



HAL
open science

Aide à la communication pour personnes handicapés moteurs : modélisation et optimisation du système homme-machine

Souhir Ghedira Laouiti

► **To cite this version:**

Souhir Ghedira Laouiti. Aide à la communication pour personnes handicapés moteurs : modélisation et optimisation du système homme-machine. Ordinateur et société [cs.CY]. Université Paul Verlaine - Metz, 2009. Français. NNT : 2009METZ040S . tel-01752664

HAL Id: tel-01752664

<https://hal.univ-lorraine.fr/tel-01752664v1>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

THESE

Aide à la communication pour personnes handicapés
moteurs : modélisation et optimisation du système
homme-machine

Présentée dans le cadre de l'école doctorale IAEM Lorraine pour l'obtention
du grade de Docteur de l'Université Paul Verlaine - Metz
mention Automatique, Traitement du Signal et
des Images, Génie Informatique

par

Souhir GHEDIRA

Soutenue publiquement le 17 novembre 2009
devant la commission d'examen :

Rapporteurs :

M. POIRIER Franck
M. CABESTAING François

Examineurs :

M. BERGER-VACHON Christian
M. BOURHIS Guy
M. PINO Pierre
M. PRUSKI Alain

Le travail présenté dans ce mémoire a été réalisé au laboratoire d'Automatique humaine et de Sciences Comportementales (LASC) de l'Université de Paul Verlaine-Metz. Je tiens à remercier l'ensemble des personnes qui ont contribué à la réalisation de ce travail.

Je tiens tout d'abord à témoigner toute ma reconnaissance à Monsieur Guy BOURHIS, Professeur de l'Université de Metz, pour avoir été à l'origine de ce travail et pour le soutien et la confiance qu'il m'a accordé pour le mener à terme.

J'exprime mon profond remerciement à Monsieur Pierre PINO, Maître de Conférences à l'Université de Paul Verlaine-Metz, pour ses aides, ses conseils et ses encouragements qu'il m'a apporté durant ma thèse.

Je remercie Monsieur Alain PRUSKI, Professeur de l'Université de Paul Verlaine-Metz de m'avoir accueilli dans son laboratoire.

Je remercie l'ensemble des membres du jury qui m'ont fait l'honneur d'accepter de lire et de juger ce mémoire : Messieurs Franck POIRIER (Professeur 27^e, Université de Bretagne Sud, Vannes), François CABESTAING (Professeur 61^e, UST Lille) et Christian BERGER-VACHON (Professeur 61^e, Université de Lyon I).

Mes remerciements s'adressent aussi à tous les membres du laboratoire LASC, pour avoir accepté d'effectuer des expériences, pour leur soutien, leur sympathie et leur aide.

Je tiens enfin à remercier particulièrement mon mari pour son soutien et ses encouragements quotidiens et mes parents pour leur encouragement tout le long de mes études.

Table des matières

INTRODUCTION.....	6
I. CHAPITRE 1 : Etat de l'art	8
I.1. Introduction.....	8
I.2. Les aides à la communication	9
I.2.1. Introduction.....	9
I.2.2. Quelques pathologies concernées par les systèmes à défilement	10
I.2.2.1. L'infirmité Motrice Cérébrale : IMC	10
I.2.2.2. Le syndrome d'enfermement : LIS	10
I.2.2.3. La Sclérose latérale Amyotrophique : SLA	11
I.2.3. La communication directe.....	11
I.2.3.1. Introduction.....	11
I.2.3.2. L'ALPHABET	12
I.2.3.3. L'alphabet ESARIN.....	12
I.2.3.4. Code alphabétique voyelles-consonnes	12
I.2.3.5. Code alphabétique voyelles-consonnes à double entrée.....	13
I.2.3.6. Codes pictographiques	13
I.2.4. La communication indirecte	14
I.2.4.1. Capteurs de type Tout-Ou-Rien	15
I.2.4.2. Les appareils dédiés à la synthèse vocale	17
I.2.4.3. Les solutions logicielles basées sur ordinateur	20
I.2.5. Méthodes d'accélération	21
I.2.5.1. Optimisation du temps d'accès à une touche.....	21
I.2.5.2. Economie du nombre de saisies	26
I.2.6. Conclusion	29
I.3. Optimisation du temps de défilement	29
I.3.1. Introduction.....	29
I.3.2. Méthodes pour adapter le temps de défilement en ligne.....	30
I.3.2.1. Méthode basée sur un système expert.....	30
I.3.2.2. Méthode basée sur un réseau bayésien	30
I.3.2.3. Méthode basée sur le temps de réaction	33
I.3.2.4. Méthode basée sur des mesures d'erreurs de sélection.....	35
I.3.3. Méthode pour adapter le temps de défilement hors ligne	37

I.3.4. Conclusion	38
I.4. Modélisation de l'interaction humain-machine	39
I.4.1. Introduction.....	39
I.4.2. Le modèle MHP.....	40
I.4.3. Le modèle Keystroke Level Model (KLM)	41
I.4.3.1. Définition de KLM	41
I.4.3.2. Les travaux basés sur le KLM	42
I.4.4. Le modèle basé sur le canal de transmission	43
I.4.5. Le simulateur basé sur un modèle.....	45
I.5. Conclusion du chapitre 1	46
II. CHAPITRE 2 : Conception et Modélisation d'un communicateur à défilement.....	48
II.1. Introduction	48
II.2. Le système EDiTH (Environnement Digital de Téléaction pour personnes Handicapées)	
.....	49
II.2.1. L'objectif d'EDiTH	49
II.2.2. Etude critique de l'existant	50
II.2.3. Choix du langage de programmation.....	50
II.2.4. Spécifications Techniques d'EDiTH	51
II.2.5. Interface utilisateur	51
II.2.6. Les fonctionnalités du système EDiTH	53
II.2.6.1. Appeler le personnel soignant	53
II.2.6.2. Lire des textes	53
II.2.6.3. Communiquer par écrit.....	54
II.2.6.4. Communiquer verbalement	56
II.2.6.5. Accéder à un environnement multimédia.....	57
II.2.6.6. Courrier électronique	58
II.2.7. Interface de configuration.....	60
II.2.8. Conclusion	62
II.3. Acquisition et traitement des données	62
II.3.1. Fichiers logs d'EDiTH.....	63
II.3.2. Analyse des données.....	64
II.4. Modélisation de l'interaction humain-machine	65
II.4.1. Estimation des paramètres du modèle MHP.....	66
II.4.1.1. Méthode	66

II.4.1.2. Protocole expérimental	69
II.4.1.3. Résultats.....	69
II.4.2. Validation de l’hypothèse MHP	77
II.4.2.1. Méthode	77
II.4.2.2. Résultats.....	78
II.4.2.3. Conclusion	81
II.4.3. Modèle de comportement à 3 zones	81
II.4.3.1. Protocole expérimental	81
II.4.3.2. Résultats en scénario imposé.....	82
II.4.3.3. Résultats en scénario libre	85
II.4.3.4. Discussion.....	86
II.4.3.5. Conclusion	88
II.5. Conclusion du chapitre2	88
III. CHAPITRE 3 : Délai de défilement adaptatif.....	90
III.1. Introduction	90
III.2. Méthodologie	91
III.2.1. Justification	91
III.2.2. Algorithme de réglage adaptatif de T_{scan}	92
III.2.3. Réglage des paramètres	92
III.2.4. Conclusion.....	95
III.3. Résultats comparatifs	96
III.3.1. Introduction	96
III.3.2. Evolution de T_{scan} pour une personne SLA en situation écologique.....	96
III.3.3. Expérimentations en laboratoire (Sujets Valides).....	97
III.3.3.1. Introduction	97
III.3.3.2. Méthodologie	98
III.3.3.3. Résultats	98
III.3.4. Discussion	106
III.3.5. Expérimentations en situation écologique (sujet handicapé).....	106
III.3.6. Conclusion.....	109
III.4. Conclusion du chapitre3.....	109
CONCLUSION ET PERSPECTIVES	110
BIBLIOGRAPHIE	112

Liste des figures

Figure I.1 : Schéma classique de la communication [JACQ]	10
Figure I.2 : Exemple de communication directe	11
Figure I.3 : Alphabet ESARIN	12
Figure I.4 : Code alphabétique voyelles-consonnes	13
Figure I.5 : Code alphabétique voyelles-consonnes à double entrée	13
Figure I.6 : Code pictogrammes d'Axelia [AXEL].....	14
Figure I.7 : Exemple de communication indirecte	15
Figure I.8 : Exemples de capteurs Tout-Ou-Rien [HMC] [SUPP]	16
Figure I.9 : Appareils de communication à voix numérisée.....	18
Figure I.10 : Appareils de communication à voix synthétique	19
Figure I.11 : Clavier optimisé (pour l'anglais) pour un défilement ligne/colonne.....	23
Figure I.12 : Modification de la disposition des touches après chaque saisie du mot « compter » [SCHA04]	24
Figure I.13 : Le système Sibylle.....	25
Figure I.14 : Le clavier ambigu UKO	26
Figure I.15 : Le clavier DotNote avec ses deux interfaces.....	26
Figure I.16 : Un exemple de réseau bayésien utilisé dans [SIMP99]	31
Figure I.17 : Le réseau bayésien utilisé dans la seconde expérimentation : TER= taux de texte saisi, Δ TER = Variation de taux de texte saisi, Erreurs = nombre d'erreurs, Δ Err = Variation du taux d'erreurs et SD = valeur du temps de défilement	32
Figure I.18 : Menu hiérarchique symétrique avec trois alternatives et trois niveaux	34
Figure I.19 : Diagramme temporel de l'algorithme pour le temps de défilement adaptatif	34
Figure I.20 : Réglage automatique de T_{scan}	37
Figure I.21 : Réaction élémentaire à un stimulus selon le modèle MHP.....	41
Figure I.22 : Architecture du simulateur	45
Figure II.1 : Menu principal d'EDiTH « version 2005 »	50
Figure II.2 : Interface principale d'EDiTH	52
Figure II.3 : Interface pour lire du texte	54
Figure II.4 : Interface « Editeur de texte » (défilement colonne/ligne).....	55
Figure II.5 : Interface « Editeur de texte » (mode pavé numérique).....	56
Figure II.6 : Interface « Soins ».....	57
Figure II.7 : Interface « Musique\MP3».....	57
Figure II.8 : Interface « Photos/Images»	58
Figure II.9 : Interface « E-mails envoyés».....	59
Figure II.10 : Interface « Carnet d'adresses»	59
Figure II.11 : Configuration des phrases audio, de la voix et du clavier virtuel	60
Figure II.12 : Configuration du temps de défilement et du « clic long »	61
Figure II.13 : Configuration des données confidentielles et du mail	62
Figure II.14 : Exemple de fichier log en format XML.....	64
Figure II.15 : Présentation des données sous forme d'un tableau sous Excel.....	64
Figure II.16 : Histogramme des temps d'action d'un utilisateur pour un délai $T_{scan} = 600ms$.	65
Figure II.17 : Expérience pour mesurer le temps de perception T_p	66
Figure II.18 : Expérience pour mesurer le temps de cognition T_c	67
Figure II.19: Expérience pour mesurer le temps moteur T_m	68
Figure II.20 : Interface d'identification et de configuration.....	69
Figure II.21.a : Histogrammes des temps T_{act} de sujet1 et sujet2	79
Figure II.21.b : Histogrammes des temps T_{act} de sujet3 et sujet4	80

