INTERFACE HOMME - MACHINE

SEQUENCE 2: Modèles cognitifs pour la description d'une IHM

Pape Mamadou Djidiack FAYE, Enseignant Chercheur

Email: djidiack88@gmail.com

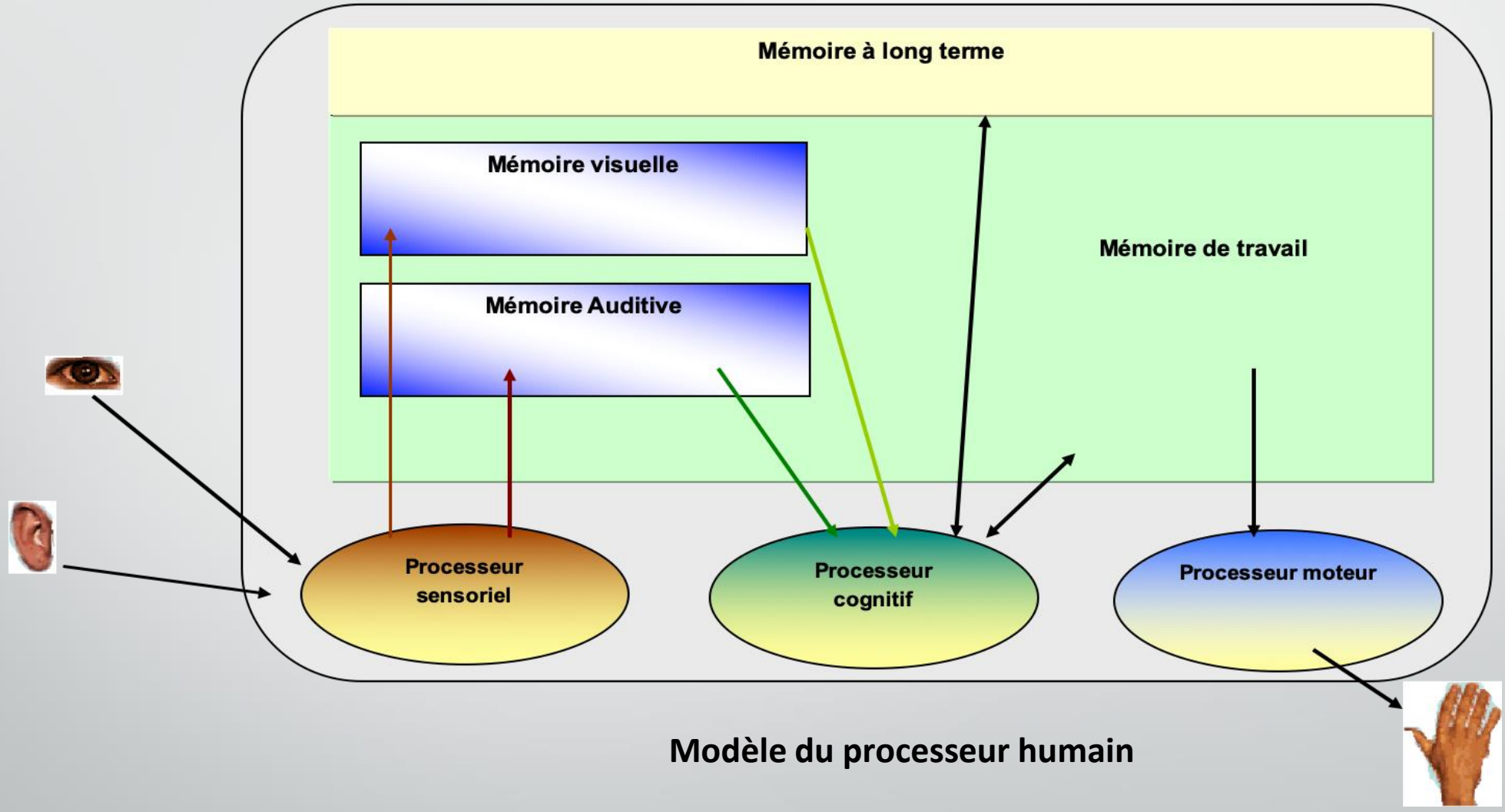
INTRODUCTION

- Une interface est un lieu et un nœud de relations complexes entre l'usager , considéré comme individu doué de cognition et de perception et la machine, considérée comme médiatrice entre l'opérateur et la tâche.
- Quand une interface **est inadaptée, inefficace,** c'est souvent du à beaucoup plus qu'à un simple problème de programmation
- La psychologie cognitive offre des modèles théoriques du fonctionnement de l'esprit humain pour tout comportement mettant en œuvre des connaissances.
- Plusieurs modèles cognitifs ont été proposés pour décrire l'interaction entre utilisateur et ordinateur

