

TABLE DES MATIÈRES

	Table des matières	i
	Liste des figures	iii
	Introduction	1
1	Le Modèle du Processeur Humain	5
1.1	Les éléments du modèle	5
1.2	Le système sensoriel	5
1.3	Le système moteur	6
1.4	Le système cognitif	8
1.4.1	La mémoire à court terme	8
1.4.2	La mémoire à long terme	9
1.4.3	Le processeur cognitif	10
1.5	Principes opératoires	11
1.6	Evaluation du modèle	11
1.6.1	L'intérêt du modèle	12
1.6.2	Trois exemples d'approximation	12
1.6.2.1	ts, le cycle du processeur sensoriel et le rafraîchissement de l'écran ..	12
1.6.2.2	La loi de Fitts et la souris du Star	13
1.6.2.3	Les limites de la mémoire cognitive	14
1.6.3	Les limites du modèle	15
1.7	Résumé du chapitre	15
	Bibliographie	17
	Index	29



TABLE DES MATIÈRES

LISTE DES FIGURES



1	Le Modèle du Processeur Humain	5
Figure 1.1	Le système sensoriel visuel et sa relation avec le système cognitif.	6
Figure 1.2	Déplacement de la main vers une cible. X0, X1, X2 indiquent les emplacements successifs de la main au cours des micromouvements.	7

INTRODUCTION

LE SUJET : L'INTERACTION HOMME-ORDINATEUR

Cet ouvrage se rapporte à l'interaction homme-ordinateur, un domaine en pleine évolution, à la confluence de deux disciplines : les sciences cognitives et les techniques informatiques. Pour les ergonomes et les psychologues, l'interaction homme-ordinateur désigne l'ensemble des phénomènes physiques et cognitifs qui interviennent dans la réalisation de tâches informatisées. Chez les informaticiens, concernés par la technologie de l'interaction, on parle d'interface homme-ordinateur ou encore, par abus de langage, d'interface homme-machine. Une telle interface est un assemblage de composants logiciels et matériels qui permet l'accomplissement de tâches avec le concours d'un ordinateur.

Nous observons ici la cohabitation de deux vues distinctes de l'interaction homme-ordinateur, l'une animée par les aspects psychologiques de la communication, l'autre dirigée par les composantes techniques de la réalisation. Jusqu'à ces dernières années, les deux approches ont évolué sur des chemins parallèles conduisant chacune à des enseignements intéressants mais souvent sans caractère unitaire et sans transfert de savoir-faire. Cet ouvrage marque un premier pas en direction d'un rapprochement entre les deux points de vue car tous deux trouvent un terrain d'analyse complémentaire dans l'étude des interfaces et de l'interaction.

INTERFACE ET INTERACTION

Au sens large, une interface est un dispositif qui sert de limite commune à plusieurs entités communicantes. Chaque entité s'exprime dans un langage spécifique : signal électrique, force mécanique, langue artificielle ou naturelle, etc. Pour que la communication soit possible, le dispositif doit assurer à la fois la connexion physique entre les entités et effectuer des opérations de traduction entre les formalismes. Dans le cas de l'interface homme-ordinateur, la connexion a lieu entre l'image du système (c'est-à-dire sa manifestation externe) et les organes sensorimoteurs de l'utilisateur ; la traduction s'effectue entre les formalismes du système et ceux de l'utilisateur. Du côté système, une traduction est assurée entre les représentations internes adaptées aux traitements du problème et la représentation externe qui participe à la définition de l'image. Une opération analogue se produit du côté utilisateur entre la représentation mentale de la situation perçue ou à atteindre et les actions physiques à entreprendre. Une fois la communication établie, l'interaction peut avoir lieu.

La réalisation d'une interface suppose donc la connaissance précise du comportement de chacune des entités à relier. Lorsque ces entités sont des objets artificiels, le comportement et les formalismes sont généralement bien maîtrisés ; lorsque l'une des entités est une cible mobile, tel le sujet humain, la définition de l'interface devient une tâche complexe, souvent teintée d'empirisme et d'arbitraire. Il y a trois raisons à cela :

la puissance fonctionnelle des systèmes informatiques, le peu d'intérêt des informaticiens pour les sciences cognitives et l'absence de modèles psychologiques adaptés à l'interaction homme-ordinateur.

LES LACUNES DE LA SITUATION

En raison de son potentiel fonctionnel, l'ordinateur peut s'entrevoir, non pas comme un simple outil, mais comme un collaborateur. Un outil est un instrument sans pouvoir décisionnel. Il est conçu pour être manipulé. A l'inverse, un collaborateur participe activement à la réalisation de l'œuvre commune. Il est conçu pour aider à la résolution d'un problème. L'efficacité du soutien dépend étroitement de la connaissance des facultés et des mécanismes de résolution du partenaire. Dans ces conditions, le concepteur d'un système interactif doit élaborer une description aussi précise que possible du problème et des processus cognitifs de l'utilisateur. Il lui faut ensuite concrétiser aussi fidèlement que possible cette représentation dans le logiciel. Si la description est "correcte" et si la transcription logicielle conserve cette "correction", le système peut être l'extension électronique des facultés intellectuelles de l'utilisateur tout comme la machine-outil est devenue l'extension mécanique des facultés musculaires et motrices de l'homme.

L'ensemble "système informatique-utilisateur" fonctionne comme des vases communicants : si l'utilisateur ne va pas vers le système, le système parcourt la branche de chemin qui conduit à la collaboration ; à l'inverse, si le système ne va pas jusqu'à l'utilisateur, l'utilisateur doit faire l'effort d'adaptation complémentaire. Lorsque l'effort demandé dépasse les capacités ou les motivations, l'interaction s'achève en situation d'échec. L'objectif du concepteur est donc de déterminer les limites du chemin mental de l'utilisateur. Cette détermination fait précisément appel au domaine des sciences cognitives.

Par ignorance, par réticence, ou en raison de contraintes conjoncturelles et techniques, la conception des systèmes se plie trop souvent aux exigences de l'informatique avant de répondre à celles de l'utilisateur. L'informaticien s'applique avant tout à définir les fonctions logicielles d'un système sans vraiment se préoccuper des besoins et des limites cognitives de l'utilisateur. Une autre composante de cette approche erronée est la croyance encore trop répandue qu'un habillage racoleur suffit à rendre le système "facile à apprendre" et "facile à utiliser". Si les fonctions d'un système ne sont pas de nature à compléter les facultés de l'utilisateur, si leur organisation ne correspond pas à la structure mentale de résolution, aucun effet de présentation ne pourra durablement maquiller les carences de fond. A la décharge de l'informaticien, les enseignements des sciences cognitives ne permettent pas, à l'heure actuelle, de déterminer scientifiquement les caractéristiques cognitives de l'utilisateur.

Les sciences cognitives proposent des théories séduisantes mais trop restreintes pour envisager de modéliser l'ensemble des processus psychologiques de l'interaction homme-machine ; elles comprennent des modèles quantitatifs formels mais trop réducteurs pour être applicables au cas complexe de l'interaction ; il existe des modèles qualitatifs généraux mais trop informels pour guider les choix sur une voie

scientifiquement sûre ; on dispose d'une multitude de recommandations pratiques qui conduisent parfois à des situations contradictoires. Les hypothèses et les démonstrations expérimentales prolifèrent mais aucune méthode de conception ne s'adresse véritablement à l'informaticien. De cette diversité, il est possible néanmoins de retenir des enseignements formateurs et constructifs que cet ouvrage tente d'organiser et de rassembler.

L'OBJET ET L'ORGANISATION DE L'OUVRAGE

L'un des objets de cet ouvrage est de sensibiliser les informaticiens à la nécessité d'ouvrir les logiciels aux concepts de l'ergonomie cognitive. La mise en pratique de cette ouverture ne relève pas d'une méthode éprouvée mais d'une compétence. Ce document présente de manière informelle une part de ce savoir-faire. Il comprend trois parties. La première organise la diversité des connaissances du domaine des sciences cognitives en une synthèse de points de référence simples et utiles à la conception des systèmes interactifs ; les parties suivantes concernent les aspects logiciels de l'interaction. Elles montrent en particulier comment construire des systèmes capables de répondre aux principes de l'ergonomie.

La première partie reflète les deux tendances du domaine des sciences cognitives en présentant trois théories influentes et un condensé de principes pratiques :

- Le Modèle du Processeur Humain, exposé au chapitre 1, définit un cadre fédérateur à la diversité des connaissances en psychologie et utilise une terminologie adaptée à l'informaticien.
- Le chapitre 2 présente les modèles GOMS et Keystroke, deux applications directes du Modèle du Processeur Humain résolument orientées vers les performances.
- Le chapitre 3 complète l'approche behavioriste de GOMS et de Keystroke avec le point de vue cognitiviste de la théorie de l'action. Cette dernière est un modèle explicatif du comportement utile à la mise en évidence des points clés de l'interaction.
- L'approche théorique trouve son complément au chapitre 4 avec l'orientation pratique des principes ergonomiques. Ceux-ci sont innombrables et difficiles à embrasser dans leur totalité. Dans cette présentation, l'objectif est de retenir les éléments les plus marquants mais aussi les plus simples à mettre en œuvre par un informaticien.
- Cette première partie conclut au chapitre 5 sur la nécessité d'un schéma directeur de méthode de conception.

