

Apports de la Psychologie Cognitive à la modélisation de l'utilisateur en IHM

MASTER INFORMATIQUE

2ème année, EID et PLS

Catherine Recanati
Université de Paris 13

Modélisation de l'utilisateur

- Caractérisation de son niveau d'expertise
 - Débutant, Confirmé, Expert
- Modélisation de ses processus cognitifs
 - Apprentissage, connaissance, croyances
- Processus psychologiques
 - Importance du nombre magique 7 plus ou moins 2

Apports de la Psychologie Cognitive à la modélisation

- Modèle du processeur humain
 - Trois sous-systèmes
- Modèles GOMS et Keystroke
- La théorie de l'action de Norman
 - comment se fait l'accomplissement d'une tâche
- Modèle de Rasmussen

Modèle du Processeur Humain

Card, Moran, Newell, 1983

"The Psychology of Human-Computer Interaction".

Modèle du processeur humain

Dans ce modèle, le sujet humain est vu comme un système de traitement de l'information qui comporte 3 sous-systèmes interdépendants

- le **système sensoriel**,
- le **système moteur**,
- le **système cognitif**.

Chaque sous-système dispose

- d'un **processeur**
- d'une **mémoire**

Paramètre d'un processeur :

- τ , **cycle de base**, qui inclut le cycle d'accès à la mémoire. En général, ~ 100 ns

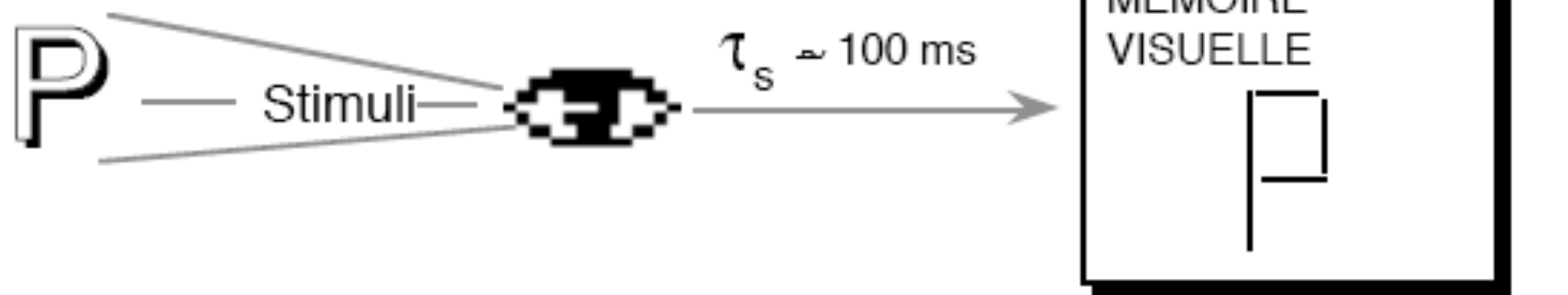
Paramètres d'une mémoire :

- μ , la **capacité** (c'est-à-dire le nombre d'éléments d'information mémorisés),
- δ , la **persistance** (c'est-à-dire le temps au bout duquel la probabilité de retrouver un élément d'information est inférieure à 0.5),
- K , le **type d'informations** mémorisés (physique, symbolique, etc.).

Systeme sensoriel :

- ensemble des sous-systèmes spécialisés chacun dans le traitement d'une classe de stimuli.
- **stimulus** = phénomène physique détectable par un sous-système sensoriel.
- chaque sous-système dispose d'une mémoire spécifique dite **mémoire sensorielle** et d'un mécanisme de traitement intégré
- les stimuli sont codés dans la mémoire sensorielle pour exprimer les propriétés physiques du phénomène.

Le codage de la lettre P traduit les courbures et les dimensions de la lettre mais n'exprime pas sa reconnaissance, qui se concrétisera dans la mémoire à court terme.



- Lorsque, par suite de transferts, la capacité d'absorption de la mémoire à court terme atteint la limite de saturation, les informations des mémoires sensorielles ne sont plus transmises et se dégradent.
- Persistance des mémoires sensorielles :
 - 200 ms pour la mémoire visuelle
 - 1500 ms pour la mémoire auditive.

- Le cycle de base d'un processeur sensoriel est de l'ordre de 100 ms et varie inversement avec l'intensité du stimulus. Ceci signifie qu'il faut en moyenne 100 ms pour qu'un stimulus soit représenté dans une mémoire sensorielle.
- La sensation de percevoir se manifeste plus rapidement lorsque le stimulus est intense. En conséquence, deux événements sensoriels similaires survenant dans le même cycle sont combinés en un seul mais la durée de ce cycle est sensible à l'intensité du stimulus.

Systeme moteur :

- Le système moteur est responsable des mouvements. Dans le cadre de l'interaction homme-ordinateur, les mouvements concernés sont les manipulations des claviers, écrans et des dispositifs de désignation.
- Un mouvement n'est pas continu mais est constitué d'une suite de micromouvements discrets. Chaque micromouvement s'accomplit en moyenne en 70 ms. Ce temps constitue le cycle de base t_m du processeur du système moteur.

- Loi de Fitts

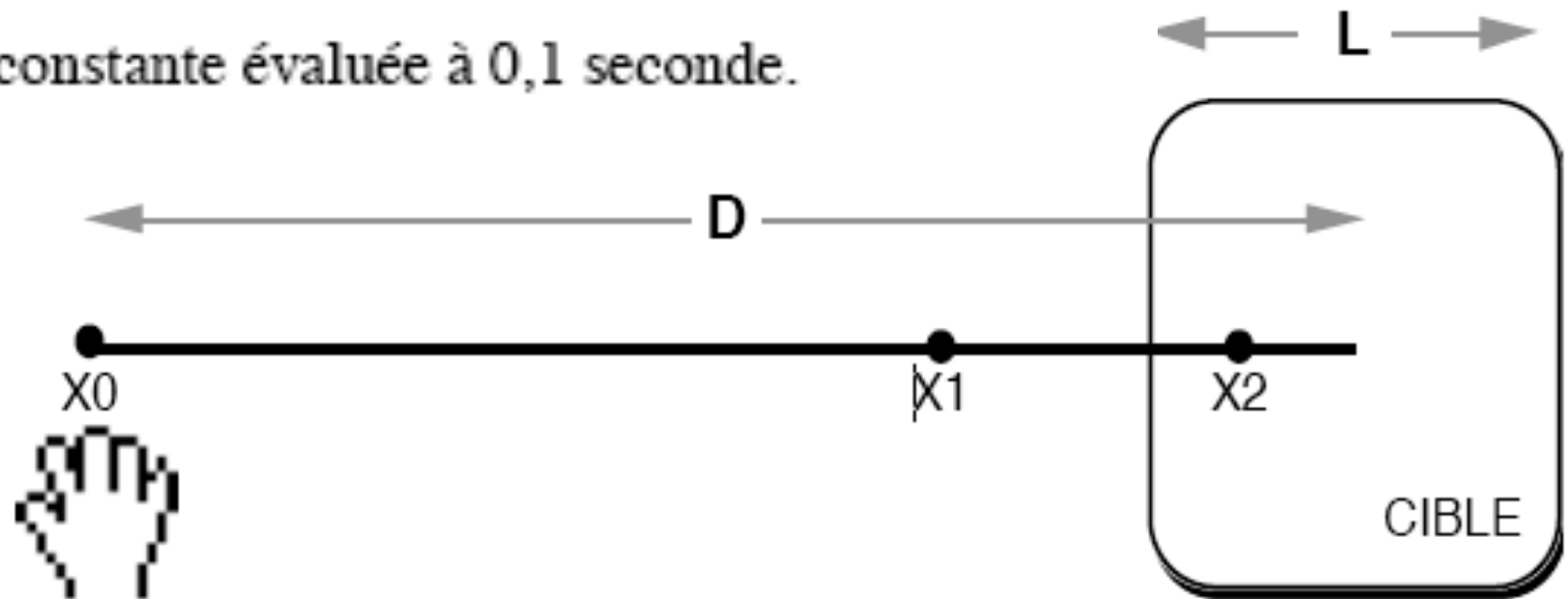
$$T = I \cdot \log_2\left(\frac{2D}{L}\right)$$

où

D est la distance à parcourir par la main,

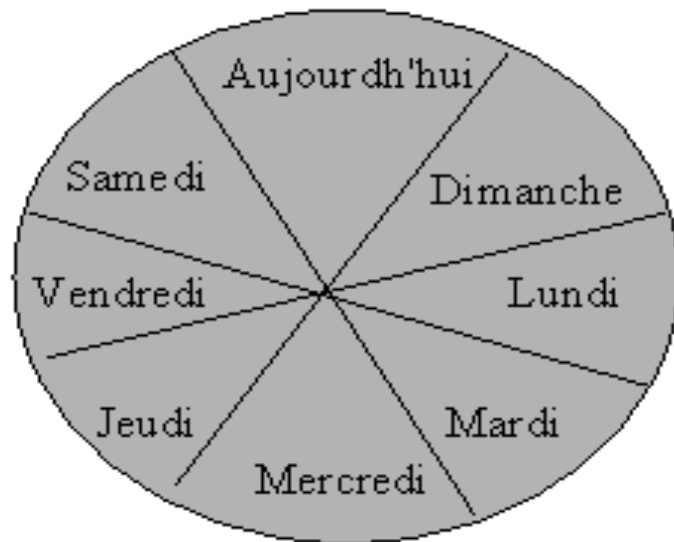
L est la largeur de la cible, et

I une constante évaluée à 0,1 seconde.



Exemple d'utilisation de la loi de Fitts

Menu Camembert



Menu linéaire

Aujourd'hui
Dimanche
Lundi
Mardi
Mercredi
Jeudi
Vendredi
Samedi

Lequel est le plus rapide en moyenne ?

Systeme cognitif :

Le processeur du système cognitif contrôle le comportement de l'individu en fonction du contenu de sa mémoire. Cette mémoire comprend :

- la **mémoire à court terme** (appelée également mémoire de travail) qui détient les informations en cours de manipulation,
- la **mémoire à long terme**. C'est le lieu de stockage des connaissances permanentes.

Mémoire à court terme

- **semblable aux registres d'un calculateur** : elle contient les opérandes d'entrée et les résultats intermédiaires des traitements en cours.

Les opérandes proviennent des mémoires sensorielles et/ou de la mémoire à long terme.

- **Les informations d'origine sensorielle sont représentées sous forme symbolique.**

Elles ne sont plus affectées des caractéristiques physiques.

ex: la représentation de P dans la mémoire à court terme traduit le fait qu'il s'agit de la lettre P.

