Introduction to how to predict user performance

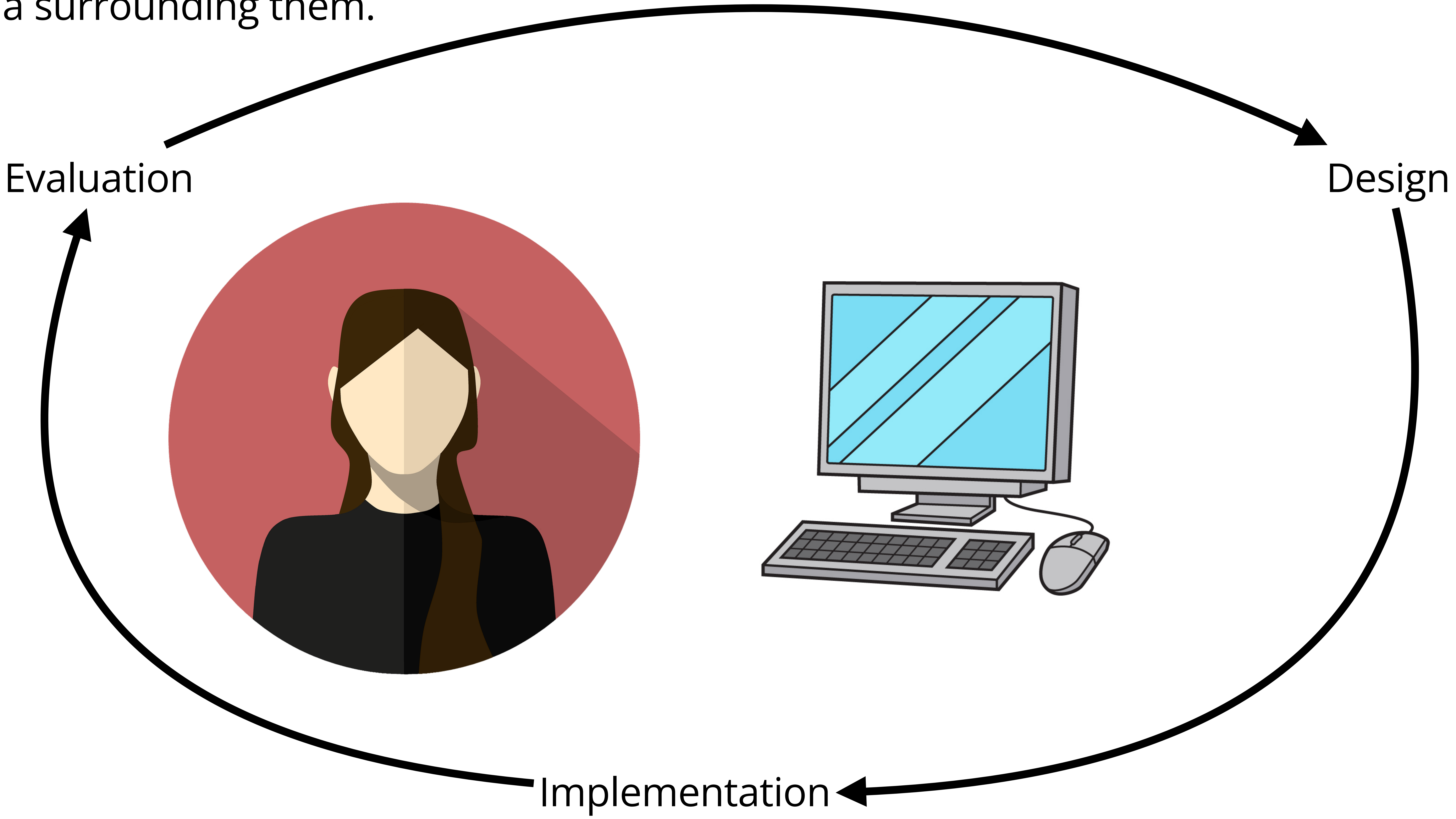
Sylvain Malacria

http://www.malacria.com/

mailto:sylvain.malacria@inria.fr

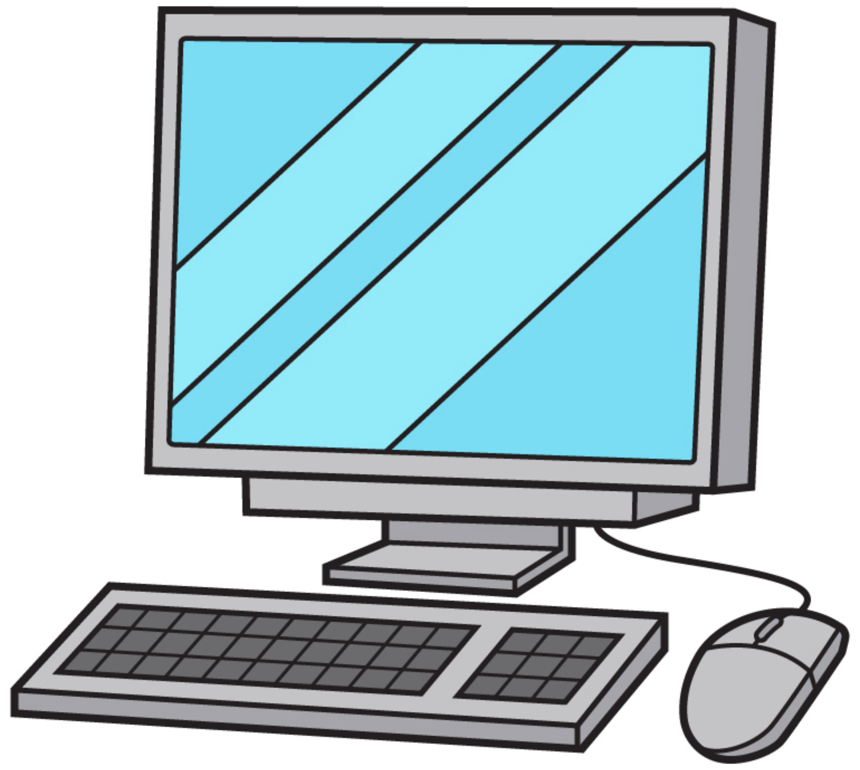
What is HCI?

HCI is concerned with the design, implementation and evaluation of interactive systems, and with the major phenomena surrounding them.



Evaluation ?

Not always possible/worth it



Paradox: cannot be run until it is "too late"

Predictive modelling techniques

Predicting user behaviour/performance

Card, Moran, Newell, “The Human as an Information Processor” 1983

GOMS, Keystroke Level Model (KLM), etc.

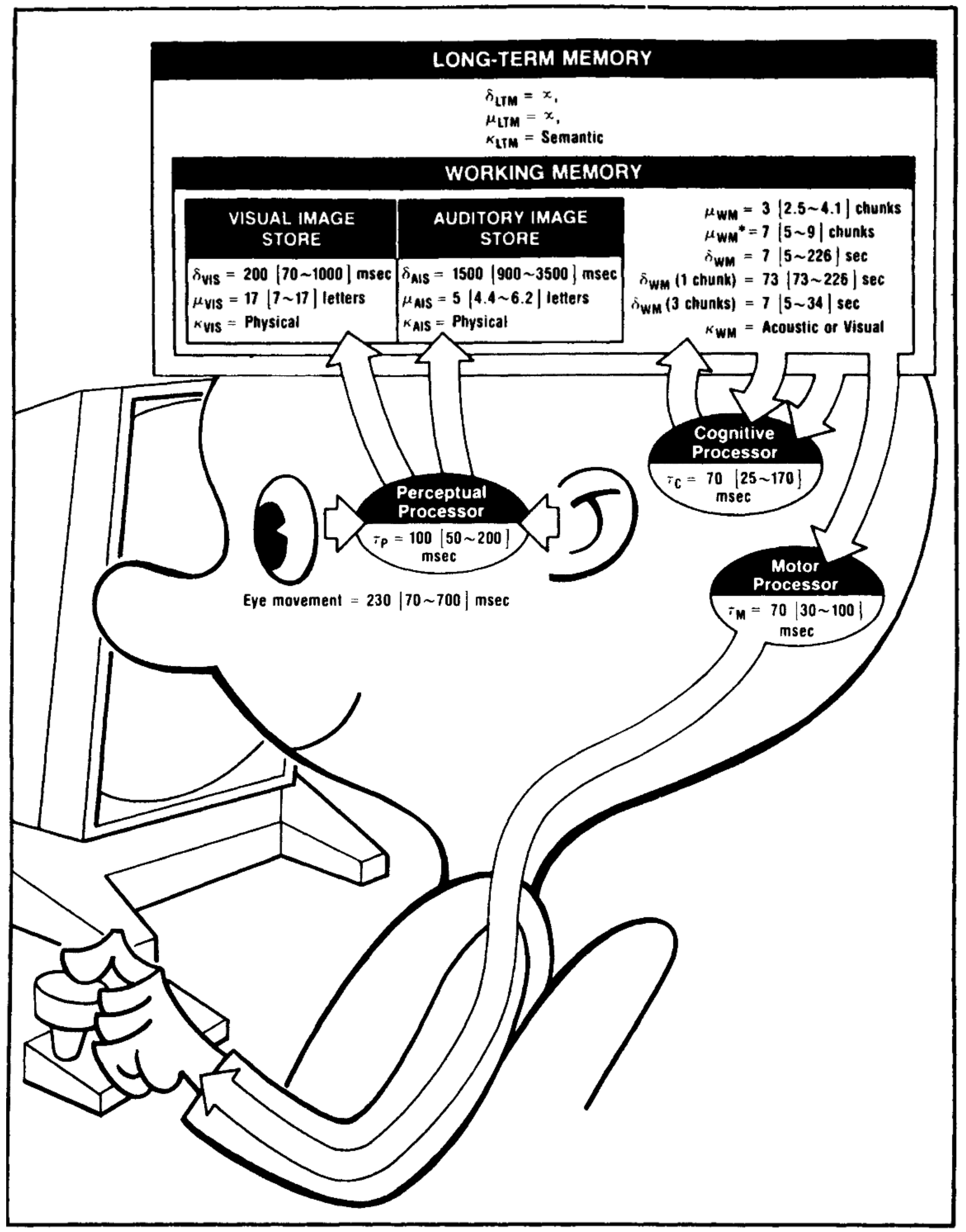
Fitts’ Law, etc.

