8	6.1	31	4
P9	7.2	32	5
P10	7.3	35	3
P11	6.4	37	4
P12	8.1	38	2
P13	7.9	40	4
P14	2.3	16	8

Figure: Source: Figure 7.39 (Mackenzie)

Fitts' Law

One of the most widely used models in HCI is Fitts' law (1954). Three primary usage are

- To see if the interaction technique follows Fitts' law
- To analyze design alternatives
- To use Fitts' index of performance (now throughput) as a dependent variable in a comparative evaluation

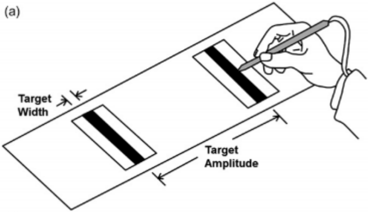


Figure: Source: Figure 7.14 (Mackenzie)

Fitts' Law

- The robustness of Fitts' law is extraordinary. In the decades since Fitts' seminal work, the relationship has been verified countless times and in remarkably diverse settings.
- Fitts proposed a variable quantifying movement task's difficulty - *ID*, *the index of difficulty*.

$$ID = log_2\left(\frac{A}{W} + 1\right)$$

- Fitts' index of performance, now called throughput (TP, in bits/s), is calculated by dividing ID by mean movement time, computed over a block of trials (W_e is computed from the standard deviation in the selection coordinates which reflect a 4% error rate.)

$$TP = \left(\frac{ID_e}{MT}\right) \text{ where } W_e = 4.133 * SD_x;$$

- To use Fitts law to predict MT, it is a linear function of ID where a and b are obtained from experiments

$$MT = a + b * ID$$

- Note that **building** model uses W_e while **using the model to predict** uses simply W

Example

A (pixels)	*W* (pixels)	*ID* (bits)	Mouse				RemotePoint			
			W_e (pixels)	ID_e (bits)	*MT* (ms)	*TP* (bits/s)	W_e (pixels)	ID_e (bits)	*MT* (ms)	*TP* (bits/s)
40	10	2.32	11.23	2.19	665	3.29	13.59	1.98	1587	1.25
40	20	1.58	19.46	1.61	501	3.21	21.66	1.51	1293	1.17
40	40	1.00	40.20	1.00	361	2.76	37.92	1.04	1001	1.04
80	10	3.17	10.28	3.13	762	4.11	10.08	3.16	1874	1.69
80	20	2.32	18.72	2.40	604	3.97	25.21	2.06	1442	1.43
80	40	1.58	35.67	1.70	481	3.53	37.75	1.64	1175	1.40
160	10	4.09	10.71	3.99	979	4.08	10.33	4.04	2353	1.72
160	20	3.17	21.04	3.11	823	3.77	19.09	3.23	1788	1.81
160	40	2.32	41.96	2.27	615	3.69	35.97	2.45	1480	1.65
		Mean	23.25	2.38	644	3.60	23.51	2.35	1555	1.46

Figure: Figure 7.16 (Mackenzie)

Example

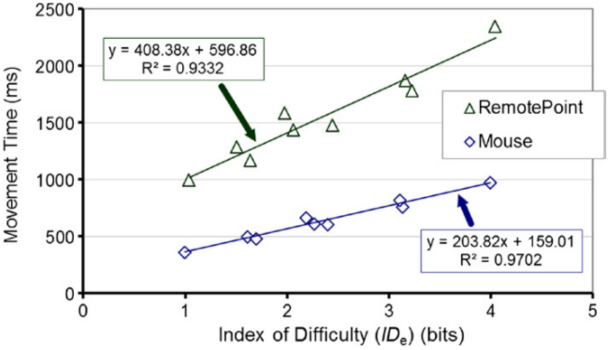


Figure: Figure 7.17 (Mackenzie)

Fitts' Law

- By using Fitts' law as objective/cost function, we can find optimal design alternatives - an area called **design optimization**

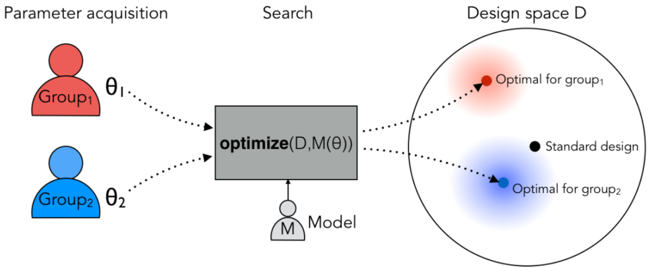


Figure: Source: Sarcar et al., **Ability-Based Optimization of Touchscreen Interactions**, IEEE Pervasive

Activities

Classwork

-Download GoFitts.jar from
`http://www.yorku.ca/mack/FittsLawSoftware`
-Perform a 2D fitts tasks with default settings (A = 100, 200, 400; W = 20, 40, 80).
-Set the Number of Trials to 5
-Construct a table similar to below (Note that Mackenzie recommend using ID_e as it accounts for both speed and accuracy)
-Create a chart showing the a). scatter plot, b). regression line, c). R-squared, d). prediction equation
-Interpret the R^2 value.

