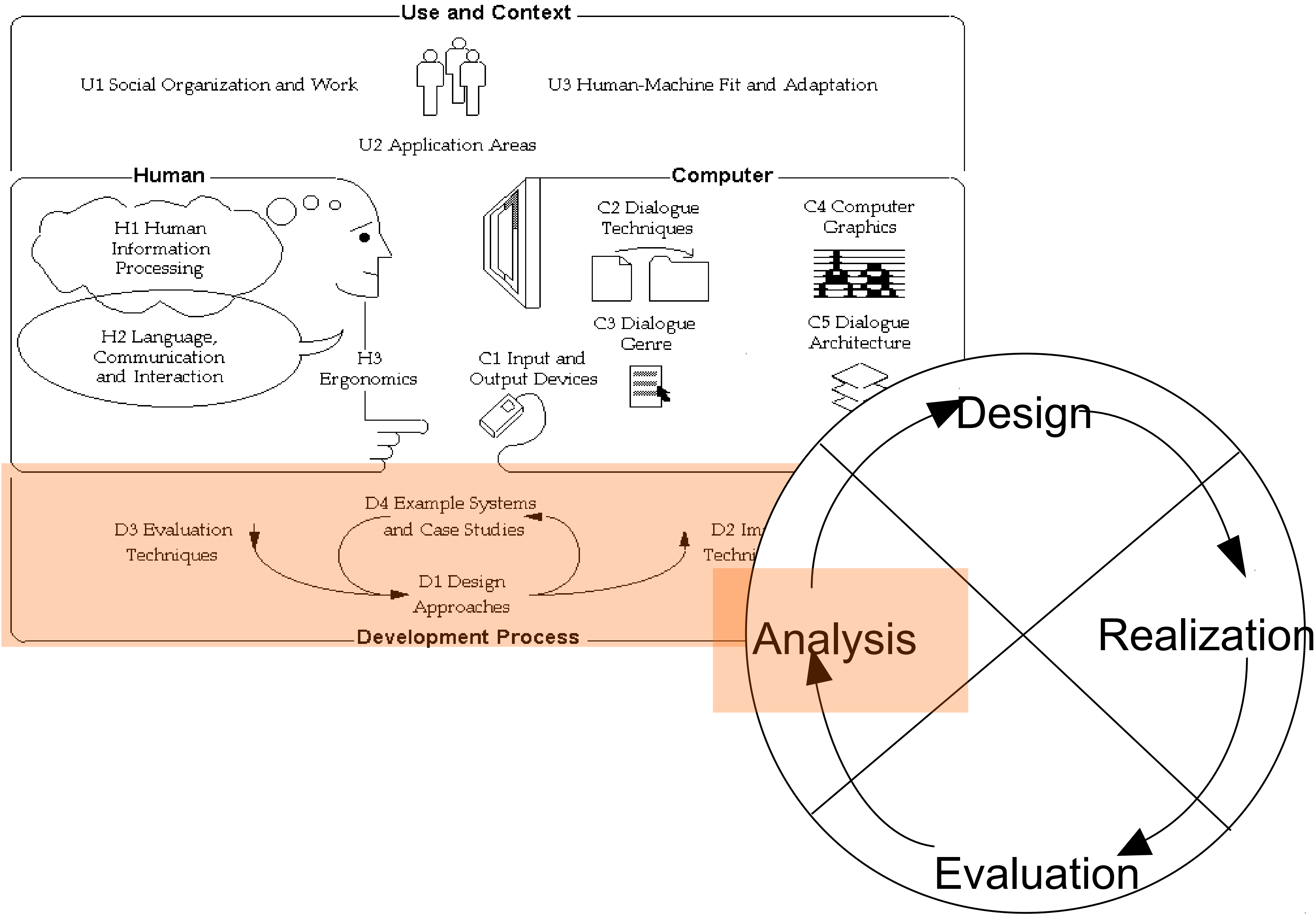
Mensch-Maschine-Interaktion 1

Chapter 9 (July 14th, 2011, 9am-12pm):

Basic HCI Models

Overview

- Introduction
- Basic HCI Principles (1)
- Basic HCI Principles (2)
- User Research & Requirements
- Designing Interactive Systems
- Capabilities of Humans and Machines
- User Study Design & Statistics
- Implementing Interactive Systems
- ***Basic HCI Models***
- User-Centered Development Process


Use and Context
U1 Social Organization and Work
U3 Human-Machine Fit and Adaptation
U2 Application Areas
Human
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
Computer
C2 Dialogue Techniques
C4 Computer Graphics
C3 Dialogue Genre
C5 Dialogue Architecture
C1 Input and Output Devices
D4 Example Systems and Case Studies
D3 Evaluation Techniques
D1 Design Approaches
D2 Im
Techni
Development Process
Design
Analysis
Realization
Evaluation

Basic HCI Models

- Predictive Models for Interaction: Fitts' / Steering Law
- Descriptive Models for Interaction: GOMS / KLM

Fitts' Law – Introduction

- Robust model of human psychomotor behavior
- Predicts movement time for rapid, aimed pointing tasks
  - Clicking on buttons, touching icons, etc.
  - Not suitable for drawing or writing
- Developed by Paul Fitts in 1954
- Describes movement time in terms of distance+size of target and device
- Rediscovered for HCI in 1978
- Subsequently heavily used and discussed

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*, 381-391.

Card, Stuart K., English, William K., Burr, Betty J. (1978). *Ergonomics, 21(8):601–613* Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT.

Fitts' Law – History

- **Paul M. Fitts** was an American psychologist and one of the pioneers in improving aviation safety. He went on to lead the Psychology Branch of Air Force Research Laboratory – later renamed, in his honor, to Fitts Human Engineering Division.
- Fitts' Law was his most famous work. It was first mentioned in a publication in 1954, and first applied to Human-Computer Interaction in 1978.
- Fitts' discovery "*was a major factor leading to the mouse's commercial introduction by Xerox*“ [Stuart Card]
- Initially derived from a theorem for analogue information transmission

http://fww.few.vu.nl/hci/interactive/fitts/

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*, 381-391.

Derivation from Signal Transmission

$$C = B \log_2 \left(1 + \frac{S}{N}\right)$$

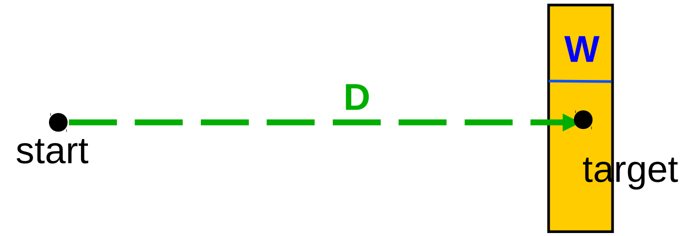
- Shannon-Hartley Theorem
- C is the channel capacity (bits / second)
- B is the bandwidth of the channel (Hertz)
- S is the total signal power over the bandwidth (Volt)
- N is the total noise power over the bandwidth (Volt)
- S/N is the signal-to-noise ratio (SNR) of the communication signal to the Gaussian noise interference (as linear power ratio – $SNR(dB)=10\log_{10}(S/N)$)

C. E. Shannon (1949). Communication in the presence of noise.
Proc. Institute of Radio Engineers vol. 37 (1): 10–21.

Fitts' Law – Formula

- The time to acquire a target is a function of the **distance** to and **size** of the target and depends on the particular pointing **system**

$$MT = a + b \log_2\left(1 + \frac{D}{W}\right)$$

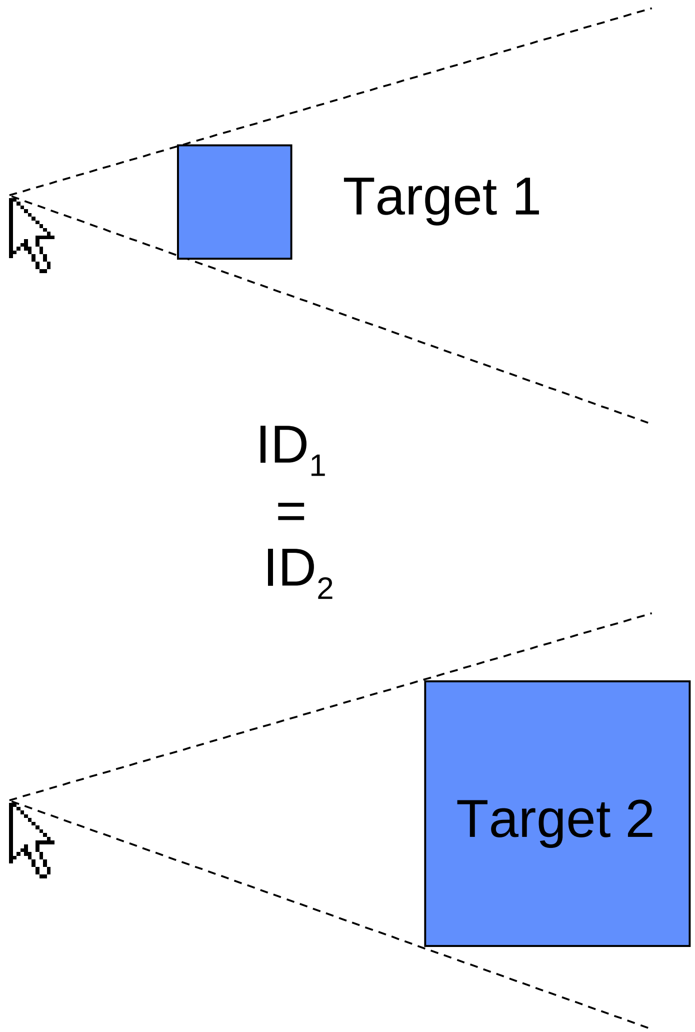


- **MT**: movement time
- **a and b**: constants dependent on the pointing system
- **D**: distance to the target area
- **W**: width of the target