MODELE DU PROCESSEUR HUMAIN (MPH)

- Dans le modèle du processeur humain, l'individu est présenté comme **un système de traitement d'informations** régi par un ensemble de règles.
  - ✓ **sous-système de perception (sensoriel)**
  - ✓ **sous-système cognitif**
  - ✓ **sous-système moteur**
- **La mémoire du système cognitif** comprend **une mémoire à court terme** (contentant les informations en cours de manipulation) et une **mémoire à long terme** (contenant les informations permanentes).
- Les performances d'une mémoire ou d'un processeur sont caractérisées par des paramètres constituant des indications intéressantes qui permettent d'adapter l'interface en fonction des performances de l'utilisateur. Ces paramètres ne permettent que d'évaluer les performances mesurables de l'individu, sans apporter des indications sur les modèles de représentation et raisonnement mis en œuvre.
- Ce modèle est suffisamment formel pour être utilisé par les informaticiens; cependant, il offre une modélisation trop simplifiée et trop réductrice.

MODELE DU PROCESSEUR HUMAIN (MPH)



Modèle du processeur humain

GOAL OPERATOR METHOD SELECTION(GOMS)

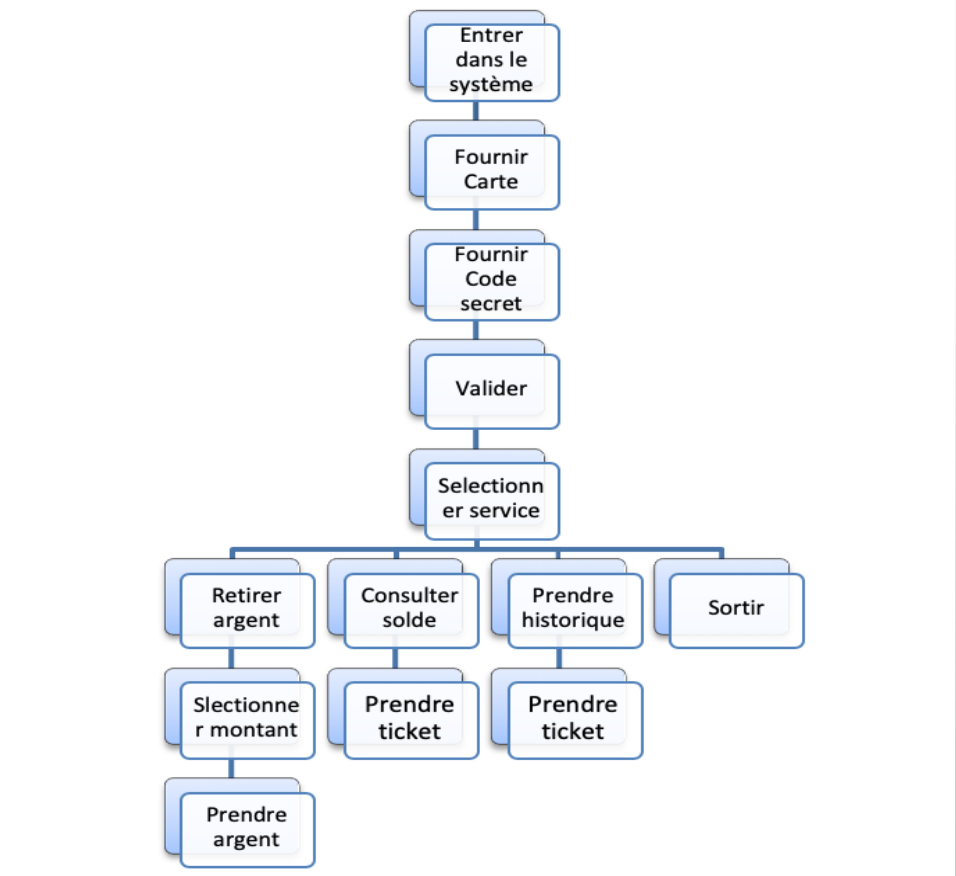
- Ce modèle se contente de modéliser le comportement observable de l'utilisateur (approche behavioriste) et ne cherche pas à décrire les états mentaux et les traitements internes.
- GOMS utilise comme point de départ le principe de rationalité du Modèle du Processeur Humain ; son apport essentiel est une structure formelle qui permet d'organiser le processus de conception.
- La méthode de conception induite par GOMS s'effectue selon deux axes : **l'analyse de tâche** (puisque c'est elle qui détermine le comportement) et **l'évaluation prédictive** du comportement de l'utilisateur dans l'accomplissement de cette tâche.