Après avoir identifié l'apport des sciences cognitives à la conception des systèmes interactifs, cet ouvrage présente les aspects logiciels de l'interaction. La deuxième partie traite de l'organisation des abstractions logicielles impliquées dans la réalisation des systèmes interactifs. Sur ce thème, la littérature est parcimonieuse et confuse.

INTRODUCTION

- Le chapitre 6 identifie les fonctions essentielles d'un système interactif et précise la terminologie.
- Le chapitre 7 est une clarification des connaissances relatives aux modèles d'architecture. Il fait observer que ces modèles ne permettent pas toujours de satisfaire aux exigences psycho-cognitives de l'interaction.
- Le modèle PAC, au centre de la contribution de l'auteur au domaine, vise à combler les lacunes des propositions précédentes. Il fait l'objet du chapitre 8.
- Au chapitre 9, la description de trois applications réalisées selon ce modèle en démontre l'intérêt.

La troisième partie présente les outils utiles à la construction des systèmes interactifs : les boîtes à outils, les techniques d'affichage, les squelettes d'application et les générateurs d'interfaces.

- Le chapitre 10 propose une classification et une étude comparative des boîtes à outils actuelles. Il en identifie les avantages et les limitations en examinant leurs effets sur la qualité ergonomique du logiciel et en considérant le point de vue du réalisateur.
- Le chapitre 11 concerne les problèmes de l'affichage.
- Le chapitre 12 est consacré aux squelettes d'application qui viennent utilement combler les lacunes des boîtes à outils. La technique des squelettes est un exemple de réutilisation de logiciel. Elle est récente et encore mal maîtrisée. L'objet de ce chapitre est d'en dégager les principes directeurs.
- Le dernier chapitre développe les aspects essentiels de la génération automatique des interfaces. Il en indique les éléments prometteurs et les lacunes.

La conclusion dresse un bilan de la situation dans le domaine de l'interaction homme-ordinateur et identifie les points de recherche sensibles.

LE MODÈLE DU PROCESSEUR HUMAIN

1.1 LES ÉLÉMENTS DU MODÈLE

Dans leur modèle, "The Model Human Processor", S. Card, T. Moran et A. Newell représentent l'individu comme un système de traitement d'informations régi par des règles [Card 83].

Le processeur humain comprend trois sous-systèmes interdépendants : les systèmes sensoriel, moteur et cognitif. Chacun d'eux dispose d'une mémoire et d'un processeur dont les performances se caractérisent à l'aide de paramètres. Pour une mémoire, les paramètres essentiels sont :

- 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),
- κ , le type d'information mémorisée (physique, symbolique, etc.).

Pour un processeur, le paramètre important est

- τ , le cycle de base qui inclut le cycle d'accès à sa mémoire locale.

Les règles qui régissent le comportement du processeur humain sont répertoriées au paragraphe 1.5. Certaines traduisent les caractéristiques des sous-systèmes considérés isolément, d'autres expriment leur interdépendance. Voyons dans les paragraphes suivants comment chaque sous-système opère et coopère.

1.2 LE SYSTÈME SENSORIEL

Le système sensoriel désigne l'ensemble des sous-systèmes spécialisés chacun dans le traitement d'une classe de stimuli. Un stimulus est un 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. Ce codage exprime les propriétés physiques du phénomène. Par exemple, dans la mémoire visuelle présentée dans la figure 1.1, le codage de la lettre P traduit les courbures et les dimensions de la lettre mais n'exprime pas sa reconnaissance.

Les mémoires sensorielles sont intimement liées à la mémoire à court terme du système cognitif. Ce dernier dispose d'un filtre qui détermine la nature des informations à transférer entre les mémoires sensorielles et 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. La persistance des mémoires sensorielles est de l'ordre de

200 ms pour la mémoire visuelle et légèrement supérieure, 1500 ms, pour la mémoire auditive.

Le cycle de base τ_s 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 (c'est-à-dire pour que l'individu ait la sensation de percevoir) et que 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 (de 50 à 200 ms, voire largement en dehors de ces limites si les conditions sont extrêmes).

Nous verrons au paragraphe 1.6 consacré à l'évaluation du modèle pourquoi ces valeurs concernent directement la réalisation des systèmes interactifs.

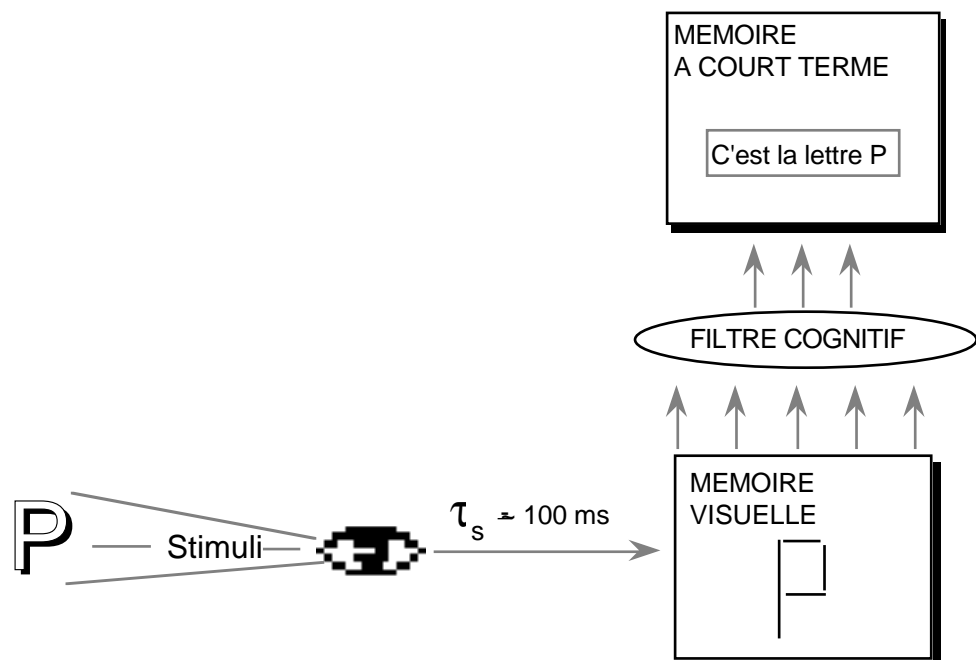


Figure 1.1 : Le système sensoriel visuel et sa relation avec le système cognitif.

1.3 LE SYSTÈME MOTEUR

Le système moteur est responsable des mouvements. Dans le cadre de l'interaction homme-ordinateur, les mouvements d'intérêt sont les manipulations des unités physiques de commande, telles que les claviers, les écrans et les 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.

Partant de l'hypothèse qu'un mouvement est le résultat de plusieurs micromouvements, il est possible de déterminer le temps théorique T nécessaire au placement de la main sur une cible donnée, soit :

$$T = I \cdot \log_2\left(\frac{2D}{L}\right) \quad (1)$$

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.

La figure 1.2 illustre la situation initiale : la main est située en X_0 .

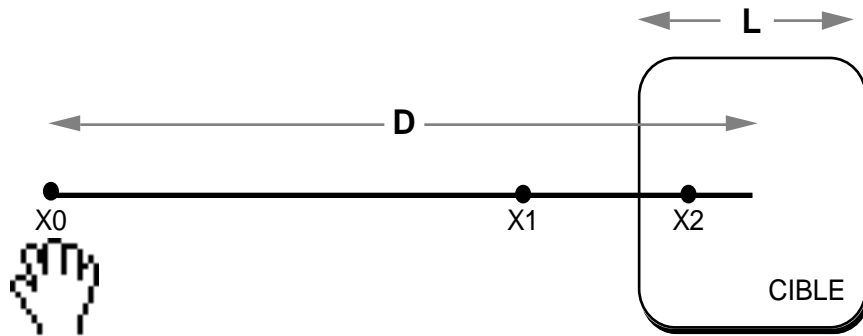


Figure 1.2 : Déplacement de la main vers une cible. X_0 , X_1 , X_2 indiquent les emplacements successifs de la main au cours des micromouvements.

Soient :

- X_i , la distance à parcourir après l'exécution du i ème micromouvement, et
- ϵ , une constante d'erreur telle que $X_i \div X_{i-1} = \epsilon$ avec $\epsilon < 1$, ce qui signifie que la précision relative du mouvement est supposée constante.

Alors :

- $X_0 = D$
- $X_n = \epsilon \cdot X_{n-1} = \epsilon^n \cdot X_0 = \epsilon^n \cdot D$

La main s'arrête lorsqu'elle est sur la cible c'est-à-dire dès que :

$$\epsilon^n \cdot D \leq \frac{L}{2} \quad \epsilon^n \leq \frac{L}{2D} \quad (2)$$

où n représente le nombre de micromouvements nécessaires pour atteindre la cible. Déterminons n à partir de l'inégalité (2).

Sachant que $\log_2(\epsilon^n) = n \cdot \log_2(\epsilon)$,

$$n = \frac{\log_2(\epsilon^n)}{\log_2(\epsilon)} \quad (3)$$

ou, en remplaçant ϵ^n par sa valeur prise dans l'équation (2) :

$$n = \frac{\log_2\left(\frac{L}{2D}\right)}{\log_2(\epsilon)} = \frac{-\log_2\left(\frac{2D}{L}\right)}{\log_2(\epsilon)} \quad (4)$$

Connaissant le nombre n de micromouvements pour atteindre la cible et le temps théorique τ mis pour effectuer un micromouvement, il est possible de déterminer T , le temps théorique nécessaire au déplacement de la main soit : $T = n \cdot \tau$.