The Psychology of Human-Computer Interaction

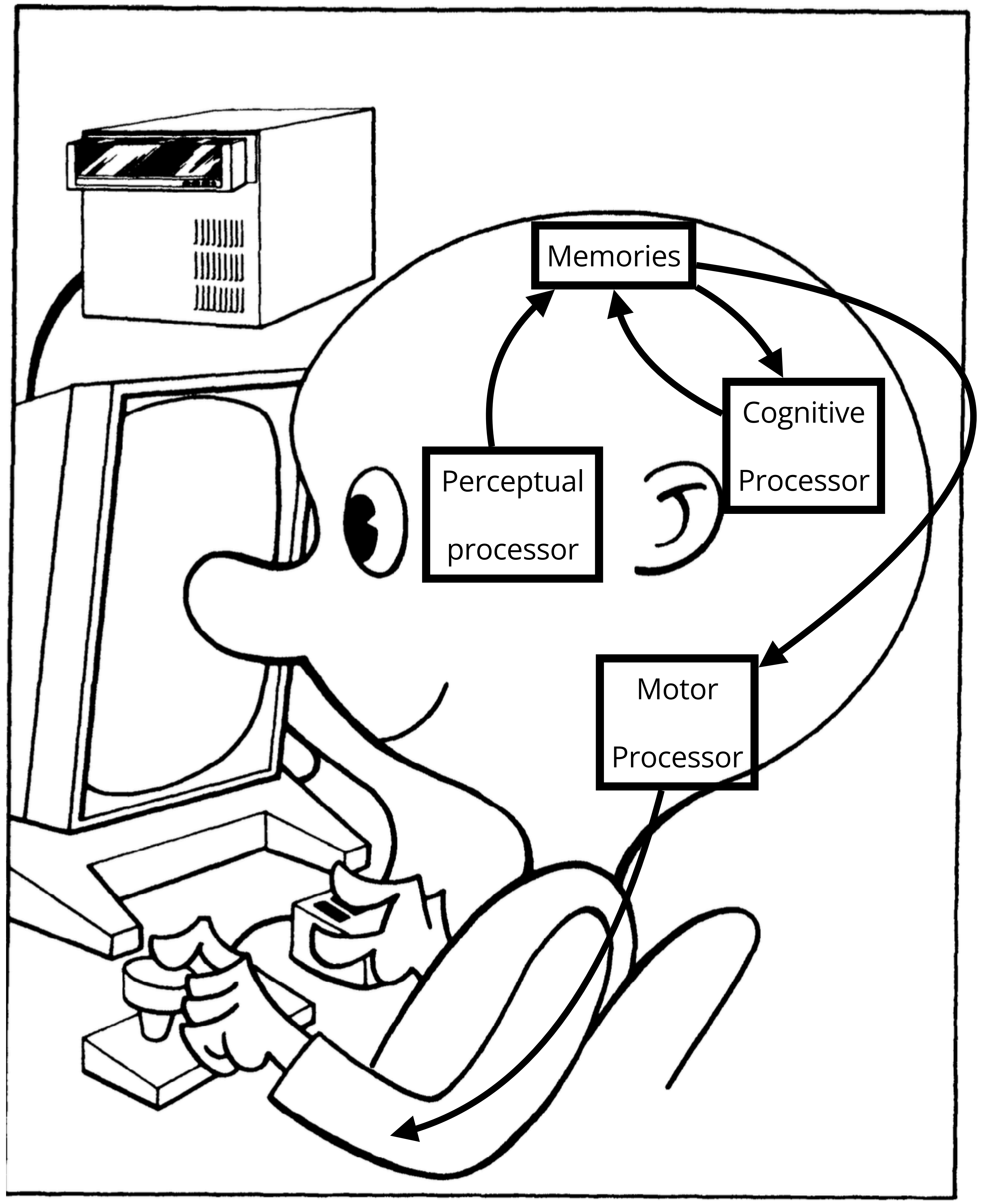
STUART K. CARD
THOMAS P. MORAN
ALLEN NEWELL