Figure II.22 : Histogrammes de T_{act} pour deux personnes parmi les six expérimentateurs valides en scénario imposé	84
Figure II.23 : Histogramme des temps d'action T_{act} pour une personne tétraplégique pour différentes valeurs de T_{scan} en scénarios imposés.....	85
Figure II.24 : Histogrammes des temps d'action T_{act} d'une personne SLA en utilisation libre sur une période d'une journée en février (à gauche) et en mars (à droite).....	86
Figure II.25 : Histogramme des temps d'action T_{act} d'un opérateur valide en scénario imposé pour $T_{scan}=600ms$	87
Figure III.1 : Analyse anticipations/erreurs (sujet 4)	94
Figure III.2 : Analyse des actions et évolution de T_{scan} (traits horizontaux) du sujet 4 ;.....	94
en abscisses : chronologie des actions ; en ordonnées : T_{act} en ms	94
Figure III.3 : Analyse anticipations/erreurs (sujet 1)	95
Figure III.4 : Evolution de T_{scan} sur une journée pour une personne SLA avec l'algorithme « anticipations/erreurs »; en abscisses : chronologie des actions; en ordonnée : T_{act}	97
Figure III.5 : Texte écrit par le sujet 1.....	99
Figure III.6.a : Chronogrammes du sujet 1. En haut : algorithme « anticipations/erreurs » avec $(nb_click_inf, nb_click_sup) = (20,25)$ et $(taux1, taux2) = (0.9, 1.3)$. En bas : algorithme de Leshet et al.	100
Figure III.6.b : Algorithme « anticipations/erreurs » (sujet 1) avec $(taux1, taux2) = (1.05, 0.95)$	101
Figure III.7 : Chronogrammes du sujet 2. En haut : algorithme anticipations/erreurs avec $(nb_click_inf, nb_click_sup) = (20,25)$. En bas : algorithme de Leshet et al.	102
Figure III.8 : Chronogrammes du sujet 3. En haut : algorithme « anticipations/erreurs » avec $(nb_clic_inf, nb_clic_sup) = (15,20)$. En bas : algorithme de Leshet et al.	103
Figure III.9 : Chronogramme du sujet 1. Algorithme de Leshet et al. sans annulation de caractère	104
Figure III.10 : Chronogramme du sujet 2. Algorithme de Leshet et al. sans annulation de caractère	105
Figure III.11 : Chronogramme du sujet 3. Algorithme de Leshet et al. sans annulation de caractère.	105
Figure III.12 : Histogrammes des temps d'action T_{act} du sujet SLA en utilisation libre sur une période d'une journée.....	108
Figure III.13 : Chronogrammes du sujet SLA : variations de T_{scan} selon l'algorithme « anticipations/erreurs »	108

INTRODUCTION

Depuis les années 1980 de nombreuses études intégrant les connaissances des sciences cognitives, de l'informatique ou de l'intelligence artificielle ont été entreprises pour concevoir des dispositifs techniques permettant à des personnes handicapées, atteintes dans leur motricité et leur capacité de production de parole, de communiquer avec leur entourage.

Ces aides à la communication sont pilotées par des dispositifs physiques qui captent des informations provenant de la personne handicapée. Il peut s'agir par exemple d'un joystick, d'une commande par souffle, d'un détecteur de mouvement de la tête ou d'un simple bouton-poussoir. Quelque soit le système de communication assistée, nous avons toujours le même principe : sélectionner les items qui vont composer le message à émettre. Ces messages sont saisis à travers un clavier simulé présenté à l'écran et sont affichés dans une zone d'édition.

Dès lors, le problème qui se pose est celui de la lenteur et de la pénibilité de la composition des messages. Ce problème majeur est particulièrement manifeste dans le cas d'un clavier virtuel piloté par une simple action tout-ou-rien sur le capteur : après chaque saisie, le sujet doit attendre que le défilement à l'écran arrive à l'item qu'il souhaite sélectionner. Il en résulte des temps d'attente très longs, qui s'accompagnent d'un effort d'attention fatigant. Le temps de défilement est en général réglé de façon empirique et n'évolue pas en fonction de l'apprentissage et/ou de l'état de fatigue de l'utilisateur.

La problématique abordée dans ce mémoire consiste alors à adapter ce temps de défilement au sujet pour optimiser sa communication. Ce réglage doit être fonction de l'état physique et cognitif de la personne au moment où il utilise l'interface de communication. Pour cela, nous cherchons à analyser l'utilisation de l'interface par le biais d'une modélisation de l'interaction personne handicapée/machine.

Dans ce travail de thèse nous nous attachons tout d'abord à concevoir une aide à la communication qui intègre des fonctionnalités essentielles de la vie quotidienne. Pour cela nous avons conçu le système EDiTH (Environnement Digital de Téléaction pour personnes Handicapées), qui est une aide technique de type «communicateur à défilement», afin d'avoir

un système ouvert, modulable et adaptable. Ceci permet en particulier d'étudier la mise en place d'une méthode d'adaptation automatique de la vitesse de défilement.

Le premier chapitre présente en premier lieu une classification des aides à la communication, puis décrit plus particulièrement les systèmes à défilement et les différentes techniques d'accélération qui visent à minimiser le nombre d'interactions entre l'utilisateur et le système. Nous abordons ensuite les méthodes proposées par quelques auteurs pour adapter le temps de défilement. Nous clôturons le chapitre par une revue des quelques modèles décrits dans la littérature concernant l'interaction entre une personne handicapée et l'aide à la communication.

Le second chapitre est tout d'abord consacré à la présentation de notre outil de travail le système EDiTH. Nous proposons ensuite une modélisation de l'interaction humain-machine pour les communicateurs à balayage déduite d'expérimentations sur des personnes valides et des personnes handicapées en scénarios imposés et en scénarios libres. Le modèle baptisé « modèle de comportement à 3 zones » est déduit du modèle du « processeur humain modèle » (MHP) de Card et al [CARD83].

Le dernier chapitre porte sur la définition et la validation expérimentale de notre algorithme de temps de défilement adaptatif. Le principe de cet algorithme est déduit de la modélisation humain/communicateur à défilement décrite au chapitre précédent. Nous présenterons tout d'abord les différents paramètres qu'il utilise. Nous décrivons et discutons ensuite les résultats des tests obtenus lors de la comparaison de notre méthode avec la méthode proposée par d'autres auteurs pour adapter le temps de défilement. Ces tests ont permis de montrer l'avantage de notre algorithme et de choisir les paramètres optimaux pour chaque sujet. Nous concluons par les résultats obtenus sur un sujet handicapé lors de l'utilisation d'EDiTH en scénario libre.

I. CHAPITRE 1 :

ETAT DE L'ART

I.1. Introduction

Pour certaines personnes handicapées moteurs souffrant de troubles de la communication verbale, le recours à des communicateurs, ou systèmes de «Communication Améliorée et Alternative» (CAA), est le seul moyen disponible pour communiquer et contrôler l'environnement d'une manière autonome. Mais la mise en place d'un outil de communication reste complexe pour plusieurs raisons : un outil de communication intègre une identification précise du besoin de l'utilisateur, un mode d'accès physique sans lequel rien n'est possible et une présentation à l'écran des éléments de communications (pictogrammes, images, etc.).

Le principe de ces systèmes est toujours le même : il faut sélectionner les icônes ou les symboles (lettres ou mots) qui vont construire le message à émettre. Pour valider un choix on peut utiliser un dispositif de pointage qui dépend des possibilités motrices de la personne handicapée (joystick, système de pointage à la tête, etc.).

Dans le cas où la personne présente un lourd handicap moteur, suite par exemple à une sclérose latérale amyotrophique (SLA) ou à un syndrome d'enfermement (LIS), elle peut n'avoir que la possibilité d'utiliser un simple contacteur (capteur de type Tout Ou Rien) pour contrôler le communicateur. Le recours à des systèmes à défilement demeure alors le seul moyen d'interaction humain-machine pour ces personnes. Le principe de ces systèmes consiste en un défilement lumineux sur un écran ou un tableau spécifique, une action sur le contacteur permettant de valider un choix ou un groupe de choix.

Le principal inconvénient des systèmes de communication à balayage est la lenteur de la communication qu'ils induisent, un délai typique de défilement pouvant varier de 0.2 à 6 secondes. Dans le cas d'une saisie de texte, la tâche de composition de messages est donc longue (1 à 5 mots par minute en moyenne) et fatigante pour les sujets. Dans ce cas, on fait

couramment appel à des techniques de prédiction pour améliorer la performance de communication : prédiction de caractères, de fins de mots, de mots compte tenu d'un contexte syntaxique. Ceci se fait cependant au détriment de la charge cognitive. De façon complémentaire, on s'attache également à présenter l'information sous une forme optimisée (balayage ligne/colonne, présentation des choix les plus fréquents en début de balayage, etc.). Le paramètre essentiel pour la vitesse de communication reste toutefois le temps élémentaire de défilement que nous appellerons T_{scan} . Un défilement trop rapide engendre un nombre important d'erreurs de sélection alors qu'un défilement trop long diminue la vitesse de communication dans des proportions importantes. Ce temps T_{scan} est généralement défini de façon empirique et garde sa valeur tout au long de l'utilisation du système.

Dans cet état de l'art nous nous intéresserons tout d'abord aux différents types d'aide à la communication et, plus particulièrement, aux systèmes à défilement. Nous citerons quelques pathologies concernées par ces systèmes. Nous verrons aussi les approches envisagées par plusieurs auteurs pour accélérer la saisie sur les claviers simulés. Nous aborderons en deuxième partie les travaux évoqués dans la littérature qui visent à régler la vitesse de défilement des systèmes de manière automatique pour l'adapter à l'utilisateur. La troisième partie sera consacrée à l'interaction homme-machine, plus particulièrement à la modélisation de l'interaction entre la personne handicapée et le dispositif d'aide à la communication.

I.2. Les aides à la communication

I.2.1. Introduction

Le schéma classique de toute communication comporte : un émetteur, un récepteur, un support de communication, un référentiel commun et le message à décoder (Figure I.1). La communication peut être mono ou plus généralement bidirectionnelle et le support peut prendre des formes diverses.

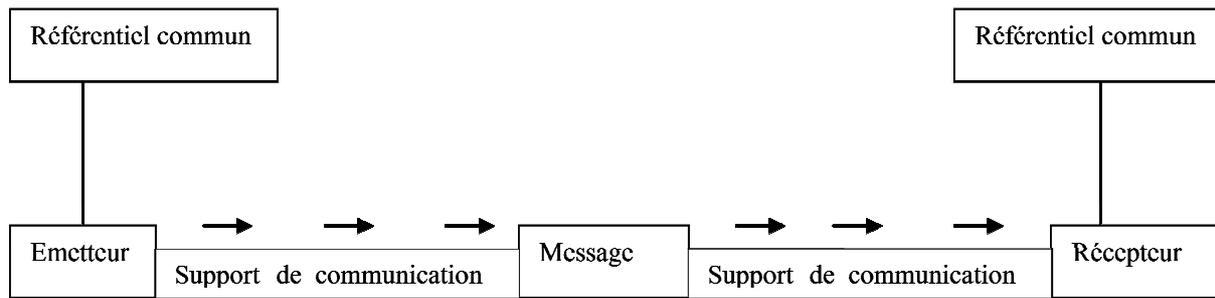


Figure I.1 : Schéma classique de la communication [JACQ]

La communication qu'elle soit orale ou écrite est indispensable à chaque personne et fait partie des activités essentielles de sa vie quotidienne. Les systèmes de communication alternative doivent offrir à l'utilisateur, quelle que soit sa pathologie, un accès rapide et efficace à la communication et à l'information. Dans le cas du clavier virtuel, la vitesse de saisie de texte pour une personne lourdement handicapée est considérablement réduite. Pour augmenter cette vitesse, de nombreuses techniques d'accélération ont été développées que nous allons détailler ci-après. Nous allons citer maintenant quelques pathologies concernées par les systèmes à défilement ainsi qu'un certain nombre d'aides à la communication que nous avons groupées en deux types : aides à la communication directe et indirecte.