GOAL OPERATOR METHOD SELECTION(GOMS)

- GOMS (Goal, Operator, Method, Sélection) introduit quatre ensembles pour représenter l'activité cognitive d'un individu engagé dans la réalisation d'une tâche :
  - ✓ Un but **(Goal) :** C'est une structure qui définit un état recherché. A chaque but est annoncé un ensemble de méthodes qui permettent de l'atteindre.
  - ✓ Un opérateur **(Operator) :** C'est une action élémentaire dont l'exécution provoque un changement d'état (état de l'utilisateur ou état de l'environnement).
  - ✓ Une Méthode **(Method) :** C'est une suite conditionnée de buts et d'opérateurs où les conditions font référence à l'état mental ou l'état de l'environnement. Une méthode est un plan acquis par l'expérience.
  - ✓ Une règle de Sélection **(Selection)** : Elle exprime le choix de la méthode lorsque plusieurs méthodes conduisent au même but.

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Exemple :** Soit le système de distribution automatique de billets suivant :



GOAL OPERATOR METHOD SELECTION(GOMS)

- **Goal**:
  - ✓ Les buts sont les services que l'utilisateur veut réaliser à travers le système. Ces buts peuvent être décomposés en sous-buts, d'une façon hiérarchique, tout en respectant le modèle humain de partition des problèmes en sus-problèmes lors de leur résolution.
  - ✓ Toutefois, dans le modèle GOMS cette hiérarchie n'est pas complètement spécifiée (il y en a des versions qui permettent la représentation de buts en parallèle). Pour montrer son fonctionnement nous nous appuyons sur notre exemple, pour laquelle nous pouvons identifier les buts suivants :
    - Retirer d'argent
    - Consulter le solde
    - Prendre l'historique
    - Sortir du système

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Operator :**
  - ✓ Les opérateurs sont des actions réalisées par l'utilisateur pour accomplir un but. Ces opérateurs peuvent être externes, c'est-à-dire des actions perceptuelles et motrices des utilisateurs sur les objets du système pour changer des informations (actions observées), ou internes, c'est-à-dire des opérateurs mentaux des utilisateurs, inférés par les théoriciens et analyseurs.
    - **La Loi de Hicks** donne une façon de prédire le temps **T** pris par l'homme pour faire le choix d'un item parmi une liste de **n items** : **$T = k \log_2 (n+1)$,** où **k** est une constante
    - **La Loi de Fitts** permet de prédire le temps de pointage T pris par l'homme : **$T = k \log_2 (A / W + 1)$,** où:
      - A : la distance entre l'objet de pointage et l'icône à être pointé
      - W : la largeur de l'icône
      - k : constante (valeur 100 msec)

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Operator**
  - ✓ …
  - ✓ Description des opérateurs externes GOMS

Opérateur	Description	Temps
H	Ramener la main au clavier	**0.40 s**
K	Presser une touche	**0.20 s**
T(n)	Taper une chaîne de caractères	**n x 0.2**
P	Pointer	**1.10 s ***
M	Localiser un objet sur l'écran ou vérifier une description	**1.35 s**
W(t)	Attendre la réponse du système	**tR ****
I	Insérer ou retirer une carte	**1.50 s**
*	temps calculé par la Loi de Fitts	
** tR	est le temps pendant lequel le système fait attendre l'utilisateur. Si on considère m comme le temps de traitement d'une commande par le système et t le temps exploité par l'utilisateur pour exécuter un opérateur pendant le traitement de la commande, tR vaut :	**tR = 0 si m ≤ t** **tR = m - t si m > t**

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Operator**
  - ✓ …
  - ✓ Description des opérateurs internes humains

Description	Temps
Accomplir un but	-
Décider	1.35 s
Rappeler à partir de la mémoire à court terme	1. 20 s
Rappeler à partir de la mémoire à long terme	1.35 s
Penser à une valeur	-