Mémoire à court terme

- Les informations en provenance de la mémoire à long terme sont des **mnèmes** (en anglais, "chunks") activés par le processeur cognitif.
- Un mnème est une unité cognitive symbolique, une abstraction qui peut être associée à d'autres unités. ex: “ S, N, C, F ”= 1 ou 4 mnèmes.
- mnème activé = disponible dans la mémoire à court terme. Les activations se propagent.
- La capacité estimée à 7 +/- 2 mnèmes (Miller 75)
- Saturation => conflit et interférence. Les éléments inutilisés se dégradent.

Mémoire à long terme

Rôle analogue aux mémoires centrales et secondaires d'un ordinateur :

- contient l'information de masse
- peut être lue ou modifiée
- son contenu est un réseau sémantique de mnèmes [qui représentent des procédures et des données appelées respectivement connaissance procédurale (ou savoir-faire) et connaissance factuelle (ou connaissance de faits)].

Mémoire à long terme

- **opération de lecture = rechercher un mnème.** Le succès de cette recherche transfère le mnème dans la mémoire à court terme avec un degré d'activation donné. L'échec a deux causes principales : soit aucune association n'est trouvée, soit plusieurs mnèmes interfèrent avec le mnème cible.
- **la persistance** des informations dans la mémoire à long terme **est infinie.**

Devinettes

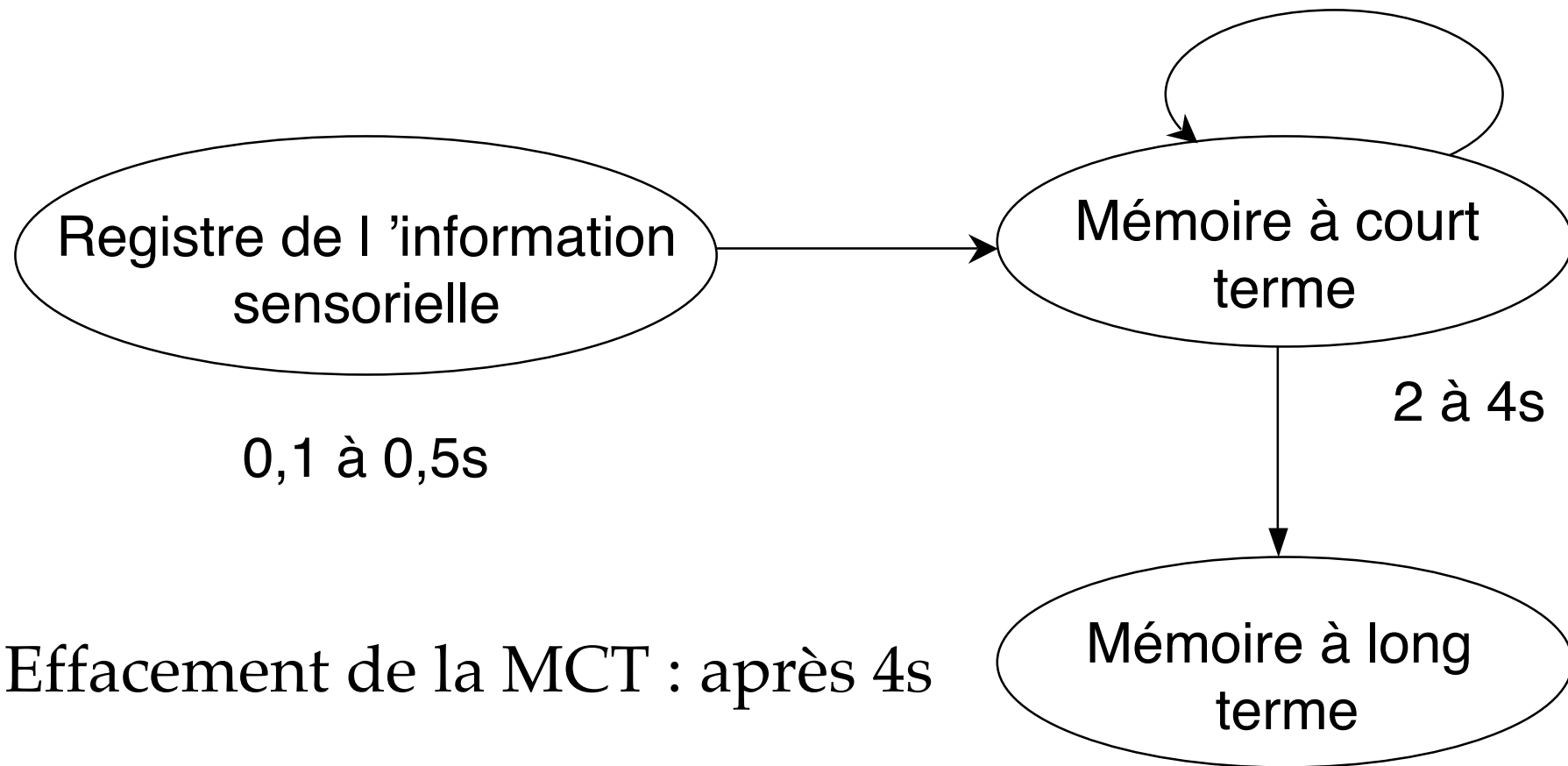
- Quelle est la couleur du cheval blanc d'Henri IV ?
- Quelle est la couleur de la neige ?
- Que boit la vache ?

Mémoire à long terme

- Pour être inscrit dans la mémoire à long terme, un mnème de la mémoire à court terme doit être associé, selon des critères définis par l'individu, à un ou plusieurs mnèmes de la mémoire à long terme.
- Les chances de retrouver un mnème croissent avec le nombre d'associations discriminantes. Elles augmentent aussi avec le temps disponible pour effectuer ces associations.
- La capacité de la mémoire à long terme n'est pas limitée.

Mémoire à court et à long terme

Auto répétition de maintien



Processeur cognitif

- Cycle "Reconnaissance-Action" (analogue au cycle "Recherche-Exécution" des calculateurs usuels)
 - Le système cognitif reçoit des informations « symboliques » de la mémoire à court terme
 - Le système cognitif utilise les informations stockés dans la mémoire à long terme pour prendre des décisions d'actions et formuler une réponse
 - Les actions modifient le contenu de la mémoire à court terme
- cycle de base : 70 ms

Processeur cognitif

- Le fonctionnement du processeur cognitif est calqué sur le modèle des systèmes de production. Il opère selon le cycle "Reconnaissance-Action".
- Dans la phase de Reconnaissance, le processeur détermine les actions de la mémoire à long terme associées aux mnèmes de la mémoire à court terme.

Processeur cognitif (suite)

- Dans la seconde phase, l'exécution, ces actions sont effectuées, provoquant une modification du contenu de la mémoire à court terme.
- Le cycle de base du processeur cognitif est de l'ordre de 70 ms.

Techniques pour favoriser la mémorisation à long terme

- Reformuler l'information
- Ajouter du sens (raconter une histoire)
- Imagination visuelle (techniques des sophistes)
- Organiser (créer un mnème)
- Faire des liens avec des connaissances existantes (catégories)

Exemple de perceptions

- Expérimentation
 - 1 volontaire SVP !
- Dire à haute voix la couleur des mots dans la liste des transparents suivants aussi vite que possible
- Dire « stop » quand c'est fini.

Voler

Agenda

Papier

Maison

Page

Modifier



Bleu

Jaune

Rouge

Blanc

Vert

Rouge

Autre exemple

- "Sloen une rhceerche mneée dnas une usiniervté aglanise, l'odrre des ltrtees dnas un mot ne snot pas fnometadanl puor la cpremoension
- des mtos, ce qui est ipormatnt c'est que la pemrière et la dreinère letrte du mot syeont dnas les pnotisios croretecs. Les ltreetes du mleilu pveeunt etre cleommppteent ierenvsés.
- Si le lteecur arirve a lrie les mtos ce prace que nuos ne lsonis pas cquhae ltrete seemparent mias le mot eientr!"

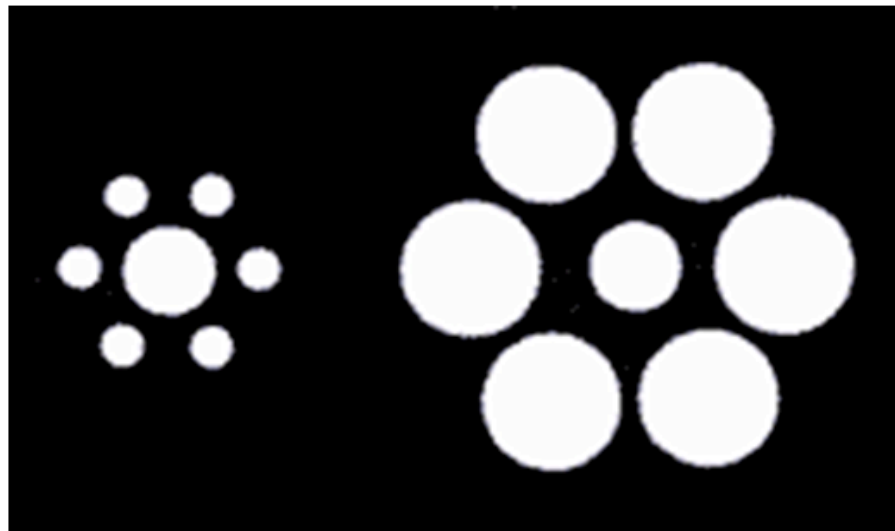
Exemple

<u>F</u> ile	<u>E</u> dit	<u>R</u> eports
<u>N</u> ew...		Ctrl+N
<u>O</u> pen...		Ctrl+O
<u>C</u> lose		Ctrl+W
<u>S</u> ave		Ctrl+S
Save <u>A</u> s...		Shft+Ctrl+S
<u>F</u> ind File		Ctrl+F
Summary <u>I</u> nfo		Ctrl+I
<u>P</u> rint...		Ctrl+P ▶
<u>E</u> xit		Ctrl+Q