CRC Press
Taylor & Francis Group

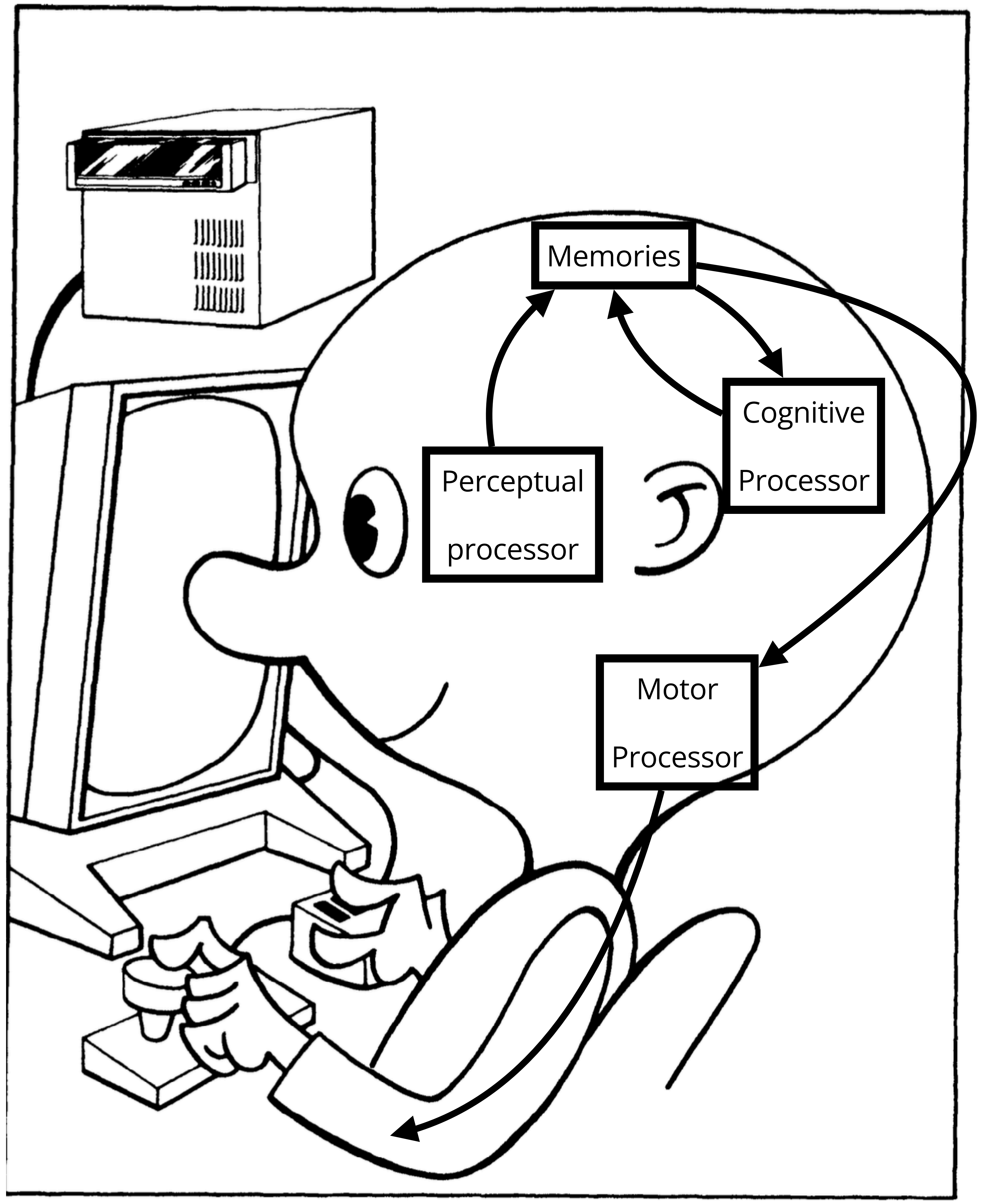
The Model Human Information Processor

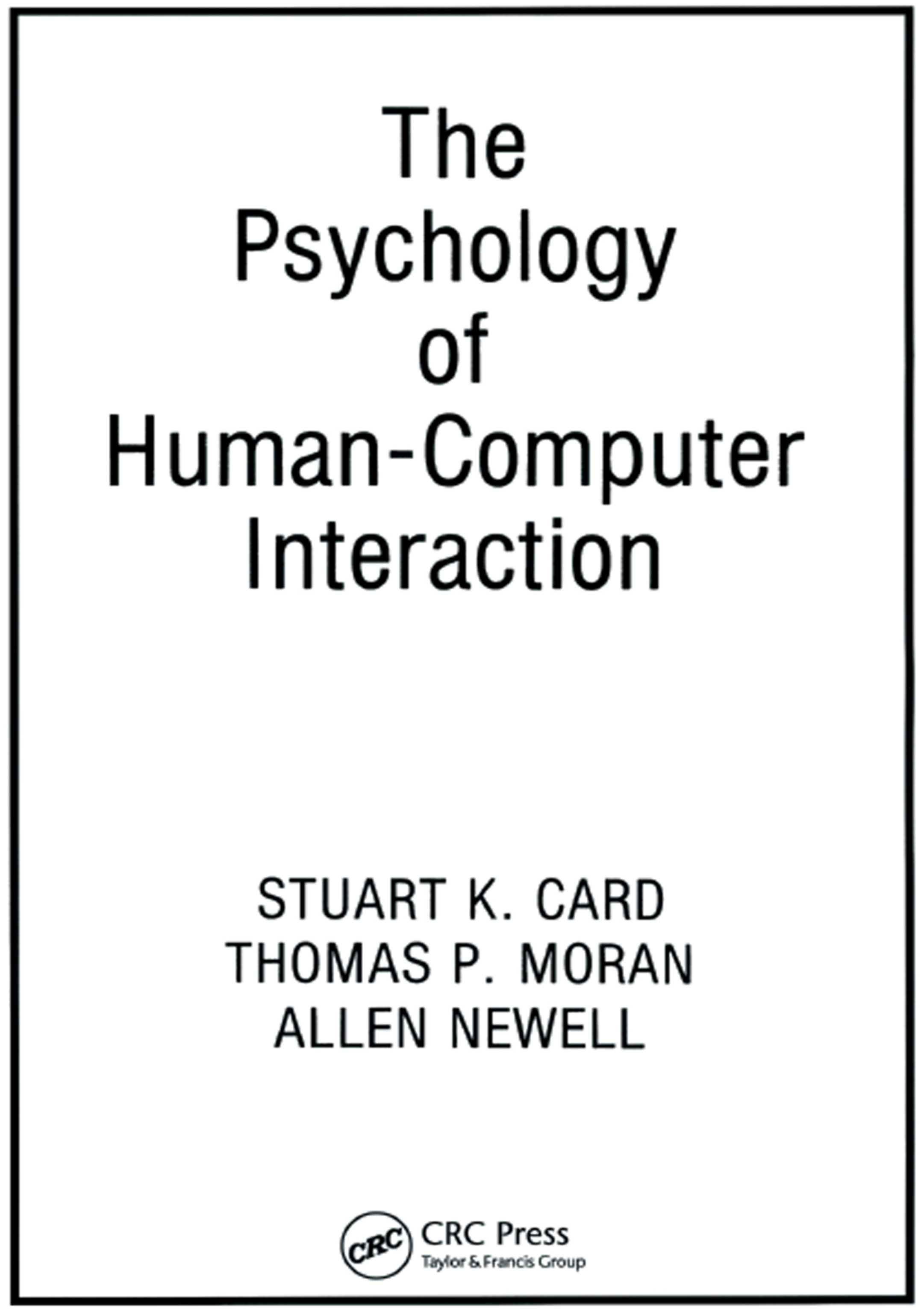


The Model Human Information Processor



The Model Human Information Processor





Main contribution

- Argues that models are important and useful
- "Teach" how to build them (with approximations)
- Models can be used for prediction and anticipatory evaluation

Example

*"A user is presented with **two symbols, one at a time**. If the second symbol is **identical** to the first, he is to push the key labeled **yes**. Otherwise he is to push **no**. What is the time between signal and response for the **yes case**?"*

(Card et al., 1983, p. 66)

If suggested text is the word in users' mind, then **yes case**

otherwise, no case.

Example

If suggested text is the word in users' mind, then **yes case**

otherwise, no case.

total execution time = Tp+2xTc+Tm

= 100 [30 ~ 200] + 2 x (70 [25 ~ 170])+ 70 [30 ~ 100]

= 310 [130 ~ 640] ms

Important high-level models

KLM (Keystroke-Level Model)

- Task (or series of sub-tasks)
- Method used
- Command language of the system
- Motor skill parameter of the user
- Response time parameters of the system

Important high-level models

GOMS (Goals, Operators, Methods, Selection Rules)

"Predicts" methods the user will adopt

- Goals: symbolic structures that define a state of affairs to be achieved and determinate a set of possible methods by which it may be accomplished
- Operators: elementary perceptual, motor or cognitive acts, whose execution is necessary to change any aspect of the user's mental state or to affect the task environment
- Methods: describe a procedure for accomplishing a goal
- Selection Rules: needed when a goal is attempted, there may be more than one method available to the user to accomplish it.

Important high-level models

GOMS (Goals, Operators, Methods, Selection Rules)

```
GOAL: EDIT-MANUSCRIPT
.   GOAL: EDIT-UNIT-TASK              repeat until no more unit tasks
.   .   GOAL: ACQUIRE-UNIT-TASK
.   .   .   GET-NEXT-PAGE             if at end of manuscript page
.   .   .   GET-NEXT-TASK
.   .   GOAL: EXECUTE-UNIT-TASK
.   .   .   GOAL: LOCATE-LINE
.   .   .   .   [select:   USE-QS-METHOD
                           USE-LF-METHOD]
.   .   .   GOAL: MODIFY-TEXT
.   .   .   .   [select:   USE-S-COMMAND
                           USE-M-COMMAND]
.   .   .   .   VERIFY-EDIT .
```