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Method :**
  - ✓ Les méthodes sont des séquences d'opérateurs qui permettent l'accomplissement d'un but. Dans le cas où on pourrait avoir plusieurs méthodes pour le même but, le modèle fournit des règles de sélection de la méthode. Dans notre application, les méthodes sont uniques pour chaque but.
    - **Méthode pour le but : Retirer d'argent**

Pas	Opérateur	Durée(s)
Pas1. Identifier l'écran	M	1.35
Pas2. Décider : si vous n'êtes pas dans le système, accomplir but : Entrer dans le système		
Pas3. Accomplir but : Sélectionner service		
Pas4. Accomplir but : Sélectionner montant		
Pas5. Accomplir but : Prendre d'argent		
Pas6. Retourner avec le but accompli		

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Method :**
  - ✓ ….
    - **Méthode pour le but : Entrer dans le système**

Pas	Operator	Durée(s)
Pas1.Insérer la carte	I	2.0
Pas2. Attendre vérification de la carte	W(t)	tR
Pas3. Rappeler de la LTM le code secret(….)		
Pas4. Ramener la main au clavier	H	0.20
Pas5. Tapez code secret	T(4)	1.12
Pas6. Accomplir le but : Valider		
Pas7. Attendre vérification du code	W(t)	tR
Pas8. Décider, si code incorrecte, aller au Pas3 (…)		
Pas9. Retourner avec le but accompli		

    - **Méthode pour le but : Sélectionner un service**

Pas	Opérateur	Durée(s)
Pas1. Localiser le service sur l'écran	M	1.35
Pas2. Pointer sur l'écran	P	1.10
Pas3. Vérifier que le service a été attendu	M	1.35
Pas4. Retourner avec le but accompli		

GOAL OPERATOR METHOD SELECTION(GOMS)

- **Method :**
  - ✓ ….
    - **Méthode pour le but : Entrer dans le système**

Pas	Operator	Durée(s)
Pas1.Insérer la carte	I	2.0
Pas2. Attendre vérification de la carte	W(t)	tR
Pas3. Rappeler de la LTM le code secret(….)		
Pas4. Ramener la main au clavier	H	0.20
Pas5. Tapez code secret	T(4)	1.12
Pas6. Accomplir le but : Valider		
Pas7. Attendre vérification du code	W(t)	tR
Pas8. Décider, si code incorrecte, aller au Pas3 (…)		
Pas9. Retourner avec le but accompli		

    - **Méthode pour le but : Sélectionner un service**

Pas	Opérateur	Durée(s)
Pas1. Localiser le service sur l'écran	M	1.35
Pas2. Pointer sur l'écran	P	1.10
Pas3. Vérifier que le service a été attendu	M	1.35
Pas4. Retourner avec le but accompli		

LE MODÈLE KEYSTROKE

- **Keystroke** concerne les aspects syntaxiques et lexicaux de l'interaction. Ses éléments relèvent des actions physiques que l'utilisateur doit effectuer pour spécifier une commande.
- Dans Keystroke, un problème se formule par :
  - ✓ **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 Méthodes de réalisation de la tâche,
- Keystroke prédit le temps d'exécution de cette **tâche** par un utilisateur expert.

LE MODÈLE KEYSTROKE

- Contrairement à GOMS, Keystroke ne prédit pas de choix de méthode : la méthode est donnée.
- 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.**

LE MODÈLE KEYSTROKE

Les opérateurs

- Keystroke introduit six operateurs 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 MODÈLE KEYSTROKE

Les opérateurs

- Le temps d'exécution T_{exec} d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

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

✓ La détermination du temps T_K dépend de la touche, du clavier et de l'aptitude de l'utilisateur. Une approximation acceptable consiste à soumettre à un test une population d'individus dont les aptitudes sont similaires puis de déterminer T_K en appliquant la formule suivante :

T_K = (durée totale des tests) / (nombre touches frappées sans erreur)

LE MODÈLE KEYSTROKE

Les opérateurs

- Le temps d'exécution T_{exec} d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

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

✓ ..

✓ L' opérateur **P** représente le déplacement du curseur de la souris vers une cible. Le temps T_P nécessaire à ce placement est déterminé à partir de la variante de la loi de **Fitts** .Cette variante nous indique que le temps T pour sélectionner une cible avec la souris s'exprime avec l'équation :

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

- **Ko** 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 décrites par **Card, Moran et Newell** indiquent que pour la souris **Ko = 1.03 s**
- **I** est une constante évaluée à : **I = 0.1 s**,
- **D** est la distance entre la position actuelle de la souris et celle de la cible,
- **L** est la taille (par exemple la largeur) de la cible.