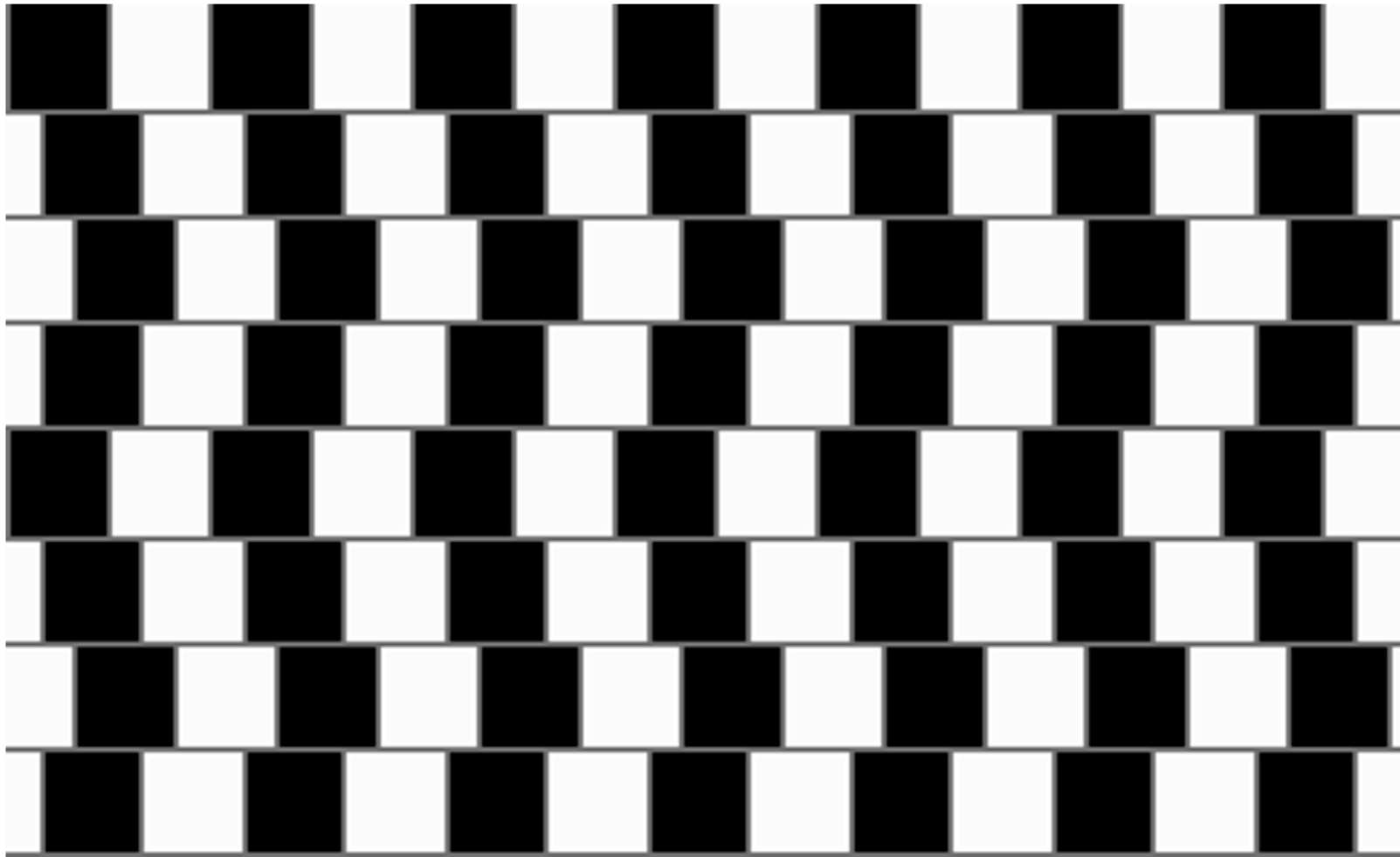
Avec séparateurs

<u>F</u> ile	<u>E</u> dit	<u>R</u> eports
<u>N</u> ew...		Ctrl+N
<u>O</u> pen...		Ctrl+O
<u>C</u> lose		Ctrl+W
<u>S</u> ave		Ctrl+S
Save <u>A</u> s...		Shft+Ctrl+S
<u>F</u> ind File		Ctrl+F
Summary <u>I</u> nfo		Ctrl+I
<u>P</u> rint...		Ctrl+P ▶
<u>E</u> xit		Ctrl+Q

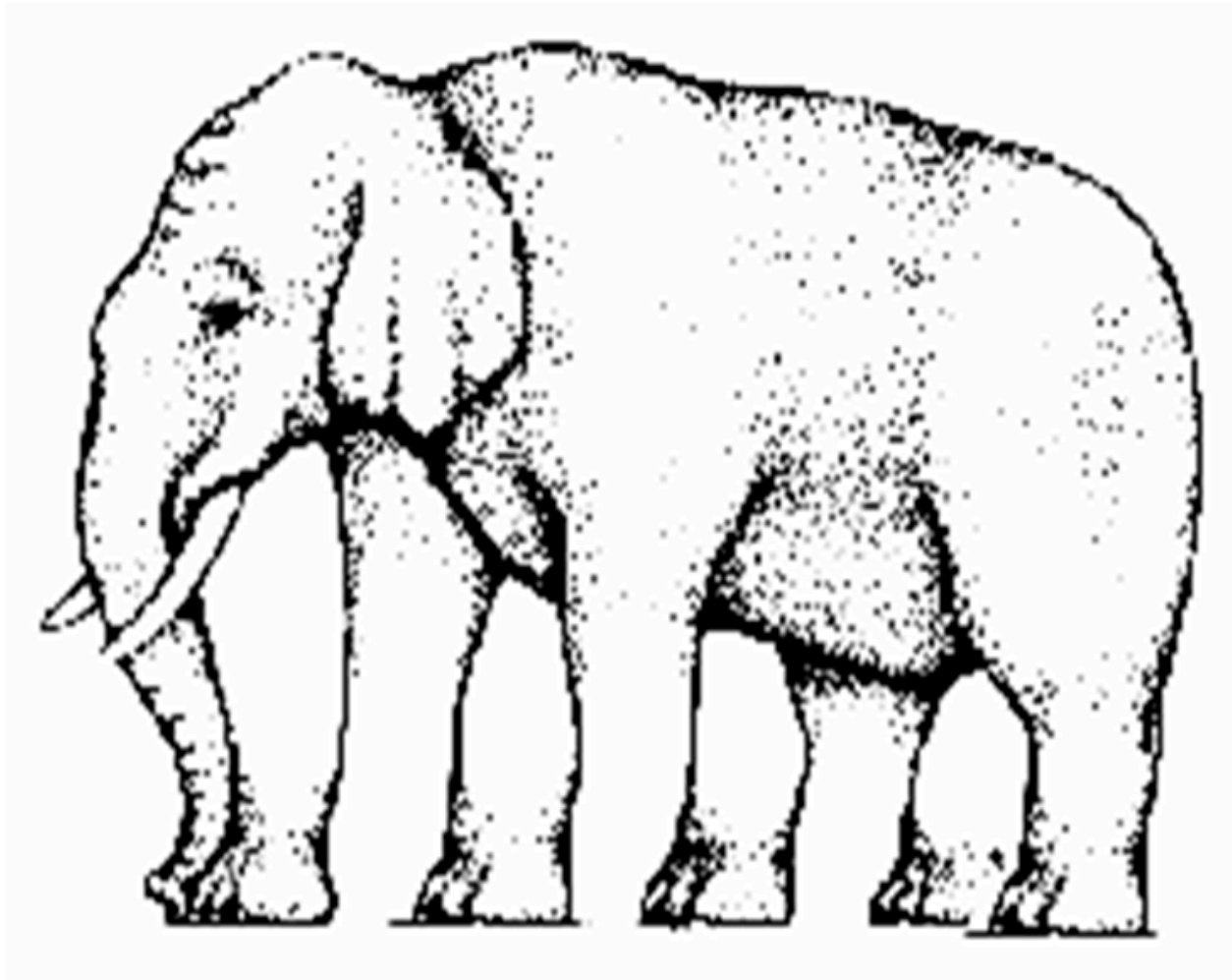
sans séparateurs



Entre les 2 figures, quel est le cercle central le plus grand ?



Les lignes « horizontales » sont-elles parallèles ou penchées ?



Combien de pieds à cet éléphant ?

Modèle du processeur humain

- Principes opératoires
- Limites du modèle

Principes opératoires

- **Principe du fonctionnement cyclique du processeur cognitif.**

Le système cognitif procède selon le cycle "Reconnaissance-Action".

- **Principe de Discrimination.**

La difficulté de retrouver une information est liée au nombre de candidats répondant aux mêmes indicateurs d'accès.

Principes opératoires (suite)

- **Principe de Rationalité.**

Pour atteindre un objectif, l'individu agit de manière rationnelle. Le comportement rationnel d'un individu résulte de l'union des ensembles : *buts* à atteindre, *structure de la tâche* à réaliser, *connaissances*.

- **Loi de Fitts.**

Le temps T pour placer la main sur une cible dépend uniquement de la précision requise, c'est-à-dire du rapport entre la distance et la dimension de la cible.

Exemples d'application du modèle

1) Le cycle du processeur sensoriel et le rafraîchissement de l'écran

En choisissant pour la valeur moyenne d'un cycle 100ms, le système répond aux conditions s'il est capable de produire au moins 10 images/s. ex: Rectangle et événement souris.

2) Loi de Fitts et conception de la souris du Star

3) Les limites de la mémoire cognitive

Si un utilisateur lit une suite d'informations arbitraires, comme une quinzaine de noms de commandes dans une langue inconnue, comment va-t-il se comporter sur le plan mémorisation juste après avoir cessé de consulter la liste ?

La réponse est liée à la capacité de la mémoire à court terme (7 ± 2 mnèmes) et aux temps de persistance moyens des mémoires : 200 ms pour la mémoire sensorielle, 7 s pour la mémoire à court terme, et l'infini pour la mémoire à long terme.

Puisque la langue est inconnue, chaque mot constitue un mnème. La liste comportant plus de 7 mnèmes, l'utilisateur va commettre des oublis, et les mots oubliés seront ceux du milieu de la liste.

Les premiers mots présentés, s'ils ont été mémorisés, sont accessibles depuis la mémoire à long terme car, au-delà de 200ms, ou bien l'information est perdue, ou bien elle est disponible dans la mémoire à long terme.

Les derniers éléments de la liste, s'ils ont été transférés depuis la mémoire sensorielle, sont encore présents dans la mémoire à court terme.

Ce sont donc les éléments du milieu qui seront oubliés.

Évaluation du modèle du processeur humain

- Cadre fédérateur à diverses connaissances de psychologie
- Terminologie informatique
- Tentative de "psychologie appliquée"
- Niveau d'abstraction inadapté
- S'intéresse aux performances (motrices et perceptuelles) mais pas à la cognition
- N'indique aucune méthode de conception

Limites du modèle

- Ce modèle concerne les performances motrices et perceptuelles mais ne dit rien des structures cognitives du sujet humain. Quid de la conceptualisation et la reconstruction mnésiques qui interviennent dans l'apprentissage et la résolution de problèmes, la capacité à traiter plusieurs niveaux d'interruption, le phénomène du parallélisme et les erreurs ?
- Le Modèle du Processeur Humain n'implique pas de méthode de conception. Aucun élément du modèle n'indique comment satisfaire les contraintes de performance qu'il permet de déduire. Les modèles GOMS et Keystroke sont des dérivés du modèle du processeur humain qui tentent de combler cette lacune.

GOMS

Card, Moran, Newell, 1983

"The Psychology of Human-Computer Interaction".

GOMS

Selon le **Principe de rationalité**, un individu s'efforce de s'adapter aux conditions de la tâche qu'il s'est fixé. Cela signifie que **le comportement est conditionné par l'environnement**.