D'après le Modèle du Processeur Humain, le temps théorique pour effectuer un micromouvement comprend :

- un cycle τ_s du processeur sensoriel pour vérifier la localisation actuelle de la main,
- un cycle τ_c du processeur cognitif pour ordonner l'exécution du micromouvement,
- un cycle τ_m du processeur moteur pour effectuer le micromouvement.

Ce qui donne : $\tau = \tau_s + \tau_c + \tau_m$.

- En posant $I = -\tau \div \log_2(\epsilon)$, on obtient :

$$T = n \cdot \tau = I \cdot \log_2\left(\frac{2D}{L}\right) \quad (5)$$

L'égalité (5) définit la loi de Fitts. Nous verrons son application au paragraphe 1.6.

1.4 LE SYSTÈME COGNITIF

La mémoire du système cognitif comprend la mémoire à court terme (appelée également mémoire de travail) et la mémoire à long terme. La mémoire à court terme détient les informations en cours de manipulation tandis que la mémoire à long terme est le lieu de stockage de la connaissance permanente. Le processeur du système cognitif contrôle le comportement de l'individu en fonction du contenu de ces mémoires.

1.4.1 La mémoire à court terme

La mémoire à court terme se comporte comme les 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. Contrairement au codage des stimuli en mémoire sensorielle, elles ne sont plus affectées des caractéristiques physiques. Par exemple, la représentation de P dans la mémoire à court terme traduit le fait qu'il s'agit de la lettre P.

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. Les associations et la nature d'un mnème dépendent de la tâche en cours et des connaissances de l'individu. Par exemple, la suite de lettres "S, N, C, F" constitue un mnème pour la plupart d'entre nous mais forme quatre mnèmes pour toute personne qui ignore la signification du sigle.

L'activation d'un mnème entraîne sa mise à disposition dans la mémoire à court terme. Cette activation se propage aux mnèmes associés ajoutant de nouveaux éléments dans la mémoire à court terme. La capacité de cette mémoire est estimée, selon la célèbre formule de Georges Miller [Miller 75], à 7 ± 2 mnèmes¹. Lorsque la mémoire à court terme est saturée, l'activation de nouveaux mnèmes efface de la mémoire ceux qui n'ont pas fait l'objet d'une réactivation. Ce phénomène de conflit s'appelle interférence avec, pour conséquence, la dégradation de l'information au bout d'un certain temps².

L'interférence se manifeste dès qu'il y a partage de ressource. Il est intéressant de relever la similitude entre le modèle de fonctionnement de la mémoire à court terme selon lequel les éléments inutilisés se dégradent, et la technique usuelle des systèmes de pagination qui réquisitionnent les pages les plus anciennes et les moins référencées (c'est-à-dire les moins "réactivées").

Nous verrons au paragraphe 1.6 les conséquences de la faible capacité de la mémoire à court terme et de sa persistance sur l'interaction homme-ordinateur.

1.4.2 La mémoire à long terme

La mémoire à long terme a le même rôle que les mémoires centrales et secondaires d'un ordinateur : elle contient l'information de masse ; elle peut être lue ou modifiée ;

-
1. En toute rigueur, deux valeurs traduisent la capacité de la mémoire à court terme :
 - la capacité pure : $3[2.5 \sim 4.1]$ mnèmes, déterminée, par exemple, à partir du nombre de chiffres que l'on est capable de retrouver au sein d'une longue suite qui s'arrête brusquement,
 - la capacité effective qui tient compte de l'utilisation de la mémoire à long terme : 7 ± 2 mnèmes, estimée, par exemple, comme étant le nombre de mots répétés d'une liste.
 2. La persistance δ_c varie inversement avec le nombre de mnèmes que l'utilisateur tente de retrouver simultanément. Card et Newell retiennent pour les moyennes suivantes :
 - $\delta_c [1 \text{ mnème}] = 73 \text{ s}$
 - $\delta_c [3 \text{ mnèmes}] = 7 \text{ s}$.

son contenu est un réseau de mnèmes qui représente des procédures et des données appelées respectivement connaissance procédurale (ou savoir-faire) et connaissance factuelle (ou connaissance de faits).

Une opération de lecture dans la mémoire à long terme consiste à 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 : ou bien aucune association n'est trouvée, ou bien plusieurs mnèmes interfèrent avec le mnème cible. Dans le second cas, il y a interférence parce que plusieurs mnèmes répondent aux mêmes critères de recherche ; ils répondent aux mêmes critères de recherche parce qu'ils ont été enregistrés avec des critères d'association identiques. Les mnèmes qui interfèrent, acquis plus récemment que la cible, ont un degré d'activation supérieur ; de ce fait, ils masquent l'accès à la cible. En résumé, l'information ne disparaît pas de la mémoire à long terme. Elle est simplement inaccessible. En conséquence, on considère que la persistance des informations de la mémoire à long terme est infinie.

Les opérations d'écriture dans la mémoire à long terme s'effectuent par association. Afin d'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. Le principe de discrimination abordé précédemment indique que les chances de retrouver ce mnème croissent avec le nombre d'associations discriminantes. L'intuition pousse à penser que ces chances augmentent également avec le temps t disponible pour effectuer ces associations. Ceci est vrai tant que $t \leq \delta_c$. Au-delà de δ_c , un mnème se dégrade s'il n'est pas réactivé. Or, la mise en place d'une association active des mnèmes de la mémoire à long terme qui entrent en compétition avec le mnème initial pour la mémoire à court terme. Ces mnèmes "plus frais" ont un degré d'activation plus élevé que le mnème initial qui s'efface. δ_c définit le rythme optimal d'arrivée de nouveaux mnèmes dans la mémoire à court terme pour qu'ils soient associés à des éléments de la mémoire à long terme. La création de nouvelles associations est toujours possible. On considère donc que la capacité de la mémoire à long terme est infinie.

1.4.3 Le processeur cognitif

Le fonctionnement du processeur cognitif est calqué sur le modèle des systèmes de production. Il opère selon le cycle "Reconnaissance-Action", analogue au cycle "Recherche-Exécution" des calculateurs usuels. 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. 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 t_c semble similaire aux cycles des autres processeurs : $t_c = 70\text{ms}$.

Dans la description qui précède, nous avons énoncé quelques-uns des principes de fonctionnement du système cognitif. Nous les rappelons au paragraphe suivant.

1.5 PRINCIPES OPÉRATOIRES

Nous distinguons deux types de principes : ceux qui reflètent la théorie du Modèle du Processeur Humain et ceux qui formalisent des observations empiriques en accord avec cette théorie. Ne sont cités ici que les principes utiles à l'exposé de l'ouvrage.

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.

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, connaissance.

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, soit :

$$T = I \cdot \log_2\left(\frac{D}{L} + 0.5\right) \quad (6)$$

où

$$I = \frac{-(\tau_s + \tau_c + \tau_m)}{\log_2(\epsilon)} \quad (7)$$

ϵ a été évalué à 0.07 ([Card 83] p. 53), ce qui donne une valeur moyenne de $I = 100\text{ms}$. La formule ci-dessus est une variante de la loi de Fitts que nous devons à Welford (1968). Diverses expérimentations tendent à conforter sa validité.

1.6 EVALUATION DU MODÈLE

Le Modèle du Processeur Humain présente trois avantages : il constitue un cadre fédérateur à la diversité des connaissances en psychologie ; il utilise la terminologie de l'informaticien et surtout, franchit un pas important en direction d'une psychologie appliquée. En revanche, le niveau d'abstraction des éléments théoriques du modèle ne correspond pas aux besoins de conception globale d'une interface homme-machine.

1.6.1 L'intérêt du modèle

Le Modèle du Processeur Humain représente l'individu comme un système de traitement d'informations. Les concepts attachés à ces systèmes sont précisément ceux que les informaticiens manipulent quotidiennement. Pour cette raison, ce modèle constitue une introduction simple et séduisante au domaine de la psychologie cognitive.

Au-delà d'une terminologie adaptée, le Modèle du Processeur Humain définit un cadre fédérateur à la diversité des connaissances en psychologie. Sur cette charpente, l'informaticien peut progressivement greffer ses connaissances concernant tel ou tel aspect de l'individu. L'objectif des auteurs du modèle est plus ambitieux que d'offrir un simple cadre de réflexion.

L'objectif de Newell est de créer une discipline qui allierait à la fois les caractéristiques de la science fondamentale et les caractéristiques de la science appliquée. Elle permettrait, comme la physique, d'effectuer des évaluations approchées. A partir d'une théorie technique [Newell 86], il serait possible de construire des modèles qui répondraient aux questions que le concepteur se poserait sur des phénomènes précis de l'interaction homme-ordinateur.

Le Modèle du Processeur Humain est une tentative en direction de cette théorie technique : des paramètres (capacité, persistance, cycle) formalisent les performances élémentaires des sous-systèmes ; des lois (comme celle de Fitts) expriment, par des formules mathématiques, les performances globales du processeur humain. Ces paramètres et ces lois ont été confirmés par l'expérimentation ; ils constituent des approximations réalistes formelles utiles à la prise de décision des compromis inhérents à la conception d'interface homme-ordinateur. Citons quelques exemples de questions qui trouvent leur réponse grâce au modèle.