Lower-level user models

Fitts' law for target acquisition time

Steering law for path following time

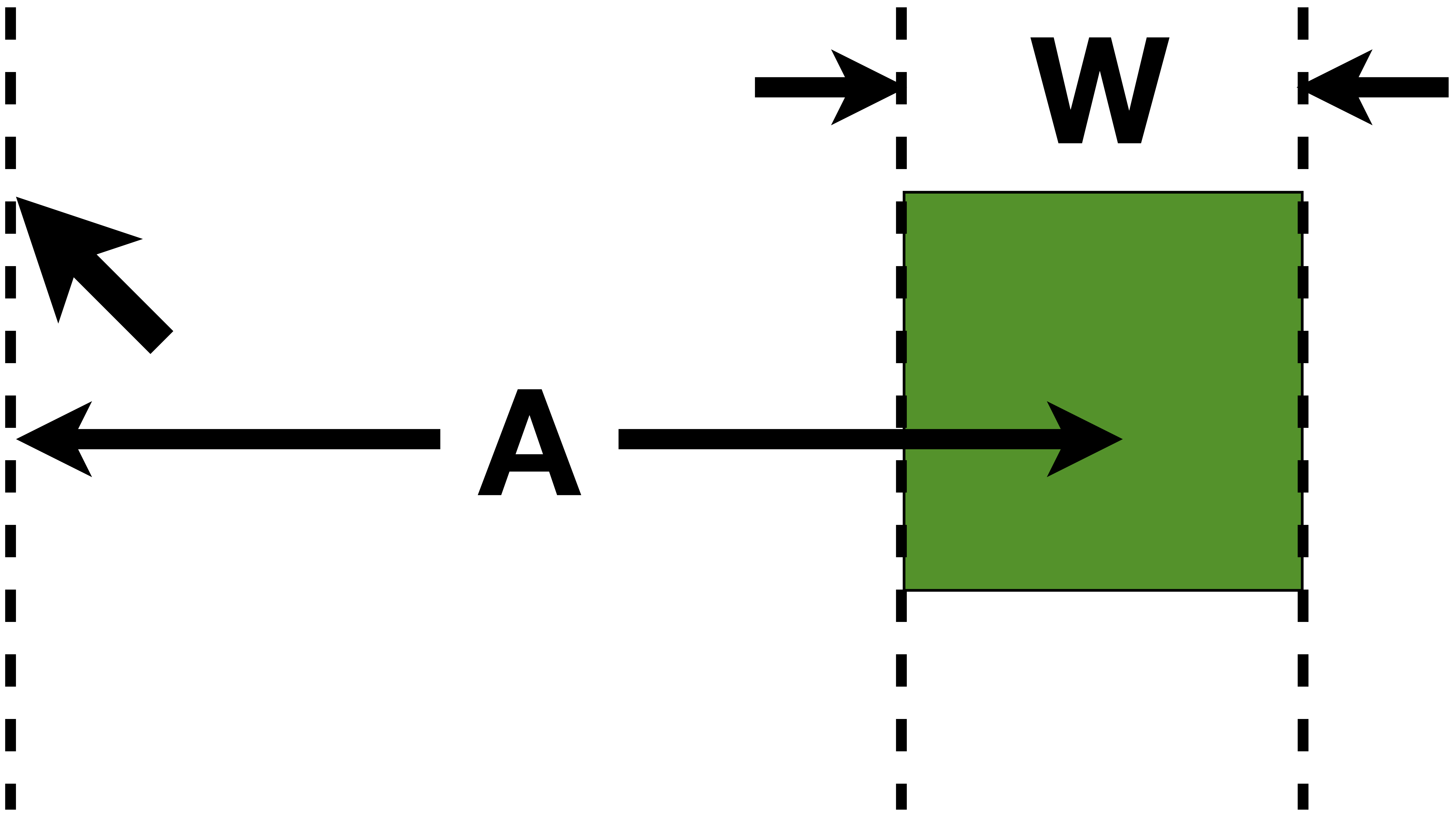
Hick-Hyman Law for choice reaction time

Power law of practice

Search, Decision and Pointing for menu selection time

Fitts' law

For target acquisition time



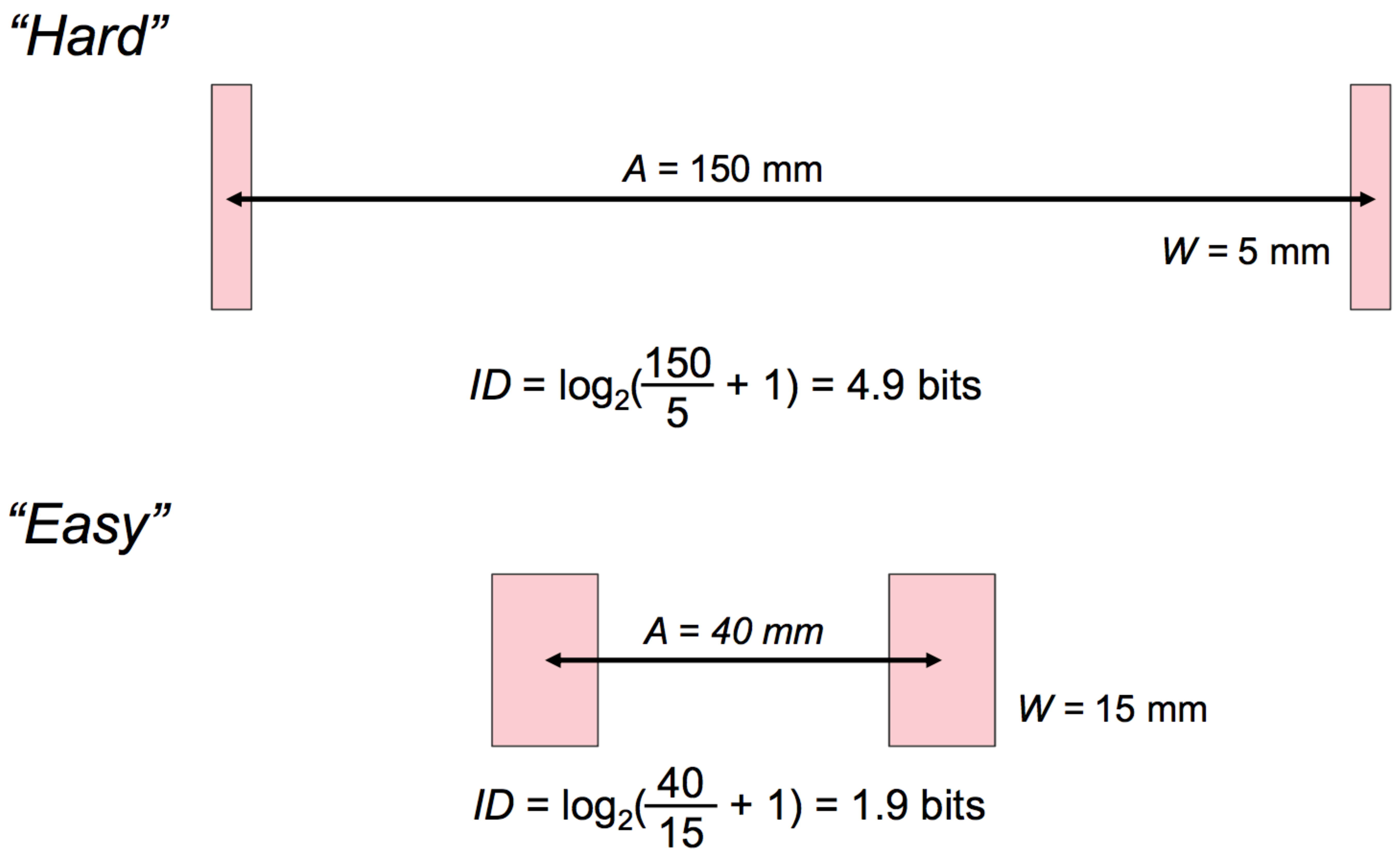
Fitts' law | Index of difficulty

For target acquisition time

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

Fitts' law | Index of difficulty

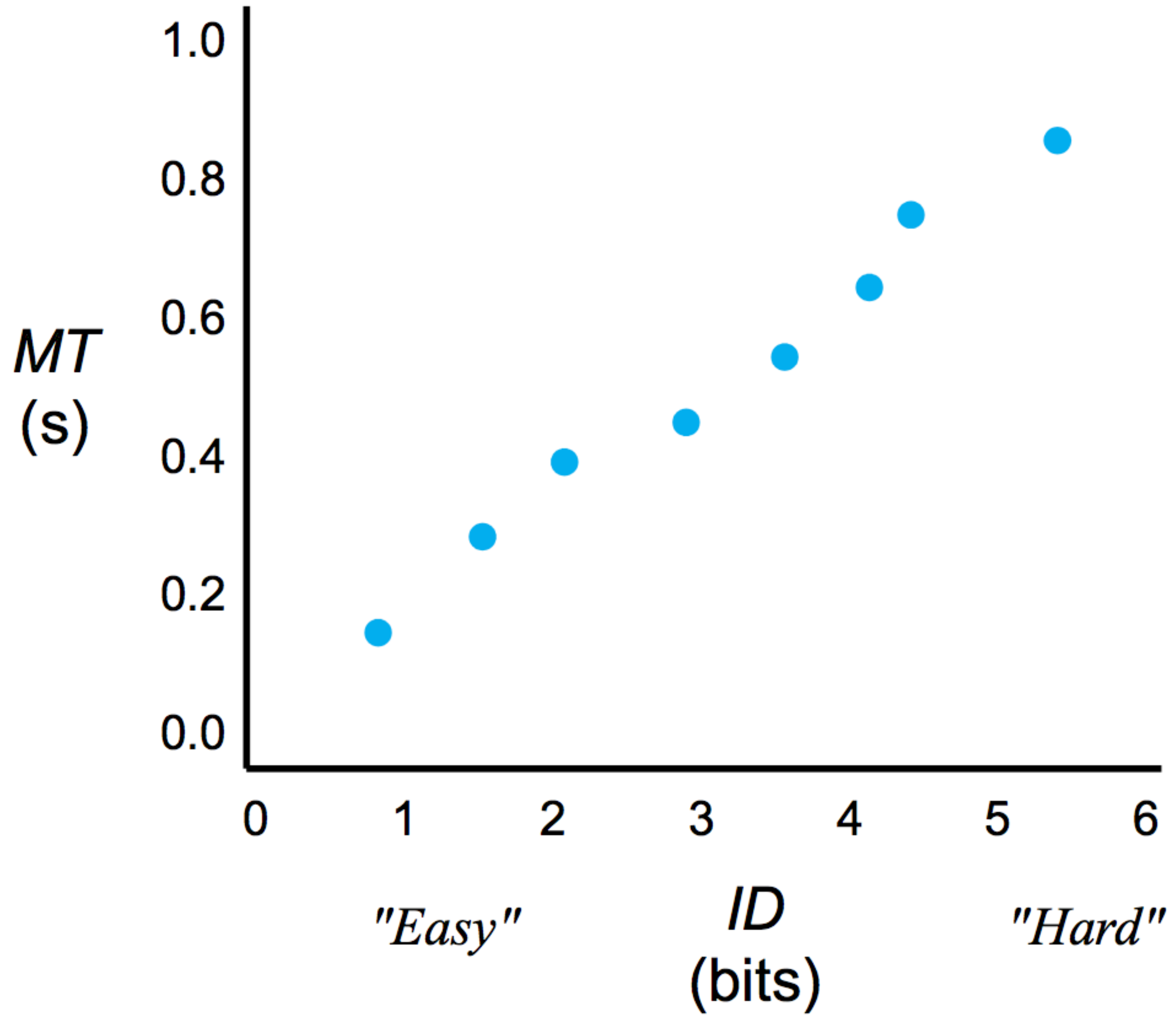
What is "hard" and what is "easy"?



Fitts' law | ID & Movement Time

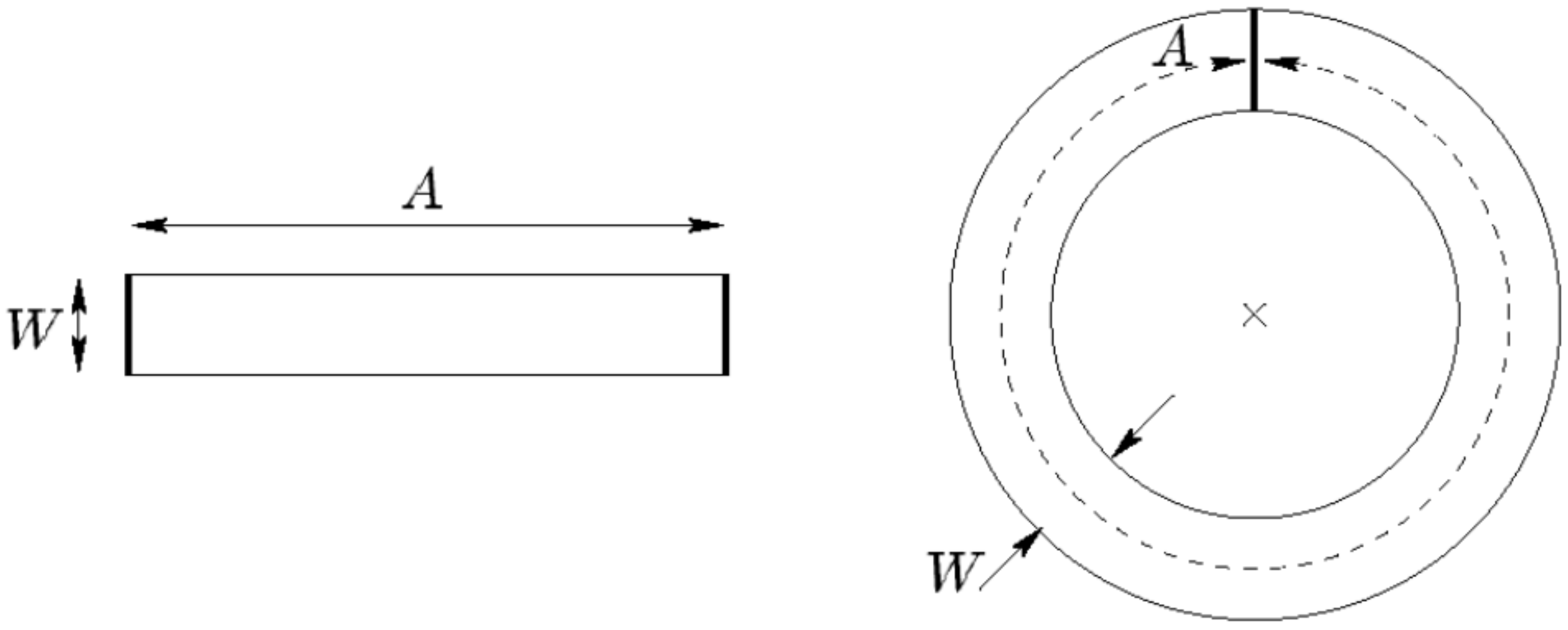
Fitts' thesis:

The ID-MT ratio is relatively constant across a wide range of IDs



Steering law

For path following time

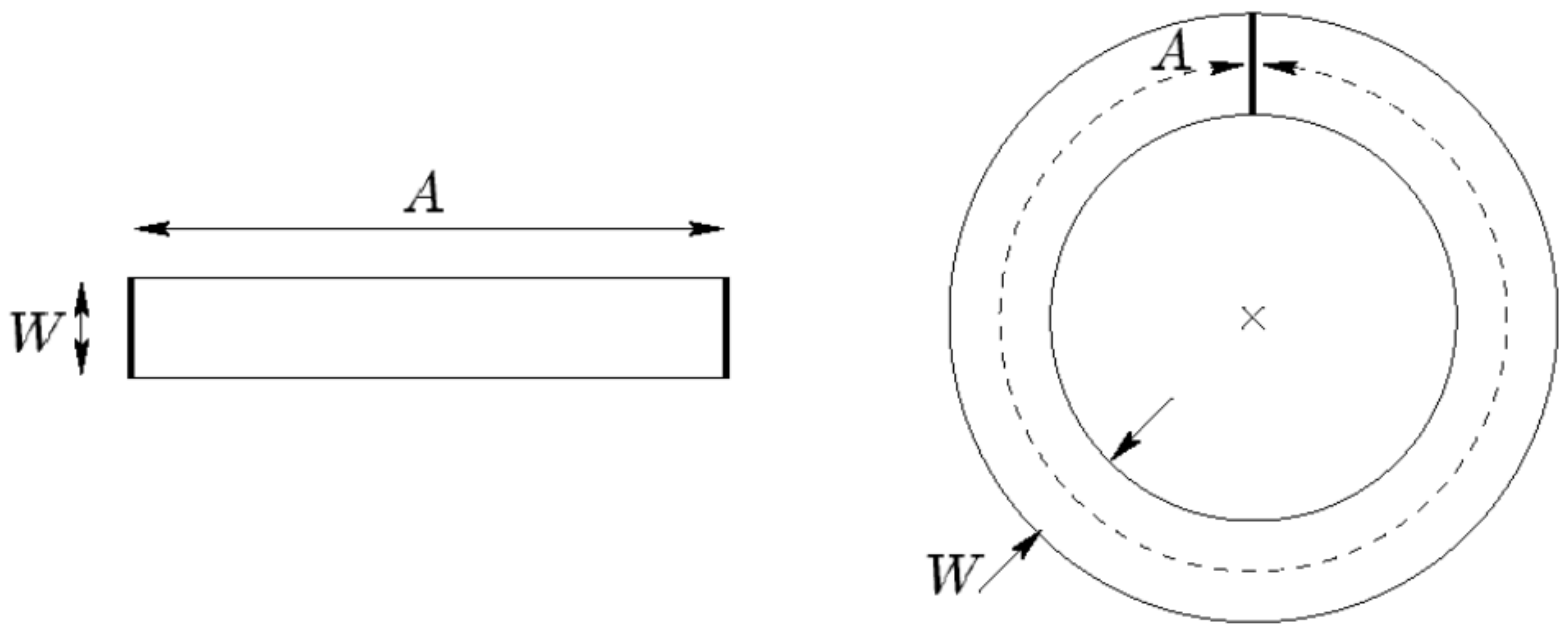


(a) Linear tunnel (b) Circular tunnel

Steering law

For path following time

$$MovementTime = a + b \times (\frac{A}{W})$$



(a) Linear tunnel (b) Circular tunnel

A is the tunnel length (minimum), W is the path width

$\frac{A}{W}$ still called Index of Difficulty

Like Fitts' Law, a and b empirically derived by regression analysis

Hick-Hyman Law

For choice reaction time

Time to select from options when optimally prepared [Hick ’52; Hyman’53]

$ReactionTime = a + b \times log_2(C)$

Where:

- C is the number of equally probable choices
- Otherwise, replace C with $\frac{1}{P_i}$ (item probability)
  - a and b are empirically derived constants

Speed-accuracy trade-offs occur

Response times typically around 160ms per bit (8 choices, 3 bits)

Power law of practice

For how task performance may improve with practice

How task performance speeds up with practice [Newell and Rosenbloom ’81]

$log(T_n) = C - \alpha log(n)$ or $T_n = Cn^{-\alpha}$

where $C = T_1$, task time on first trial, α = learning curve steepness

Applies to simple and relatively complex tasks

In practice, expertise is dependent on previous experience selecting the item (trials, t_i) and on interface learnability (L, from 0 to 1):

$$e_i = L \times (1 - \frac{1}{t_i})$$

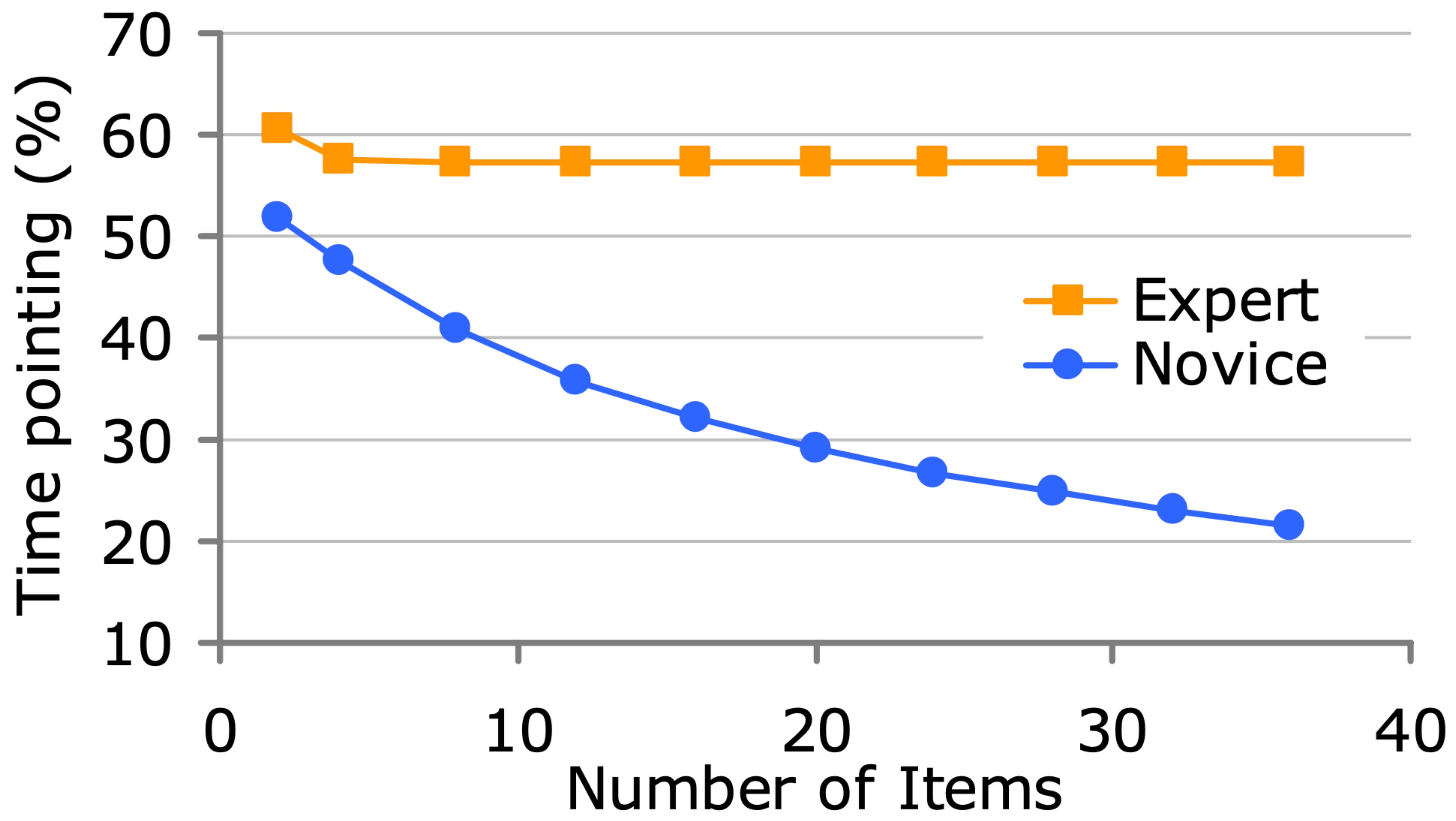
The Search, Decision, and Pointing (SDP) Model

Integrates several low level models to predict performance with 1-level linear menus [Cockburn et al. 07]