L'idée selon laquelle la complexité du comportement n'est pas due à la complexité interne de l'individu mais à celle de l'environnement, vient de Herbert Simon (84, ex. fourmi et sable).

ex: Traitement de texte avec Emacs "Ctrl-@ Esc-f Esc-f Ctrlw Ctrl-y Ctrl-n Ctrl-n Ctrl-n Ctrl-y" ou MacWrite.

GOMS

GOMS permet de modéliser le comportement à différents niveaux d'abstraction, depuis la *tâche* jusqu'aux *actions physiques*.

GOMS = (**G**oal, **O**perator, **M**ethod, **S**election)
introduit quatre ensembles pour représenter l'activité cognitive d'un individu engagé dans la réalisation d'une tâche : les Buts, les Opérateurs, les Méthodes et les règles de Sélection.

Buts (Goals)

- But = structure symbolique qui définit un état recherché. Il lui est associé un ensemble de méthodes qui conduisent à cet état. En cas d'échec, il constitue un point de reprise à partir duquel il est possible d'amorcer d'autres tentatives.
- Les buts sont organisés de manière hiérarchique : un but complexe est atteint lorsque plusieurs sous-buts sont satisfaits
- Les buts élémentaires sont réalisés par l'exécution d'une suite d'opérateurs.
- Les buts forment donc une structure arborescente dont les feuilles sont des opérateurs.

Opérateurs

- Un opérateur est une action élémentaire dont l'exécution provoque un changement d'état (état mental de l'utilisateur et/ou état de l'environnement).
- Analogie entre l'ensemble des opérateurs et celui des instructions d'une machine abstraite. Un opérateur se caractérise par des opérands d'entrée et de sortie et par le temps nécessaire à son exécution. Lorsque l'analyse est fine, l'opérateur reflète des mécanismes psychologiques élémentaires (sensoriels, moteurs ou cognitifs). Lorsqu'elle s'effectue à un niveau d'abstraction élevé, les opérateurs sont des unités d'action spécifiques à l'environnement (par exemple les commandes du système).

Méthodes

- Une méthode décrit le procédé qui permet d'atteindre un but.
- Elle s'exprime sous la forme d'une suite conditionnelle de buts et d'opérateurs où les conditions font référence au contenu de la mémoire à court terme et à l'état de l'environnement.
- Les méthodes représentent un savoir-faire : elles constituent la **connaissance procédurale**. Elles ne sont pas des plans d'action construits dynamiquement pendant l'accomplissement de la tâche. Elles sont le résultat de l'expérience acquise.

Exemple

Par exemple, pour un éditeur pleine-page qui permet à la fois les commandes à la Emacs et les commandes à la Macwrite avec la souris. Le choix entre les méthodes clavier ou souris peut s'exprimer en fonction de la distance entre la position actuelle du curseur et la localisation visée :

si le but à atteindre est placer le curseur au bas de la fenêtre

et

si la position actuelle du curseur est loin du bas de la fenêtre

alors utiliser la méthode M1;

Exemple (suite)

si le but à atteindre est placer le curseur au bas de la fenêtre

et

si la position actuelle du curseur est près du bas de la fenêtre

alors utiliser la méthode M2;

avec M1: prendre la souris; déplacer la souris au point désiré; sélectionner;

et M2: tant que le curseur n'est pas sur la ligne désirée taper ctrl-n; tant que le curseur n'est pas au point désiré taper esc-f;

Card, Moran et Newell définissent quatre niveaux d'analyse : les niveaux tâche, fonctionnel, argument et physique.

- *Le niveau tâche* structure l'espace de travail en une **hiérarchie de sous-tâches** dont la nature dépend uniquement du domaine. Les éléments terminaux de la décomposition sont des **tâches conceptuelles élémentaires**.
- *L'analyse fonctionnelle* modélise les tâches élémentaires en termes de fonctions du système. A ce niveau de modélisation, **l'accomplissement d'une tâche est décrit par une suite de fonctions**.

- *Le niveau argument* précise, pour chaque fonction, sa réalisation par une suite de commandes. A ce niveau de modélisation, **l'accomplissement d'une tâche est décrit par une suite de commandes.**
- *Le niveau physique* décrit en termes **d'actions physiques** la spécification des commandes. Keystroke est une illustration de cette dernière classe de modélisation.

Apports de GOMS

- GOMS véhicule une **méthode de conception compatible avec celle des informaticiens**. La modélisation d'une tâche peut être raffinée ou, au contraire, élaborée à partir de constituants élémentaires.
- GOMS fournit un support formel pour des **évaluations prédictives de performance**. En effet, la description d'une tâche donnée définit la suite des opérateurs que l'utilisateur va employer pour la réaliser. Connaissant le temps d'exécution de chaque opérateur, il est possible de prédire le temps nécessaire à la réalisation de la tâche.

Limites

- GOMS n'offre pas de support théorique d'aide à la structuration d'une tâche. Nous avons vu que l'informaticien retrouve dans GOMS le repère familier de l'analyse descendante et ascendante. Mais ici, le sujet d'analyse n'est pas un programme mais une tâche, notion que l'informaticien n'a pas l'habitude de manipuler.
- La réussite d'une analyse de tâche suppose la connaissance approfondie des mécanismes de représentation mentale. GOMS n'offre aucun support théorique dans ce sens : il est un modèle prédictif et quantitatif de performance.

Limites

- Avec GOMS, le phénomène observé est l'accomplissement de tâches de routine réalisées sans la moindre erreur. Or, l'erreur est inévitable et le traitement des erreurs est un casse-tête, y compris dans le cas simple des systèmes déterministes.
- Dans le cas du sujet humain, le traitement d'une erreur peut se voir comme la réalisation d'une tâche particulière. S'il s'agit d'une tâche de routine, alors il lui correspond un plan qui peut être greffé sur l'arbre de résolution. La question qui se pose maintenant est le lieu d'insertion du sous-plan. A ce problème, GOMS n'apporte aucun élément de réponse.

Keystroke

Card, Moran, Newell, 1983

"The Psychology of Human-Computer Interaction".

Keystroke

Formulation du Pb résolu par Keystroke :

Etant donné

- une tâche (constituée éventuellement de plusieurs sous-tâches),
- le langage de commande du système,
- les paramètres caractéristiques des capacités motrices de l'utilisateur,
- les paramètres des temps de réponse du système, et
- la méthode de réalisation de la tâche,

prédire le temps d'exécution de cette tâche par un utilisateur expert.

- Comme dans GOMS, Keystroke s'intéresse aux performances sans erreur.
- Contrairement à GOMS, Keystroke ne prédit pas de choix de méthode : celle-ci est donnée.
- Seconde restriction par rapport à GOMS, Keystroke évalue le temps d'exécution, non pas le temps total d'accomplissement d'une tâche. Le temps d'accomplissement d'une tâche est la somme du temps d'acquisition et du temps d'exécution. Pendant l'acquisition, l'utilisateur construit une représentation mentale de la tâche.

- L'exécution est la réalisation effective physique de la tâche. Temps d'acquisition et temps d'exécution sont considérés indépendants. Keystroke s'intéresse uniquement au temps d'exécution.
- Puisque Keystroke se situe au niveau lexical et que la méthode de réalisation de la tâche est donnée, les notions de buts et de règles de GOMS deviennent inutiles.
- Keystroke utilise donc seulement deux ensembles d'entités : les opérateurs et les méthodes.

Opérateurs

Keystroke introduit six opérateurs pour décrire l'exécution d'une tâche élémentaire :

- K ("Keystroking", frappe de touches du clavier ou de la souris),
- P ("Pointing", désignation),
- H ("Homing", rapatriement de la main),
- D ("Drawing", action de dessiner), et
- M ("Mental activity", activité mentale).
- R ("Response time", temps de réponse du système)

Le temps d'exécution d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

$$T_{\text{exec}} = T_K + T_P + T_H + T_D + T_M + T_R$$

- L'opérateur K représente la frappe d'une touche du clavier ou l'acte d'appuyer sur un bouton de la souris (ou tout autre dispositif de désignation).

on peut estimer T_K sur une population à partir de tests avec

$$T_K = (\text{durée totale des tests}) / (\text{nombre touches frappées sans erreur})$$

- L'opérateur P représente le déplacement du curseur de la souris vers une cible. Une variante de la loi de Fitts donne le temps de saisie

$$T = K_0 + I \log_2(D/L + 0.5) \text{ secondes}$$

où K_0 , est une constante qui tient compte du temps nécessaire pour ajuster la saisie initiale de la souris et pour appuyer sur un bouton de sélection. Les mesures expérimentales de Card, Moran et Newell (83) fournissent $K_0 = 1.03 \text{ s}$

- I est une constante évaluée à : $I = 0.1 \text{ s}$,
- D est la distance entre la position actuelle de la souris et celle de la cible,
- L est la largeur de la cible.

et on en déduit $T_P = (K_0 - T_K) + I \log_2(D/L + 0.5)$. En remplaçant les constantes et temps par leurs valeurs, on obtient pour T_P

borne inférieure : $T_P = 0.8$ s,

borne supérieure : $T_P = 1.5$ s (avec $D/L = 128$), et

valeur moyenne : $T_P = 1.1$ s

- L'opérateur H représente les aspects pragmatiques de l'interaction homme-machine, en particulier le changement d'utilisation d'un dispositif physique.

$$T_H = 0.4 \text{ s (d'après Card 83)}$$

- L'opérateur D représente l'utilisation de la souris pour construire un dessin sur l'écran. Avec des tests de tracé de segments de droite, Card, Moran et Newell ont obtenu :

$$T_D = 0.9 n + 0.16 l \quad \text{où } n = \text{nb de segments,} \\ \text{et } l = \text{somme de leurs longueurs}$$

- L'opérateur M représente l'activité mentale dont l'individu a besoin pour se préparer à exécuter un opérateur physique K, P, H ou D. Card, Moran et Newell ont simplifié la situation et proposé une seule valeur

$$T_M = 1.35 \text{ s}$$

- L'opérateur R a trait aux temps de traitement des commandes par le système. T_R est le temps pendant lequel le système fait attendre l'utilisateur.