A (pixels)	*W* (pixels)	*ID* (bits)	*MT* (ms)
192	16	3.70	654
192	32	2.81	518
192	64	2.00	399
320	16	4.39	765
320	32	3.46	613
320	64	2.58	481
512	16	5.04	872
512	32	4.09	711
512	64	3.17	567

Hick-Hyman Law

- Given n stimuli, associated one-for-one with n responses, the time to react (RT) to the onset of a stimulus is given by, where a and b are empirically determined constants. Typical values for a is 200ms and b is 150ms/bit

$$RT = a + b * log_2(n+1)$$

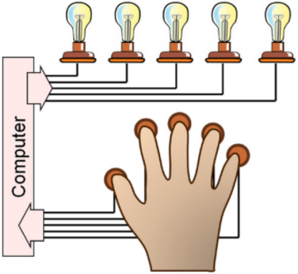


Figure: Source: Figure 7.18 (Mackenzie)

Hick-Hyman Law

- If **some choice is more probable than others**, this reduces the information content, thus in turn reduces the choice reaction time
- For a set of alternatives with different probabilities, the information H is

$$H = \sum p_i log_2 \left(\frac{1}{p_i} + 1 \right)$$

- Consider a choice selection task where the choice is among 26 alternatives and all appear with equal probability, the information content of the task is simply

$$H = log_2 (27) = 4.75 bits$$

Hick-Hyman Law

Letter	Frequency	Probability (p)	$p\log_2(1/p+1)$
a	24373121	0.0810	0.3028
b	4762938	0.0158	0.0950
c	8982417	0.0299	0.1525
d	10805580	0.0359	0.1742
e	37907119	0.1260	0.3981
f	7486889	0.0249	0.1335
g	5143059	0.0171	0.1008
h	18058207	0.0600	0.2486
i	21820970	0.0725	0.2819
j	474021	0.0016	0.0147
k	1720909	0.0057	0.0427
l	11730498	0.0390	0.1846
m	7391366	0.0246	0.1322
n	21402466	0.0711	0.2783
o	23215532	0.0772	0.2935
p	5719422	0.0190	0.1092
q	297237	0.0010	0.0099
r	17897352	0.0595	0.2471
s	19059775	0.0633	0.2578
t	28691274	0.0954	0.3358
u	8022379	0.0267	0.1404
v	2835696	0.0094	0.0636
w	6505294	0.0216	0.1203
x	562732	0.0019	0.0170
y	5910495	0.0196	0.1119
z	93172	0.0003	0.0036
		$H=$	4.25

Figure: Source: Figure 7.19 (Mackenzie)

Hick-Hyman Law

- Card et al. (1983, 74) describe an example of a **telephone operator** selecting among ten buttons
- Landauer and Nachbar (1985) applied the Hick-Hyman law in measuring and predicting the time to select items in **hierarchical menus** - breadth should be favored over depth in hierarchical menus (*Note that choice reaction is different from visual search, which is a linear function of the number of items. Landauer and Nachbar eliminated visual scanning in the task by ensuring that "the sets of choice alternatives were well practiced and well-ordered"*)
- Ruiz et al. (2008) used the Hick-Hyman law to model the perception, planning, and activation time for users to **switch modes with their non-dominant hands** in a tablet interface.

Activities

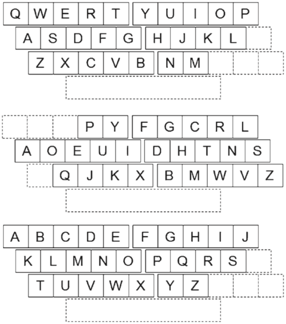
Classwork

A human operator attends to eight stimulus lights and presses one of eight keys when the corresponding light turns on. Two of the lights turn on more frequently than the others, accounting for 40 percent and 30 percent of all activations, respectively. The other lights activate with the same frequency. What is the information content of the task?