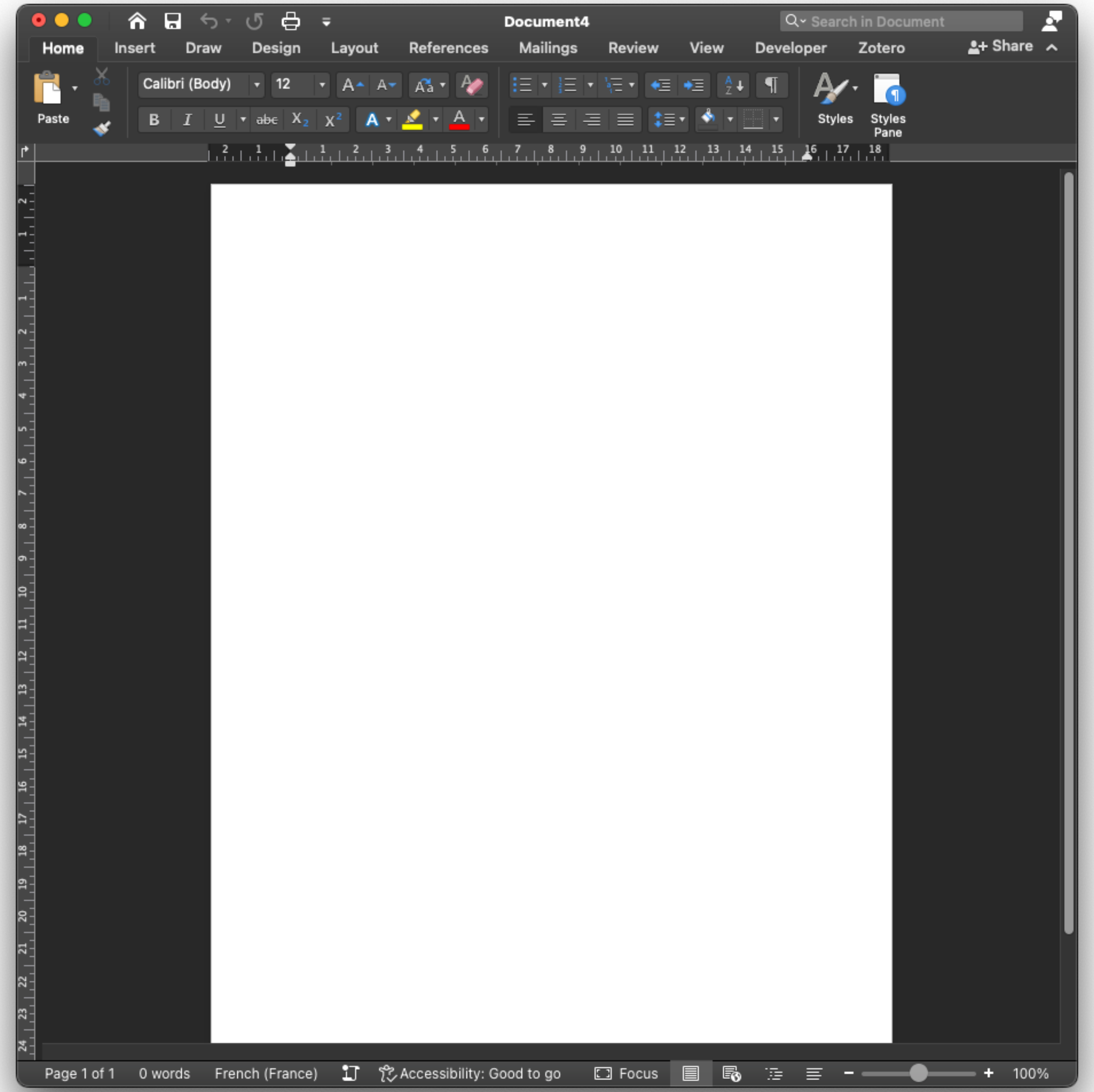
Selecting a command consists in visually locating a target and then selecting it

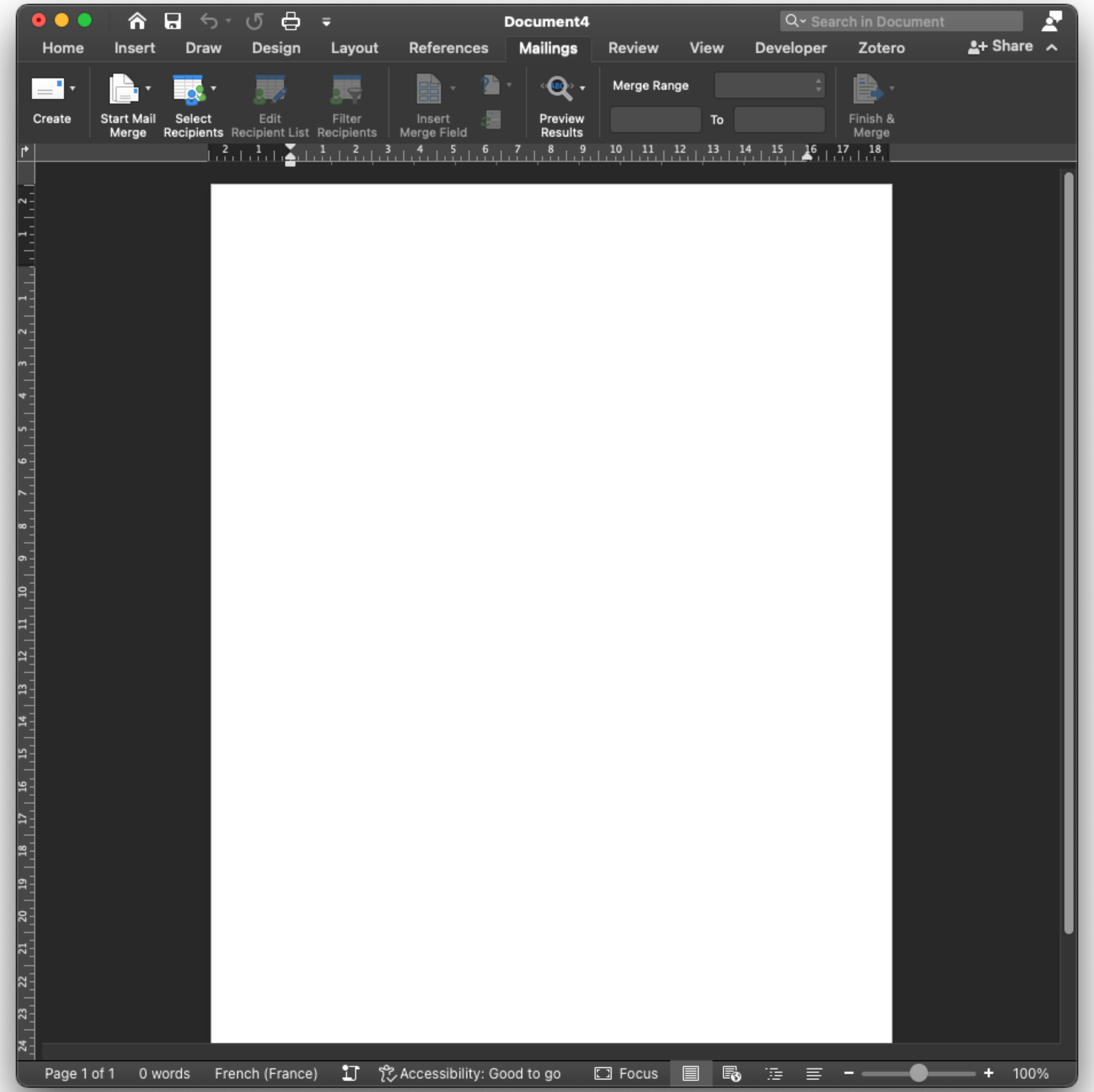
- The average time T_{avg} to select items in a menu is calculated as the probabilistic sum of times T_i for its constituent entries
  - $T_{avg} = \sum_{i=1}^{n} P_i T_i$
- Item selection time T_i is calculated as the sum of the two sub-tasks that involve first finding the item (T_{dsi}) and then acquiring it (T_{pi}). For menus using traditional cursor movement for target acquisition, pointing time T_{pi} is calculated with **Fitts' Law:**
  - $T_i = T_{dsi} + T_{pi}$
- the time to find the item $(T_{dsi} = (1 - e_i)_{vsi} + e_i T_{hhi})$ depends on the user's level of expertise e_i rfrom 0 to 1): novices visually search, experts decide about location, and intermediates do some of both
- Visual search time ($T_{vsi} = b_{vs}n + a_{vs}$) is a linear function of menu length (n), where a_{vs} and b_{vs} are empirically derived intercept and slope values
- Expert decision time (T_{hhi}) is calculated using the **Hick- Hyman Law of choice reaction time**, where a_{hhi} and b_{hhi} are empirically-derived intercept and slope constants, and H is termed the information entropy of the decision
- Item expertise (e_i) is calculated as a power law of practice dependent on previous experience selecting the item (trials, t_i) and on interface learnability (L, from 0 to 1): $e_i = L \times (1 - \frac{1}{t_i})$