Si n est le temps de traitement d'une commande par le système, et si t est le temps exploité par l'utilisateur pour exécuter un opérateur pendant le traitement de la commande, T_R vaut :

- $T_R = 0$ si $n \leq t$
- $T_R = n - t$ si $n > t$

Codage des méthodes

Une méthode s'exprime sous la forme d'une suite d'opérateurs. Par exemple, pour entrer la commande unix **ls** au clavier, la méthode correspondante s'écrit :

```
M K[l] K[s] K[retour-chariot]
```

ou, de manière plus condensée :

```
M 3K[l s retour-chariot].
```

Si maintenant, la commande **ls** est spécifiée avec la souris, la méthode devient :

```
H[souris] M P[souris] K[bouton-souris] H[clavier]
```

- Les occurrences de M (= activité mentale) dans une méthode dépendent des opérateurs physiques et du savoir-faire de l'utilisateur. Le savoir-faire est une donnée spécifique à chaque utilisateur. Pour modéliser cette spécificité aux caractéristiques imprécises, Card, Moran et Newell utilisent des règles heuristiques.
- La définition de ces règles s'appuie sur la théorie suivante (confirmée par des observations expérimentales) : l'utilisateur tend à partitionner une méthode en sous-méthodes (c'est-à-dire en mnèmes) et à insérer une activité mentale entre chaque sous-méthode. Il se trouve que les mnèmes d'une méthode correspondent essentiellement aux unités syntaxiques d'une commande.

Les règles suivantes déterminent de manière approximative la décomposition d'une méthode :

- **Règle 0** Insérer M devant tous les K qui ne font pas partie de chaînes argument. Insérer M devant un P qui correspond à la désignation d'un nom de commande.
- **Règle 1** Supprimer M si l'opérateur qui suit M peut être anticipé avec l'opérateur qui précède M (par exemple, dans PMK, K représente l'acte d'appuyer sur un bouton de la souris. On considère que l'activité mentale pour K est anticipée dans P).

- **Règle 2** Si une chaîne de la forme MKMK....MK, constitue un mnème, par ex., le nom d'une commande, supprimer tous les M sauf le premier.
- **Règle 3** Si K est un symbole de terminaison redondant, supprimer le M qui le précède. K est un symbole de terminaison redondant s'il suit un autre symbole de terminaison, par exemple s'il termine la commande et s'il est précédé d'un symbole de fin d'argument.
- **Règle 4** Si K termine une constante (par exemple un nom de commande, non pas un argument), supprimer le M qui le précède. Si K termine une variable (par exemple, un argument), alors conserver M.

ex: Appliqué à la tâche précédemment évoquée qui consiste à placer le curseur en bas de la fenêtre dans un éditeur aux commandes souris / clavier à la Emacs, le usage du modèle sera de prédire dans quels cas la méthode M1 (souris) est préférable à M2 (clavier).

Le codage d'une méthode s'effectue en 3 étapes: codage sous forme d'opérateurs physiques, introduction de M avec la règle 0, puis application de règles 1,2,3 et 4 pour supprimer les M.

M1: prendre la souris; déplacer la souris au point désiré; sélectionner;

1. H[souris] P[souris] K[bouton-souris] H[clavier]

2. H[souris] M P[souris] M K[bouton-souris]
H[clavier]

3. H[souris] M P[souris] K[bouton-souris]
H[clavier]

d'où

$$T_{M1} = 2t_H + t_P + t_K + t_M$$

M2 : tant que le curseur n'est pas sur la ligne
désirée taper ctrl-n; tant que le curseur n'est pas
sur le mot désiré taper esc-f;

*Sachant que l'utilisateur a tapé m fois ctrl-n et p fois
esc-f :*

1. K[touche control] m {K[touche n]} p {K[touche
esc] K[touche f]}
2. M K[touche control] m {M K[touche n]} p {M
K[touche esc] M K[touche f]}
3. M K[touche control] m {K[touche n]} M p
{K[touche esc] K[touche f]}

d'où

$$T_{M2} = 2t_M + (m + 2p + 1) t_K$$

M1 est préférable à M2 si : $T_{M1} < T_{M2}$

c'est-à-dire si :

$$2t_H + t_P + t_K + t_M < 2t_M + (m + 2p + 1) t_K$$

en remplaçant par les temps moyens de Card 83

$$(t_H = 0.4 \text{ s}, t_P = 1.1 \text{ s}, t_K = 0.2 \text{ s}, t_M = 1.35 \text{ s})$$

on arrive à : $m + 2p > (2t_H + t_P - t_M) / t_K$

$$\text{soit } m + 2p > 2.75 \quad (1)$$

⇒ pour $m=1$ et $p=1$, la souris est mieux adaptée.

⇒ résultat hatif. Dans la relation, le facteur influent est t_M (1.35 s). Pour un expert Emacs, ce nb est surévalué, et peut-être absent !

Si on supprime le terme t_M on obtient que M1 est préférable à M2 si :

$$m+2p > 9.5 \quad (2)$$

résultat compatible avec le comportement observé sur des utilisateurs expérimentés.

La différence importante entre (1) et (2) révèle deux difficultés : celle de définir une heuristique adaptée et celle de déterminer la valeur des paramètres.

Évaluation de Keystroke

- outil d'analyse quantitative et prédictif
- modélisation simple et compréhensible (bémol: exemple du curseur => simplicité apparente)
- notion d'opération mentale (« pause entre deux mnèmes ») imprécise
- l'ajustement des règles requiert des compétences non informatiques
- pas de distinction touche ordinaire <> touche de contrôle
- niveau local lexical; pas de niveau sémantique global

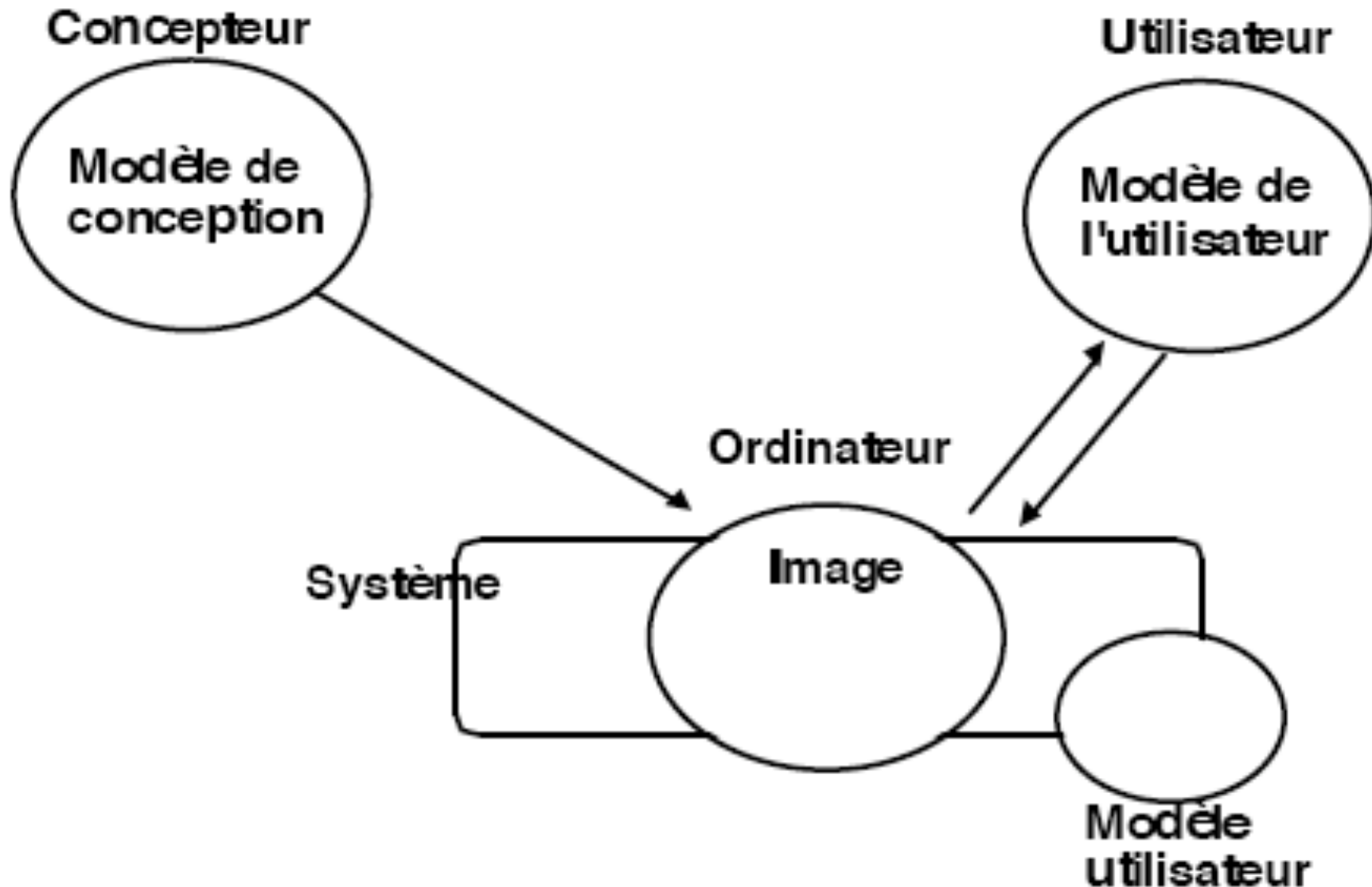
La Théorie de l'Action

D. Norman, « Cognitive Engineering, User Centered System Design », New Perspectives on Computer Interaction, 1986.

La théorie de l'action

- [Norman 86] : théorie fondée sur la notion de modèle conceptuel
 - Modèle de l'utilisateur : variables psychologiques
 - Modèle de conception : variables physiques
 - Image : représentation physique du système
- Permet de structurer l'accomplissement d'une tâche
 - Décomposition en 7 activités

Modèle Conceptuel



L 'accomplissement d 'une tâche

- 1^{ère} étape : Établir un but
 - Le but est la représentation mentale d'un état désiré
 - Exemple : détruire un fichier
- 2^{ème} étape : Formation d'une intention
 - résulte de l'évaluation de la distance entre le but et l'état actuel
 - Exemple : détruire un fichier → enlever l'objet sélectionné

L 'accomplissement d 'une tâche

- 3^{ème} étape : Spécification de la suite d'actions
 - traduction de l'intention en une suite d'actions
 - exemple : Mettre le fichier à la poubelle
- 4^{ème} étape : Exécution des actions
 - met en jeu le savoir-faire moteur

L 'accomplissement d 'une tâche

- 5^{ème} étape : Perception de l 'état du système
 - exemple :
 - état antérieur : liste de fichiers avec le fichier à supprimer
 - état actuel : liste de fichiers sans le fichier à supprimer
 - perception possible : le fichier à supprimer a disparu
- 6^{ème} étape : Interprétation
 - exemple :
 - le fichier a disparu → le fichier a été détruit

L 'accomplissement d 'une tâche

- 7^{ème} étape : Évaluation
 - établit une relation entre le but et la sémantique de l 'expression de sortie
 - peut conduire à modifier le plan
 - exemple : comparer le fichier détruit avec le but

La théorie de l'action (résumé)

L'objectif du concepteur :

réduire les distances mentales par le biais de l'image du système

- **distance d'exécution** : l'effort cognitif de l'utilisateur pour la mise en correspondance entre la représentation mentale de sa tâche et la représentation physique induite de l'image du système
- **distance d'évaluation** : l'effort inverse