Activities

Classwork

Below are the layouts for the Qwerty and Dvorak keyboards, as well as an alphabetic layout proposed by Card et al. (1983, 63). Assuming the layouts are implemented as standard physical keyboards, which design provides the most even split between lefthand and righthand keying?



Keystroke level model

- Card et al. (1980; 1983, ch. 8) developed KLM that predict **error-free task completion time** using
  - Task
  - Method used
  - Command language of the system
  - Motor skill parameters of the user
  - Response time parameters of the system
- Requires a task to be broken down into a series of subtasks, or primitive operations. The models works with four motor-control operators (K = keystroking, P = pointing, H = homing, D = drawing), one mental operator (M), and one system response operator (R):)

$$t_{EXECUTE} = t_k + t_p + t_H + t_D + t_M + t_R$$

Keystroke level model

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) Good typist (90 wpm) Average skilled typist (55 wpm) Average non-secretary typist (40 wpm) Typing random letters Typing complex codes Worst typist (unfamiliar with keyboard)	 0.08 0.12 0.20 0.28 0.50 0.75 1.20
P	POINT WITH A MOUSE Empirical value based on Fitts' law. Range from 0.8 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	0.40
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

Figure: Source: Figure 7.20 (Mackenzie)

Keystroke level model

- To validate the KLM, Card, Moran, and Newell conducted an experiment using 14 tasks performed using various methods
- Task T1, for example, was "Replace one 5-letter word with another (one line from previous task)."
- On one system, POET, the required sequence of subtasks was:

Jump to next line	**M K**[LINEFEED]
Issue Substitute command	**M K**[S]
Type new word	5**K**[word]
Terminate new word	**M K**[RETURN]
Type old word	5**K**[word]
Terminate old word	**M K**[RETURN]
Terminate command	**K**[RETURN]

- The task required four mental operations (M) and 15 keystroking operations (K):

$$t_{EXECUTE} = 4 * t_M + 15 * t_k -> t_{EXECUTE} = 4 * 1.35 + 15 * 0.23 = 8.85s$$

Keystroke level model

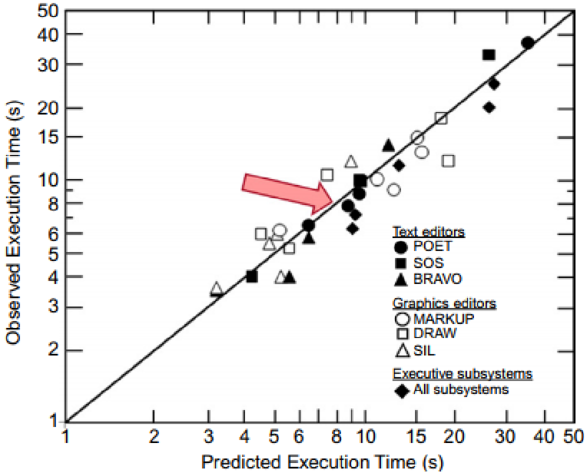


Figure: Source: Figure 7.21 (Mackenzie)

Keystroke level model

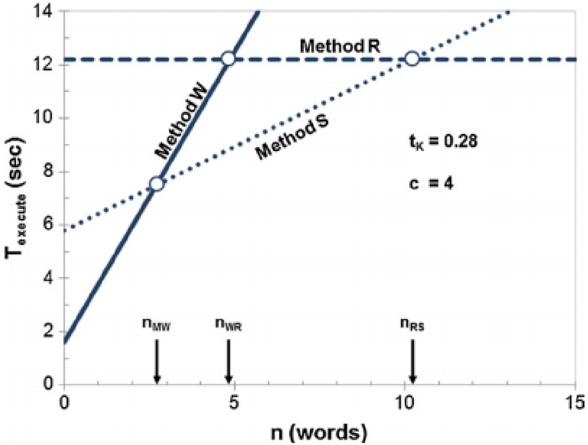


Figure: Source: Figure 7.22 (Mackenzie)

Keystroke level model

- Using the mouse model, we can also derive t_p
$$t_p = 0.159 + 0.204 * log_2\left(\frac{A}{W} + 1\right)$$
- If we are clicking a 1.2cm wide toolbar button, and is 3.2cm away, the the pointing time is
$$t_p = 0.159 + 0.204 * log_2\left(\frac{3.2}{1.2} + 1\right) = 0.45s$$