1.6.2 Trois exemples d'approximation

Les trois exemples choisis concernent chacun un sous-système différent. Le premier démontre le lien entre le cycle du processeur sensoriel et le rafraîchissement de l'écran. Les trois exemples choisis concernent chacun un sous-système différent. Le premier démontre le lien entre le cycle du processeur sensoriel et le rafraîchissement de l'écran. Le second relève la dépendance entre les performances du système moteur et la conception d'unités de désignation ; le dernier exemple met en garde le concepteur contre la faible capacité de la mémoire à court terme.

1.6.2.1 τ_s , LE CYCLE DU PROCESSEUR SENSORIEL ET LE RAFFRAÎCHISSEMENT DE L'ÉCRAN

D'après le modèle, les images produites dans un même cycle du processeur visuel sont confondues. L'utilisateur a donc l'illusion d'un dessin animé si le taux d'images produites est supérieur à $1/\tau_s$. En choisissant pour τ_s la valeur moyenne de 100ms, le système répond aux conditions s'il est capable de produire au moins 10 images/s. La satisfaction de cette contrainte dépend, pour un matériel donné, des techniques de réalisation.

John Uebbing et Charles Young montrent dans [Uebbing 86] comment combiner judicieusement la technique de la programmation par objets et celle des langages algorithmiques traditionnels. En effet, il est souvent reproché aux langages à objets le coût de la communication par message. Dans le cas qui nous intéresse, l'objectif est de déterminer les conditions où ces coûts sont compatibles avec les performances du système visuel. Connaissant :

- τ_m , le temps de transfert d'un message entre deux objets, et
- n le nombre d'objets graphiques constituant d'un objet O ,

l'affichage de O implique un temps $t = n \cdot \tau_m$ de communication par message. Si t dépasse un seuil, fonction de τ_s , alors il est souhaitable de :

- 1) Redéfinir l'organisation structurelle de O afin de réduire le nombre de messages nécessaires à l'affichage de O .
- 2) Produire tout ou partie de O en appliquant des techniques brutales comme l'usage de l'assembleur.

Le rapport entre le coût de transfert d'un message et le cycle du processeur visuel nous permet de constater que le signal d'un événement souris est tout à fait réalisable en utilisant la communication par message. A titre d'illustration, X Window [Scheifler 86], le système de fenêtrage conçu à l'origine dans le cadre du projet Athena [Balkovich 85], est capable de gérer sur un réseau local l'affichage de rectangles élastiques avec des performances telles, que l'utilisateur n'imagine pas l'existence d'un ordinateur distant.

1.6.2.2 LA LOI DE FITTS ET LA SOURIS DU STAR

[Card 83] (p. 252-253) rapporte une anecdote intéressante sur l'application de la loi de Fitts à la conception matérielle de la souris du Star [Smith 82].

Initialement, les concepteurs du Star souhaitaient que la vitesse maximale du suivi du curseur fût de 50 cm/s pour un écran de 35 cm de diagonale. Les évaluations effectuées sur la base du modèle ont révélé que la vitesse envisagée était de 2 à 3 fois inférieure à la vitesse théorique de la main. Le délai entre les mouvements de la souris et ceux du curseur, se serait traduit, pour l'utilisateur, en une difficulté réelle à désigner rapidement et avec précision. La conception matérielle de la souris du Star a donc été revue. Les calculs qui ont permis d'aboutir à ces décisions se présentent comme suit.

Au $n^{\text{ième}}$ micromouvement, la distance X_n qui sépare le curseur de la cible située initialement à une distance D s'exprime avec l'égalité $X_n = \varepsilon^n \cdot D$ où ε est la constante d'erreur entre deux micromouvements introduite au paragraphe 1.3. Puisque $\varepsilon < 1$, la vitesse maximale V_{max} de la main agissant sur la souris est atteinte au premier micromouvement. Sachant que le temps τ d'exécution d'un micromouvement est la somme des temps de cycle des trois processeurs sensoriel, cognitif, moteur : $\tau = \tau_s + \tau_c + \tau_m$

$$V_{max} = \frac{(X_0 - X_1)}{\tau} = \frac{(D - \varepsilon \cdot D)}{\tau} = D \cdot \frac{(1 - \varepsilon)}{\tau} \quad (8)$$

En prenant $\varepsilon = 0.07$, $D = 35\text{cm}$ c'est-à-dire la diagonale de l'écran, $\tau = 240\text{ms}$ c'est-à-dire $\tau_s = \tau_c = \tau_m =$ une valeur moyenne de 70ms , on obtient :

$$V_{max} = 35 \cdot \frac{1-0.07}{240} = 0.135 \text{ cm/ms} = 135 \text{ cm/s} \quad (9)$$

soit de l'ordre de deux fois la vitesse initialement envisagée pour le suivi du curseur.

En prenant pour τ la valeur minimale estimée (celle qui correspond à un individu rapide, soit 190ms), $V_{max} = 171 \text{ cm/s}$, soit à peu près trois fois la vitesse prévue.

1.6.2.3 LES LIMITES DE LA MÉMOIRE COGNITIVE

Si un utilisateur lit une suite d'informations arbitraires, par exemple une quinzaine de noms de commande dans une langue inconnue, comment cet utilisateur va-t-il se comporter sur le plan mémorisation immédiatement 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 δ_s , δ_c et δ_{lt} des mémoires :

- $\delta_s = 200 \text{ ms}$ pour la mémoire sensorielle,
- $\delta_c [3 \text{ mnèmes}] = 7 \text{ s}$ pour la mémoire à court terme, et
- $\delta_{lt} = \infty$ pour la mémoire à long terme.

Puisque, pour l'utilisateur, les mots n'ont pas de relation (la langue est inconnue), chaque mot constitue un mnème. La liste comportant plus de 7 mnèmes, il est clair que l'utilisateur va commettre des oublis. Les mots oubliés seront ceux du milieu de la liste. Pourquoi cela ?

Les premiers mots présentés, s'ils ont été mémorisés, sont accessibles depuis la mémoire à long terme car, au-delà de δ_c , 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.

Cette analyse prédit le comportement de l'utilisateur immédiatement après avoir cessé de consulter la liste des noms de commande. Si on le soumet au même test quelques dizaines de secondes plus tard, les premiers noms de la liste seront tout aussi bien retrouvés (puisque $\delta_{lt} = \infty$) ; en revanche, les derniers mots ne seront pas retrouvés avec autant de succès que précédemment : l'utilisateur, dans son effort de recherche, aura réactivé de nouveaux mnèmes qui auront interféré avec les derniers éléments de la liste.

Des expérimentations ([Card 83] p.77) confirment cette analyse. La conséquence, pour le concepteur d'interface, est de suppléer aux limitations de la mémoire à court terme. Une première technique, celle des menus et des formulaires, constitue une extension satisfaisante à titre de support syntaxique. Un second moyen concerne le choix d'une terminologie adaptée qui facilite l'association de mnèmes et qui donc réduit le nombre d'éléments arbitraires isolés.

1.6.3 Les limites du modèle

Les quelques exemples qui précèdent démontrent l'utilité du modèle pour expliquer et prédire les performances d'un utilisateur. Ces performances, bien qu'évaluées de manière approximative, permettent de répondre à des questions précises. Les réponses permettent à leur tour de déterminer les contraintes qui devront être respectées lors de la réalisation du système.

Le modèle présente cependant deux sortes de limitations :

- 1) Les questions possibles concernent les performances motrices et perceptuelles mais ne peuvent se rapporter aux structures cognitives du sujet humain. Le modèle passe sous silence les processus cognitifs les plus fondamentaux : 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 concepteur n'a donc pas la possibilité de vérifier que l'organisation du système qu'il vient de créer est compatible avec la structure cognitive de l'utilisateur.
- 2) Le Modèle du Processeur Humain ne véhicule pas de méthode de conception. Aucun élément du modèle n'indique comment satisfaire les contraintes de performance qu'il permet cependant de déduire. Les modèles GOMS et Keystroke présentés au chapitre suivant tentent de combler cette lacune.

1.7 RÉSUMÉ DU CHAPITRE

Le Modèle du Processeur Humain présente le sujet humain comme un système de traitement d'informations. Ce système comprend trois sous-systèmes interdépendants (sensoriel, moteur et cognitif) munis chacun d'un processeur et d'une mémoire. Le cycle de base de ces processeurs est lent (70 ms), la capacité de la mémoire à court terme est faible (7 ± 2 mnèmes), et les performances motrices qui interviennent dans l'interaction homme-ordinateur peuvent s'exprimer formellement avec la loi de Fitts.

Le Modèle du Processeur Humain constitue un cadre général de pensée et un point de départ culturel simple dans le domaine de l'ergonomie cognitive. Sa représentation de l'individu sous forme d'un système de traitement de l'information sert de paradigme à la psychologie contemporaine. Cette vision et la théorie des systèmes de production sur laquelle le modèle s'appuie, ont contribué pour une large part à rénover les techniques de laboratoire de la psychologie traditionnelle.

La simplicité du modèle a sa contrepartie. Le Modèle du Processeur Humain ne fournit aucune indication sur les représentations mentales alors que ces structures ont une incidence directe sur le comportement ou la prédiction de ce comportement. Bien qu'il se situe à un haut niveau d'abstraction, il se limite à la prédiction ou à l'explication de phénomènes de performance de bas niveau. Le Modèle du Processeur Humain est donc trop réducteur pour assister l'informaticien dans sa tâche de conception de systèmes interactifs. En fait, il ne véhicule aucune méthode de conception.