Exemple du bain

- Tâche de l'utilisateur :
 - remplir une baignoire avec deux robinets indépendants d'eau chaude - eau froide
- Objectif (besoins) de l'utilisateur :
 - avoir une certaine température t et un certain débit d
- Variables psychologiques :
 d et t



Constat : exemple du bain

- Variables physiques (du système)
 - d_c et t_c : débit et température eau chaude
 - d_f et t_f : débit et température eau froide
- commandes physiques : robinets liés à d_c et d_f
- relations entre les variables physiques et psychologiques :
 - $d = d_c + d_f$
 - $t = (d_c.t_c + d_f.t_f) / (d_c + d_f)$

Exemple du bain

- Etape 1 : fixer le but
 - remplir la baignoire avec une température spécifique et un débit spécifique
- Etape 2 : Comment atteindre le but ?
 - en tournant les 2 robinets
- Etape 3 : Planification
 - tourner le robinet d'eau chaude entièrement
 - tourner le robinet d'eau froide pas à pas

Exemple du bain

- Etape 4: exécuter les actions
- Etape 5: perception de l'état du système
 - mettre la main dans la baignoire ou sous le robinet pour percevoir la température de l'eau
- Etape 6: interprétation de l'état du système
 - la température à une certaine température
- Etape 7 : évaluer l'état du système par rapport au but (et peut-être redéfinir des intentions)
 - l'eau n'est pas assez chaude, ce n'est pas la température voulue
 - en conséquence, je dois diminuer le débit d'eau froide

Exemple du bain

- Pour atteindre le but, il est nécessaire de faire les 7 étapes plusieurs fois !!!
 - Evaluer l'état du système et planifier d'autres actions
 - C'est trop chaud
 - C'est trop froid
 - Le débit n'est pas suffisant
 - Etc.

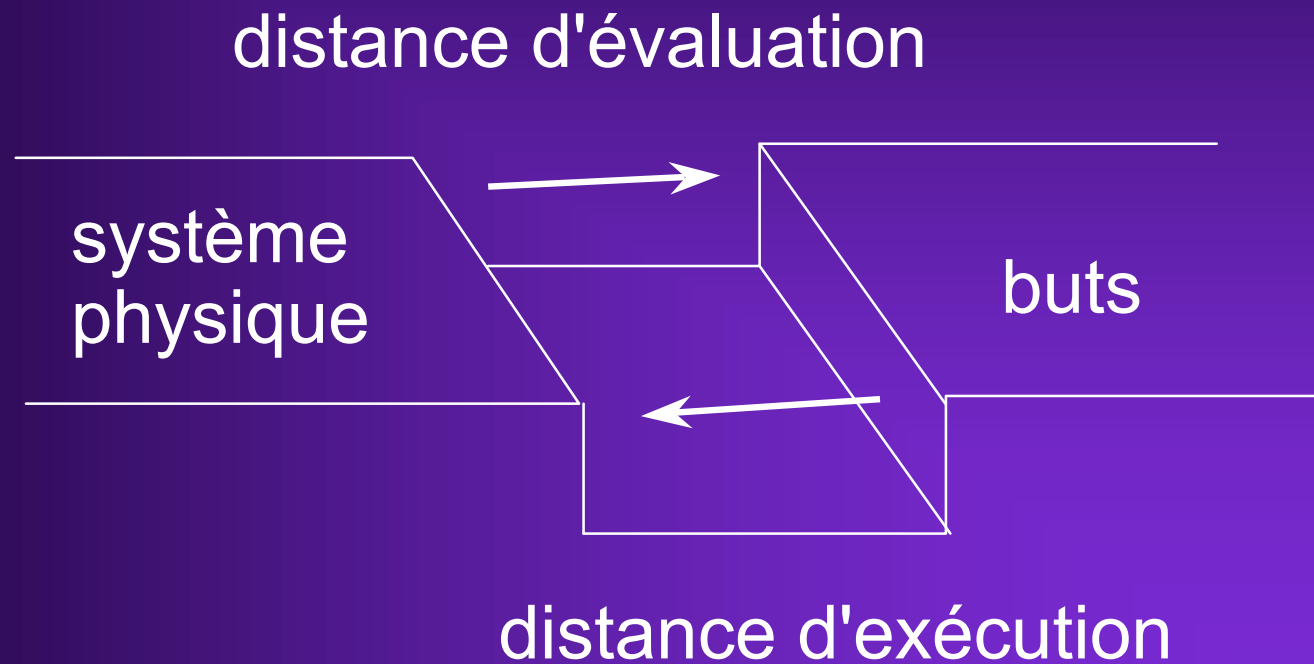
Constat : exemple du bain

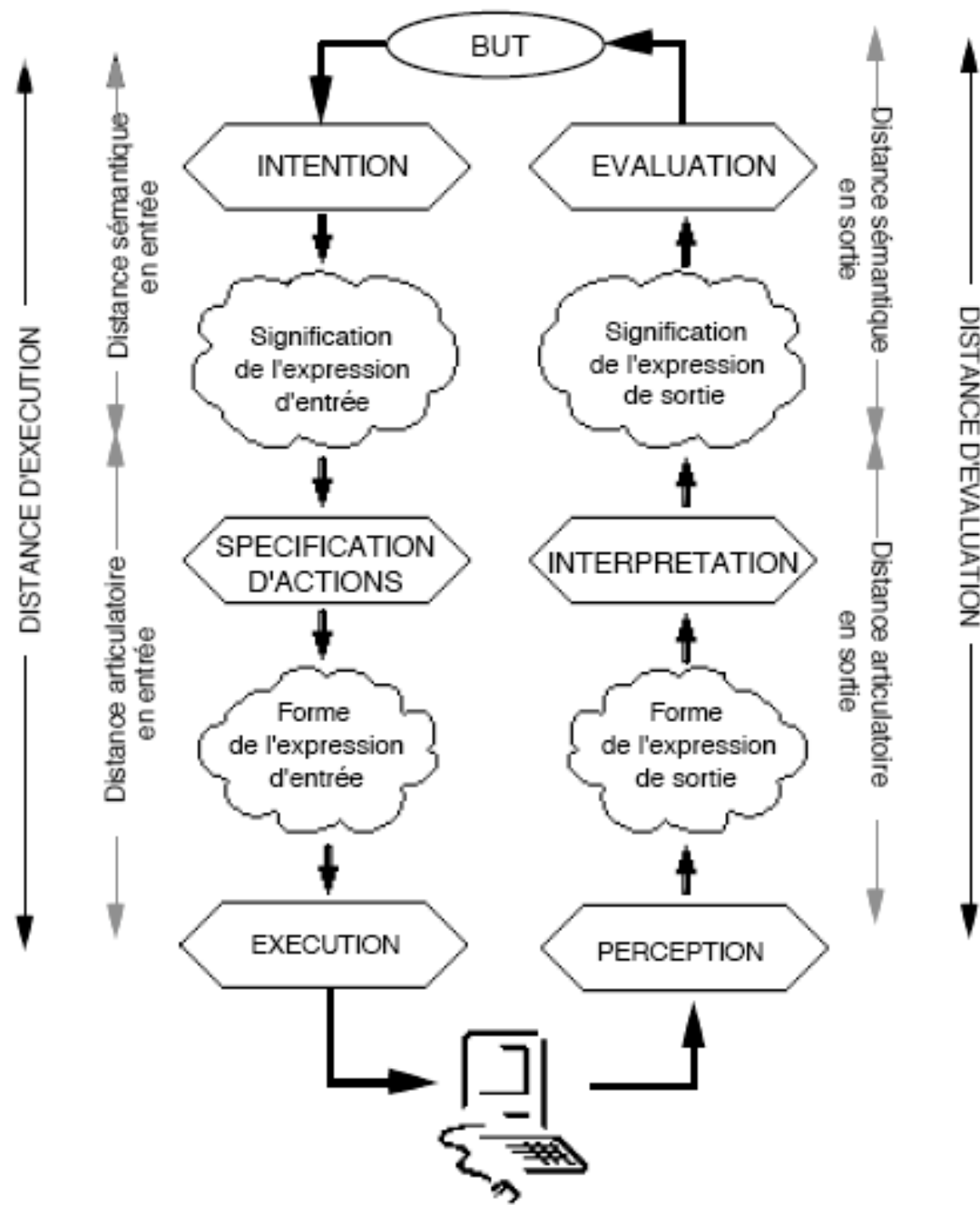
- Problèmes rencontrés par l'utilisateur
 - correspondance entre variable physique et dispositif physique
 - Quel robinet dispense l'eau froide ?
 - comment faire varier le débit (dans quel sens tourner ?)
 - correspondance variables physiques et psychologiques
 - refroidir le bain tout en gardant le débit ?
 - diminuer le débit en gardant la température constante ?

Constat : exemple du bain

- Évaluation du résultat :
 - évaluer la valeur du débit
 - évaluer la valeur de la température
- ➔ Problème avec la réalisation de la tâche
- ➔ Le dispositif physique du bain n'est pas adapté, il est orienté système mais pas utilisateur

Gouffre de l'exécution et de l'évaluation



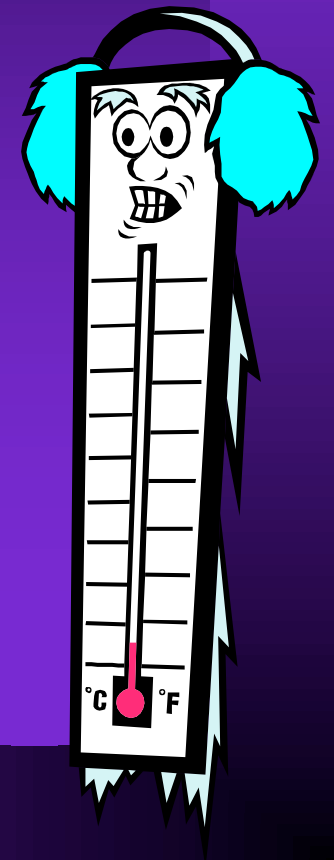
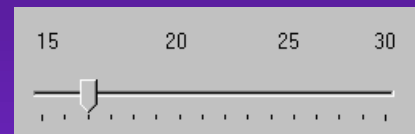


Réduire le gouffre de l'exécution

- Améliorer la mise en correspondance entre intention et sélection
- Jouer sur la «Visual Affordance» des éléments d'interaction
 - Capacité à suggérer leur fonction visuellement
- Feedback proactif des éléments d'interaction

Exemple : Thermostat de chauffage central

- Vous rentrez chez vous et il fait froid. Que faites-vous avec le thermostat du chauffage central ?
 - A : Je le monte à fond
 - B : Je le monte à 20° C