I.2.2. Quelques pathologies concernées par les systèmes à défilement

I.2.2.1. L'infirmité Motrice Cérébrale : IMC

L'infirmité motrice cérébrale est liée à un dommage du cerveau survenue dans la période anténatale ou périnatale. Elle constitue un trouble moteur non progressif à des degrés variables, ou une atteinte sur un cerveau en maturation qui peut s'accompagner aussi de troubles cognitifs. Ainsi la personne a des mouvements lents, involontaires et non contrôlés lors de la réalisation d'actions motrices volontaires.

I.2.2.2. Le syndrome d'enfermement : LIS

Le syndrome d'enfermement (Locked-in Syndrom : LIS) résulte d'un Accident Vasculaire Cérébral (AVC) majeur qui détruit le tronc cérébral responsable de la transmission des ordres moteurs. Le sujet dans ce cas est atteint par une paralysie complète, mais il a la possibilité de contrôler encore les clignements de paupières, il entend tout et il comprend tout. Ses facultés cognitives et perceptives restent intactes [SCHA03].

I.2.2.3. La Sclérose latérale Amyotrophique : SLA

La sclérose latérale amyotrophique est une maladie de cause inconnue. Elle concerne les deux sexes et son incidence augmente avec l'âge à partir de 40 ans. Elle se manifeste par une paraplégie spasmodique (paralysie partielle ou totale des membres inférieurs avec raideur ou contracture des membres en extension). Elle se caractérise par une perte progressive des facultés motrices et finit par paralyser les muscles utiles à la phonation. Le malade garde tout au long de l'évolution une lucidité et une conscience intactes [BRAN97].

I.2.3. La communication directe

I.2.3.1. Introduction

La communication directe est le mode de communication le plus ancien et un moyen simple à mettre en place et peu onéreux. Ce mode nécessite une information préalable orale ou écrite. L'élaboration d'un message respecte des règles lexicales et syntaxiques et cherche à créer un signifiant. Il peut avoir des non-concordances entre l'émission et la réception : celles-ci sont liées au degré de formation, de culture, d'expérience des personnes, des ambiguïtés peuvent donc survenir. Quel que soit le code de communication, il faut définir avec la personne handicapée le OUI et le NON avant toute utilisation [CATA98]. Divers codes sont utilisés, citons par exemple les codes alphabétiques, pictographiques ou gestuels. Nous allons détailler quelques uns d'entre eux parmi les plus représentatifs.



Figure I.2 : Exemple de communication directe

I.2.3.2. L'ALPHABET

L'ALPHABET est le plus classique des codes, il consiste à épeler l'alphabet habituel à une vitesse adaptée et avec une attention visuelle importante en notant les lettres validées par la personne non parlante. Il ne nécessite aucun apprentissage particulier, il donne donc aux interlocuteurs occasionnels la possibilité d'échanger sans avoir à apprendre un code spécifique. Mais compte tenu de la lenteur de cette technique d'égrenage, nous avons souvent intérêt à recourir à des codes accélérant sensiblement la fabrication des mots et des phrases [CATA98].

I.2.3.3. L'alphabet ESARIN

L'alphabet ESARIN est fondé sur l'ordre de fréquence d'apparition des lettres dans la langue française (Figure I.3). Cet alphabet va être visualisé sur un support matériel et lu par l'interlocuteur. Citons l'exemple de Jean-Dominique Bauby qui a utilisé cette méthode pour écrire son livre "Le scaphandre et le papillon".



E S A R I N T U L O M D P C F B V H G J O Z Y X K

Figure I.3 : Alphabet ESARIN

I.2.3.4. Code alphabétique voyelles-consonnes

L'alphabet voyelles-consonnes sépare l'alphabet en deux groupes, comme le montre la Figure I.4, l'interlocuteur propose "voyelles" puis "consonnes". La personne handicapée valide un des quatre groupes puis une des lettres du groupe.

Voyelles	Consonnes1	Consonnes2	Consonnes3
A	B	J	R
E	C	K	S
I	D	L	T
O	F	M	V
U	G	N	W
Y	H	P	X
		O	Z

Figure I.4 : Code alphabétique voyelles-consonnes

I.2.3.5. Code alphabétique voyelles-consonnes à double entrée

Ces codes partent d'une présentation alphabétique disposée en tableau matriciel. Citons par exemple le code « Vigand » qui est un code hybride réalisé à partir de l'alphabet voyelles-consonnes. Dans ce cas les voyelles et les consonnes sont traitées séparément dans deux tableaux différents (Figure I.5). M. Vigand utilise le clignotement des yeux pour renseigner l'interlocuteur sur le numéro de ligne puis le numéro de colonne, sans lui désigner préalablement lequel des deux tableaux est concerné [CATA98]. Nous avons donc un décodage en termes de comptage d'abord, puis de référence spatiale et mémorisation des deux tableaux. L'interlocuteur doit présupposer à chaque pas de quel tableau il s'agit.

Le code "Vigand" :										
	1	2			1	2	3	4	5	
1	A	O			1	B	G	L	Q	V
2	E	U			2	C	H	M	R	W
3	I	Y			3	D	J	N	S	X
					4	F	K	P	T	

Figure I.5 : Code alphabétique voyelles-consonnes à double entrée

I.2.3.6. Codes pictographiques

Les pictogrammes sont des représentations graphiques figurées ou symboliques qui visent à exprimer un mot, une idée (Figure I.6). Un code pictographique s'adresse en priorité aux personnes qui ont des troubles du langage ou qui ne peuvent ni lire, ni écrire ou à des enfants

en phase d'apprentissage. Un intérêt de cette représentation est qu'elle ne nécessite qu'un nombre minimal de symboles pour communiquer.

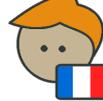
				
Ambulance	Orthophoniste	Matin	Marcher	Courir
				
Là	Pansement	Radar	Aimer	Français

Figure I.6 : Code pictogrammes d'Axelia [AXEL]

I.2.4. La communication indirecte

Aujourd'hui de nombreuses aides technologiques permettant à la personne handicapée de communiquer avec son environnement pour exprimer ses choix et ses besoins sont disponibles dans le commerce (Figure I.7). Ces aides sont pilotées par des interfaces adaptées aux possibilités physiques de l'utilisateur (joysticks, contacteurs, claviers et souris adaptés, etc.). Ces interfaces sont de deux types : de type pointeur ou de type « Tout Ou Rien » (TOR). Nous allons dans ce qui suit nous focaliser sur les systèmes à défilement associés à une interface TOR et nous allons dessiner un panorama non exhaustif des appareils dédiés à la synthèse vocale et des solutions logicielles basées sur ordinateur.

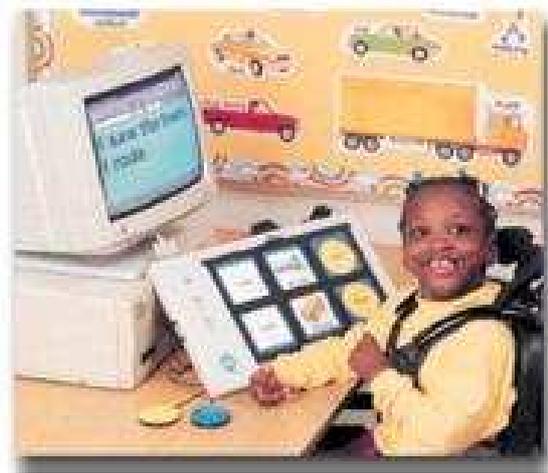


Figure I.7 : Exemple de communication indirecte

I.2.4.1. Capteurs de type Tout-Ou-Rien

Le choix du dispositif d'interaction est un point essentiel pour pouvoir manipuler l'aide à la communication. Dans le cas de sujets avec des capacités motrices très réduites, des commandes de type « Tout Ou Rien » sont le seul moyen pour utiliser une aide à la communication. Une gamme importante est commercialement disponible : contacteurs mécaniques, pneumatiques, au souffle, électromyographiques (EMG) (Figure I.8). Le choix de cette interface matérielle dépend du geste libre laissé par la déficience motrice, permettant une utilisation efficace du système d'aide à la communication.

<p>Contacteur au souffle : BLOW PB30</p>	<p>Contacteur pneumatique</p>
	
<p>Contacteurs mécaniques:</p>	
	
<p>BUDDY BUTTON</p>	<p>SOFT SWITCH TASH (intégré dans un coussin)</p>
	<p>Contacteur EMG : Ablenet</p>

Figure I.8 : Exemples de capteurs Tout-Ou-Rien [HMC] [SUPP]

Les contacteurs mécaniques sont caractérisés par leurs formes, leurs couleurs, leurs dimensions, la pression nécessaire pour établir le contact qui peut être réglable, la matière de la surface d'appui ou encore la résistance aux chocs. Ils peuvent être fixés sur un support velcro, une surface autocollante, être vissés ou simplement déposés sur une surface antidérapante. Ils peuvent offrir un feedback auditif et tactile. L'usage de ce type de contacteur ne se restreint pas aux doigts et à la main, il peut convenir à d'autres parties du corps : joue, nuque, cuisse, etc. [HMC].

Le contacteur au souffle utilise la fonction aspiration/expiration pour 2 contacts, l'ensemble est muni d'un tube d'acier pour permettre un positionnement précis [SUPP]. Plus complexes, les capteurs de signaux EMG détectent les contractions musculaires contrôlables de la personne. Ce capteur peut s'utiliser directement sur n'importe quel muscle du sujet. Citons par exemple le contacteur Impulse Switch d'Ablenet qui peut capter de simple mouvement des

sourcils et permettre au sujet de communiquer avec l'ordinateur. Ce système est composé de deux électrodes, d'une alimentation par piles, et d'un émetteur « bluetooth » qui permet une transmission sans fil jusqu'à dix mètres de l'équivalent d'un clic souris [ABLE].

I.2.4.2. Les appareils dédiés à la synthèse vocale

Comme leur nom l'indique, les appareils dédiés à la synthèse vocale sont des machines uniquement consacrées à la communication verbale, de petites tailles et légères. Ils sont intégrables facilement sur les fauteuils roulants. Ces appareils se présentent sous la forme de claviers alphanumériques, phonétiques ou composés de pictogrammes. Ils utilisent le plus souvent des symboles, des pictogrammes ou un alphabet phonétique pour construire des messages [BIRD07]. La voix émise peut être :

- **Numérisée** : voix humaine préenregistrée et restituée sur commande par la personne handicapée. La liste de messages demeure limitée et l'utilisateur n'est pas autonome en ce qui concerne la manière de les formuler. Ce type d'appareils s'adresse en priorité aux enfants et aux personnes ne sachant pas écrire et fonctionnent donc généralement avec des pictogrammes. Citons par exemple les systèmes « Springboard » ou « LightTalker » (Figure I.9).