Fitts' Law – Index of Difficulty

$$MT = a + b \log_2\left(1 + \frac{D}{W}\right)$$

- Index of Difficulty, ID = $\log_2\left(1 + \frac{D}{W}\right)$
  - MT = a + b ·ID
  - ID describes the difficulty of the task independent of the device / method
- Units
  - Constant a measured in seconds
  - Constant b measured in seconds / bit
  - Index of Difficulty, ID measured in bits



Fitts' Law – Advanced Topics

- Throughput
  - Also known as index of performance or bandwidth
  - Single metric for input systems
  - One definition: TP = ID / MT ('average' values of ID and MT are used)
  - Another definition: TP = 1 / b (equals ID / MT only if a=0)
  - Probably still the best approach:
    - Use regression analysis to compute a and b
    - Use 1 / b as throughput cautiously

- See detailed discussion in [Zhai 2004]

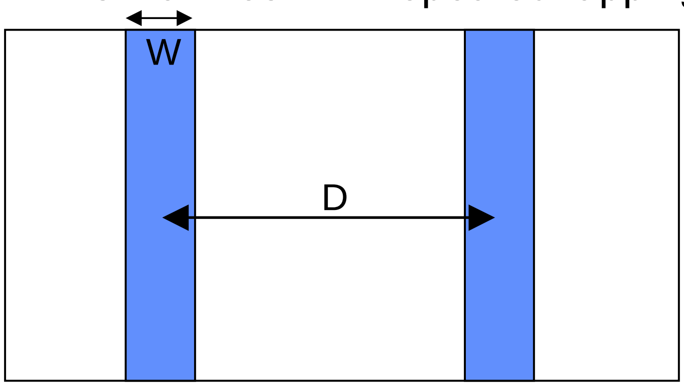
Zhai, S. 2004. Characterizing computer input with Fitts' law parameters: the information and non-information aspects of pointing. *Int. J. Hum.-Comput. Stud. 61, 6 (Dec. 2004), 791-809*

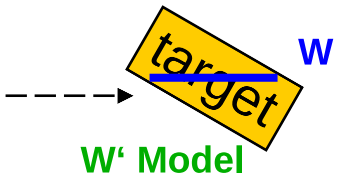
Fitts' Law Experiment

- Extension to 2D
  - "Status Quo": use horizontal width
  - "Sum Model": *W* = width + height
  - "Area Model": *W* = width * height
  - "Smaller Of": *W* = min(width, height)
  - "W ' Model": width in movement direction
  - See also [MacKenzie, Buxton 1992] and [Zhai et al. 2004] who refer to

$$ID = \log_2\left(\sqrt{\left(\frac{D}{W}\right)^2 + \eta\left(\frac{D}{H}\right)^2} + 1\right)$$

Original Fitts' Law test: 1D repeated tapping





MacKenzie, I. S. and Buxton, W. 1992. Extending Fitts' law to two-dimensional tasks. *In Proceedings CHI '92. 219-226.*

Zhai, S., Accot, J., and Woltjer, R. 2004. Human action laws in electronic virtual worlds: an empirical study of path steering performance in VR. *Presence: Teleoper. Virtual Environ. 13, 2 (Apr. 2004), 113-127.*

(Simple) Linear Regression

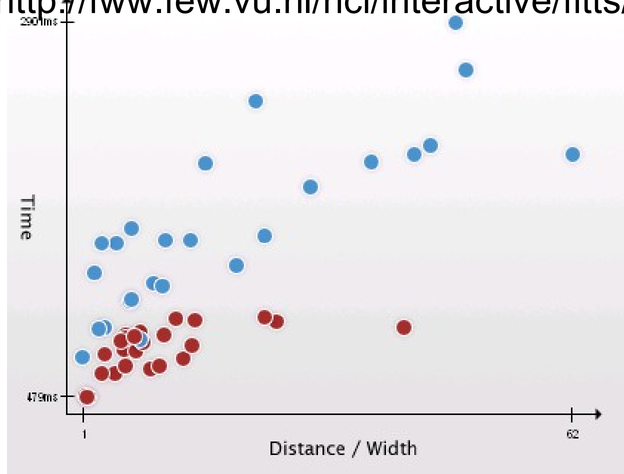
- How to measure a and b for a new pointing device / menu / etc.?

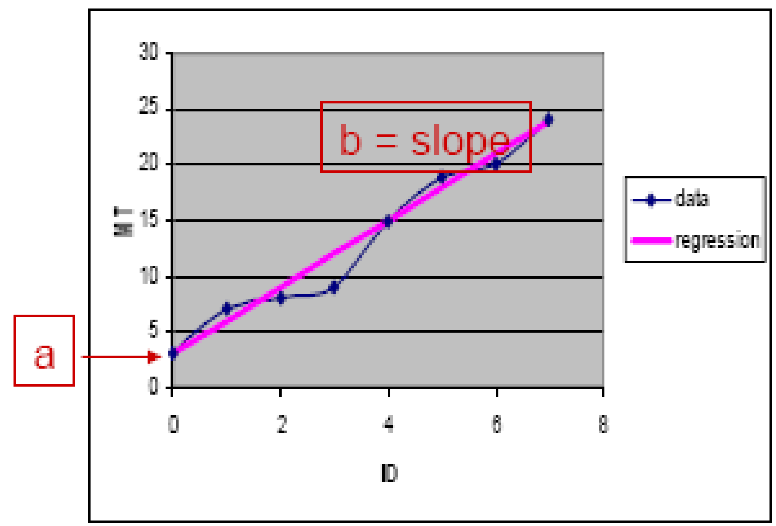
$$MT = a + b \log_2\left(1 + \frac{D}{W}\right)$$

- Setup an experiment with varying D and W and measure MT
- Fit a line through the measured points: a = intercept, b = slope

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

http://fww.few.vu.nl/hci/interactive/fitts/

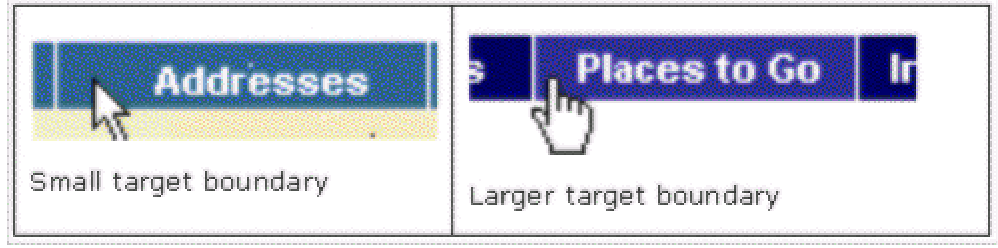




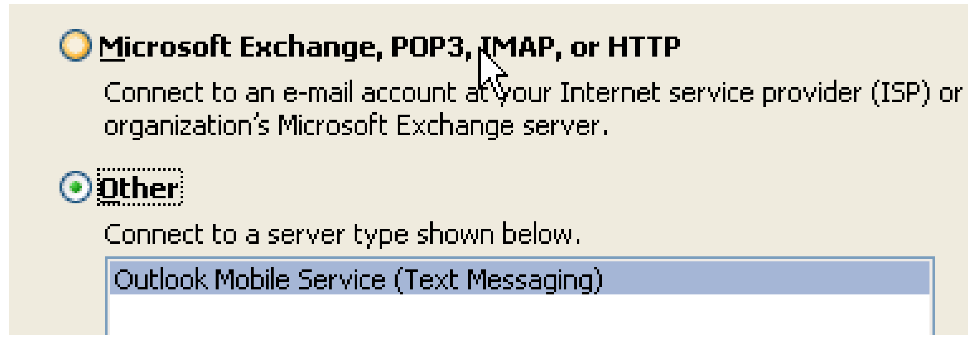
linear regression model

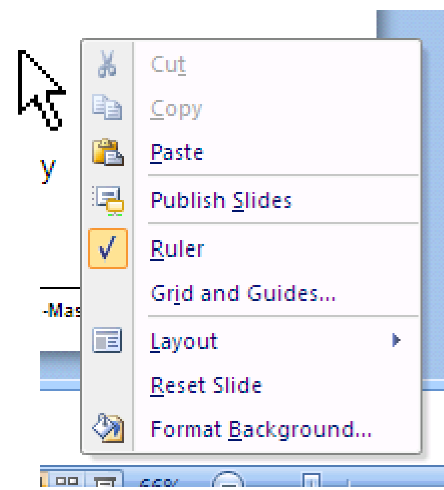
Implications for HCI (1)