BIBLIOGRAPHIE

- [Adobe 85] Adobe, *PostScript Language Reference Manual*, Addison Wesley, Reading, Mass., (1985)
- [Acetta 86] M. Acetta, R. Baron, W. Bolowsky, D. Golub, R. Rashid, A. Tevanian, M. Young, *Mach, A New Kernel Foundation for UNIX Development*, Carnegie-Mellon University Technical Report, (May 1986)
- [Ahlers 86] K.L. Ahlers, A. Dwelly, OUTILS, *Towards a User Interface Management System for Graphical Interaction*, Technical Report ECRC (European Computer-Industry Research Centre), (October 1986)
- [Amodeus 89] *Assimilating Models of Designers, Users and Systems*, ESPRIT Basic Research Action Project 3066, Technical Annex, (1989)
- [Anderson 83] J.R. Anderson, *The Architecture of Cognition*, Harvard University Press, Cambridge, Massachusetts, (1983)
- [Balkovich 85] E. Balkovich, S. Lerman, R.P. Parmelee, Computing in Higher Education, The Athena Experience, *Comm. ACM*, vol. 28,11 (1985), pp. 1214-1224
- [Barnard 81] P.J. Barnard, N.V. Hammond, J. Morton, J.B. Long, I.A. Clark, Consistency and Compatibility in Human-Computer Dialogue, *International Journal of Man-Machine Studies*, vol. 15 (1981), pp. 87-134
- [Barnard 86] P.J. Barnard, *Cognitive Resources and the Learning of Human-Computer Dialogue*, MRC Applied Psychology Unit, 15 Chaucer Road, Cambridge, England, Technical report, (March 1986)
- [Barnard 87] P.J. Barnard, M. Wilson, A. MacLean, Approximate Modelling of Cognitive Activity, Towards an Expert System Design Aid, *CHI+GI Conference Proceedings*, ACM, New York, (April 1987), pp. 21-26
- [Barth 86] P. S. Barth, An Object-Oriented Approach to Graphical Interfaces, *ACM Transactions on Graphics*, vol. 5,2 (April 1986)
- [Barthet 86] M. F. Barthet, C. Sibertin-Blanc, La modélisation d'applications interactives adaptées aux utilisateurs par des réseaux de Petri à structure de donnée, *Actes du Troisième Colloque-Exposition de Génie Logiciel*, AFCET, (Mai 1986), pp. 117-136
- [Bass 88] L. Bass, E. Hardy, K. Hoyt, R. Little, R. Seacord, *The Serpent run time architecture and dialogue model*, Carnegie Mellon Technical Report, CMU/SEI-88-TR-6 (January 1988)
- [Bernard 87] S. Bernard, L. Lys, P. Marchal, C. Po, Projet MOUSE, *Outils pour la Construction d'Interfaces Homme-Machine, Spécifications externes, Architecture logicielle*, Rapports de projet DESS-GI, Université Joseph Fourier de Grenoble, (Septembre 1987)
- [Berry 86a] G. Berry, F. Boussinot, P. Couronné, G. Gonthier, *ESTEREL V2.2 System Manuals*, Rapports techniques, Ecole des Mines, Sophia Antipolis, (1986)

BIBLIOGRAPHIE

- [Berry 86b] G. Berry, P. Couronné, G. Gonthier, Synchronous Programming of Reactive systems, an introduction to ESTEREL, *Proceedings of the First France-Japan Symposium on Artificial Intelligence and Computer Science*, North Holland, (October 1986)
- [Binding 89] C. Binding, S. Schmandt, K. Lantz, M. Arons, Workstation audio and window based graphics : similarities and differences, *Proceedings of the 2nd Working Conference IFIP WG 2.7*, Napa Valley, (1989), pp. 120-132
- [Bly 86] S. Bly, J.K. Jarret, A Comparison of Tiled and Overlapping Windows, *Computer Human Interaction Conference Proceedings*, ACM, New York, (April 1986), pp. 101-106
- [Bobrow 83] D.G. Bobrow, M. Stefik, *The Loops Manual*, Tech. report KB-VLSI-81-13, Knowledge Systems Area, Xerox, Palo Alto Research Center, (1981)
- [Borning 86a] A.H. Borning, Graphically Defining New Building Blocks in ThingLab, *Human Computer Interaction*, vol. 2,4 (1986), pp. 269-295
- [Borning 86b] A.H. Borning, Defining Constraints Graphically, *Computer Human Interaction Conference Proceedings*, ACM, New York, (April 1986), pp. 137-143
- [Borras 87] P. Borras, D. Clément, T. Despeyroux, J. Incerpi, G. Kahn, B. Lang, V. Pascual, *CENTAUR, the system*, Rapport de Recherche INRIA no 777, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, (Décembre 1987)
- [BULL 85] *Introduction générale à EMERAUDE*, Société Bull, Route de Versailles, Louveciennes, (1985)
- [Buxton 82] W. Buxton, An Informal Study of Selection Positioning Tasks, *Graphics Interface'82*, (1982), pp. 323-328
- [Buxton 83] W. Buxton, M.R. Lamb, D. Sherman, K.C. Smith, Towards a Comprehensive User Interface Management System, *Computer Graphics*, vol. 17, 3 (July 1983), pp. 35-42
- [Card 83] S. Card, T. Moran, A. Newell, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, (1983)
- [Cardelli 85] L. Cardelli, R. Pike, Squeak, a Language for Communicating with Mice, *ACM Computer Graphics*, vol. 19,3 (1985), pp. 199-204
- [Cardelli 87] L. Cardelli, *Building User Interfaces by Direct Manipulation*, Digital Systems Research Center, Technical Report 22, (October 1987)
- [Carroll 83] J.M. Carroll, Presentation and Form in User Interface Architecture, *Byte*, vol. 8,12 (December 1983), pp. 113-122
- [Carroll 84] J.M. Carroll, C. Carrithers, Training Wheels in a User Interface, *Comm. ACM*, vol. 27,8 (August 1984), pp. 800-807
- [Carroll 85a] J.M. Carroll, D.S. Kay, Prompting, Feedback and Error Correction in the Design of a Scenario Machine, *Proceedings of the CHI'85 Conference*, ACM, (April 1985), pp. 149-153
- [Carroll 85b] J.M. Carroll, R.L. Mack, Metaphor, Computing Systems, and Active Learning, *International Journal of Man-Machine Studies*, vol. 22,1 (January 1985), pp. 39-57

BIBLIOGRAPHIE

- [Ciccarelli 84] E.C. Ciccarelli, *Presentation Based User Interfaces*, Technical Report 794, Artificial Intelligence Laboratory, Massachusetts Intelligence Laboratory, (August 1984)
- [Clément 88] D. Clément, J. Incerpi, *Specifying Behavior of Graphical Objects Using Esterel*, Rapport de Recherche INRIA no 836, (Avril 1988)
- [Cohen 86] E.S. Cohen, E.T. Smith, L.A. Iverson, Constraint-Based Tiled Windows, *IEEE Computer Graphics and Applications*, vol. 6,5 (May 1986), pp. 35-45
- [Collet 87] C. Collet, *Les Formulaires Complexes dans les Bases de Données Multimédia*, Thèse de Doctorat de l'Université Scientifique Technologique et Médicale de Grenoble, (Novembre 1987)
- [Conchon 86] A. Conchon, J. Camacho, A.M. Rasser, Une session sur le poste de travail Concerto, *Actes du 3ème Colloque-Exposition de Génie Logiciel*, AFCET, (Mai 1986), pp. 57.68
- [Conklin 87] J. Conklin, Hypertext, An Introduction and Survey, *IEEE Computer*, vol. 20,9 (September 1987), pp. 17-41
- [Coutaz 84] J. Coutaz, M. Herrmann, Adèle et le Médiateur-Compositeur ou Comment rendre une Application Interactive indépendante de l'Interface Usager, *Actes du deuxième colloque de Génie Logiciel*, AFCET, (Juin 1984), pp. 195-212
- [Coutaz 85a] J. Coutaz, A Layout Abstraction for User System Design, *ACM SIGCHI*, (January 1985), pp. 18-24. Paru également dans Carnegie-Mellon University Technical Report, CMU-CS-84-167, (December 1984)
- [Coutaz 85b] J. Coutaz, Abstractions for User Interface Design, *IEEE Computer*, vol. 18, 9 (September 1985), pp. 21-34
- [Coutaz 86a] J. Coutaz, Abstractions for User Interface Toolkits, *Foundation for Human-Computer Communication*, K. Hopper et I.A. Newman eds., North Holland, (1986), pp. 335-354
- [Coutaz 86b] J. Coutaz, Abstractions pour la construction d'interfaces Homme-Machine, *Techniques et Science Informatiques*, vol. 5, 4 (Juillet-août 1986), pp. 239-250
- [Coutaz 87 a] J. Coutaz, F. Berthier, The construction of User Interfaces, *Office Systems, Methods and Tools*, G. Bracchi, D. Tsihrizis eds., North Holland, (1987), pp. 59-66
- [Coutaz 87b] J. Coutaz, The Construction of User Interface and the Object Oriented Paradigm, *Proceedings of ECOOP'87, The European Conference on Object Oriented Programming*, (June 1987), pp. 135-144
- [Coutaz 87c] J. Coutaz, PAC, an Implementation Model for Dialog Design, *Proceedings of the Interact'87 conference*, H.J. Bullinger, B.Shackel ed., North Holland, (September 1987), pp. 431-436
- [Coutaz 88a] J. Coutaz, De l'ergonome à l'informaticien, pour une méthode de conception et de réalisation des interfaces homme-machine, Conférence invitée, *Actes du colloque européen ERGO-IA'88, Ergonomie et Intelligence Artificielle*, (octobre 1988)
- [Coutaz 88b] J. Coutaz, *Interface Homme-Ordinateur, Conception et Réalisation*, Thèse de doctorat d'état, Université Joseph Fourier de Grenoble, (Décembre 1988)