Figure I.9 : Appareils de communication à voix numérisée

- **Synthétique :** La voix est synthétisée au moment où l'utilisateur compose le mot. Ces appareils sont plus ouverts que les systèmes numérisés. Ils s'adressent aux personnes sachant lire et écrire et fonctionnent généralement avec un clavier alphabétique ou phonétique comme par exemple « Vanguard », « Dialo » ou « Leblatphone » (Figure I.10). Ils offrent, selon les modèles, la possibilité de communiquer plus vite grâce notamment à :

- **Une entrée avec des mots abrégés usuels :** un exemple d'encodage de mot *Monsieur* est *M..*
- **Un système de prédiction de mots :** lorsque l'utilisateur tape un mot, une liste de mots commençant par les mêmes lettres apparaît. Il suffit alors de sélectionner le mot voulu.

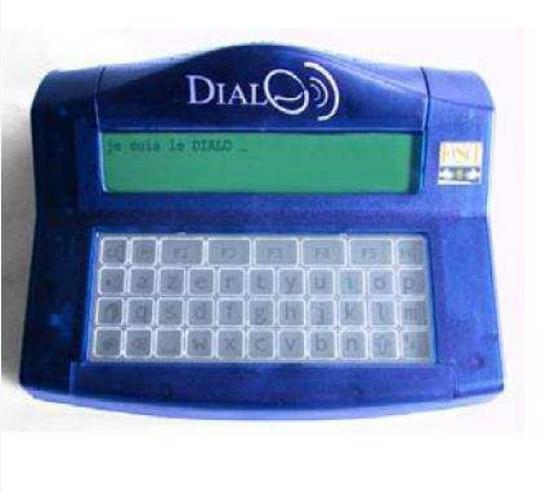
 <p>The Vanguard device is a green, rectangular handheld device with a large screen displaying a grid of colorful icons. It has several buttons at the top and a small speaker grille.</p>	<p>Vanguard : il intègre les avantages des écrans dynamiques et le concept de codage sémantique du vocabulaire Minspeak. Sa synthèse vocale permet de lire n'importe quel mot en se basant sur l'entrée orthographique avec prédiction de mots. Vanguard couvre un large vocabulaire en se basant sur l'utilisation d'une synthèse vocale et la possibilité d'enregistrement digital audio. Il est utilisable par un joystick ou par l'intermédiaire d'un contacteur [FST].</p>
 <p>The Dialo device is a blue, handheld device with a small screen at the top displaying the text 'Je suis le DIALO'. Below the screen is a large, light-colored touch-sensitive keyboard. The brand name 'DIALO' is visible on the top left.</p>	<p>Dialo : dans le clavier les lettres sont classées par ordre de fréquence d'utilisation en français et il intègre la technique de prédiction de mots. Il a la possibilité de mémoriser des mots ou des phrases sous une touche ou une combinaison de touches. Son clavier est un écran tactile. Il est utilisable par un accès direct ou par l'intermédiaire d'un contacteur [FST].</p>
 <p>The Leblatphone is a black, handheld device with a grid of colorful, touch-sensitive keys. It is being held by a person's hands, showing its ergonomic shape.</p>	<p>Leblatphone : il s'agit d'une machine à parler qui permet un pointage direct sur le clavier ou l'écran tactile. Elle intègre l'ensemble du dictionnaire, soit près de 110000 mots avec un minimum de touches. Le clavier est composé de soixante touches tactiles : 36 syllabes d'un côté, 24 de l'autre. Par association, l'utilisateur obtient les mots voulus qui seront prononcés par une voix digitalisée. Deux voix sont disponibles pour la synthèse vocale : une masculine et une féminine [LEBL].</p>

Figure I.10 : Appareils de communication à voix synthétique

I.2.4.3. Les solutions logicielles basées sur ordinateur

La saisie du texte est une activité essentielle lors de l'utilisation d'un ordinateur. Pour les personnes ayant un handicap sévère, cette activité est rendue possible grâce aux claviers dits logiciels qui sont des représentations numériques de claviers physiques permettant de saisir des données et remplaçant toutes les fonctionnalités d'un clavier standard. La validation d'un choix se réalise par plusieurs modes de pointage qui dépendent des facultés motrices du sujet.

De nombreuses études décrites dans la littérature portent sur l'optimisation des claviers virtuels associés à un dispositif d'entrée de type proportionnel : dans le cas d'utilisateurs ayant la capacité de déplacer un pointeur sur un écran, l'optimisation va consister à limiter la distance de déplacement du curseur. Citons les travaux de recherche du centre Enkidu à New York qui ont développé un clavier réarrangé nommé KNITS [LESH00]. Il s'agit d'un clavier optimisé sur les bigrammes de lettres observés dans un corpus de trois millions de mots. La méthode proposée est la suivante : les auteurs affectent un coût à chaque transition entre deux touches, ce coût dépendra principalement de la distance entre le centre des deux touches. Citons encore une disposition plus optimale dans le clavier « KeyGlass » [RAYN07] qui utilise le contexte courant de saisie : la prédiction dans ce cas se base sur deux modèles complémentaires, un bigramme de lettres et un arbre lexicographique probabiliste. Dès qu'un caractère est sélectionné, le système affiche autour de la touche les quatre lettres qui lui semblent les plus probables. Ainsi, le système d'aide à la communication « PCA » est un outil de mise en œuvre d'une communication iconique facilement adaptable et personnalisable. Il possède une base d'icônes regroupant environ 750 pictogrammes. Le message est composé dans ce cas d'une séquence d'icônes [BELL04b] [BELL06].

Dans le cas d'un sujet ne pouvant utiliser directement le pointeur de la souris, la sélection des items ne peut se faire que par un défilement lumineux sur un écran. L'utilisateur doit attendre que son choix soit activé, une action sur le contacteur permettant de valider l'item sélectionnable à cet instant. Le principal inconvénient de ces claviers virtuels à balayage est la lenteur de construction des messages. Plusieurs paramètres concernant le type de défilement ou le type de clavier interviennent pour les optimiser. A partir de ces paramètres des techniques ont été développées visant à accélérer la communication. Elles sont décrites dans ce qui suit.

I.2.5. Méthodes d'accélération

Dans les systèmes à défilement proposer des techniques d'optimisation de saisie de texte s'avère nécessaire pour améliorer le temps de communication. Deux approches peuvent être utilisées dans cette optique :

- La première consiste à accélérer la sélection sur le clavier virtuel en optimisant le temps d'accès à la touche recherchée pour réduire le temps de défilement. Cette optimisation dépend du type de balayage, du clavier et également de la prise en compte du contexte de la saisie [SCHA03] [LESH00a] ;
- La seconde approche vise à économiser le nombre de saisies, en complétant d'une manière automatique des parties du message en fonction du contexte. Interviennent dans ce cas des techniques de prédiction de texte et des abréviations [ZAGL99] [BOIS03].

Ces deux approches sont complémentaires et peuvent être envisagées en même temps. Nous allons maintenant les décrire plus en détail.

I.2.5.1. Optimisation du temps d'accès à une touche

Cette approche est fortement liée au déficit moteur de la personne et au type de clavier utilisé. Parmi les méthodes proposées par les logiciels pour minimiser le nombre de défilements, on peut dans un premier temps adopter un balayage optimisé des touches comme le défilement « ligne/colonne ». Une autre possibilité est d'organiser les touches sur le clavier en utilisant leur fréquence d'utilisation et de présenter au sujet les lettres les plus probables en fonction de ce qu'il a déjà tapé [BOIS03] [LESH98b] [LESH00a].

a) Types de défilement

Trois modes de balayage sont couramment utilisés dans les claviers virtuels :

- Le balayage linéaire ;
- Le balayage ligne/colonne ;
- Le balayage bloc par bloc.

Dans le premier cas, pour accéder à une touche, la méthode consiste à parcourir séquentiellement chaque case jusqu'à atteindre la case désirée. Ce défilement est le plus simple mais aussi le plus long. Pour un clavier de T touches, le Nombre de Défilements Moyen (NDM) est donné par la formule suivante : $NDM = \frac{T+1}{2}$ en considérant que chaque touche est équiprobable [SCHA03].

En balayage colonne/ligne (ou ligne/colonne), la sélection d'un item se fait en deux actions : dans un premier temps, le balayage se fait colonne par colonne, une action sur un contacteur permet de valider la colonne désirée. Le défilement reprend alors dans la colonne en parcourant chacun de ses items. Une nouvelle action sur le contacteur permet de valider un item. Pour un tel balayage, une sélection nécessite deux actions pour activer successivement la colonne et la ligne de la touche désirée. Le nombre de défilements moyen (NDM) pour atteindre une case est donné par la formule suivante : $NDM = \frac{L+1}{2} + \frac{C+1}{2}$ où L est le nombre de lignes du clavier et C le nombre de colonnes.

Le dernier type de défilement qui permet encore d'améliorer la vitesse d'accès aux items est le balayage bloc par bloc. Dans ce cas, le clavier est composé de blocs, eux-mêmes composés de sous-blocs qui permettent d'atteindre finalement les touches. La sélection d'un item est donc réalisée en trois étapes : sélection d'un bloc, puis d'un sous-bloc, et enfin d'une touche.

Dans ce cas le nombre de défilements moyen devient : $NDM = \frac{\sqrt[3]{T+1} \times 3}{2}$ [SCHA03]

Le tableau I.1 donne quelques exemples de nombres de défilements moyens selon le type de balayage et pour différentes tailles de clavier (L est le nombre de lignes du clavier et C le nombre de colonnes). La dernière colonne présente le nombre d'appuis nécessaires pour sélectionner une case. Les chiffres de ce tableau montrent le saut qualitatif entre le défilement linéaire et le défilement ligne/colonne. Quant au défilement bloc par bloc l'ajout d'une validation supplémentaire le pénalise et son utilisation devient intéressante à partir d'un grand nombre de touches.

Type de défilement	L - C 25-27	L - C 64 - 64	L - C 121-125	Nombre d'appuis
Linéaire	13,5	32,5	62,5	1
Ligne/Colonne	6	9	12	2
Bloc par bloc	4,5	7,5	6	3

Tableau I.1 : NDM selon le type de balayage et la taille du clavier [SCHA03]

Le défilement colonne/ligne permet une sélection plus rapide du caractère recherché que le balayage linéaire mais nécessite une action supplémentaire. Par ailleurs, le défilement linéaire ne demande qu'une attention sur la case qui change de couleur, alors que le défilement colonne/ligne exige une observation complète du clavier, pour choisir la colonne, puis la ligne et implique par conséquent une augmentation de la charge cognitive.

Quel que soit le défilement utilisé, nous pouvons trouver une configuration optimale du clavier fonction du nombre de touches. Dans [CANT01], les auteurs ont montré que dans le cas d'un défilement ligne/colonne la matrice optimale des touches forme un triangle rectangle dont les côtés opposés à l'hypoténuse se rejoignent en haut à gauche de l'écran (Figure I.11). L'emplacement des lettres sur la matrice du clavier dépend de leur fréquence d'apparition dans la langue.

SP	e	a	n	l	f	v	-
r	o	s	d	w	k		
i	h	u	y	'			
r	c	g	x				
m	p	j					
b	q						
z							

Figure I.11 : Clavier optimisé (pour l'anglais) pour un défilement ligne/colonne

b) Réorganisation dynamique des touches sur le clavier

Le principe des claviers dynamiques est de réorganiser le clavier après chaque sélection de lettre afin que les caractères les plus probables soient accessibles en un nombre minimal de défilements (Figure I.12). La méthode consiste à présenter les lettres dans un ordre correspondant à leur fréquence d'utilisation, et à effectuer la sélection de manière à ce que

plus un élément est utilisé, plus sa sélection est rapide. L'étude des fréquences d'utilisation doit se faire sur un échantillonnage suffisamment élevé pour pouvoir obtenir les probabilités d'usage. Cette idée peut être prolongée en utilisant non seulement les fréquences des symboles hors contexte, mais aussi les fréquences des lettres en contexte en fonction des saisies précédentes réalisées par le sujet. Chaque fois que le clavier est remis à jour, le sujet doit parcourir visuellement la totalité de l'écran pour retrouver la nouvelle position du caractère qu'il veut saisir. Cette méthode permet d'augmenter de façon importante la vitesse de la construction des messages, mais elle est plus difficile à appréhender car elle augmente la charge cognitive [LESH02]. Dans le cas d'un défilement linéaire, le caractère dynamique de l'affichage est moins gênant puisque le sujet peut se contenter de considérer la position courante du curseur et celle qui va lui succéder.