Keystroke level model

- Consider the editing operations to change the font style and font family for text, e.g., the word "M K"
- For mouse, four pointing operations are required: select the text, select the Bold, select the drop-down arrow in the Font list, and select Arial
- Table belows show KLM operators, where P is written in the format P[A, W]. The total time is

$$t_{EXECUTE} = 4 * t_M + \sum t_p = 4 * 1.35 + 2.71 = 8.11s$$

Mouse Subtasks	KLM Operators	t_P (s)
Drag across text to select "M K"	**M P**[2.5, 0.5]	0.686
Move pointer to Bold button and click	**M P**[13, 1]	0.936
Move pointer to Font drop-down button and click	**M P**[3.3, 1]	0.588
Move pointer down list to Arial and click	**M P**[2.2, 1]	0.501
	$\sum t_P$ =	2.71

Figure: Source: Figure 7.24 (Mackenzie)

Keystroke level model

- The total time when using a keyboard is

$$t_{EXECUTE} = 4 * t_M + 12 * t_k = 4x1.35 + 12x0.75 = 14.40s$$

Keyboard Subtasks	KLM Operators
Select text	**M P**[shift] 3**K**[→]
Convert to boldface	**M K**[ctrl] **K**[b]
Activate Format menu and enter Font sub-menu	**M K**[alt] **K**[o] **K**[f]
Type *a* ("Arial" appears at top of list)	**M K**[a]
Select "Arial"	**K**[↓] **K**(Enter)

Figure: Source: Figure 7.25 (Mackenzie)

Keystroke level model

- $t_{EXECUTE}$ in response to t_k

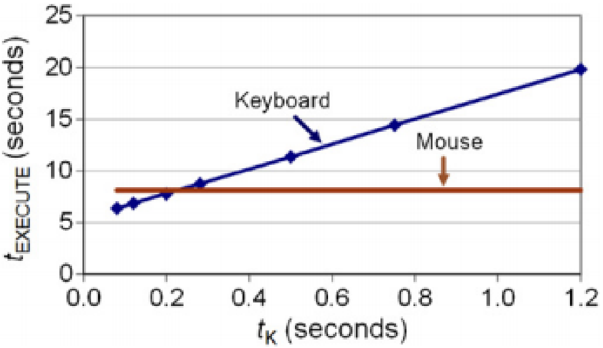


Figure: Source: Figure 7.26 (Mackenzie)

Keystroke level model

KLM has been since extended for modern HCI research

- Ruiz et al. (2008) defined a new operator t_{INT}, the interval between a mode switch with the non-dominant hand and the beginning of the next action
- Pettitt et al. (2007) define $t_R F$ for "reach far", the time to reach from a car's steering wheel to an IVIS (in-vehicle information system).
- Whenever a extension of the model is proposed, an experiment is required to validate the extension
- The key challenge lies in the t_k which includes reacting, planning, pausing, and attending, and also lies in individual differences (novice vs. experts) which is the main limitation of the KLM model
- To tackle this limitation, HCI modeling is limited to primitive actions such as taps, clicks, gaze shifts, etc., instead of the whole execution time

Keystroke level model

- **Physical matching**: matching word
- **Name matching**: same as physical matching but vary in appearance, eg., uppercase/lowercase, font-size/family
- **Class matching**: matching certain properties such as font-size/family

Proposed Mnemonic	Task	Execution Time (ms)	
		Card et al.	Figure 2-28 & Figure 2-30
M_S	Simple Reaction	240 [105 – 470]	277 [±44]
M_P	Physical Matching	310 [130 – 640]	510 [±59]
M_N	Name Matching	380 [155 – 810]	485 [±52]
M_L	Class Matching	450 [180 – 980]	566 [±96]
M_C	Choice Reaction	$200 + 150 \log_2(N + 1)$	
M_V	Visual Search		$498 + 41\ N$

Figure: Source: Figure 7.33 (Mackenzie)

Mobile text entry

- KLM is a classic HCI example for **mobile text entry**
- A relevant statistic is KSPC (**keystrokes per character**). Typically, mobile phone has KSPC ≈ 2.023 for multi-tap and KSPC ≈ 1.007 for predictive text entry (T9)
- If word completion or word prediction is added, KSPC < 1. However, these figures do not account for the performance costs of **visually attending** to the interface.
- One important consideration in mobile text entry is the way users interaction with a phone either with a **thumb** or **index finger**
- For touch typing on a Qwerty keyboard or multi-tapping on a mobile phone, **the KLM's mental operator (M) is not needed**. The user knows what to do and does it according to his or her keystroking expertise with the method.
- This is not the case for predictive text entry as it requires mental operations since uncertainty is present