The Search, Decision, and Pointing (SDP) Model

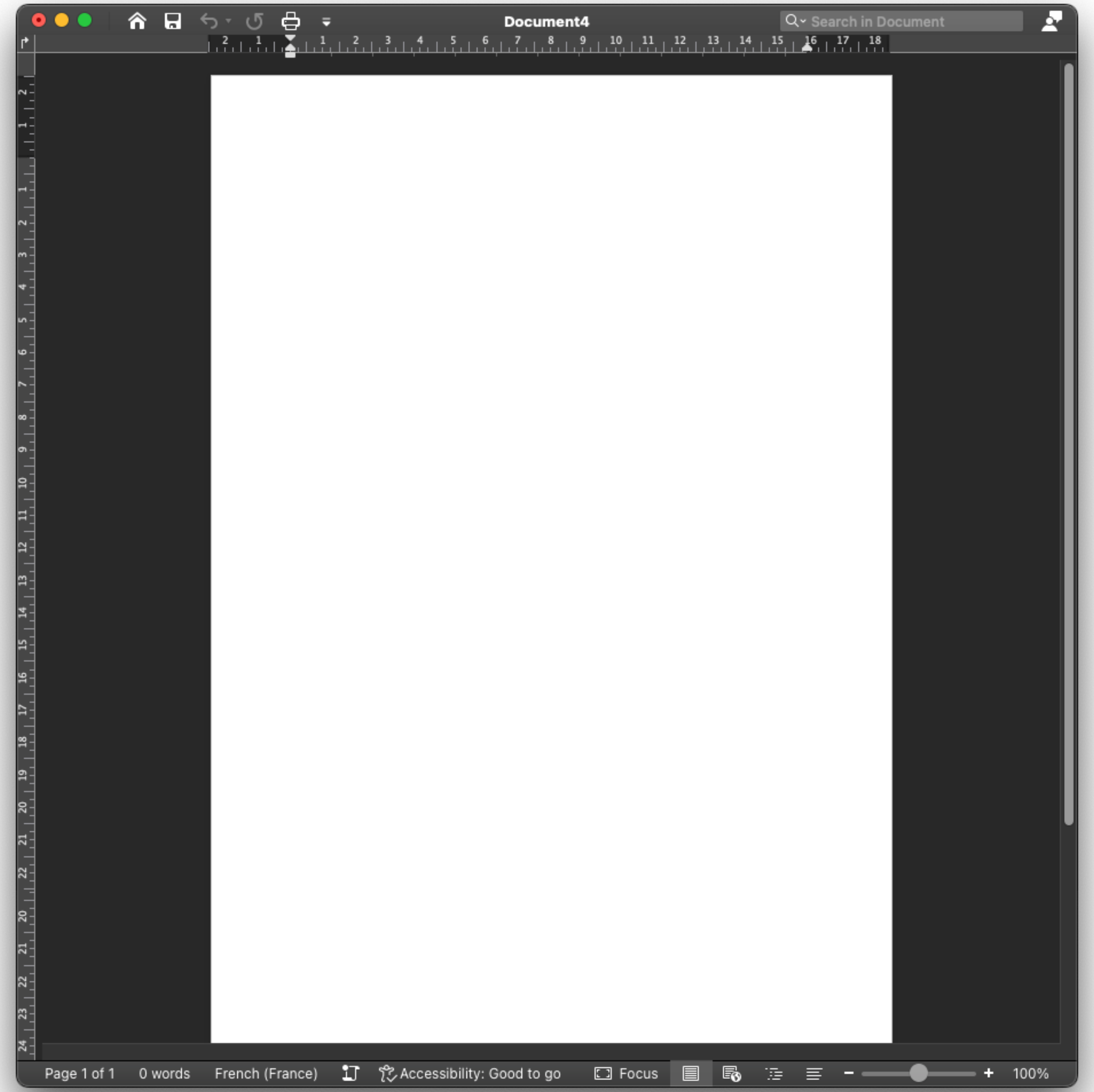


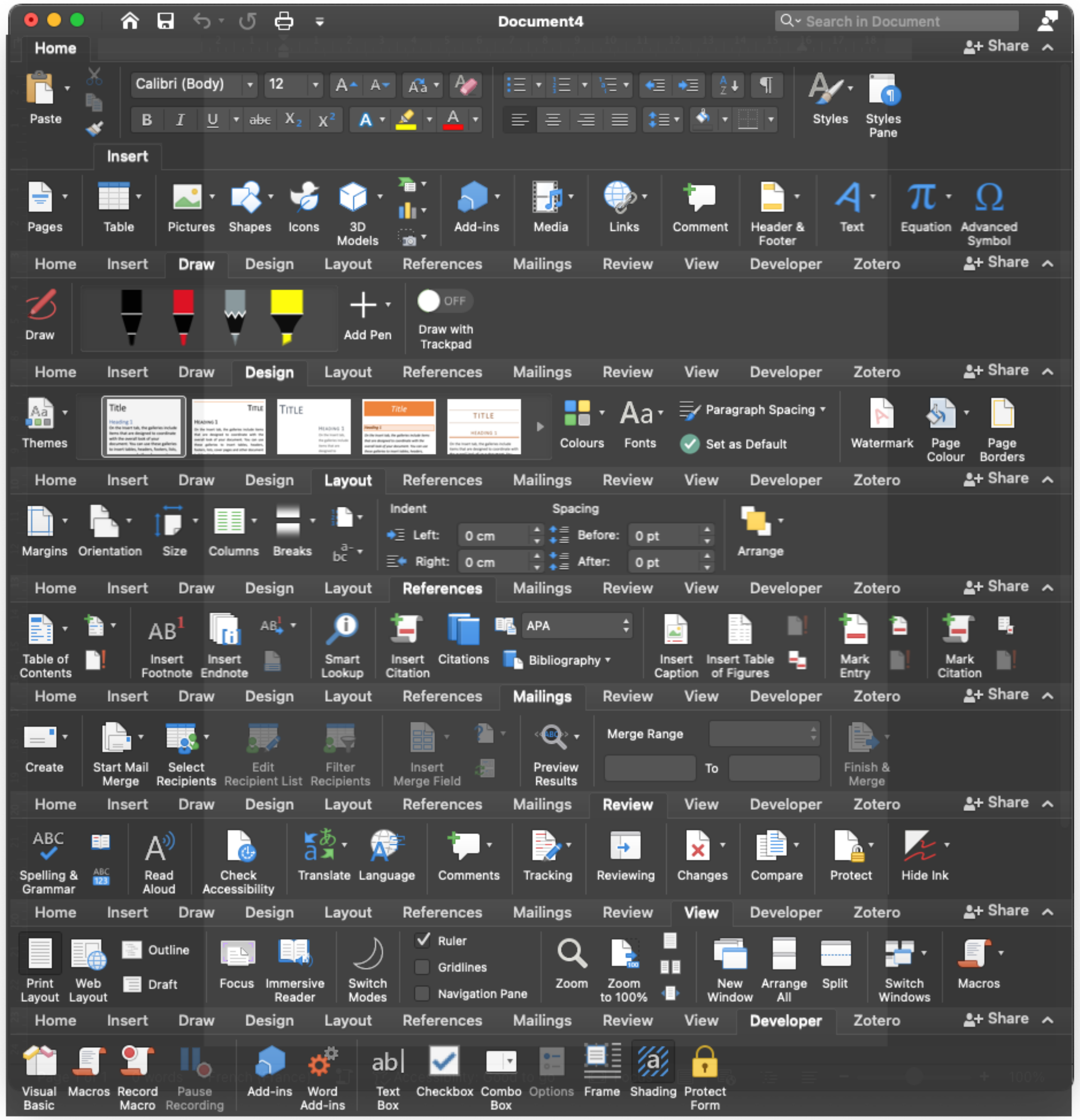
Interaction technique| tab-based command hierarchy