Figure I.12 : Modification de la disposition des touches après chaque saisie du mot « compter » [SCHA04]

Deux approches ont été étudiées pour permettre une organisation contextuelle des claviers dynamiques. La première méthode est basée sur les connaissances lexicales. Elle permet de construire l'arborescence d'un lexique et de factoriser les fréquences associées à chaque mot. Par conséquent un lexique fréquentiel élaboré sous la forme d'un graphe probabiliste de lettres est parcouru. Chaque nœud de l'arbre représente un caractère. La probabilité d'occurrence d'un caractère est donnée par la somme des probabilités de ses nœuds fils. La probabilité finale des mots complétés, qui se trouvent dans les nœuds feuilles, est leur fréquence relative d'occurrence dans la langue [ANTO07].

L'autre approche s'appuie sur une modélisation markovienne. Elle est fondée sur la fréquence des occurrences des n-uplets de lettres et permet d'estimer la probabilité d'une lettre en fonction des n-1 dernières lettres. Le point fort de l'approche markovienne réside dans le fait que la prédiction n'est pas troublée par l'existence des mots hors dictionnaire, ni par les

erreurs de saisies réalisés par le sujet. Cette méthode est utilisée par le système Sibylle développé au Laboratoire VALORIA [SCHA04] (Figure I.13) qui utilise un pentagramme de lettres, c'est-à-dire qu'il se base sur le contexte de quatre lettres pour prédire la suivante, et présente un système de prédiction de mots et de lettres.

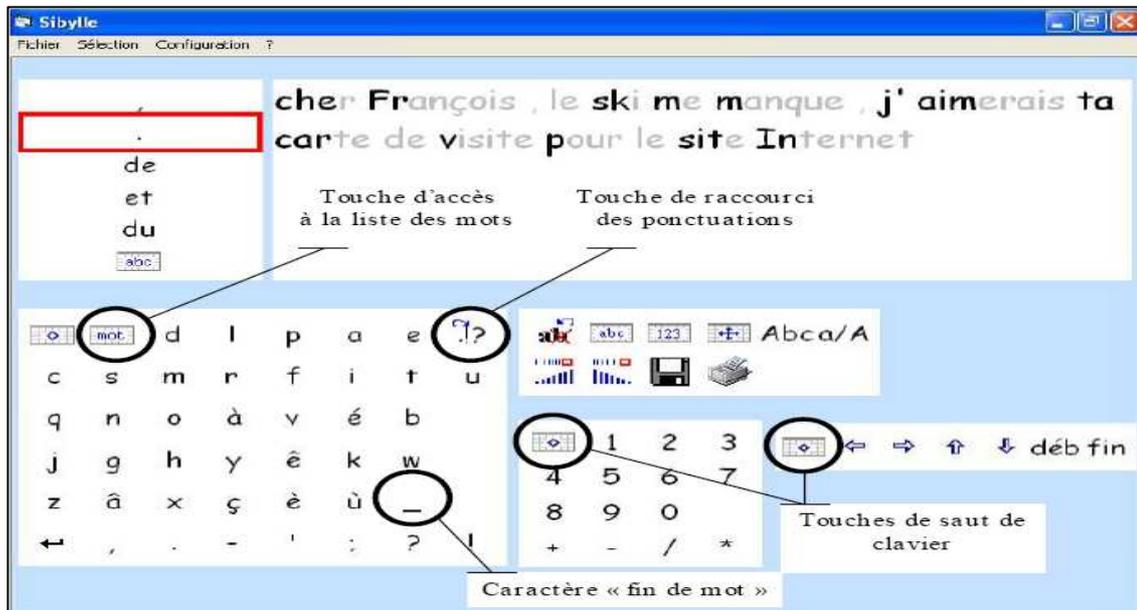


Figure I.13 : Le système Sibylle

c) Réduction du nombre de touches du clavier

On peut accéder à l'ensemble des caractères d'un clavier complet à partir d'un nombre de touches moins important. Ainsi les claviers ambigus possèdent un nombre réduit de touches, chaque touche représentant plusieurs lettres comme sur les téléphones mobiles. Dans le cas d'une commande Tout Ou Rien, le nombre de défilements moyen est réduit et par conséquent la réalisation d'une sélection plus rapide.

A titre d'exemple, le clavier UKO (« Unbekanntes KommunikationObjekt », ie objet de communication non identifié) [KUHN01] de l'université de Coblenz en Allemagne intègre dans son interface six touches dont quatre utilisées pour les lettres (Figure I.14). Dans le cadre de droite apparaît les mots possible et dans celui de gauche la saisie en cours. Un défilement automatique parcourt d'une manière continue ces touches. Le point fort du clavier UKO est que le nombre de touches est configurable et que les lettres ne sont pas classées de manière

classique (ordre alphabétique), mais organisées de manière à simplifier la tâche de désambiguïsation.



Figure I.14 : Le clavier ambigu UKO

Un autre exemple, le clavier DotNote [UTIL], réduit de moitié le nombre de touches à l'écran. Les caractères sont présentés sur deux interfaces (Figure I.15), la première interface contient les caractères les plus fréquemment utilisés et la deuxième interface contient les caractères les moins utilisés. Le sujet passe d'une interface à une autre grâce à la touche Shift.

					←						←
A	C	D	E	↑	B	F	J	K	↑		
G	H	I	L	123	Q	V	W	X	123		
M	N	O	P	A↔Q	Y	Z	.	,	A↔Q		
R	S	T	U	Space	"	-	&	↵	Space		

Figure I.15 : Le clavier DotNote avec ses deux interfaces

I.2.5.2. Economie du nombre de saisies

Nous venons de voir différentes approches qui visent à réduire le temps d'accès aux touches du clavier virtuel. Malgré les améliorations qu'elles procurent la saisie de texte reste cependant fatigante pour les utilisateurs, d'où le désir d'accélérer encore la composition des messages en réduisant le nombre de saisies de lettres. Pour ce faire deux méthodes sont

envisageables : la première est basée sur la prédiction de mots et la deuxième fait usage de l'abréviation.

a) Prédiction de mots

Le principe est le suivant : le logiciel dispose d'un dictionnaire doté d'informations de fréquence. En se basant sur le préfixe donné au cours de la saisie, les mots sont extraits du lexique puis ils sont rangés en se basant sur leurs valeurs fréquentielles. Les mots qui ont une fréquence plus grande vont être affichés au début. Dans ce cas la prédiction est hors contexte, elle ne gère l'accord ni en genre ni en nombre. Si le sujet veut retenir une des prédictions proposées, il n'a qu'à valider le mot cherché, ce mot va ensuite s'intégrer automatiquement dans le texte en cours. L'affichage des prédictions peut se faire de deux manières :

- **Complétion intégrée** : dans ce cas le système ne propose la prédiction que d'un seul mot à chaque saisie. L'utilisateur n'a qu'à valider par un appui la sélection éventuelle de la proposition. Citons le système français « VITIPI » qui propose d'insérer d'une manière automatique les lettres des mots dès qu'il n'y a plus d'ambiguïté [BOIS01].
- **Liste de mots** : dans ce cas le système retient plusieurs propositions. L'affichage de la liste qui contient le mot va se faire dans une zone définie dans le clavier. Par conséquent le sujet a besoin d'un appui supplémentaire pour accéder à cette liste. Citons par exemple le système Sibylle qui intègre en plus du clavier dynamique une prédiction de mots [SCHA04]. Dans ce cas l'utilisateur cherche dans une liste de mots directement affichée le mot désiré. La prédiction est réalisée en contexte, en fonction des mots et des lettres déjà saisis.

L'utilisation de la technique de prédiction peut accélérer d'une manière significative la composition du message sans aucun effort de mémorisation, ce gain peut atteindre un caractère sur deux en moyenne [ANTO07]. Par contre cela va engendrer une augmentation de charge cognitive due à la recherche de mots dans la liste [KOES94a].

b) Accélération par contraintes syntaxique ou sémantique

La méthode d'accélération syntaxique se base sur une analyse syntaxique des messages. Cette méthode consiste à présenter uniquement à l'opérateur les éléments utiles du point de vue grammatical en tenant compte des éléments sélectionnés dans le message en construction. Par exemple, après la sélection de l'article déterminant « le », il est raisonnable de n'accéder qu'aux noms singuliers masculins d'un dictionnaire [CANT01].

La méthode d'accélération sémantique consiste à tenir compte du sens des éléments sélectionnés dans le message en cours d'élaboration et à ne présenter à l'opérateur que les éléments utiles d'un point de vue sémantique. Elle complète la méthode d'accélération syntaxique. Par exemple, la sélection de l'élément verbe « boire » va nous ramener à la visualisation des éléments de même domaine, le nom des boissons. Ainsi le système HANDIAS, développé à l'Institut de Recherche en Informatique de Nantes, utilise un prédicteur de mots qui fonctionne à partir de dictionnaires fréquentiels et de la modélisation d'un ensemble de structures du français auxquelles sont affectées des probabilités d'utilisation. HANDIAS doit rechercher dans le dictionnaire la liste des mots et des lettres les plus fréquents dans la (ou les) catégorie syntaxique la plus probable tout en respectant le début de la saisie [LEPE98].

c) Abréviation

L'utilisation de l'écriture abrégée est ancienne dans le domaine des aides techniques à la communication. Cette méthode permet de stocker des paires (abréviation, mot) et ainsi, dès qu'une abréviation est détectée, elle est remplacée par son mot équivalent. L'inconvénient est que cela demande un effort supplémentaire d'apprentissage et par conséquent augmente la charge cognitive de l'utilisateur. Toutefois nous trouvons des méthodes d'apprentissage progressif de l'abrégé qui facilitent ces opérations [RICC01].

En pratique, le sujet utilise une large couverture lexicale qui se trouve dans le dictionnaire d'abréviation. Par exemple, le sujet pourra décider d'utiliser l'abréviation *mt* pour représenter le morphème *ment* qui termine la plupart des adverbes français. Cette méthode est a priori très puissante. Cependant elle présente des inconvénients du fait qu'elle nécessite une écriture non naturelle qui peut perturber un sujet ayant une bonne maîtrise de la langue ou un enfant en phase d'apprentissage. Elle pose également la question de l'ambiguïté de la saisie de formes non fléchies (qui peut se traduire par des erreurs d'expansion) et elle augmente la charge

cognitive de l'utilisateur qui doit mémoriser la liste d'abréviations. Enfin le nombre d'abréviations qui peuvent être créées est limité.