BIBLIOGRAPHIE

- [Coutaz 89a] J. Coutaz, Architecture Models for Interactive Software, *Proceedings of ECOOP'89, The European Conference on Object Oriented Programming*, (July 1989)
- [Coutaz 89b] J. Coutaz, UIMS, Promises, Failures and Trends, *HCI'89, People and Computers V*, A. Sutcliffe, L. Macauley eds., Cambridge University Press, (1989), pp. 71-84
- [Crowley 86] J. Crowley, J. Coutaz, Navigation et Modélisation pour un Robot Mobile, *Techniques et Science Informatiques*, vol. 6,4 (Septembre-octobre 1986)
- [Duce 87] D.A. Duce, F.R.A. Hopgood, The Graphical Kernel System, *Computer-Aided Design*, vol. 19,8 (October 1987), pp. 396-409
- [Duisberg 86] R.A. Duisberg, Animated graphical interfaces using temporal constraints, *Computer Human Interaction Conference Proceedings*, ACM, New York, (April 1986) pp. 131-136
- [Ege 87] R.K. Ege, D. Maier, A. Borning, The Filter Browser Defining Interfaces Graphically, *Proceedings of ECOOP'87, The European Conference on Object Oriented Programming*, (June 1987), pp. 155-165
- [Estublier 83] J. Estublier, S. Krakowiak, J. Mossière, Y. Rouzaud, Design Principles of the Adèle Programming Environment, *International Computing Symposium*, (ACM), (March 1983), pp. 149-156
- [Estublier 86] J. Estublier, N. Beltakir, Experience with a Data Base of Programs, *Proceedings of the Second Symposium on Practical Software Environments Sigplan/Sigsoft*, (December 1986)
- [Foley 81] J.D. Foley, W.L. Wallace, P. Chan, *The Human Factors of Graphic Interaction Tasks and Techniques*, Technical Report GWU-IIST-81-3, Institute for Information Science and Technology, The George Washington University, (1981)
- [Foley 84] J. D. Foley, A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Addison Wesley, (1984)
- [Foley 87] J.D. Foley, W.C. Kim, C.A. Gibbs, Algorithms to Transform the Formal Specification of a User-Computer Interface, *Conference Proceedings of Human-Computer Interaction Interact'87*, H.J. Bullinger, B.Shackel ed., North Holland, (1987), pp. 1001-1006
- [Foley 88] J.D. Foley, W.C. Kim, S. Kovacevic, K. Murray, *The User Interface Design Environment*, Technical Report GWU-IIST-88-04, Department of Electrical Engineering and Computer Science, The George Washington University, Washington D.C. 20052, (January 1988)
- [Forgy 84] C.L. Forgy, *The OPS83 Report*, Technical Report CMU-CS-84-113, Carnegie-Mellon University, Computer Science, (1984)
- [Goldberg 84] A. Goldberg, Smalltalk-80, *The Interactive Programming Environment*, Addison-Wesley, (1984)
- [Gosling 86a] J. Gosling, Partitioning of Functions in Window Systems, *Methodology of Window Management*, Hopgood ed., Springer Verlag, (1986), pp. 101-106

- [Gosling 86b] J. Gosling, D. Rosenthal, A Window Manager for BitMapped Displays and Unix, *Methodology of Window Management*, Hopgood ed., Springer Verlag, (1986), pp. 115-128
- [Gosling 89] J. Gosling, D.S.H. Rosenthal, M. Arden, *The NeWS Book, An Introduction to the Network/extensible Window System*, Springer-Verlag, (1989)
- [Gould 88] J.D. Gould, How to Design Usable Systems, *Handbook of Human-Computer Interaction*, M. Helander ed., Elsevier Science B. V. (North Holland), (1988), pp. 757-789
- [Granor 86] T.E. Granor, *A User Interface Management System Generator*, PhD Thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA. Technical Report MS-CIS-86-42 Graphics Lab 12, (May 1986)
- [Green 85] M.W. Green, *The Design of Graphical Interfaces*, PhD Thesis, Tech. Report CSRI-170, Computer Systems Research Institute, University of Toronto, Canada, M5S 1A1, (April 1985)
- [GSPC 79] Status Report of the Graphics Standards Committee, *Computer Graphics*, vol. 13, 3 (August 1979)
- [Halasz 87] F.G. Halasz, T.P. Moran, R.H. Trigg, NoteCards in a Nutshell, *CHI+GI Conference Proceedings*, ACM, New York, (April 1987), pp. 45-53
- [Harel 87] D. Harel, Statecharts, A visual formalism for complex systems, *Science of Computer Programming*, vol. 8,3 (June 1987), pp. 231-274
- [Harvey 88] G. Harvey, *Understanding HyperCard for Version 1.1*, Sybex Books Publishers, (1988)
- [Hayes 83] P.J. Hayes, P. Szekely, Graceful Interaction through the Cousin Command Interface, *International Journal of Man Machine Studies*, vol. 19,3 (September 1983), pp. 285-305
- [Hayes 85] P.J. Hayes, P. Szekely, R. Lerner, Design Alternatives for User Interface Management Systems Based on Experience with Cousin, *Proceedings of the CHI'85 Conference*, ACM, (April 1985), pp. 169-175
- [Hayes-Roth 79] B. Hayes Roth, F. Hayes-Roth, A Cognitive Model for Planning, *Cognitive Science*, vol. 3 (1979), pp. 275-310
- [Helfman 87] J.I. Helfman, Panther, A Specification System for Graphical Controls, *CHI+GI Conference Proceedings*, ACM, (April 1987), pp. 279-284
- [Henderson 86] A. Henderson, The Trillium User Interface Design Environment, *Proceedings SIGCHI'86 Human Factors in Computing Systems*, ACM, (April 1986), pp. 221-227
- [Herrmann 84] M. Herrmann, *Interface Usager-Application dans un Atelier de Génie Logiciel*, Thèse Doctorat de Troisième Cycle, Université Scientifique Technologique et Médicale de Grenoble, (Octobre 1984)
- [Hibbard 84] P. Hibbard, *Mint User's Manual*, Carnegie-Mellon University, (1984)
- [Hill 87a] R.D. Hill, *Supporting Concurrency, Communication and Synchronization in Human-Computer Interaction*, PhD Thesis, Department of Computer Science, University of

BIBLIOGRAPHIE

- Toronto, January, 1987 ; voir aussi, Supporting Concurrency, Communication and Synchronization dans Human-Computer Interaction-The Sassafras UIMS, *ACM Transactions on Graphics*, vol. 5, 2 (April 1986), pp. 179-210
- [Hill 87b] R.D. Hill, Event Response Systems-A Technique for Specifying Multi-thread Dialogues, *Proceedings of the CHI+GI'87 Conference*, ACM, (1987), pp. 241-248
- [Hoare 78] C.A.R. Hoare, Communicating Sequential Processes, *Comm. of the ACM*, vol. 21, 8 (1978), pp. 666-678
- [Hodges 89] M. E. Hodges, R. M. Sasnett, M. S. Ackerman, A Construction Set for Multimedia Applications, *IEEE Software*, (January 1989), pp. 37-43
- [Hullot 86] J.M. Hullot, SOS Interface, un Générateur d'Interfaces Homme-Machine, *Actes des Journées Afcet-Informatique sur les Langages Orientés Objet, Bigre+Globule*, vol. 48, IRISA, Campus de Beaulieu, 35042 Rennes, (Janvier 1986), pp. 69-78
- [Hutchins 86] E. L. Hutchins, J. D. Hollan, D. A. Norman, *Direct Manipulation Interfaces, User Centered System Design*, Lawrence Erlbaum Associates, Publishers, (1986), pp. 87-124
- [Ilog 89] Ilog, Aïda, *Environnement de développement d'applications*, Manuel de référence Aïda, Version 1.32, Copyright Ilog, (Mars 1989)
- [Ingalls 88] D.H. Ingalls, S. Wallace, Y. Chow, F. Ludolph, K. Doyle, Fabrik, A visual Programming Environment, *OOPSLA'88 proceedings*, (1988), pp. 176-190
- [ISO 85] International Organization for Standardization, *Information Processing Systems - Computer Graphics - Graphical Kernel System (GKS) functional description*, ISO IS 7942, (July 1985)
- [ISO 86a] International Organization for Standardization, *Information Processing Systems - Computer Graphics - Programmer's Hierarchical Interface to Graphics (PHIGS) functional description*, ISO DP 9592, (October 1986)
- [ISO 86b] International Organization for Standardization, *Information Processing Systems - Computer Graphics - Techniques for interfacing graphical devices (CGI) functional description*, ISO DP 9636, (December 1986)
- [Krakowiak 87] S. Krakowiak, M. Meysembourg, M. Riveill, C. Roisin, *Modèle d'Objet et Langage du Système Guide*, Rapport Guide R2, Laboratoire de Génie Informatique (IMAG), (Novembre 1987)
- [Jacob 84] R.J.K. Jacob, An Executable Specification Technique for Describing Human-Computer Interaction, *Advances in Human Computer Interaction*, H.R. Hartson, ed. Alex Publishing Co., (1984)
- [John 85] B.E. John, P.S. Roseblom, A Theory of Stimulus-Response Compatibility Applied to Human Computer Interaction, *CHI'85 Conference Proceedings*, ACM, New York, (April 1985), pp. 213-220
- [John 87] B.E. John, A. Newell, Predicting the Time Recall Computer Command Abbreviations, *CHI+GI Conference Proceedings*, ACM, New York, (April 1987), pp. 33-40