- Bigger buttons
  - e.g. web links
  - e.g. check / radio boxes
- Proportional to amount of use?!
  - See principle (and golden rule) of **consistency**!
- Use current location of the cursor
  - distance is close to zero
- Use edges and corners (for examples see next slide)
  - edges of the screen have infinite height or width, respectively
  - corners have infinite height and width



http://msdn.microsoft.com/en-us/library/ms993291.aspx





Implications for HCI (1)

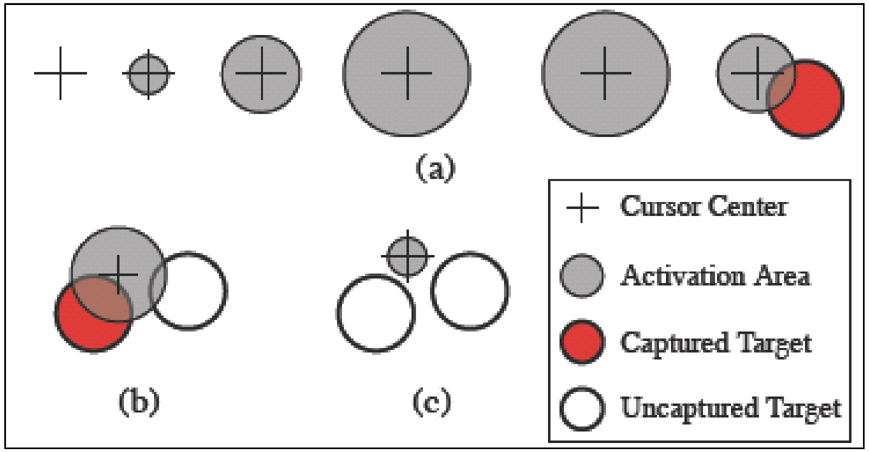
- Edges and corners

Windows

Mac OS X

Implications for HCI (2)

- Compare and evaluate input devices
- Current examples
  - Behind the display cursor
  - Dynaspot



Yang, X., Irani, P., Boulanger, P., and Bischof, W. 2009.
One-handed behind-the-display cursor input on mobile devices.
In Proceedings CHI EA '09. 4501-4506.

Chapuis, O., Labrune, J., and Pietriga, E. 2009. DynaSpot: speed-dependent area cursor.
In Proceedings CHI '09. 1391-1400

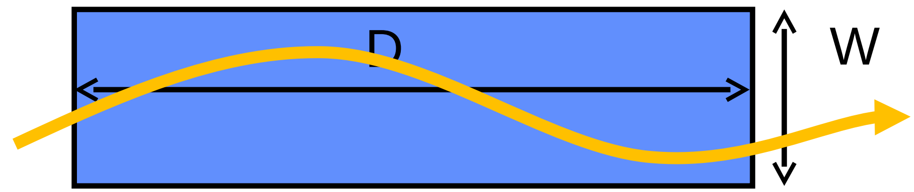
Additional Literature for Fitts' Law

- A Cybernetic Understanding of Fitts' Law:
  http://www.hcibook.com/e3/online/fitts-cybernetic/
- Bibliography of Fitts' Law Research
  (to get an impression about research in the HCI community):
  http://www.yorku.ca/mack/RN-Fitts_bib.htm
- Fitts' Law: Modelling Movement Time in HCI
  http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/fitts.html

Steering Law

- Equally early discovery: 1959 by Nicolas Rashevsky
- For HCI rediscovered in 1997 and there sometimes called the Accot-Zhai steering law

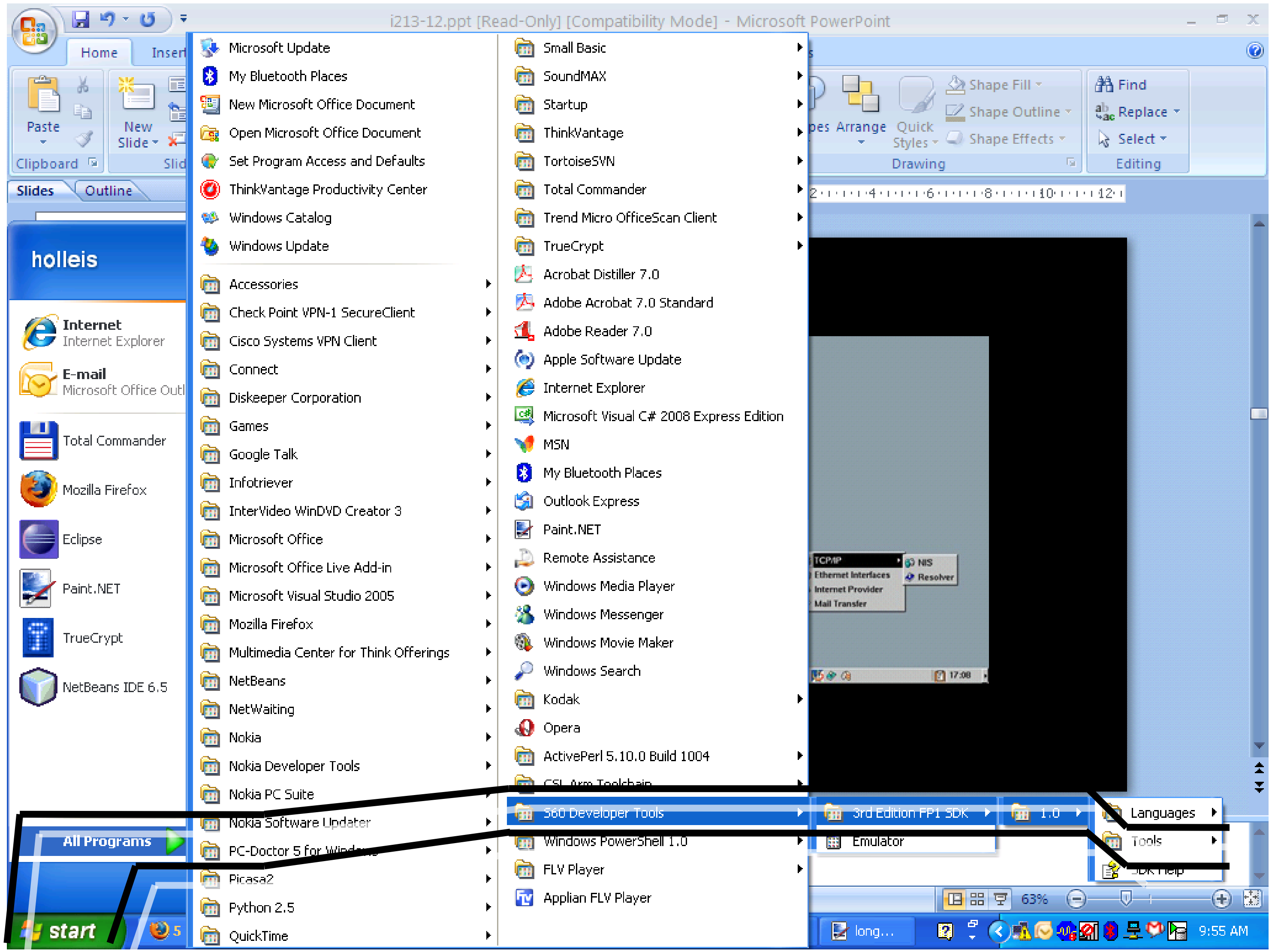
- Models the movement time of a pointer through a 2D tunnel
- Can be seen as an extension to Fitts' Law



Rashevsky, N. (1959). Mathematical biophysics of automobile driving. *In The Bulletin of Mathematical Biophysics 21:375-385*

Accot, J. and Zhai, S. (1997). Beyond Fitts' law: models for trajectory-based HCI tasks. *In Proceedings CHI '97. 295-302.*