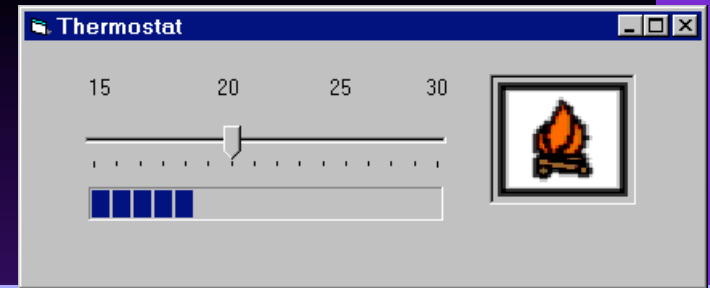
Interprétation de ce comportement

- Trois modèles de l'utilisateur possibles :
 - Le thermostat régule directement la température de l'eau dans les tuyaux (ou la température de la résistance électrique)
 - Le thermostat régule la proportion de temps ou le chauffage fonctionne (0% du temps en bas, 100% du temps à fond)
- ⇒ Le thermostat agit comme un interrupteur, ouvrant le chauffage si la température est inférieure à celle programmée

Problème : « gouffre de l'évaluation »

- Manque de feedback
 - L'action sur le thermostat n'a pas d'effet immédiatement perceptible
 - Difficile de savoir si le but est atteint (perception de la température)

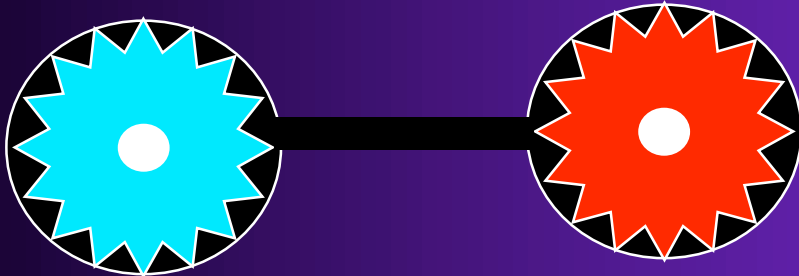
Améliorations



- Permettre à l'utilisateur de construire un meilleur modèle du système
- Améliorer le feedback
 - Indication visuelle du fonctionnement du chauffage (feedback immédiat)
 - Indication de la température réelle par rapport à la température programmée
 - Permet de déterminer facilement si le but est atteint

Mapping entre intention et sélection

- Intention : «Supprimer un fichier»
 - Mettre l'icône du document sur celle de la poubelle
 - «Supprimer» dans le menu «Fichier»
 - `rm /users/recanati/enseignement/ihm/cours.ppt`
- Robinets mitigeurs

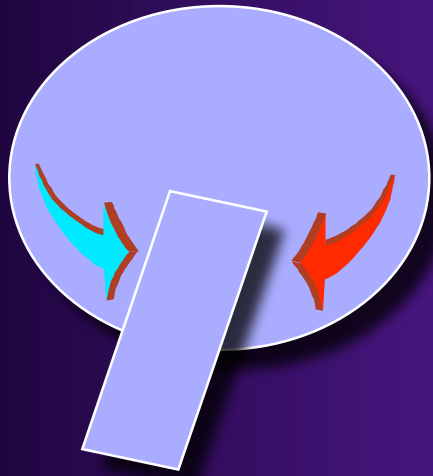


Affordance

- Fait référence à l'attribut d'un objet qui permet de savoir comment utiliser cet objet
 - Ex : une poignée (classique) de porte « invite » à appuyer dessus pour ouvrir la porte
- Norman (1988) utilise ce terme pour discuter de la conception des objets physiques de tous les jours
- Le terme à été popularisé dans le cadre du design des objets des interfaces
 - Ex : les scrollbars « invitent » à les bouger vers le haut et vers le bas

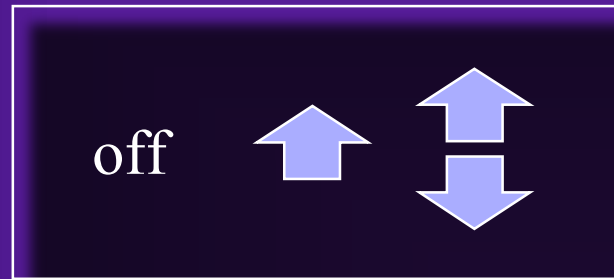
Affordance

Mon robinet mitigeur



Comment avoir
de l'eau plus chaude ?

Mon four électrique

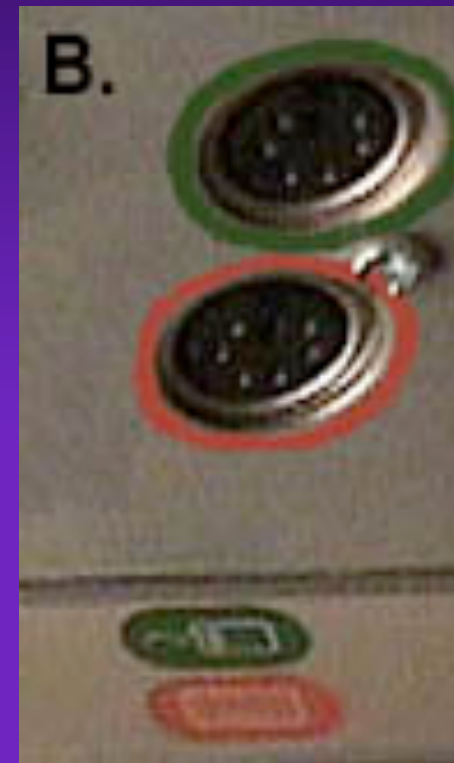


Grille du haut ou
du bas ?

Windows'95



Souris/clavier



Feedback Proactif

- Indiquer visuellement les actions significatives dans un contexte donné
 - Boutons ou zones grisées
- Signaler la complétion d'une action terminée
 - Signal d'acheminement Télécom

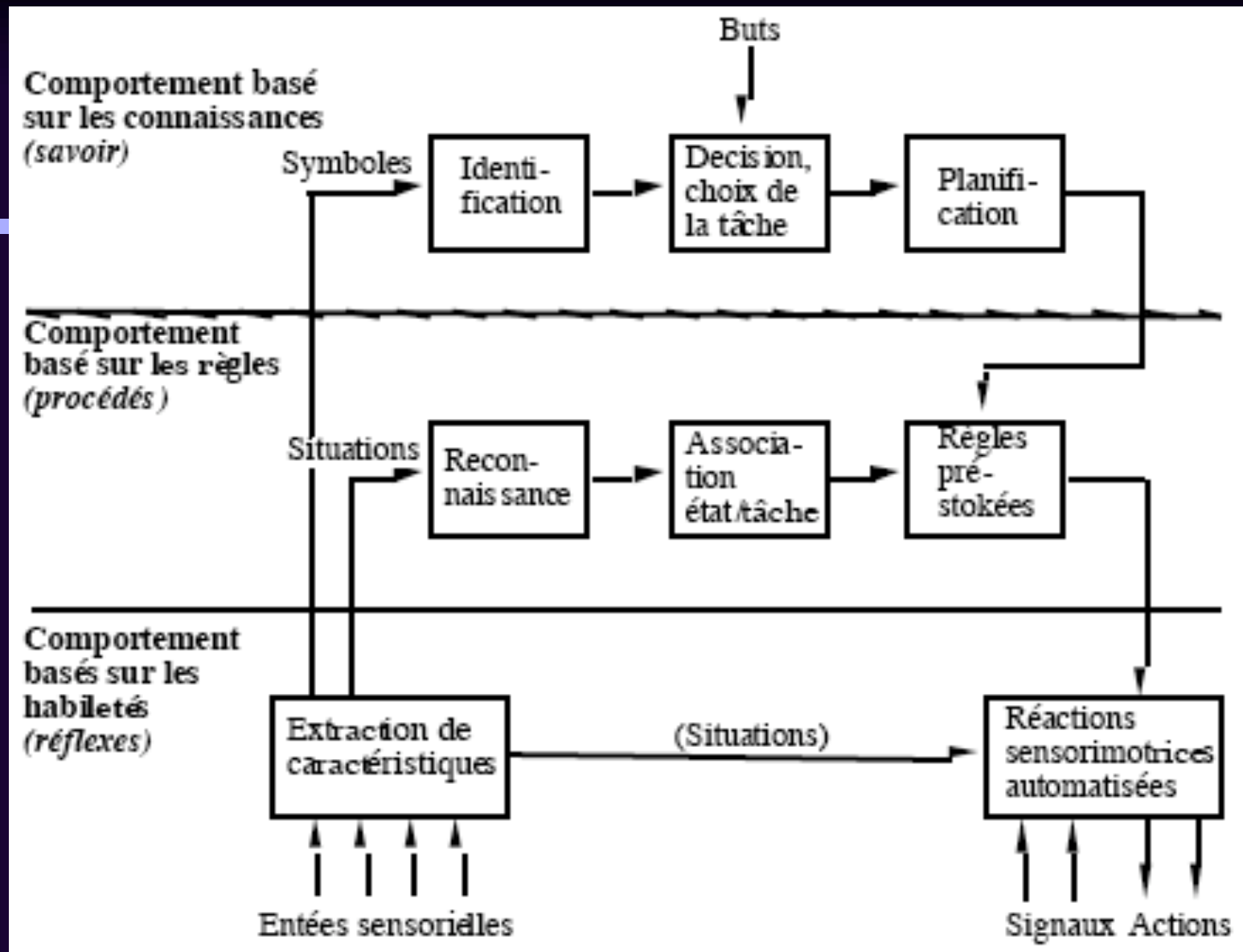
Modèle de Norman et Design

Avec quelle facilité un utilisateur peut-il :

- Déterminer les fonctions du système ?
- Connaître les actions disponibles ?
- Trouver les correspondances intention / action ?
- Exécuter les actions ?
- Percevoir l'état du système ?
- Déterminer si le système est dans l'état désiré ?