Dans ces conditions, T_P s'obtient en soustrayant de **T** le temps T_K nécessaire à la pression du bouton soit : $T_P = (Ko - tK) + I \log_2(D/L+0.5)$

LE MODÈLE KEYSTROKE

Les opérateurs

- Le temps d'exécution T_{exec} d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

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

✓ ..

✓ L' opérateur **H** représente les aspects pragmatiques de l'interaction homme-machine, en particulier le changement d'utilisation d'un dispositif physique. Les mesures effectuées par Card, Moran et Newell indiquent que le temps T_H nécessaire à la main pour changer de dispositif et se placer correctement sur le nouveau dispositif est : $T_H = 0.4$ s.

LE MODÈLE KEYSTROKE

Les opérateurs

- Le temps d'exécution T_{exec} d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

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

- ✓ ..
- ✓ L' opérateur **D** représente l'utilisation de la souris pour construire un dessin sur l' écran. Le temps T_D nécessaire à la construction d'un dessin dépend des fonctions disponibles. Si le tracé de droite est la seule possibilité, alors T_D est une fonction linéaire du nombre n de segments tracés et de la somme **l** de leur longueur.
- ✓ En appliquant la technique des moindres carrés à un ensemble de données expérimentales recueillies sur des tests de tracé de segments de droite, Card, Moran et Newell ont obtenu : $T_D = 0.9\ n + 0.16\ l$
- ✓ **D** représente un ensemble d' opérateurs graphiques. Si l'on a besoin d'effectuer une étude précise, **TD** devrait être raffinée pour tenir compte de chaque cas.

LE MODÈLE KEYSTROKE

Les opérateurs

- Le temps d'exécution T_{exec} d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

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

- ✓ ..
- ✓ 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**. Cette préparation revêt plusieurs formes et son temps de réalisation TM peut de ce fait varier d'un cas à l'autre.
- ✓ Comme pour l'évaluation de T_D, **Card, Moran** et **Newell** simplifient délibérément la situation et proposent une valeur unique pour T_M: $T_M = 1.35$ s
- ✓ kk

LE MODÈLE KEYSTROKE

Les opérateurs

- Le temps d'exécution T_{exec} d'une tâche est la somme des temps passés à exécuter chaque classe d'opérateurs.

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

✓ ..

✓ 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.

✓ Sachant que :

- **n** est le temps de traitement d'une commande par le système, et que
- **t** est le temps exploité par l'utilisateur pour exécuter un operateur pendant le traitement de la commande

✓ **T_R = o si n <= t; T_R=n-t si n>t.**

LE MODÈLE KEYSTROKE

Codage des méthodes

- Une méthode s'exprime sous la forme d'une suite d' opérateurs. Par exemple, si l'utilisateur doit 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** 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.

LE MODÈLE KEYSTROKE

Codage des méthodes

- 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 identifient la décomposition d'une méthode en mnèmes :
  - ✓ **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' operateur 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 exemple, 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.

LOI DE FITTS

- La **loi de Fitts** détermine le temps pour atteindre un objet et est définie par l'équation ci-dessous.

$$T = a + b \log_2(D/(s+1))$$

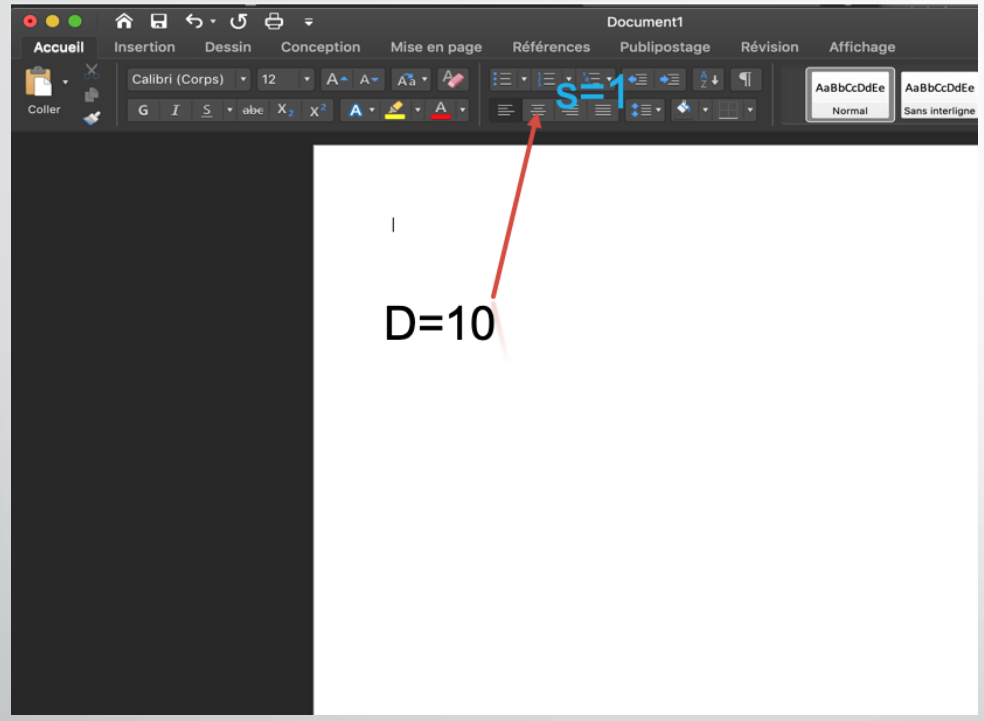
✓ **D** représente la distance entre la position de la souris et le centre du bouton à cliquer.