Steering Law in Practice



Steering Law Equation

- The time to acquire a target through a tunnel is a function of the **length** and **width** of the tunnel and depends on the particular pointing **system**

$$MT = a + b\frac{D}{W}$$

- **MT**: movement time
- **a and b**: constants dependent on the pointing system
- **D**: distance, i.e. length of the tunnel
- **W**: width of the tunnel

Steering Law Equation – Index of Difficulty

- The time to acquire a target through a tunnel is a function of the **length** and **width** of the tunnel and depends on the particular pointing **system**

$$MT = a + b\underbrace{\frac{D}{W}}$$

- ID (Index of Difficulty): ID = D / W

- Index of Difficulty is now linear, not logarithmic as in Fitts' Law
  - Steering is more difficult then pointing

Steering Law Extension to Arbitrary Tunnels

- The time to acquire a target through a tunnel is a function of the **length** and **width** of the tunnel and depends on the particular pointing **system**

- The previously shown formula applies only for constant width W

$$MT = a + b\frac{D}{W}$$

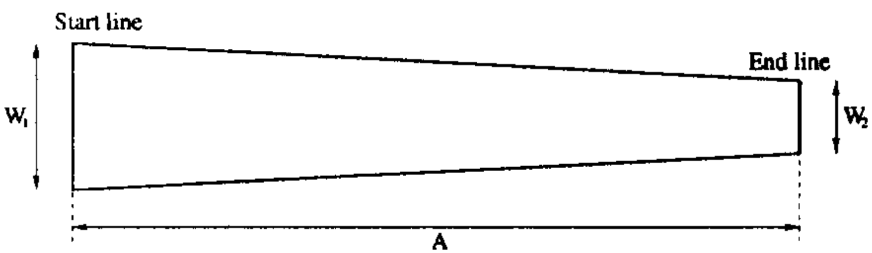
- Let the width W(s) be parameterized by s running from 0 to D

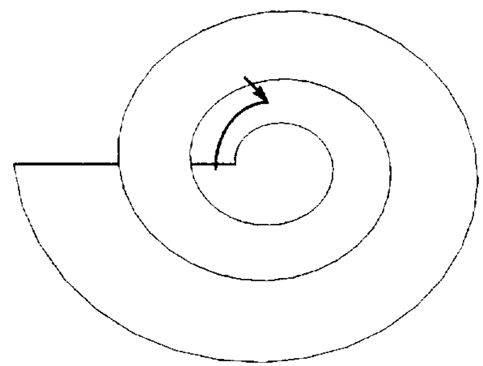
$$MT = a + b\int_C \frac{ds}{W(s)}$$

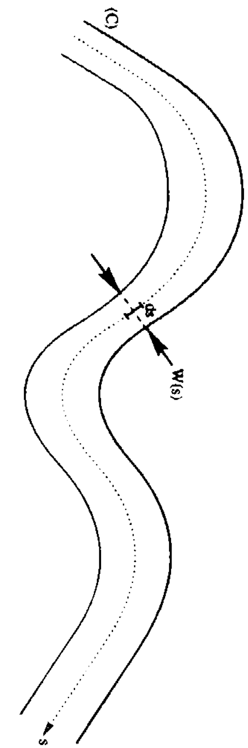
- **C**: path characterised by s
- **W(s)**: width dependent on s

Steering Law Applied

- Early work focused on car driving scenarios and models with straight tunnels
- Various example tunnel shapes have been explored



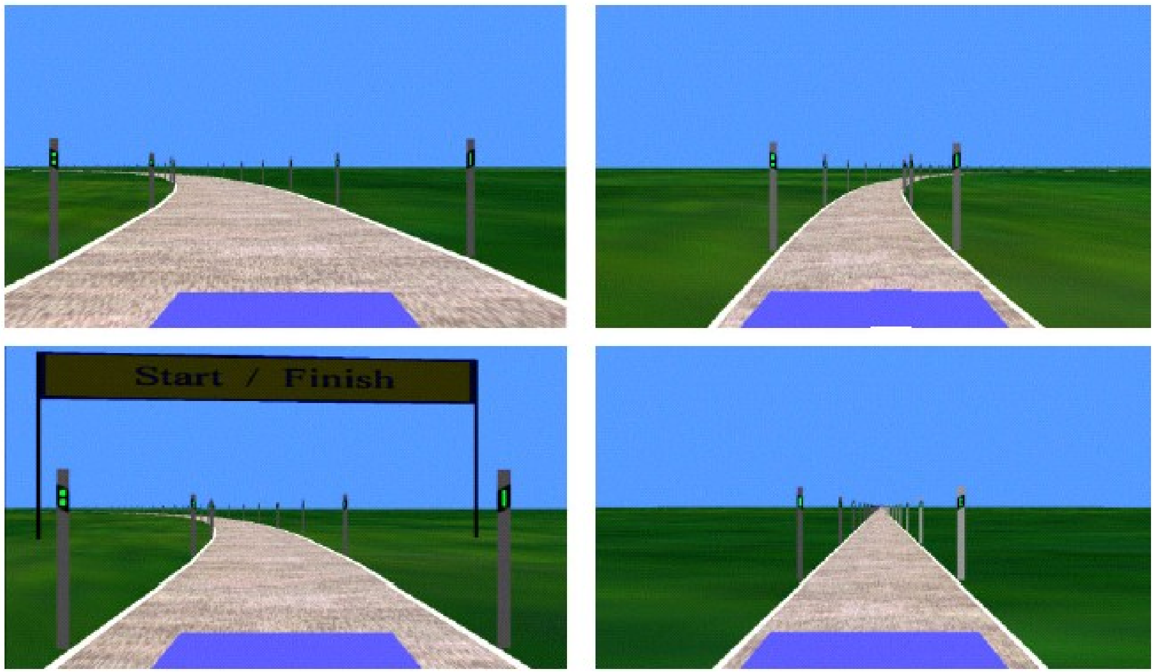




Accot, J. and Zhai, S. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. *In Proceedings CHI '97. 295-302.*

Steering Law Applied

- Further extension to 3D e.g. virtual reality applications



Zhai, S., Accot, J., and Woltjer, R. 2004. Human action laws in electronic virtual worlds: an empirical study of path steering performance in VR. *Presence: Teleoper. Virtual Environments 13, 2. 113-127.*

Looking Back: Fitts' Law

- Predicts movement time for rapid, aimed pointing tasks
- One of the few stable observations in HCI
- $MT = a + b\log_2\left(1+\frac{D}{W}\right)$ Index of Difficulty: $\log_2\left(1+\frac{D}{W}\right)$
- How to get a and b for a specific device / interaction technique
  - vary D and W and measure MT; fit a line by linear regression
- Various implications for HCI
  - Consider button sizes
  - Use edges and corners
  - Use current location of the cursor
  - Use average location of the cursor(?)
  - **Possibility to compare different input devices**

Looking Back: Steering Law

- Models the movement time of a pointer through a 2D tunnel
- Extension of Fitts' Law
- Tunnels with constant width: Index of Difficulty: D / W $MT = a + b\frac{D}{W}$
- Extension for arbitrary tunnel shapes: $MT = a + b\int_C \frac{ds}{W(s)}$
- Implications for HCI
  - Nested menus
  - Navigation tasks
  - Extensions for virtual reality / 3D movements possible

Basic HCI Models

- Predictive Models for Interaction: Fitts' / Steering Law
- Descriptive Models for Interaction: GOMS / KLM

To Recap: *Predictive* Models

- Model:
  - Simplification of a complex situation / action, e.g. human interaction
- Predictive:
  - Make educated guesses about the future
    - » relying on knowledge about past actions / states
    - » relying on a model of interaction
- Examples:
  - Fitts' Law (directed aimed movement)
  - Law of Steering (navigation through a tunnel)
  - Hick's Law / Hick-Hyman Law (choose an item within a menu)
  - ...