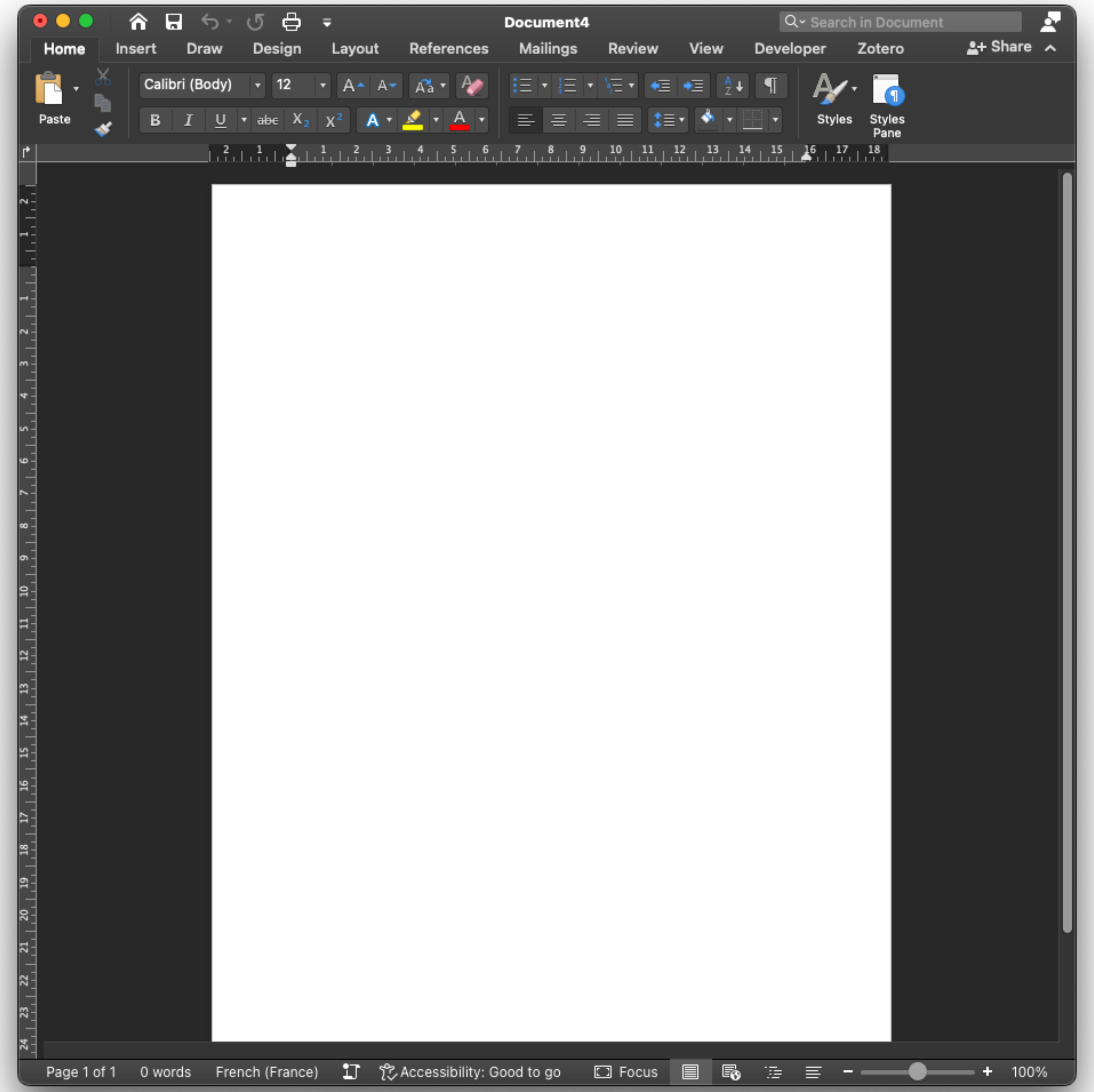


Interaction technique| transient-flat-based hierarchy

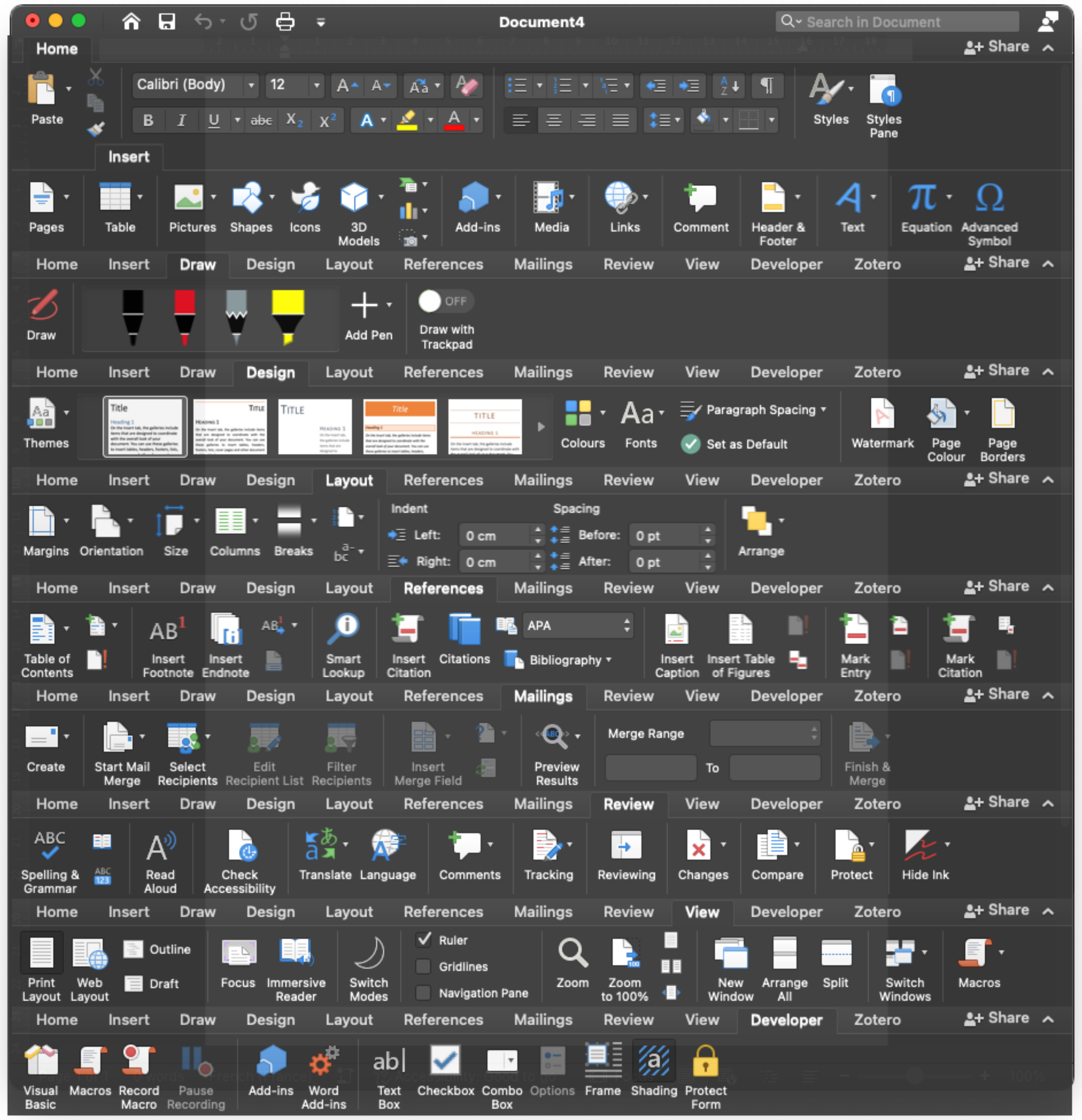




Exercise | two interfaces



tab-based command hierarchy



transient-flat-based hierarchy

Predict time command selection based on proportion of tab switch

Exercise | Goals and methods? =>task modelling

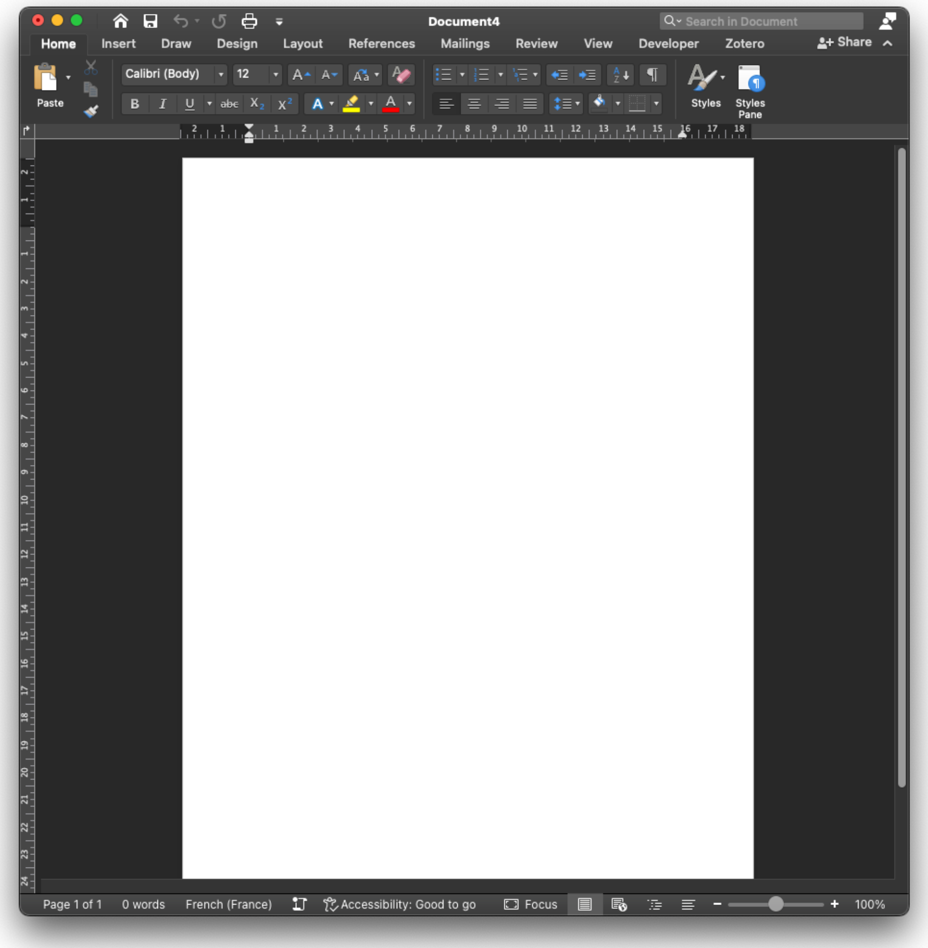
A task consists in:

- a goal (target state);
- a process to reach that goal

A process consists in a sequence of elementary tasks that can be split into physical actions

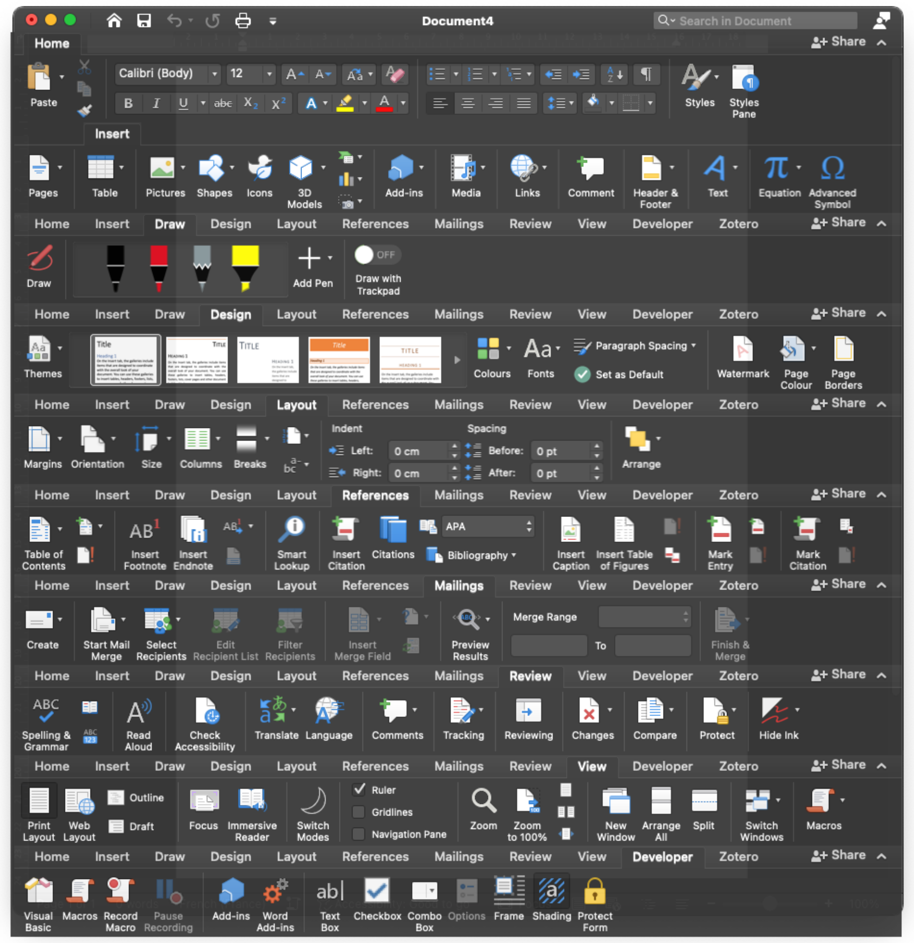
A physical action is an operation on an input/output device that results in a change of state either from the user (perception, decision, reflection) or the system (input actions)

Exercise | two interfaces



tab-based command hierarchy

sometimes necessary to switch tab



transient-flat-based hierarchy

flat-based-transient interface

Q: what are the goals and methods?
Q: how to model unpracticed and practiced performance?
Q: impact of tab switching needed (or not)

Simplify modelling

Make the following assumptions (even though different in TP)

- model *N* commands that are evenly divided across 10 groupings
- command selections begin with the cursor located at the centre of the workspace
- tabs items are *XXXX* pixels wide
- error-free performance