✓ **S** représente la hauteur du bouton calculée parallèlement à la droite reliant le centre du bouton et l'emplacement de la souris.

- Cette équation suppose une relation directe entre la grosseur du bouton et le temps nécessaire pour l'atteindre et l'enfoncer.
- L'ordinateur **Macintosh** utilise ce principe pour sa **barre de menu**. La barre de menu peut être activée si la souris est au haut de l'écran. Le pointeur ne doit pas être positionné directement sur le texte du menu. Ce design donne une grosseur de bouton « infinie ». Ce qui fait tendre la partie droite de l'équation (ci-dessus) vers o et minimise le temps nécessaire pour effectuer la commande.

LOI DE FITTS

- Exemple: *le pointage d'une icône dans une fenêtre.*



- a et b sont des constantes empiriques
- D : la distance de la cible
- s : la taille de la cible

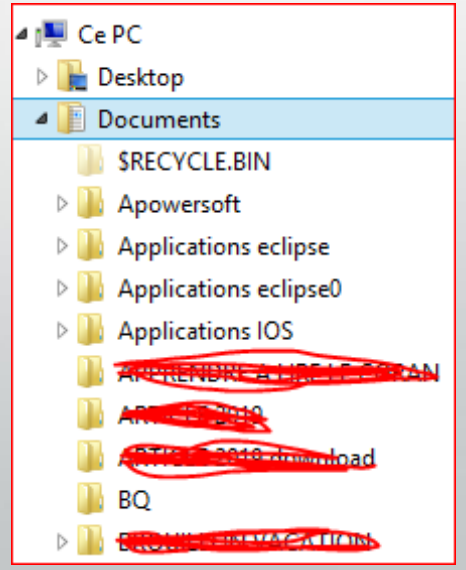
$T = a + b \log_2(D/S + 1)$

$T = 0{,}1 + 0{,}1 \log_2(10/1+1)$

$T = \mathbf{0{,}45\ s}$

LOI DE FITTS

- La **loi de Fitts** présente un temps de pointage typique situé entre 0,5 et 1 seconde.
- Si l'on accumule les temps de pointage sur une longue période d'interaction, on découvre que tout gain sur ces temps de pointage est important pour l'utilisateur.
- **Exemple :** Dans Windows 7, les cibles permettent de voir ou de cacher le contenu d'un dossier rapidement. C'est une technique d'interaction qui fait gagner du temps, car elle permet de diminuer le temps de pointage.



LOI DE HICK

- **La loi de Hick** permet de calculer le temps nécessaire pour prendre une décision parmi une multitude de choix.
- H représente l'entropie de la prise de décision.
- Le temps nécessaire pour prendre une décision est **T = kH** où **k~150ms**.
- **Pi** représente la probabilité de sélection de l'alternative **i**.

$$H=\log_2(n+1)$$

$$H=\sum_i Pi\ log2\left(\frac{1}{pi+1}\right)$$

- Ces deux équations supposent qu'une décision parmi un nombre élevé de possibilités **(n)** prend beaucoup plus de temps qu'une décision parmi un nombre réduit de possibilités.
- Ces deux lois donnent des moyens pour améliorer et calculer l'efficacité de diverses solutions de conception d'interfaces.

Référence

- http://inf1420.teluq.ca/