BIBLIOGRAPHIE

- [Jones 82] M.B. Jones, R.F. Rashid, M. Thompson, Sesame, *The Spice File System*, Tech. Report, Carnegie-Mellon University, (October 1982)
- [Karsenty 87] S. Karsenty, *Graffiti, un outil interactif et graphique pour la construction d'interfaces homme-machine adaptables*, Thèse de doctorat de 3ème cycle Informatique, Université de Paris-Sud, Centre d'Orsay, (Décembre 1987)
- [Kieras 85] D. Kieras, P.G. Polson, An Approach to the Formal Analysis of User Complexity, *International Journal of Man-Machine Studies*, 22 (1985), pp. 365-394
- [Kiger 84] J.I. Kiger, the depth/breadth trade-off in the design of menu-driven user interfaces, *International Journal of Man-Machine Studies*, 20 (1984), pp. 201-213
- [Knuth 79] D.E. Knuth, *TEX and Metafont*, New Directions in Typesetting, Digital Press, (1979)
- [Krakowiak 87] S. Krakowiak, M. Meysembourg, M. Riveill, C. Roisin, *Modèle d'Objet et Langage du Système Guide*, Rapport Guide R2, Laboratoire de Génie Informatique (IMAG), (Novembre 1987)
- [Lantz 86] K.A. Lantz, *On User Interface Reference Models*, ACM SIGCHI Bulletin, vol. 18, 2 (1986), pp. 36-44
- [Lee 89] K. Lee, *Automatic Speech Recognition*, Kluwer Academic, (1989)
- [Lieberman 85] H. Lieberman, There's More to Menu Systems than Meets the Screen, SIGGRAPH'85, *Computer Graphics*, vol. 19,3 (1985), pp. 181-189
- [Lieberman 87] H. Lieberman, Reversible Object-Oriented Interpreters, ECOOP'87, *The European Conference on Object Oriented Programming*, (June 1987), pp. 13-22
- [Linton 89] M.A. Linton, J.M. Vlissides, P.R. Calder, Composing User Interfaces with Interviews, *IEEE Computer*, (February 1989), pp. 8-22
- [Lopez 83] M. Lopez, J. Palazzo, F. Velez, *The Tigre Data Model*, Rapport de Recherche IMAG/Centre de Recherche Bull RR-Tigre no 2, (Novembre 1983)
- [Lunati 88] J.M. Lunati, V. Normand, *Un Serveur d'Interaction Centré Objet*, Projet de fin d'études, Institut National Polytechnique de Grenoble, E.N.S.I.M.A.G, (Juin 1988)
- [Lynch 86] G. Lynch, J. Meads, In Search of a User Interface Reference Model, Report on the SIGCHI workshop on User Interface Reference Models, *SIGCHI Bulletin*, vol. 18, 2 (October 1986), pp. 25-33
- [Maraninchi 90] F. Maraninchi, *Argos, Un langage graphique pour la conception, la description et la validation des systèmes réactifs*, Thèse de doctorat, Université Joseph Fourier, Grenoble, (Janvier 1990)
- [Meyer 87] B. Meyer, Reusability, The Case for Object-Oriented design, *IEEE Software*, (March 1987), pp. 50-59
- [Mical 86] R.J. Mical, S. Deyl, *Amiga Intuition Reference Manual*, Commodore Business Machines, Inc., Addison Wesley, (1986)
- [Michard 82] A. Michard, Graphical presentation of boolean expressions in a data base query language, design notes and an ergonomic evaluation, *Behaviour and Information Technology*, vol. 1,3 (1982), pp. 279-288

BIBLIOGRAPHIE

- [Mikelsons 81] M. Mikelsons, Prettyprinting in an Interactive Programming Environment, *Proceedings of the ACM Sigplan SIGOA Symposium on Text Manipulation*, (June 1981)
- [Miller 75] G.A. Miller, *The Psychology of Communication*, Basic Books, New York, second edition, (1975)
- [Miller 81] D.P. Miller, The depth/breadth trade-off in hierarchical computer menus, *Proceedings of the 25th Annual Meeting of the Human Factors Society*, (1981), pp. 296-300
- [MIT 87] X Toolkit Library - C Language interface, X protocol Version 11, (1987)
- [Moran 81] T. Moran, The Command Language Grammar, a representation for the user interface of interactive computer systems, *International Journal of Man-Machine Studies*, vol. 15, (1981), pp. 3-50
- [Moran 83] T.P. Moran, Getting into a System, External-Internal Task Mapping Analysis, *Computer Human Interaction'83, ACM SIGCHI Bulletin special issue*, (1983), pp. 45-49
- [Morcos 86] E. Morcos-Chounet, A. Conchon, PPML, A general formalism to specify pretty-printing, *Proceedings of IFIP Congress 1986*, Dublin, North Holland, (1986)
- [Morris 86] J.H. Morris, M. Satyanarayanan, M.H. Conner, J.H. Howard, D.S.H. Rosenthal, F.D. Smith, Andrew, A Distributed Personal Computing Environment, *Comm. of the ACM*, vol. 29, vol. 3 (March 1986), pp. 184-201
- [Myers 84] B.A. Myers, The User Interface of Sapphire, *IEEE Computer Graphics and Applications*, vol. 4,12 (December 1984), pp. 13-23
- [Myers 87a] B.A. Myers, Creating dynamic interaction techniques by demonstration, *Proceedings of ACM CHI+GI'87 Conference Human Factors in Computing Systems and Graphics Interface*, (April 1987), pp. 271-278
- [Myers 87b] B.A. Myers, *Creating User Interfaces by Demonstration*, PhD thesis, Department of Computer Science, University of Toronto, Technical Report CSRI-196, Computer Systems Research Institute, (May 1987)
- [Nanard 84] J. et M. Nanard, Manipulation Interactive de Documents, *Techniques et Science Informatiques*, vol. 3, 6 (1984), pp. 443-451
- [Nelson 85] G. Nelson, Juno, a Constraint-Based Graphics System, *Computer Graphics, SIGGRAPH'85 Conference Proceedings*, vol. 19,3 (July 1985), pp. 235-243
- [Newell 86] A. Newell, S. Card, Straightening Out Softening Up, Response to Carroll and Campell, *Human Computer Interaction*, vol. 2, 3 (1986), pp. 251-267
- [Norman 86] D. A. Norman, S. W. Draper, *User Centered System Design*, Lawrence Erlbaum Associates, Publishers, (1986)
- [Olsen 83] D.R. Olsen, E.P. Dempsey, Syngraph, A Graphical User Interface Generator, *Computer Graphics*, (July 1983), pp. 43-50

- [Olsen 89] D. R. Olsen, A Programming Language Basis for User Interface Management, *CHI'89 Conference Proceedings, special issue of the SIGCHI Bulletin*, ACM Press, (1989), pp.171-176
- [OSF 89] *OSF/Motif, Programmer's Reference Manual, Revision 1.0*, Open Software Foundation, Eleven Cambridge Center, Cambridge, MA 02142, (1989)
- [Pato 84] J.N. Pato, S.P. Reiss, M.H. Brown, *The Brown University Environment*, Technical Report CS-84-03, Department of Computer Science, Brown University, Providence, RI 02912, USA, (January 1984)
- [Payne 86] S.J. Payne, T.R.G. Green, Task-Action Grammars, A model of the mental representation of task languages, *Human Computer Interaction*, vol. 2 (1986), pp. 93-133
- [Petoud 89] I. Petoud, Y. Pigneur, Des spécifications fonctionnelles à l'interface-utilisateur sans programmation, *Premier Colloque sur l'Ingénierie des Interfaces Homme-Machine*, Sophia Antipolis, (Mai 1989)
- [Pfaff 85] *User Interface Management Systems*, G. E. Pfaff ed., Eurographics Seminars, Springer-Verlag, (1985)
- [Polson 85] P.G. Polson, D.E. Kieras, A Quantitative Model of the Learning and Performance of Text Editing Knowledge, *CHI'85 Conference Proceedings*, ACM, New York, (April 1985), pp. 207-212
- [Price 83] L.A. Price, C.A. Cordova, Use of Mouse Buttons, *CHI'83 Conference Proceedings*, ACM, New York, (1983), pp. 263-266
- [Quint 83] V. Quint, Un système interactif pour la production des documents mathématiques, *Techniques et Science Informatiques*, vol. 2, 3 (1983), pp. 179-190
- [Quint 85] V. Quint, I. Vatton, Grif, *un Editeur Interactif Structuré*, Rapport de Recherche TIGRE no 27, IMAG-Laboratoire de Génie Informatique, BP 53 X, 38042 Grenoble Cedex, (1985)
- [Quint 87] V. Quint, *Une Approche de l'Edition Structurée des Documents*, Thèse de doctorat d'Etat, Université Scientifique Technologique et Médicale de Grenoble, (1987)
- [Reisner 82] P. Reisner, Further developments toward using formal grammar as a design tool, *Proceedings of the Human Factors in Computer System Conference*, ACM, (1982), pp. 304-314
- [Robertson 81] C.K. Robertson, D.L. McCracken, A. Newell, *Experimental Evaluation of the ZOG Frame Editor*, Technical report CMU TR 81-112, Carnegie-Mellon University, 1981. Voir aussi, G.D Robertson, D.L. McCracken, A. Newell, The ZOG approach to Man-Machine Communication, *International Journal of Man-Machine Studies*, vol. 14,4 (1981), pp. 461-488
- [Rose 86] C. Rose et al., *Inside Macintosh*, Addison Wesley, (1986)
- [Rosson 83] M.B. Rosson, Patterns of Experience in Text Editing, *Proceedings of CHI'83 conference on Human Factors in Computer Systems*, ACM, (December 1983), pp. 177-183