I.2.6. Conclusion

L'optimisation des claviers virtuels est un domaine de recherche abordé par de nombreux laboratoires. Comme nous l'avons vu précédemment, dans le cas d'une saisie de textes on fait couramment appel à des techniques de prédiction pour améliorer leurs performances. De façon complémentaire, on s'attache également à présenter l'information sous une forme optimisée. C'est par défilements puis par clics successifs que l'utilisateur va sélectionner les éléments du message sur le clavier. Par conséquent, le paramètre essentiel pour optimiser la communication est le temps de défilement T_{scan} , qui doit être choisi de façon optimale pour un utilisateur donné. Nous allons voir dans la partie suivante les méthodes proposées par quelques auteurs pour adapter automatiquement la vitesse de défilement.

I.3. Optimisation du temps de défilement

I.3.1. Introduction

Un des paramètres essentiel pour les systèmes dits « à balayage » est le temps de défilement. Ce temps est généralement défini de façon empirique et garde sa valeur tout au long de l'utilisation du système. Un réglage trop court de T_{scan} engendre un nombre important d'erreurs de sélection alors qu'un réglage trop long diminue la vitesse de communication dans des proportions importantes. Quelques auteurs se sont intéressés à adapter le temps de défilement en temps réel. Cronk et Schubert ont développé un système expert pour régler le temps de défilement qui utilise les erreurs de sélection comme facteur de réglage [CRON87]. Leshner et al. [LESH00] utilisent le même principe basé sur les erreurs de sélection pour augmenter T_{scan} mais mesurent par contre le temps de réaction de la personne pour le diminuer. Dans un même ordre d'idée Simpson et Koester [SIMP99] règlent dynamiquement T_{scan} à l'aide d'un réseau bayésien. Enfin Evreinov et Raisamo [EVRE04] adaptent la vitesse de défilement en se basant uniquement sur l'analyse temporelle des temps de réaction dont ils calculent une moyenne toutes les 5 réalisations. Nous allons voir maintenant plus en détails le principe de chaque méthode citée ci-dessus.

I.3.2. Méthodes pour adapter le temps de défilement en ligne

I.3.2.1. Méthode basée sur un système expert

Cronk et Schubert ont choisi de développer un système expert pour adapter le temps de défilement [CRON87]. Ce système expert est construit à l'aide d'un modèle mathématique et comportemental. Le modèle mathématique est basé sur la moyenne et l'écart-type du temps de réaction de chaque sujet. Il est utilisé pour prédire le taux de défilement. Le modèle comportemental gère les situations spéciales telles que décider de diminuer le taux de défilement si le sujet fait un nombre d'erreurs excessif et si ces erreurs apparaissent de manière soudaine.

Les paramètres du système ont été ajustés pour obtenir la performance optimale du sujet. Les auteurs font une comparaison entre le taux de défilement fixé par les pathologistes et par le système expert pour évaluer la fiabilité de leur méthode. Le système expert fixe le temps de défilement en se basant sur la valeur du temps de réaction et le taux d'erreurs du sujet. Les auteurs ont trouvé que le système expert avait sélectionné exactement le même taux de défilement que les pathologistes sur les 59.9% des actions réalisées par les sujets. Le temps moyen de sélection est de 1.27 secondes pour le système expert et de 1.2 secondes pour le pathologiste [CRON87].

I.3.2.2. Méthode basée sur un réseau bayésien

Simpson et Koester se sont basés dans leurs recherches sur la possibilité d'augmenter le taux de texte saisi en adaptant dynamiquement les paramètres suivants pour contrôler le système de défilement durant l'exécution des tâches pour chaque utilisateur [SIMP00] [SIMP99]:

- Temps de pause de la première ligne ;
- Temps de pause de la première colonne ;
- Temps de défilement ligne ;
- Temps de défilement colonne ;
- Nombre de cycles de balayage d'une colonne si aucune sélection n'est faite une fois que la ligne est validée.

L'approche utilisée par Simpson et Koester [SIMP99] diffère de celle proposée par Cronk et Schubert sur deux aspects :

- L'adaptation est gérée par la performance de l'utilisateur et non par des personnes expertes ;
- Ils utilisent un raisonnement de probabilité basé sur un réseau bayésien et non un système expert aux règles bien définies.

Le réseau bayésien se base sur la formulation suivante :

$$\Pr(H/e) = \frac{\Pr(e/H) \Pr(H)}{\Pr(e)} \text{ où :}$$

- H représente le choix d'augmenter, diminuer, ou maintenir le délai du défilement courant ;
- e est un ensemble de mesures de performance de l'utilisateur ;
- $\Pr(H/e)$ représente la probabilité qu'exécuter une action particulière (par exemple, augmenter le temps de défilement) est la décision correcte compte tenu de la preuve présentée au réseau bayésien.

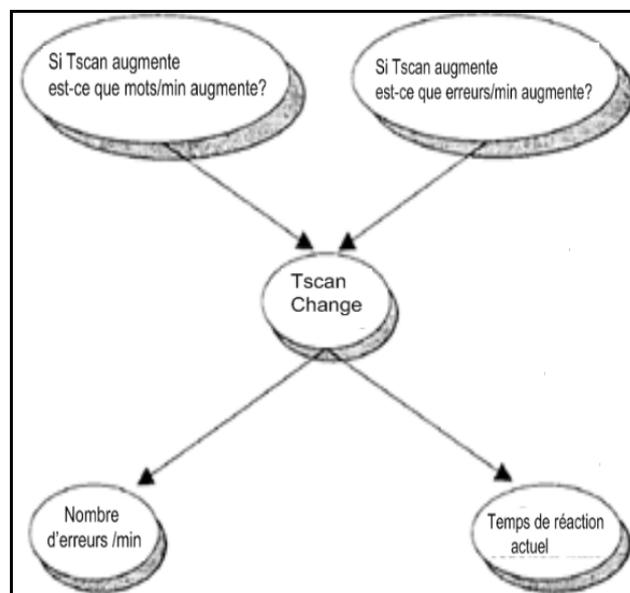


Figure I.16 : Un exemple de réseau bayésien utilisé dans [SIMP99]

La Figure I.16 présente un exemple de réseau utilisé dans [SIMP99]. Chaque nœud dans le réseau représente une variable probabiliste. Les nœuds qui présentent le nombre d'erreurs par

minute et le temps de réaction sont des variables observables dont les valeurs sont déterminées directement par la performance de l'utilisateur sur un intervalle de temps donné (30s pour les expériences menées par ces auteurs).

Pour évaluer l'efficacité de leur réseau, les auteurs ont effectué dans un premier temps une expérience sur des sujets valides qui consiste à réaliser deux tests :

- 1) Tester une interface colonne/ligne en utilisant l'intervention humaine pour varier le temps de défilement ;
- 2) Utiliser le réseau bayésien pour une variation automatique de temps de défilement.

Les résultats de cette expérience ont montré que le deuxième test n'a pas augmenté d'une manière significative le taux de texte saisi par rapport au premier test. Les sujets du premier test étaient assez habiles à déterminer correctement leur meilleur délai de défilement. Pour éliminer quelques uns des défauts perçus dans la première expérience et déterminer si la performance du réseau bayésien pouvait être améliorée, les auteurs l'ont redéfini en ajoutant plusieurs paramètres et l'ont évalué dans une deuxième expérience (Figure I.17). L'objectif de celle-ci est de démontrer que le réseau bayésien peut ajuster le temps de défilement selon la tâche réalisée sans aucune intervention de l'utilisateur.

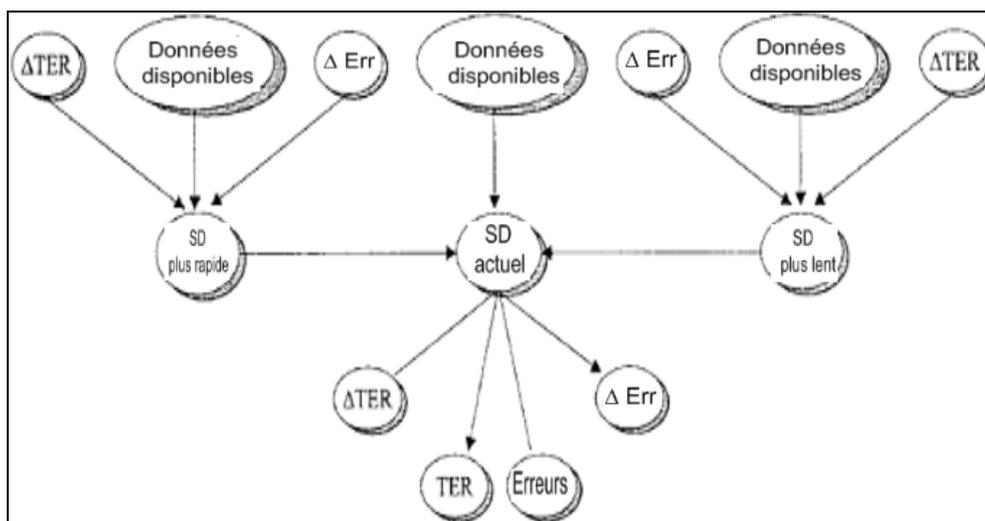


Figure I.17 : Le réseau bayésien utilisé dans la seconde expérimentation : TER= taux de texte saisi, Δ TER = Variation de taux de texte saisi, Erreurs = nombre d'erreurs, Δ Err = Variation du taux d'erreurs et SD = valeur du temps de défilement

L'algorithme représenté par le réseau bayésien est une comparaison entre d'une part la performance utilisateur au temps de défilement actuel, d'autre part la performance utilisateur au prochain temps de défilement le plus rapide et au prochain temps de défilement le plus lent. Ces performances peuvent être estimées sur la base :

- 1) Du changement moyen du taux d'entrée de texte et des erreurs observées au temps de défilement le plus rapide (ou le plus lent) comparé avec le temps de défilement actuel ;
- 2) De la moyenne de ces données.

Les résultats obtenus par les auteurs indiquent que le système peut prendre des décisions raisonnables pour adapter la vitesse de défilement sans intervention humaine, mais cette méthode n'a pas augmenté de manière significative le taux de texte saisi par minute par rapport au moment où les sujets avaient le contrôle complet sur le délai du défilement. De plus, il n'y avait pas de différences statistiquement significatives entre le nombre total des erreurs réalisées sans et avec intervention humaine. Cependant quelques sujets ont fait légèrement plus d'erreurs quand l'adaptation automatique était active, celle-ci les ayant perturbés. Plusieurs sujets ont rapporté que leur meilleure performance est venue quand le délai du défilement est resté fixe suffisamment longtemps pour synchroniser leurs actions à ce délai.999

L'avantage de l'approche utilisée par Koester et al., contrairement aux techniques d'accélération, est d'augmenter le taux de texte écrit sans augmenter la complexité de la tâche. Elle a donné un système adaptatif sensible aux changements permanents (dus à l'apprentissage) et transitoires (dus à la fatigue) de la capacité du sujet. Les résultats obtenus sont cependant limités du fait du manque de sujets handicapés: dans ces expériences les 16 sujets étaient valides.

I.3.2.3. Méthode basée sur le temps de réaction

Le problème de l'adaptation en temps réel du temps de défilement a aussi été abordé par Grigori Evreinov [EVRE04]. La méthode utilisée dans ses travaux pour sélectionner du texte est basée sur une structure hiérarchique symétrique avec trois alternatives et trois niveaux comme le montre la Figure I.18.