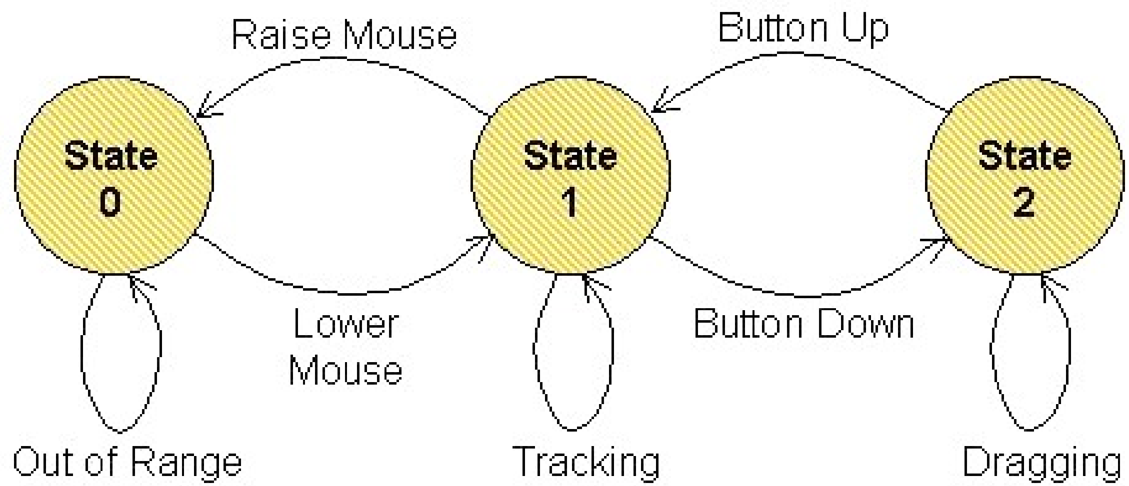
Descriptive Models

- *(The categorisation is not sharp, for more insights, see [MacKenzie 2003])*
- Descriptive models
  - provide a basis for understanding, reflecting, and reasoning about certain facts and interactions
  - provide a conceptual framework that simplifies a, potentially real, system
  - are used to inspect an idea or a system and make statements about their probable characteristics
  - used to reflect on a certain subject
  - can reveal flaws in the design and style of interaction
- Examples:
  - Descriptions, statistics, performance measurements
  - Taxonomies, user categories, interaction categories

MacKenzie, I. S., 2003, Motor Behaviour Models for Human-computer Interaction
In *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science (Book), 27-54*

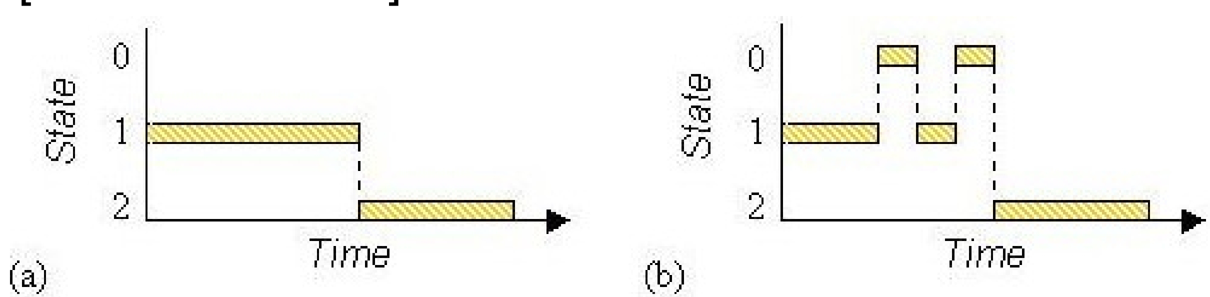
Example: Three-State Model (W. Buxton)

- Describes graphical input
- Simple, quick, expressive
- Possible extensions:
  - multi-button interaction
  - stylus input
  - direct vs. indirect input



Buxton, W, 1990, A Three-State Model of Graphical Input
In INTERACT'90, 449-456

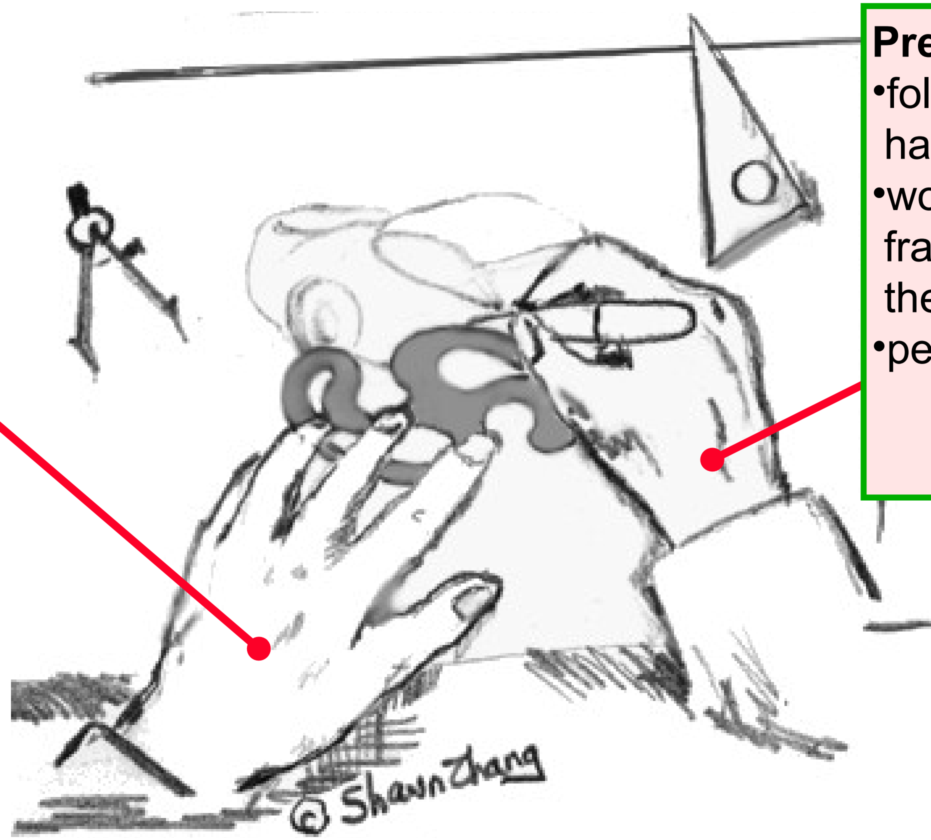
Dragging tasks: (a) mouse (b) lift-and-tap touchpad.
[MacKenzie 2003]



Example: Guiard's Model of Bimanual Skill (1 / 2)

- Many tasks are asymmetric with regard to left / right hand
- Guiard's model identifies the roles and actions of the non-preferred and preferred hands

Non-preferred hand
- leads the preferred hand
- sets the spatial frame of reference for the preferred hand
- performs coarse movements



Preferred hand
- follows the non-preferred hand
- works within established frame of reference set by the non-preferred hand
- performs fine movements

Example: Guiard's Model of Bimanual Skill (2 / 2)

Microsoft Office Keyboard

Task	Characteristics
Scrolling	• precedes/overlaps other tasks • sets the frame of reference • minimal precision needed (coarse)
Selecting, editing, reading, drawing, etc.	• follows/overlaps scrolling • works within frame of reference set by scrolling • demands precision (fine)

The GOMS Model

- **G**: goals
  - (Verbal) description of what a user wants to accomplish
  - Various levels of complexity possible
- **O**: operators
  - Possible actions in the system
  - Various levels of abstraction possible (sub-goals / ... / keystrokes)
- **M**: methods
  - Sequences of operators that achieve a goal
- **S**: selection rules
  - Rules that define when a user employs which method
- User tasks are split into goals which are achieved by solving sub-goals in a divide-and-conquer fashion

Card, S. K.; Newell, A.; Moran, T. P., 1983, The Psychology of Human-Computer Interaction (Book)

GOMS Example: Move Word (1 / 2)

Goal: move the word starting at the cursor position to the end of the text
[select **use-keyboard**
delete-and-write
use-mouse]
verify move

Main goal with methods

Goal: **use-keyboard**

Goal: select word
[select use <shift> and *n**<cursor right>
use <shift> and <ctrl> and <cursor right>]
verify selection

Sub-goal

...

Method 1

Goal: **delete-and-write**
...

Method 2

Goal: **use-mouse**

Goal: select word
[select click at beginning and drag till the end of the word
double-click on the word]
verify selection

Goal: move word
[select click on word and drag till end of text
Goal: **copy-paste-with-mouse**
...]

Method 3

GOMS Example: Move Word (2 / 2)

- Selection rules:
  - Rule 1: use method **use-keyboard** if no mouse attached
  - Rule 2: use method **delete-and-write** if length of word < 4
  - Rule 3: use method **use-mouse** if hand at mouse before action
  - ...
- Selection rules depend on the user ($\rightarrow$ remember user diversity?)
- GOMS models can be derived in various levels of abstraction
  - e.g. goal: write a paper about X
  - e.g. goal: open the print dialog

GOMS Example: Closing a Window

```
GOAL: CLOSE-WINDOW
    [select GOAL: USE-MENU-METHOD
                MOVE-MOUSE-TO-FILE-MENU
                PULL-DOWN-FILE-MENU
                CLICK-OVER-CLOSE-OPTION
            GOAL: USE-ALT-F4-METHOD
                PRESS-ALT-F4-KEYS]

For a particular user:
   Rule 1: Select USE-MENU-METHOD unless another
           rule applies
   Rule 2: If the application is GAME,
           select ALT-F4-METHOD
```

GOMS Example: ATM Machine

```
GOAL: GET-MONEY
. GOAL: USE-CASH-MACHINE
   .     INSERT-CARD
   .     ENTER-PIN
   .     SELECT-GET-CASH
   .     ENTER-AMOUNT
   .     COLLECT-MONEY
   .     COLLECT-CARD
```

(outer goal satisfied!)

```
GOAL: GET-MONEY
. GOAL: USE-CASH-MACHINE
   .     INSERT-CARD
   .     ENTER-PIN
   .     SELECT-GET-CASH
   .     ENTER-AMOUNT
   .     COLLECT-CARD
   .     COLLECT-MONEY
```

(outer goal satisfied!)