Mobile text entry

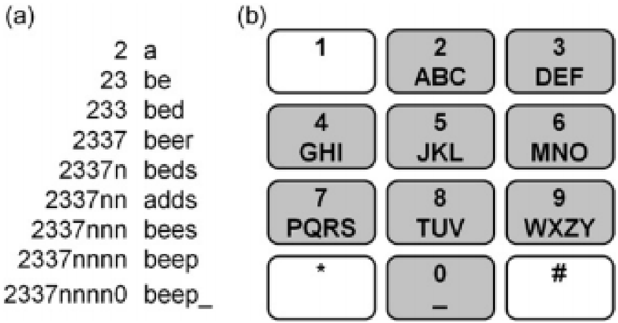


Figure: Source: Figure 7.28 (Mackenzie)

Figure: Source: Figure 7.29 (Mackenzie)

Predictive text entry

Let's say "Vegetables". Two possibilities of models are

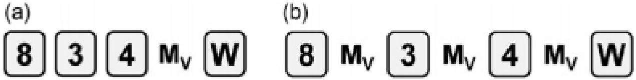


Figure: Source: Figure 7.32 (Mackenzie)

Skill acquisition

- The relationship between **skill and practice is non-linear**. In the beginning, a small amount of practice yields substantial improvement. After a lot of practice, the same "small amount" produces only a slight improvement
- This relationship can be written as below, where x is amount of practice and y is performance.

$$y = b * x^a$$

- If the dependent variable is the time to do a task, T, the equation can be rewritten as below, where n is number of trial, and T_1 is the time on the first trial (*Note that* **a is negative** *because completion time decreases over practice*)

$$T_n = T_1 * n^a$$

- Let's consider speed which takes a similar form, but **a is positive**, because speed increases over more practice)

$$S_n = S_1 * n^a$$

Skill acquisition

- Let's consider a comparison between two soft keyboard layouts: QWERTY and Opti
- Since users are familiar with QWERTY, it is essential to perform a longitudinal study to understand the potential of Opti
- The experiment involved 20 sessions of text entry

(a)

Session	Qwerty	Opti
1	28.44	16.98
2	29.98	21.06
3	31.56	24.80
4	32.62	27.26
5	33.20	29.18
6	34.52	31.04
7	34.14	33.08
8	34.68	34.16
9	35.66	34.72
10	36.77	35.89
11	36.95	37.22
12	37.36	38.75
13	38.50	39.95
14	38.65	40.73
15	39.12	40.68
16	39.00	41.85
17	39.20	42.74
18	39.85	43.55
19	40.16	43.91
20	40.30	44.29

(b)

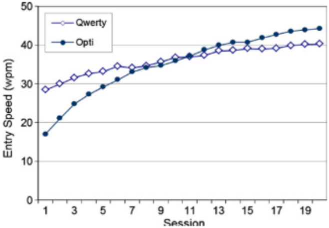


Figure: Source: Figure 7.36 (Mackenzie)

Skill acquisition

For example, the speed after 50 trials are
$$S_{50} = 17.24 * 50^{0.3219} = 60.7wpm$$

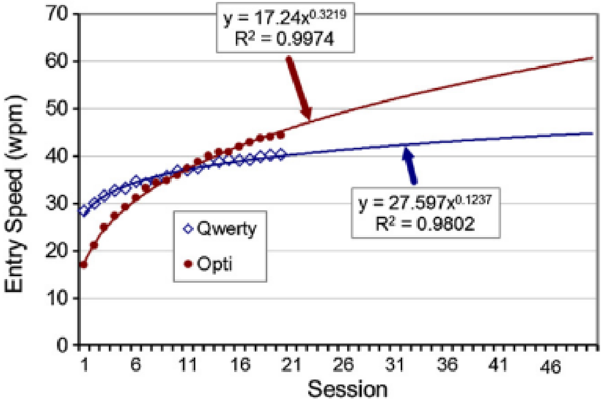


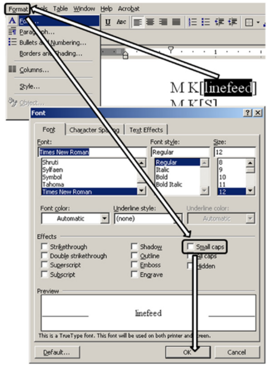
Figure: Source: Figure 7.37 (Mackenzie)

Activities

Classwork

Perform the classwork on KLM This task requires you to predict the time of mouse vs. keyboard using KLM.

For this question, assume $t_K = 0.4$ seconds. For both input methods, provide a KLM breakdown of all operations.



The End