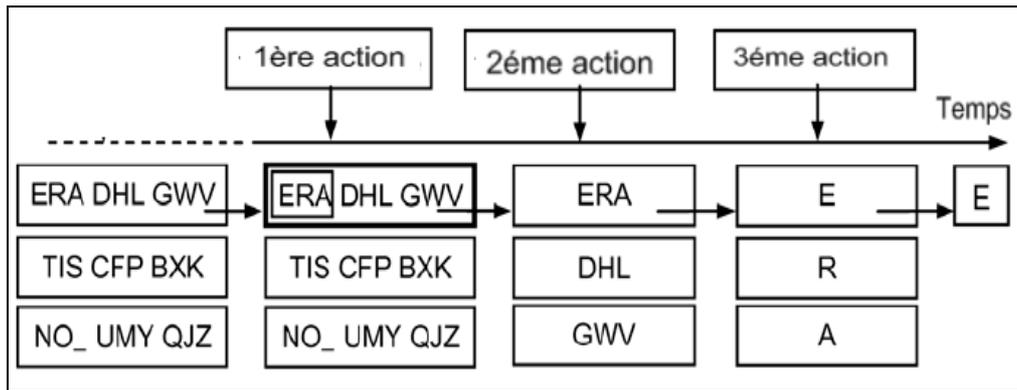


Figure I.18 : Menu hiérarchique symétrique avec trois alternatives et trois niveaux

Les auteurs se basent sur le fait que le comportement d'un sujet face à un stimulus est géré par une séquence d'actions réalisées par le processeur de cognition et le processeur moteur. Ces actions sont synchronisées du fait que le sujet attend que son choix soit validable pour appuyer sur le contacteur. La méthode utilisée pour adapter la vitesse de défilement est alors basée sur l'analyse en temps réel du temps de réaction moteur (RT) dont on calcule une moyenne toutes les 5 réalisations.

Le diagramme temporel de l'algorithme mesurant la performance de l'utilisateur à travers le temps de réaction moteur et la correction automatique de temps de défilement est présenté dans la Figure I.19. Les auteurs ont décomposé l'intervalle T_{scan} en $T_0+T_1+T_2$ avec :

- T_0 = le premier intervalle variable ;
- T_1 = le deuxième intervalle variable ;
- $T_2 = T_1$ où T_1 et T_2 présentent une fenêtre temporelle symétrique qui vise à établir un seuil dynamique : $T_{thresh} = T_0 + T_1$.

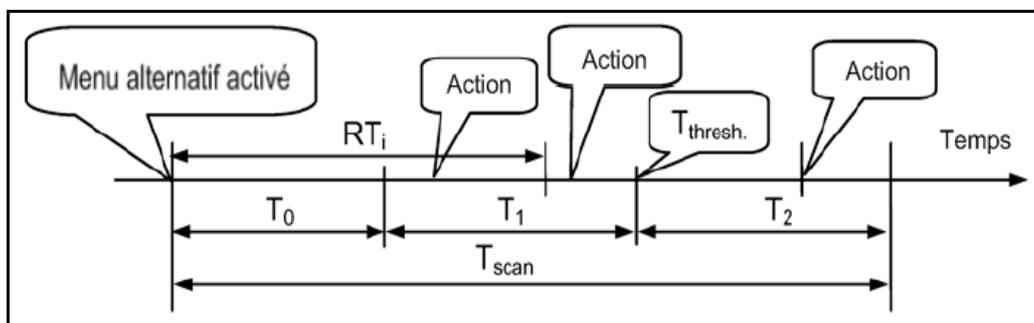


Figure I.19 : Diagramme temporel de l'algorithme pour le temps de défilement adaptatif

L'algorithme se base sur la valeur moyenne de RT (AveRT) calculée toutes les 5 réalisations et sa comparaison avec les autres variables pour décider si T_{scan} change ou pas. Chaque fois que T_{scan} change, tous les paramètres de l'algorithme s'adaptent à ce changement.

Si $AveRT < T_{thresh} - \Delta T$ la nouvelle valeur de T_{scan} est $T_{scan} - \Delta T$.

Si le sujet change de temps de réaction, et si $AveRT < T_{thresh}$, T_{scan} va diminuer en permanence en raison du décroissement de T_0 et par conséquent T_{thresh} va changer.

En pratique, pour fixer les paramètres initiaux de l'algorithme (T_0 , T_1), les auteurs observent les 40 premières actions réalisées par le sujet avec un temps de défilement initial égal à 2s. La valeur de ΔT est fixée à 20ms.

I.3.2.4. Méthode basée sur des mesures d'erreurs de sélection

Leshner et al. ont également développé une méthode pour adapter la vitesse de défilement sans intervention humaine basée sur des mesures quantitatives de la performance de l'utilisateur [LESH00b]. Leur système rassemble toutes les mesures obtenues dans une fenêtre de temps bien définie de 40 sélections d'items (le choix d'un caractère nécessite 2 sélections). Ensuite le système analyse ces renseignements pour déterminer si le délai de défilement devrait être augmenté, diminué, ou rester inchangé.

Les paramètres mesurés par ces auteurs sont :

- **Erreurs réalisées** : pour détecter les choix erronés, les auteurs ont utilisé une stratégie simple qui consiste à surveiller l'utilisation du retour arrière ou de l'annulation d'un choix. Ils examinent le choix qui vient juste après l'annulation pour vérifier que l'erreur s'est bien produite.
- **Sélections ratées** : représente le nombre de cycles successifs de balayage identiques dans une fenêtre temporelle donnée sans que le sujet ne valide une sélection. Si ce nombre est supérieur à une valeur fixée d'avance (ex : fixé à deux fois au minimum), cela indique généralement que le temps de défilement devrait être augmenté.
- **Sélections ratées dans la liste de prédiction** : dans le cas d'une liste de prédiction les sujets ont besoin d'un temps de défilement plus grand sur les éléments prédits pour pouvoir chercher dans la liste. Un nombre important de choix ratés dans la liste de prédiction indique que le temps de défilement est trop court et doit augmenter. En l'absence d'autres indicateurs d'ajustement (erreurs réalisées ou

sélections ratées), seul le délai de défilement associé à la liste de prédiction est ajusté.

- **Erreurs répétées** : les auteurs se basent sur le principe selon lequel si ces erreurs (erreurs réalisées, sélections ratées, sélection ratée dans la liste de prédiction) se produisent fréquemment, le délai devrait être augmenté. Inversement, si ces erreurs se produisent rarement il est plus approprié de diminuer le temps de défilement.

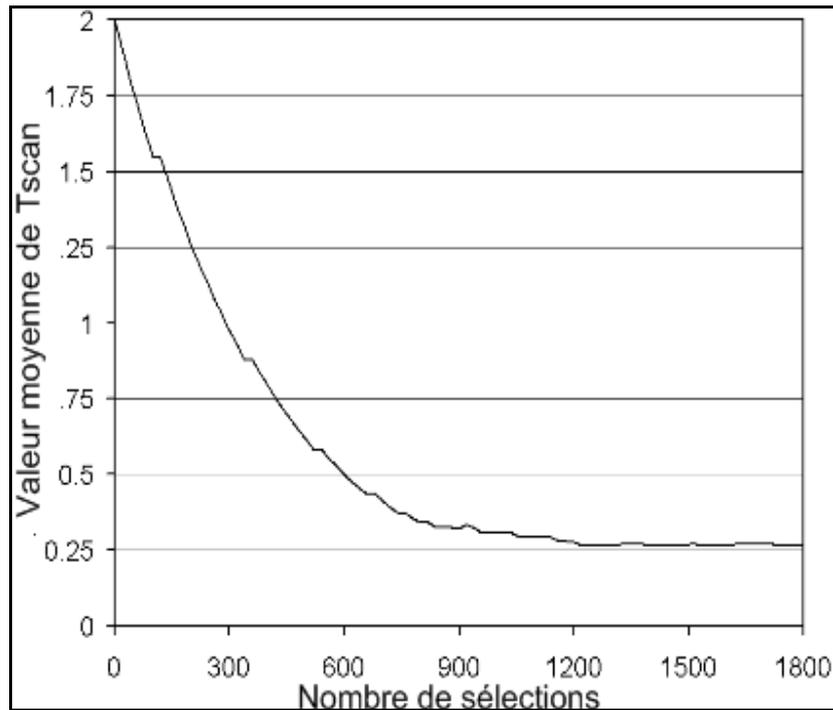
A l'évaluation de ces erreurs les auteurs associent une mesure temporelle en analysant la fraction moyenne du temps d'action réalisé pour chaque sélection correcte, les auteurs déterminent si le délai peut être raccourci sans mettre en danger l'exactitude de la sélection. En pratique les auteurs ont trouvé que les délais pourraient être réduits sans introduire des erreurs de sélection si le temps de sélection moyen était moins que 65% du délai de défilement.

La méthode de Leshner et al propose un plan d'ajustement du délai dans lequel celui-ci est multiplié par un facteur d'ajustement toutes les fois qu'un changement est exigé. Ils analysent, lors de 20 dernières sélections d'un caractère, si l'une au moins des trois conditions suivantes est vérifiée :

- « **Erreurs réalisées** » est validé 3 fois ou plus,
- « **Sélections ratées dans la liste de prédiction** » est validé 3 fois ou plus,
- « **Sélections ratées** » est validé 3 fois ou plus.

Si c'est le cas alors le délai de défilement va augmenter en étant multiplié par un facteur 1.05. Inversement si on n'est pas dans cette situation et que la moyenne des temps d'action T_{act} est inférieure à 0.65 du temps de défilement alors le délai de défilement va diminuer d'un facteur 0.95.

Un exemple de résultats donnés par cette méthode est présenté Figure I.20. Le système d'ajustement a optimisé le temps de défilement à 270ms après approximativement 1200 sélections (~240 mots) en commençant à 2s.

Figure I.20 : Réglage automatique de T_{scan}

I.3.3. Méthode pour adapter le temps de défilement hors ligne

Dans un article récent Simpson et Koester évaluent sur un panel d'utilisateurs handicapés les recommandations émises par l'outil logiciel IDA (Input Device Agent) concernant le délai de défilement [SIMP06a] [SIMP06b]. IDA règle la valeur initiale de T_{scan} en évaluant les erreurs de sélection puis calcule en dynamique le délai T_{scan} optimal en divisant le temps moyen de réaction de la personne par le facteur 0,65. Pour valider cette règle, les auteurs ont réalisé une série de tests sur des sujets habitués à l'utilisation du clavier. L'étude conçue par ces auteurs a pour objectif de faire une comparaison entre le T_{scan} recommandé par IDA et le T_{scan} sélectionné par les sujets. Il est demandé aux sujets d'écrire des lettres qui s'affichent sur l'écran. Les sujets exécutent la tâche sous 4 conditions :

- Phase de recommandation avec IDA (A1) : dans ce cas IDA contrôle T_{scan} . Après 30 validations, IDA propose une valeur de T_{scan} en divisant le temps moyen de réaction par 0.65.
- Phase de choix du sujet (B1): dans ce cas le sujet a la main pour augmenter ou diminuer T_{scan} par tranches de 25 millisecondes.
- Evaluation de la performance de IDA (A2) et des choix du sujet (B2) : il est demandé au sujet d'écrire plusieurs caractères pour enregistrer toutes les données

issues de cette utilisation (taux de texte écrit (TER), pourcentage des cibles correctement choisies (SA), pourcentage des cibles ratées (TE)).

L'ordre des conditions pour chaque sujet est A1, A2, B1, B2 et B1, B2, A1, A2. Les résultats trouvés par les auteurs sont donnés dans le Tableau I.2. Ce tableau montre que les valeurs de T_{scan} recommandées par IDA sont plus petites que celles choisies par les sujets. Ceci n'a toutefois pas augmenté significativement le taux d'erreurs.