BIBLIOGRAPHIE

- [Sacerdoti 74] E.D. Sacerdoti, Planning in a Hierarchy of Abstraction Spaces, *Artificial Intelligence*, vol. 5 (1974), pp. 115-135
- [Scapin 87] D.L. Scapin, *Guide ergonomique de conception des interfaces homme-machine*, Rapport INRIA 87, (Octobre 1987)
- [Scheifler 86] R.W. Scheifler, J. Gettys, The X Window System, *ACM Transactions on Graphics*, vol. 5, 2 (April 1986), pp. 79-109
- [Schmucker 86a] K. Schmucker, MacApp, An Application Framework, *Byte*, vol. 11, 8 (1986), pp. 189-193
- [Schmucker 86b] K. Schmucker, Object-Oriented Languages for the Macintosh, *Byte*, vol. 11, 8 (1986), pp. 177-188
- [Schulert 85] A.J. Schulert, G.T. Rogers, J.A. Hamilton, ADM - A Dialog Manager, *Proceedings of the CHI'85 Conference*, ACM, (April 1985), pp. 177-183
- [SEMS 85] *SPS7, Système Graphique, Ensemble des primitives pour l'écriture d'applications graphiques*, Bull-Sems, 20 893 759 REF 01/FR, (Juin, 1985)
- [Shafer 88] D. Shafer, *HyperTalk Programming, MacIntosh Library*, Hayden Books, Indianapolis, Indiana, USA, (1988).
- [Shneiderman 82] B. Shneiderman, Multiparty Grammars and Related Features for Defining Interactive Systems, *IEEE Transactions on Systems, Man and Cybernetics*, SMC-12, vol. 2 (1982), pp. 148-154
- [Shneiderman 87] B. Shneiderman, *Designing the User Interface, Strategies for Effective Human-Computer Interaction*, Addison Wesley Publishing Company, (1987)
- [Shuey 86] D. Shuey, D. Bailey, T.P. Morrissey, PHIGS, A Standard, Dynamic, Interactive Graphics Interface, *IEEE Computer Graphics and Application*, (August 1986), pp. 50-57
- [Shuey 87] D. Shuey, PHIGS, a graphics platform for CAD application development, *Computer-Aided Design*, vol. 19, 8 (1987), pp. 410-417
- [Sibert 86] J.L. Sibert, W.D. Hurley, T.W. Bleser, An Object Oriented User Interface Management System, *SIGGRAPH'86*, vol. 20, 4 (1986), pp. 259-268
- [Simon 84] H. A. Simon, *The Sciences of the Artificial*, The MIT Press, third edition, (1984)
- [Sisson 86] N. Sisson, Dialogue Management Reference Model, *ACM SIGCHI*, vol. 18, 2 (1986), pp. 34-35
- [Smith 82] D.C. Smith, C. Irby, R. Kimball, B. Verplank, Designing the Star User Interface, *Byte*, vol. 7, 4 (1982), pp. 242-282
- [Smith 83] R.G. Smith, Strobe, Support for structured object knowledge representation, *Proceedings of the 8th Joint Conference on Artificial Intelligence*, Karlsruhe, William Kaufman ed., (1983), pp. 855-858
- [Smith 87] R.B. Smith, Experiences with the Alternate Reality Kit, an Exemple of the Tension between Literalism and Magic, *Computer Human Interaction Conference Proceedings*, ACM, New York, (April 1987), pp. 61-67

- [Stallman 79] R.M. Stallman, *Emacs, The Extensible Customizable Self Documenting Display Editor*, MIT Artificial Intelligence Laboratory, AI Memo 519, Cambridge, Mass., (1979)
- [Stefik 87] M. Stefik, G. Foster, D.G. Bobrow, K. Hahn, S. Lanning, L. Suchman, Beyond the Chalkboard, Computer Support for Collaboration and Problem Solving in Meetings, *Comm. of the ACM*, vol. 30, 1(1987), pp. 32-47
- [SUN 85] SUN Microsystems Inc., *Programmer's Reference Manual for the Sun Window System*, SUN Microsystems Inc., 2250 Garcia Avenue, Mountain View, CA 94043, (1985)
- [Sweetman 86] D. Sweetman, A Modular Window System for Unix, dans *Methodology of Window Management*, Hopgood ed., Springer Verlag, (1986), pp. 73-79
- [Szekely 88] P.A. Szekely, B.A. Myers, A User Interface Toolkit Based on Graphical Objects and Constraints, *OOPSLA'88 Proceedings*, ACM Conference on Object-Oriented Programming Systems and Applications, San Diego, California, (1988), pp. 36-45
- [Tanner 83] P. Tanner, W. Buxton, *Some Issues in Future User Interface Management Systems (UIMS) Development*, IFIP Working Group 5.2 Workshop on User Interface Management, Seeheim, West Germany, (1983)
- [Tesler 86] L. Tesler, Programming Experiences, *Byte*, vol.11, 8 (August, 1986), pp. 195-206
- [Trigg 87] R.H. Trigg, T.P. Moran, F.G. Halasz, Adaptability and Tailorability in NoteCards, *Conference Proceedings of Human-Computer Interaction Interact'87*, H.J. Bullinger, B.Shackel ed., North Holland, (1987), pp. 723-728
- [Tufte 83] E.R. Tufte, *The Visual Display of Quantitative Information*, Graphics Press, Box 430, Cheshire, Connecticut 06410, (1983)
- [Tullis 85] T.S. Tullis, Designing a Menu-Based Interface to an Operating System, *CHI'85 Conference Proceedings*, (1985), pp. 79-84
- [Uebbing 86] J. Uebbing, C. Young, User Interface Performance Issues, *Byte*, vol. 11,8 (1986), pp. 176-176
- [Valdès 89] R. Valdès, A Virtual Toolkit for Windows and the Mac, *Byte*, vol. 14, 3 (1989), pp. 209-216
- [VIP 86] *Visual Interactive Programming*, distrib. EMDAY, 46 av. Tervieren, B-1040 Bruxelles, (1986)
- [Warnock 82] J. Warnock, D.K. Wyatt, A Device Independent Graphics Imaging Model for Use with Raster Devices, *Computer Graphics*, vol. 16, 3 (1982), pp. 313-319
- [Wasserman 85] A. Wasserman, Extending State Transition Diagrams for the Specification of Human-Computer Interaction, *IEEE Transactions on Software Engineering*, vol. 11, 8 (1985)
- [Webster 89] B. F. Webster, *The NeXT Book*, Addison Wesley, (1989)
- [Wiecha 89] C. Wiecha, W. Bennett, S. Boies, J. Gould, Generating Highly Interactive User Interfaces, *CHI'89 Conference Proceedings, special issue of the SIGCHI Bulletin*, ACM Press, (1989), pp. 277-282



BIBLIOGRAPHIE

- [Young 89] D. A. Young, *X Window Systems Programming and Applications with Xt*, Prentice-Hall, (1989)
- [Zellweger 88] P. Zellweger, D. Terry, D. Swinehart, An overview of the Etherphone system and its applications, *Proc. 2nd IEEE Computer Workstations Conf.*, Santa Clara, CA, (1988), pp. 160-168



INDEX



C

- Chunk
 - voir "Modèle du processeur humain:mnèmes"
- Comportement
 - moteur 6

I

- Introduction 1

L

- Loi de Fitts 8, 11, 13

M

- Mémoire
 - à court terme 5
 - visuelle 5
- Modèle du processeur humain 5
 - capacité 5, 9
 - cycle 5, 6, 10
 - interférence 9, 10
 - mémoire à court terme 5, 8, 14
 - mémoire à long terme 8, 9
 - mémoire sensorielle 5
 - mémoire visuelle 5
 - mnème 9, 10
 - persistance 5, 10
 - principe de rationalité
 - voir "Principe de rationalité"
 - rafraîchissement 12
 - stimuli 5
 - système cognitif 5, 8, 10
 - système moteur 5, 6
 - système sensoriel 5

P

- Principe
 - de rationalité 11