GOMS Example: ATM Machine

- GOMS gives an early understanding of interactions
- "How *not* to loose your card"

```
GOAL: GET-MONEY
. GOAL: USE-CASH-MACHINE
  .     INSERT-CARD
  .     ENTER-PIN
  .     SELECT-GET-CASH
  .     ENTER-AMOUNT
  .     COLLECT-MONEY
            (outer goal satisfied!)
  .     COLLECT-CARD
```

```
GOAL: GET-MONEY
. GOAL: USE-CASH-MACHINE
  .     INSERT-CARD
  .     ENTER-PIN
  .     SELECT-GET-CASH
  .     ENTER-AMOUNT
  .     COLLECT-CARD
  .     COLLECT-MONEY
        (outer goal satisfied!)
```

Some GOMS Variations

GOMS

(CMN-)GOMS
- Plain GOMS
- Pseudo-code
- First introduced by Card, Moran and Newell
- (This is the version we looked at)

KLM
- Keystroke-Level Model
- Simplified version of GOMS
- (See next slides)

NGOMSL
- Natural GOMS Language
- Stricter version of GOMS
- Provides more well-defined, structured natural language
- Estimates learning time

CPM-GOMS
- Cognitive Perceptual Motor analysis of activity
- Critical Path Method
- Based on the parallel multi-processor stage of human information processing

John, B., Kieras, D., 1996, Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction, 3, 287-319.*

GOMS – Characteristics

- Usually one high-level goal
- Measurement of performance: high depth of goal structure
  → high short term-memory requirements
- Predict task completion time (see KLM in the following)
  → compare different design alternatives

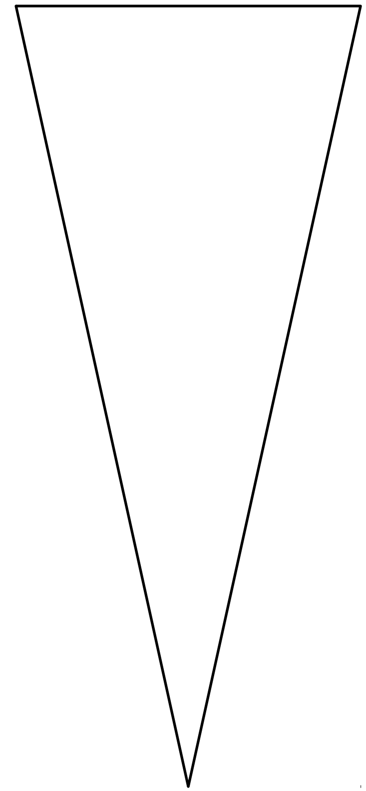
Keystroke-Level Model

- Simplified version of GOMS
  - only operators on keystroke-level
  - no sub-goals
  - no methods
  - no selection rules
- KLM predicts how much time it takes to execute a task
- Execution of a task is decomposed into primitive operators
  - Physical motor operators
    - » pressing a button, pointing, drawing a line, …
  - Mental operator
    - » preparing for a physical action
  - System response operator
    - » user waits for the system to do something

Models: Levels of Detail

- Different levels of detail for the steps of a task performed by a user

- Abstract: `correct wrong spelling`
- Concrete: `mark-word`
  `delete-word`
  `type-word`
- Keystroke-Level: `hold-shift`
  `n·cursor-right`
  `recall-word`
  `del-key`
  `n·letter-key`



KLM Operators

- Each operator is assigned a duration
  (amount of time a user would take to perform it):

Operator	Description	Associated Time
K	Keystroke, typing one letter, number, etc. or function key like 'CTRL', 'SHIFT'	Expert typist (90 wpm): 0.12 sec Average skilled typist (55 wpm): 0.20 sec Average non-secretarial typist (40 wpm): 0 .28 sec Worst typist (unfamiliar with keyboard): 1.2 sec
H	'Homing', moving the hand between mouse and keyboard	0.4 sec
B / BB	Pressing / clicking a mouse button	0.1 sec / 2*0.1 sec
P	Pointing with the mouse to a target	0.8 to 1.5 sec with an average of 1.1 sec Can also use Fitts' Law
$D(n_D, l_D)$	Drawing n_D straight line segments of length l_D	$0.9*n_D + 0.16*l_D$
M	Subsumed time for mental acts; sometimes used as 'look-at'	1.35 sec (1.2 sec according to [Olson and Olson 1995])
R(t) or ***W(t)***	System response (or 'work') time, time during which the user cannot act	Dependent on the system, to be determined on a system-by-system basis

Predicting the Task Execution Time

- Execution Time
  - OP: set of operators
  - n_{op}: number of occurrences of operator op

$$T_{execute} = \sum_{op \in OP} n_{op} \times op$$

- Example task on Keystroke-Level:

	Sequence:
– `hold-shift`	K (Key)
– `n·cursor-right`	n · K
– `recall-word`	M (Mental Thinking)
– `del-key`	K
– `n·letter-key`	n · K

- Operator Time Values: K = 0.28 sec. and M = 1.35 sec
  2n · K + 2 · K + M = 2n·0.28 + 1.91 sec

- → time it takes to replace a n=7 letter word: T = 5.83 sec

Keystroke-Level Model – Example Task

Task: in MS Word, add a 6pt space after the current paragraph

→ Word 2003:

Actions	Operator (keyboard)	Time allocated	Operator (mouse)	Time allocated
Locate menu 'Format'	***M***	1.35	***M***	1.35
Press ALT-o or mouse click	***K***,***K***	2*0.28	***P***,***B***	1.10+0.10
Locate entry 'Paragraph'	***M***	1.35	***M***	1.35
Press 'p' or mouse click	***K***	0.28	***P***,***B***	1.10+0.10
Locate item in dialogue	***M***	1.35	***M***	1.35
Point to item	***K***,***K***	0.28	***P***,***B***	1.10+0.10
Enter a 6 for a 6pt space	***K***	0.28	***K***	0.28
Close the dialogue (ENTER)	***K***	0.28	***K***	0.28
	Sum (keyboard): 5.73 sec.		**Sum (mouse): 8.21 sec.**	

→ Word 2007: **Sum (keyboard): 7.22 sec.** **Sum (mouse): 7.65 sec.**

GOMS vs. KLM

(CMN-)GOMS

- Pseudo-code (no formal syntax)
- Very flexible
- Goals and subgoals
- Methods are informal programs
- Selection rules
  ⇨ tree structure: use different branches for different scenarios
- Time consuming to create

KLM

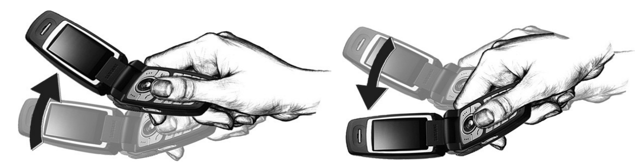
- Simplified version of GOMS
- Only operators on keystroke-level
  ⇨ focus on very low level tasks
- No multiple goals
- No methods
- No selection rules
  ⇨ strictly sequential
- Quick and easy

Problem with GOMS in general

- Only for well defined routine cognitive tasks
- Assumes statistical experts
- Does not consider slips or errors, fatigue, social surroundings, …

Extensions for Novel Mobile Interactions

- Current mobile interactions use
  - Keypad, hotkeys
  - Microphone, camera (marker detection)
  - Sensors like accelerometers
  - Tag readers (NFC)
  - Bluetooth



- Method
  - Large set of studies
  - Software on the phone
  - Video frame-by-frame analysis
  - Eye-tracker

  → Total number of actions measured: 2134

KLM – Original and New Operators

- Mental Act, M
- System Response, R

← unchanged

- Keystroke / button press, K
- Homing, H
- Pointing, P

← adopted

- Micro attentions Shift, S_{Micro}
- Macro attention shift, S_{Macro}
- Finger movement F
- Distraction X
- Gesture G
- Initial preparation I