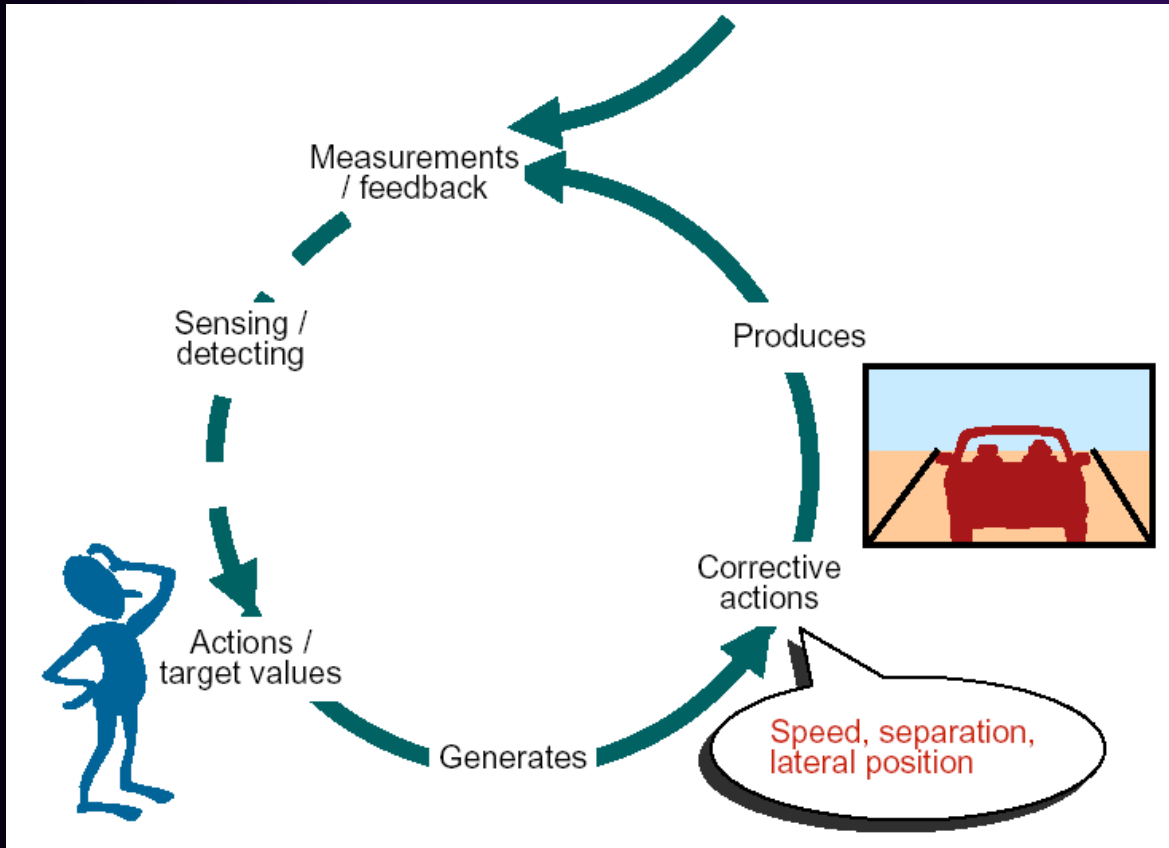
Modèle de Rasmussen

J. Rasmussen, « Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering », 1986



Modèle simplifié des 3 niveaux de contrôle des comportements humains (Modèle de Rasmussen)

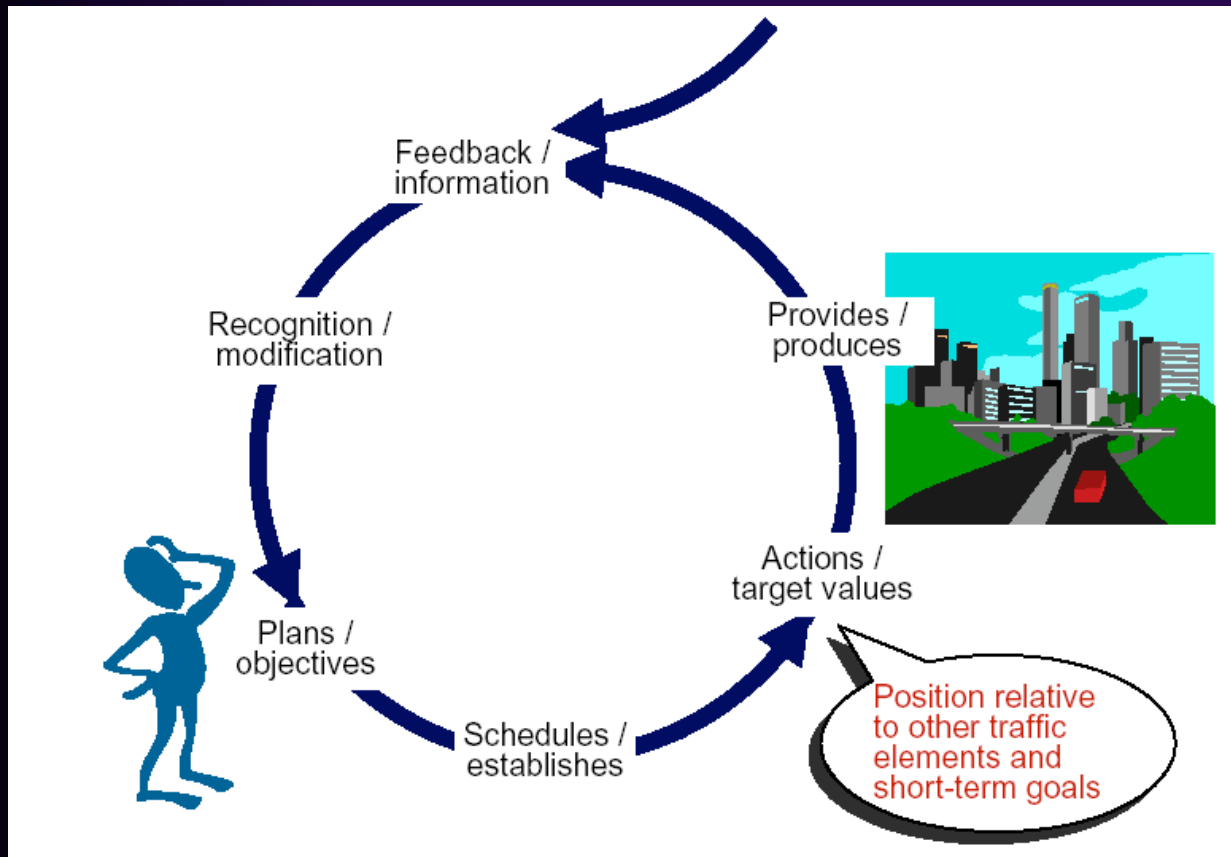
Ex de la conduite : niveau « réflexe »



A ce niveau, pour un utilisateur expérimenté, les activités sont réalisées automatiquement sans nécessité d'y porter beaucoup d'attention et avec peu d'effort

Ex : suivre la route latéralement, rouler à une certaine vitesse, etc...

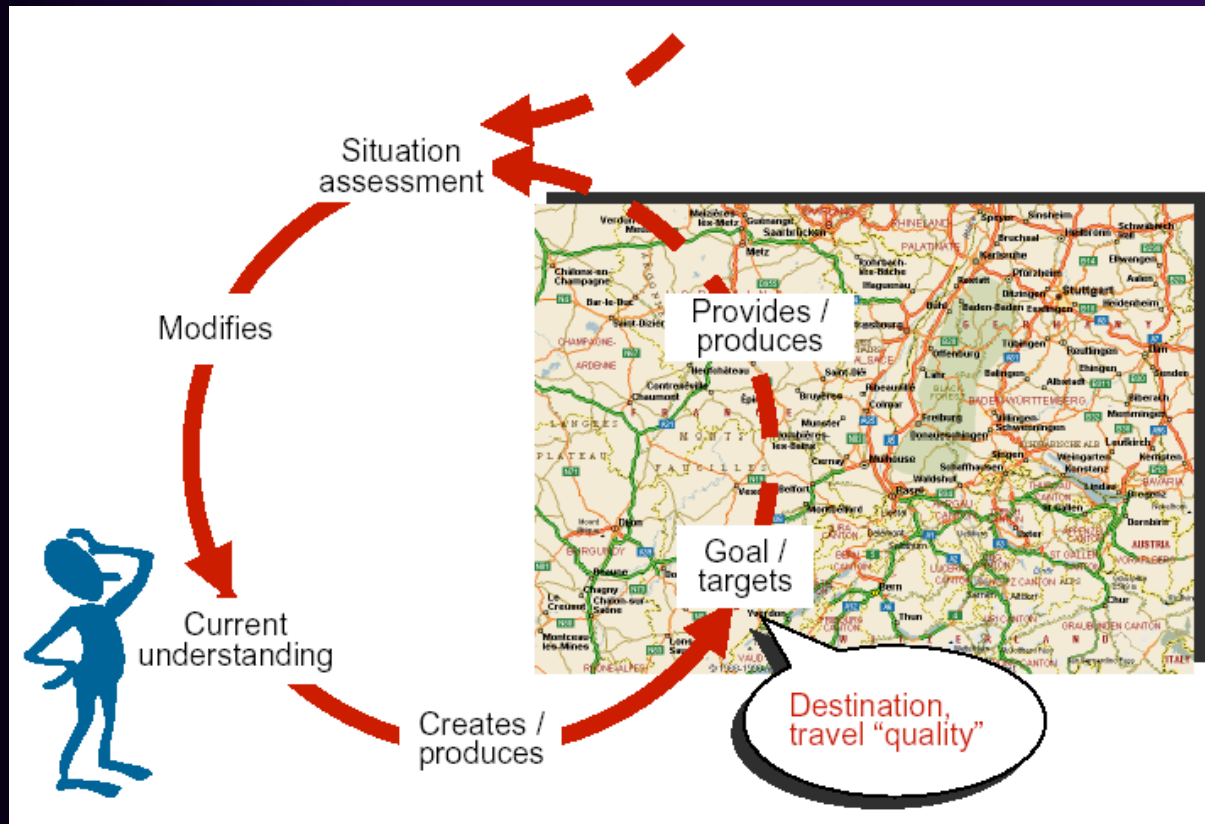
Ex de la conduite : niveau « procédés »



A ce niveau, les activités ne sont pas réalisées automatiquement. L'utilisateur sait ce qu'il fait et veut.

Ex : doubler une voiture, éviter un obstacle (détecté à l'avance)

Ex de la conduite : niveau « savoir »



A ce niveau, les activités doivent être déterminées, choisies et réalisées dans un but. L'utilisateur détermine ce « qu'il doit faire » pour atteindre ce but.

Ex : l'utilisateur pense qu'il va y avoir des bouchons, il change de trajet en cours de route en fonction de l'endroit où il se trouve.

Le modèle de Rasmussen

- Avantages

- fournit un cadre pour la modélisation de l'utilisateur
- complète la théorie de l'action de Norman

- Inconvénients

- le niveau de détail ne permet pas de dépasser les trois classes : novice/intermédiaire/expert
- ne distingue pas les différents types de connaissances de l'utilisateur : ex, connaissances sur le domaine/connaissances sur l'interface

Merci à

Christelle Farenc de l'IRIT
pour son cours de DEA

www.irit.fr/recherches/HIC/farenc/documents/cours_dea_2il_2003.ppt

et à Joëlle Coutaz, pour les 2 chapitres de son livre

<http://iihm.imag.fr/nigay/ENSEIG/M2ProGI/MODELESIHM/Chap1-ProcHumain.pdf>

et

<http://iihm.imag.fr/nigay/ENSEIG/M2ProGI/MODELESIHM/Chap2-GomsKestroke.pdf>