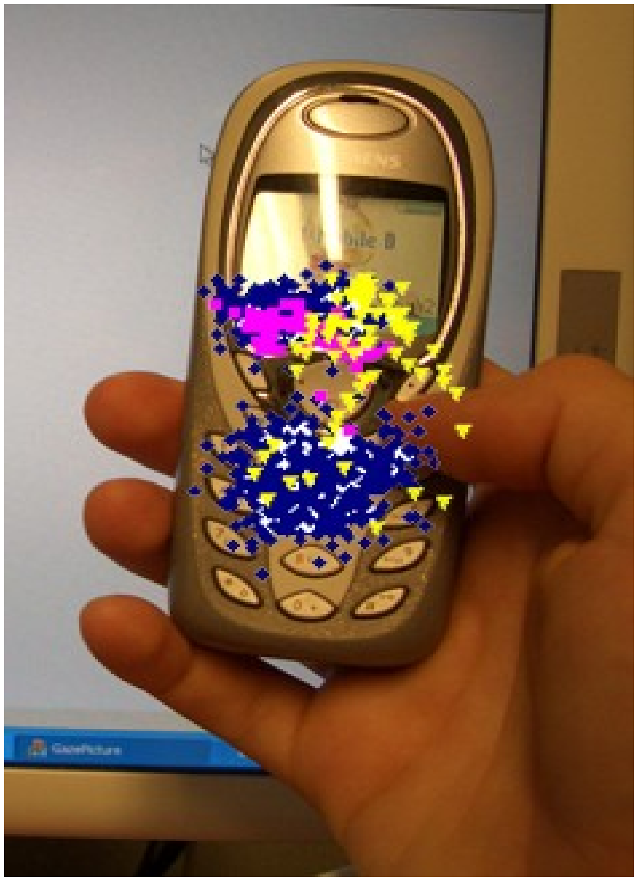
← added

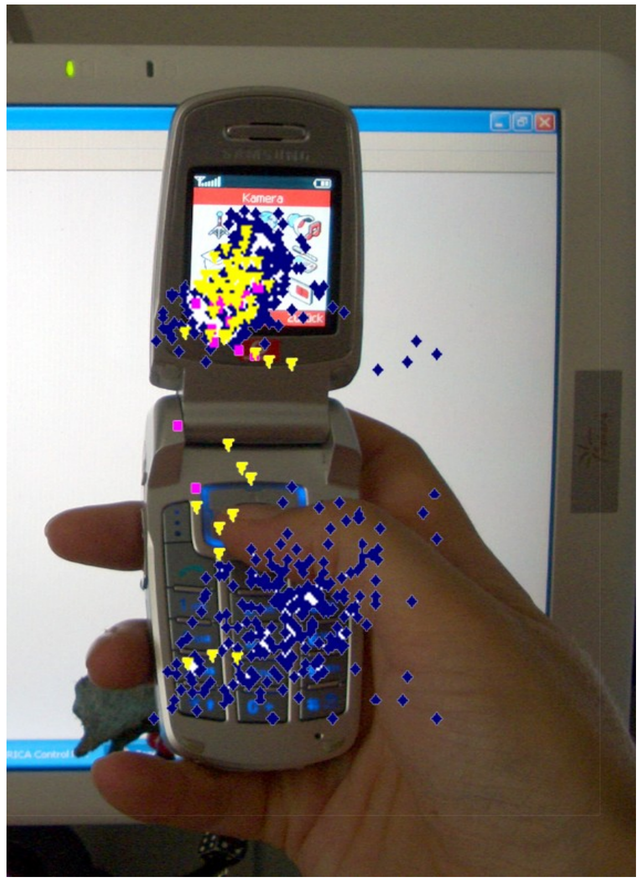
Micro Attention Shift, S_{Micro}

Switch attention between phone parts

S_{Micro} – Operator Time Estimation

- Measured with a standard eye tracker
- Mobile phone in front of the monitor (blue: text entry, yellow: menu navigation)



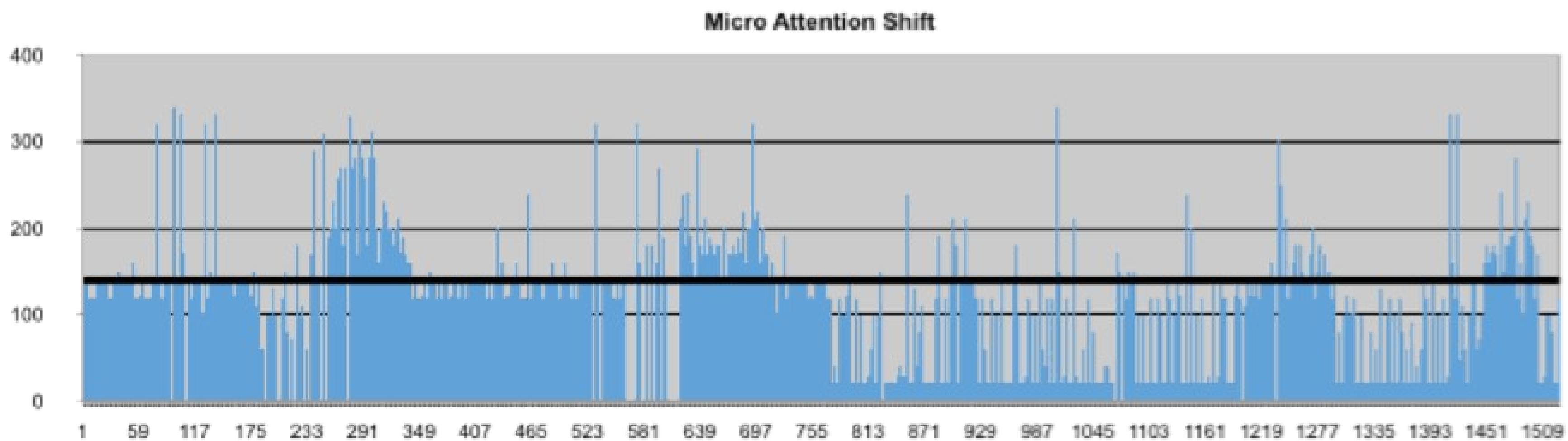


S_{Micro} – Operator Time Estimation

Study

- 10 participants, 24-34 years, 6 female
- 1500 shifts detected
- Using automatic eye-tracking
- 3 pre-set tasks

display ↔ hotkeys: 0.14 sec.
display ↔ keypad: 0.12 sec.
keypad ↔ hotkeys: 0.04 sec.



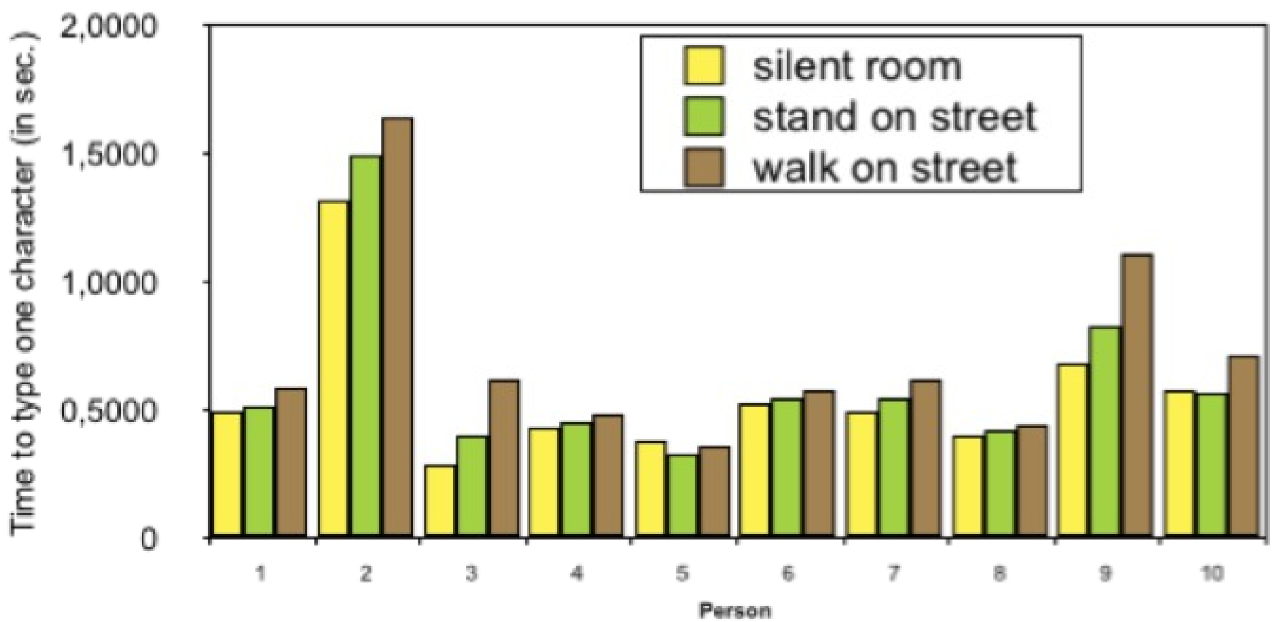
Distraction, X

Distraction: multiplicative

X_{slight} = 6%, X_{strong} = 21%

Study

- 10 participants, 24-33 years, 3 female
- Short message in 3 settings (quiet room, standing outside, walking)
- Relative slow-down (significant: $t=2.23$, $p<0.03$ and $t=3.28$, $p<0.01$)



Extended KLM – Time Prediction

Total Execution Time:

$$T_{execute} = \sum_{op \in OP} (n_{op} + d_{op} \times X_{slight} + D_{op} \times X_{strong}) \times op$$

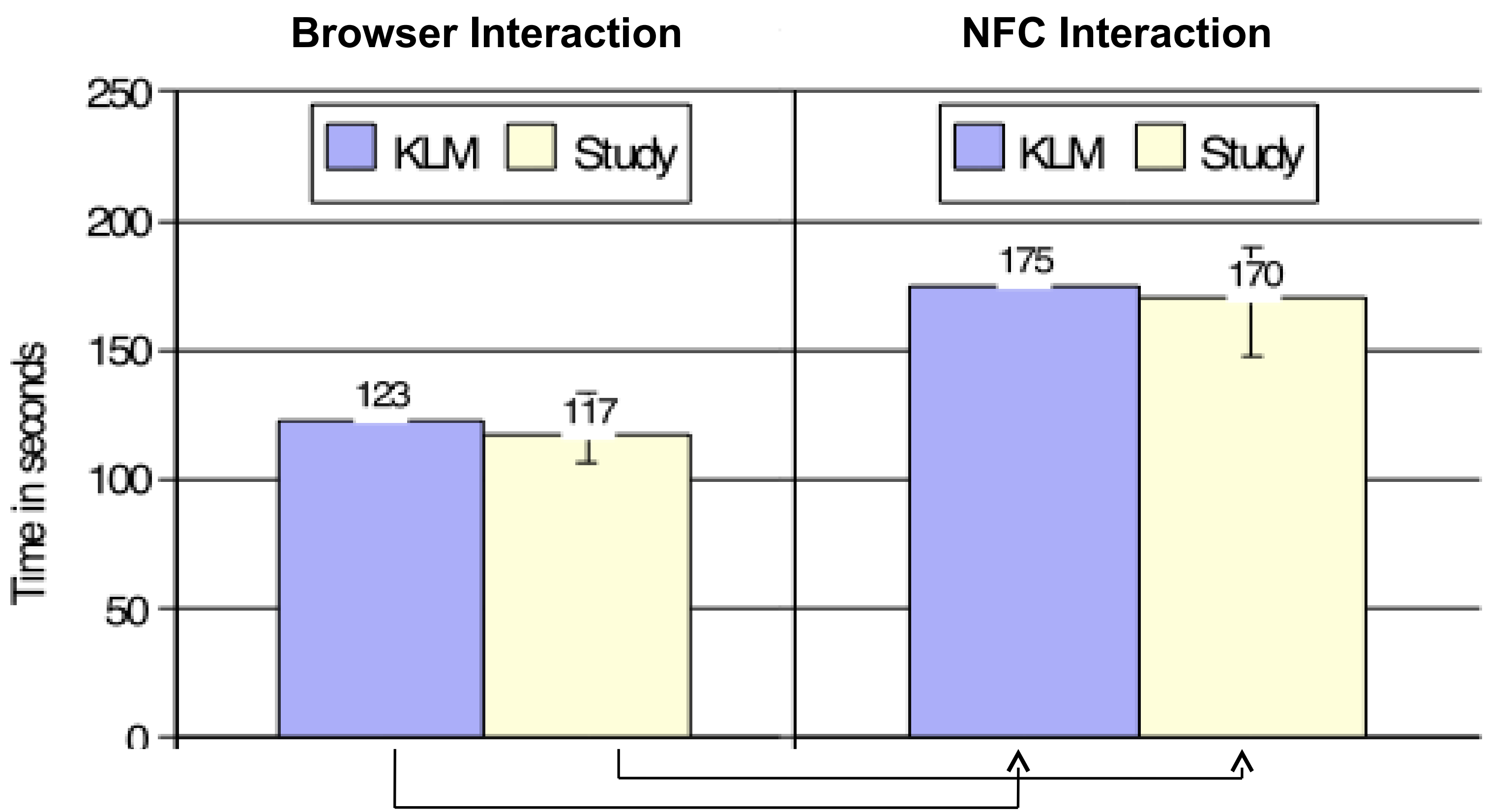
Set of Available Operators:

$$OP = \{A, F, G, H, I, K, M, P, R, S_{Micro}, S_{Macro}\}$$

Extended KLM – Empirical Validation

- Task: buy a public transportation ticket from A to B
- Implemented 2 ways of performing the task
  - Access through mobile web browser
  - Direct interaction with NFC tags
- Created the two Keystroke-Level Models
- Study: 9 people, 23-34 years, 3 female

Extended KLM – Empirical Validation



Predicted speed loss: 17%

Actual speed loss: 14%

Advanced Mobile Phone KLM – Values

Operator		Time	Qu. 1	Qu. 3
A*, *Action	picture / marker	**1.23**	0.61	1.44
	NFC	**0.00**	-	-
	in general	*variable, input to model*		
B, *Mouse Button Press*		*not applicable*		
D, *Mouse Drawing*		*not applicable*		
F*, *Finger Movement		**0.23**	0.20	0.29
G*, *Gestures		**0.80**	0.73	0.87
H, *Homing*		**0.95**	0.81	1.00
I*, *Initial Act	external trigger	**5.32**	3.98	7.51
	self triggered	**3.89**	2.23	4.89
	optimal setting	**1.18**	1.10	1.26
	no assumptions	**4.61**	-	-
K, *Keystroke*	keypad average	**0.39**	0.37	0.48
	keypad quick	**0.33**	0.32	0.37
	hotkey	**0.16**	0.15	0.20

Operator		Time	Qu. 1	Qu. 3
M, *Mental Act*		**1.35**	-	-
P, *Pointing*		**1.00**	0.84	1.20
R, *System Response Time*	NFC	**2.58**	2.46	2.80
	visual marker	**2.22**	2.09	2.82
	in general	*variable, input to model*		
S_{Macro}*, *Macro Attention Shift		**0.36**	0.28	0.44
S_{Micro}*, *Micro Attention Shift	keypad ↔ display	**0.14**	0.14	0.19
	hotkey ↔ display	**0.12**	0.02	0.14
	keypad ↔ hotkey	**0.04**	0.02	0.12
	in general	**0.14**	0.10	0.16
X*, *Distraction	slight	**6 %**	3 %	13 %
	strong	**21 %**	11 %	25 %

Using KLM

- KLM can help evaluate UI designs, interaction methods and trade-offs
- If common tasks take too long or consist of too many statements, shortcuts can be provided
- Predictions are mostly remarkable accurate: +/- 20%

Weaknesses of GOMS et al.

- Just spending time is not modelled
- Difficult to target specific users
- No real users
- Difficult to model novel interactions
- Various variable parameters
- Users like to have impact

Strengths of GOMS et al.

- Good treatment of learning effects
  - Measurement of learnability
  - Independence of sequences
  - Measurement of knowledge requirements
- Good results
  - Gives reasons
  - Helps in decision making
  - Identifies bottlenecks
  - Provides illustrative figures
  - Combines various views
  - Treats feasibility and cognitive load
- Less cost in money and time
  - Quick to apply
  - Quick to prepare
  - Helpful to design
  - Cheap to apply
  - Easy to repeat
  - Quick to analyse
  - Precise to interpret
  - Easy to convey

GOMS / KLM Summary Example

- Example prototype: the Combimouse

- Ergonomic models followed
- Follows Guiard's model of bimanual control
  (for right handed people scrolling with the non-preferred hand)
- Removes KLM's Homing operator (H ~ 1 sec.)

http://www.combimouse.com

References

GOMS

- Card S. K., Newell A., Moran T. P. (1983). The Psychology of Human-Computer Interaction. *Lawrence Erlbaum Associates Inc*.
- Card S. K., Moran T. P., Newell A. (1980). The Keystroke-level Model for User Performance Time with Interactive Systems. *Communication of the ACM 23(7)*. 396-410
- John, B., Kieras, D. (1996). Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction, 3,* 287-319.

KLM

- Kieras, D. (1993, 2001). Using the Keystroke-Level Model to Estimate Execution Times. *University of Michigan. Manuscript.*

Mobile Phone KLM

- Holleis, P., Otto, F., Hussmann, H., Schmidt, A. (2007). Keystroke-Level Model for Advanced Mobile Phone Interaction, *CHI '07*