Sujets	Tscan		Taux de texte (mots/mn)		Sélections valides		Erreurs répétées	
	IDA	Utilisateur	IDA	Utilisateur	IDA	Utilisateur	IDA	Utilisateur
JP	859	1450	12.19	13.03	76	98	36	18
JM	1199	1325	6.54	13.45	84	78	6	32
MC	1263	2250	8.17	8.26	92	84	10	0
DS	1110	1875	8.56	8.24	98	98	16	6
DY	891	1200	5.86	7.99	92	100	8	8
RW	1293	1200	11.06	10.82	100	90	16	20
Moyenne	1102	1550	8.73	10.30	90.3	91.3	15.3	14.0

Tableau I.2 : Les valeurs de T_{scan} recommandées par IDA et par l'utilisateur

I.3.4. Conclusion

De cet état de l'art nous pouvons constater que deux paramètres sont utilisés pour adapter en temps réel le délai de défilement : le temps de réaction de l'utilisateur et les erreurs de sélection. Ce dernier paramètre suppose toutefois que les erreurs soient pour la plupart effectivement repérables. Nous nous apercevons cependant en pratique que pour gagner en vitesse de communication beaucoup d'utilisateurs ne corrigent pas leurs fautes de frappe tant que la phrase reste globalement intelligible. C'est pourquoi dans notre étude nous nous sommes basés uniquement sur le temps de réaction pour adapter la vitesse de défilement. Nous pouvons noter également que tous les travaux précédemment cités ont été testés, à notre connaissance, en laboratoire. Il serait certainement souhaitable de compléter ces évaluations en situation écologique sur de longues périodes.

I.4.Modélisation de l'interaction humain-machine

I.4.1. Introduction

L'optimisation des performances du système humain-machine, constitué de la personne handicapée et du communicateur à balayage est l'objectif de plusieurs travaux [ABAS04] [BISW07b]. Ces performances sont fonctions des possibilités physiques et cognitives de l'utilisateur, en particulier de son temps moyen de réaction qui induit le temps de défilement du balayage de l'interface graphique ou alphabétique. Elle est également fonction de l'agencement des icônes sur les écrans ainsi que de la méthode permettant d'y accéder [ABAS04], [BISW07b], [EVRE04], [SANG07], [SCHA04]. Dans ce contexte, la modélisation de l'interaction humain-machine peut d'une part faciliter l'analyse du fonctionnement du système humain-machine, d'autre part permettre de prédire ses performances.

En matière de modélisation dans le domaine des aides techniques les quelques études relatées dans la littérature visent à adapter à des personnes handicapées un modèle défini initialement pour des personnes valides. Koester et Levine par exemple ont modélisé la performance d'un utilisateur pour une tâche d'entrée de texte avec prédiction de mots à l'aide d'une commande directe au clavier via un bâton de tête [KOES04b]. Dans [BISW07a] Biswas décrit un simulateur visant à évaluer les aides à la communication. L'utilisateur est ici modélisé à l'aide du modèle MHP (Model Human Processor) développé initialement par Card, Moran et Newell pour des tâches de microinformatique à faible niveau cognitif (réponses à des stimuli) [CARD83]. Sanger et Henderson, dans l'objectif d'optimiser la configuration d'un clavier de communication, modélisent l'interaction humain-machine en évaluant la capacité du canal de transmission selon le paradigme issu de la théorie de l'information de Shannon [SANG07]. Enfin Plos et Buisine dans [PLOS06] utilisent une méthode de Décomposition Multimodale de l'Activité (DMA) pour analyser une tâche bien définie. Le principe de cette méthode est de décomposer la tâche en actions jusqu'à obtenir des actes élémentaires. Cette décomposition poussée permet d'identifier, pour chaque action élémentaire, les modalités perceptives (visuelle, auditive, etc.), motrices (geste, parole, etc.) et cognitives (un choix, une décision, etc.) de l'interaction homme-système. Nous allons voir dans la partie suivante plus en détails le principe de chaque méthode citée ci-dessus.

I.4.2. Le modèle MHP

Le modèle du « Processeur Humain Modèle » (MHP) développé par Card [CARD83] présente une image simplifiée mais synthétique de notre structure mentale. Le MHP comprend trois sous-systèmes interdépendants : les systèmes sensoriel, moteur et cognitif, constitués chacun d'un processeur et d'une mémoire. Les processeurs sont caractérisés par un cycle de base, l'ordre de grandeur de ces cycles étant de 100ms pour le temps élémentaire de perception et de 70ms pour les temps élémentaires moteur et de cognition. Ces valeurs sont des valeurs moyennes : les temps de cycle des processeurs varient d'un individu à l'autre et également en fonction de la situation dans les intervalles suivants [CARD83] :

$T_{\text{cognition}}=T_c \in [25\text{ms}-170\text{ms}]$, $T_{\text{perception}}=T_p \in [50\text{ms}-200\text{ms}]$ et $T_{\text{moteur}}=T_m \in [30\text{ms}-100\text{ms}]$.

Ces temps ont été mesurés pour des personnes valides et ont fait l'objet d'une étude concernant plusieurs groupes de personnes handicapées [KEAT00]. Il en ressort que les temps de perception et de cognition sont comparables entre un panel de personnes valides et les différents groupes de personnes handicapées, une différence notable se situant au niveau moteur (T_m varie selon le type de déficience motrice de 100 à 300ms contre 70ms en moyenne pour un sujet valide).

Nous pouvons décrire le fonctionnement de ce modèle à partir d'une expérience simple (Figure I.21) : il est demandé à un opérateur d'appuyer sur une touche lorsqu'apparaît une icône « OK » à l'écran. Pendant un cycle élémentaire T_p du processeur de perception une représentation physique de cette icône va être chargée en mémoire visuelle et transférée sous forme symbolique en mémoire de travail ou mémoire à court terme. L'étape suivante concerne le processeur de cognition. Il travaille suivant le principe association-action et va associer les mnèmes (unité cognitive symbolique) de la mémoire à court terme à ceux de la mémoire à long terme. Il en résulte une action qui modifie la mémoire à court terme. En l'occurrence ici un seul temps de cognition élémentaire T_c est nécessaire pour associer « OK » à l'action d'appui sur le contacteur. Cette action est exécutée par le processeur moteur en un temps élémentaire T_m .

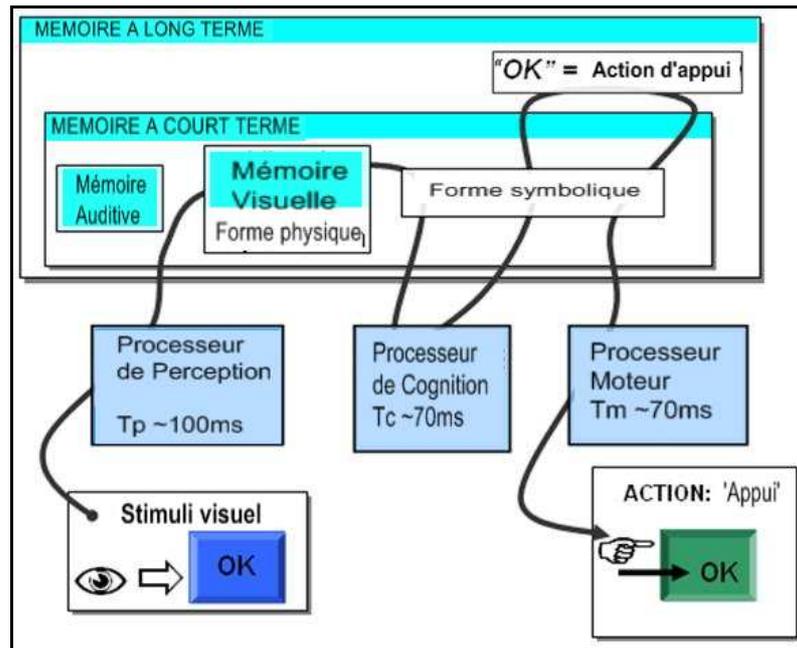


Figure I.21 : Réaction élémentaire à un stimulus selon le modèle MHP

Selon le modèle MHP le temps de réponse d'un opérateur à un stimulus de façon générale est une combinaison linéaire à coefficients entiers de ces trois temps. Le temps de réponse aux stimuli peut être décrit par l'équation suivante :

$$T = x * T_{\text{perception}} + y * T_{\text{cognition}} + z * T_{\text{moteur}} \quad \text{où } x, y, z \text{ sont des entiers.}$$

La simplicité du modèle a ses inconvénients : le modèle du « Processeur Humain Modèle » ne fournit aucune indication sur les représentations, il se limite à la prédiction ou à l'explication de phénomènes de performance de bas niveau. Le Modèle MHP est donc trop réducteur pour assister l'informaticien dans sa tâche de conception de systèmes interactifs, en effet, il ne véhicule aucune méthode de conception [COUT90]. Le modèle Keystroke présenté ci-dessous tente de combler cette lacune.

I.4.3. Le modèle Keystroke Level Model (KLM)

I.4.3.1. Définition de KLM

Le modèle KLM [CARD83] [COUT90] concerne les aspects syntaxiques et lexicaux de l'interaction. Ses éléments relèvent des actions physiques que l'utilisateur doit effectuer pour spécifier une commande. Le modèle KLM estime selon un scénario donné la durée d'une

tâche à exécuter, décomposable en plusieurs sous tâches. Ce modèle est constitué des opérateurs de temps suivants :

$$T = TK + TP + TH + TD + TM + TR \text{ où}$$

T : temps total nécessaire à l'exécution d'une tâche;

TK : temps nécessaire pour effectuer un clic;

TP : temps de pointage ;

TH : temps nécessaire pour saisir un dispositif ;

TD : temps de déplacement du curseur d'un point à un autre ;

TM : temps de l'activité mentale de l'utilisateur ;

TR : temps de réponse du système.

I.4.3.2. Les travaux basés sur le KLM

Koester et Simpson se sont basés dans leurs travaux sur le modèle KLM. Cette technique fournit un moyen pour identifier l'activité cognitive, perceptuelle et motrice qu'un sujet doit accomplir pour une tâche bien définie. Le temps nécessaire pour réaliser une tâche est la somme des prédictions de temps de chaque sous tâche. La première caractéristique de KLM est sa capacité à représenter correctement la performance d'un utilisateur en utilisant seulement un petit nombre de paramètres.

Koester et Simpson [SIMP00] ont modélisé la performance d'un utilisateur pour une tâche d'entrée de texte avec prédiction de mots en utilisant un capteur de type Tout Ou Rien. Les facteurs clés représentés dans ce modèle sont :

- Les caractéristiques de l'utilisateur ;
- La stratégie pour chercher les mots dans la liste ;
- La configuration du système de prédiction des mots.

Deux paramètres de ce modèle représentent les caractéristiques de l'utilisateur :

t_s : temps mis par l'utilisateur pour chercher dans la liste des mots.

t_k : temps nécessaire pour appuyer sur la touche.

Les deux autres facteurs sont déterminés par le système de configuration :

S : nombre de recherches nécessaires par caractère ;

k_{sva} : pourcentage de touches appuyées en utilisant la prédiction de mots.

Le temps moyen nécessaire pour produire chaque caractère est alors :

$$T_{wp} = S * ts + (1 - Ksva) * tk \text{ (en secondes par caractère)}$$

Ce modèle est utile au développement d'une interface qui utilise la prédiction de mots selon deux points de vue :

- D'un point de vue qualitatif, le modèle illustre les facteurs qui influencent l'utilisateur quand il utilise la prédiction de mots et la puissance relative de ces facteurs ;
- D'un point de vue quantitatif, la simulation du modèle permet de mesurer l'influence des paramètres sur la performance de l'utilisateur.

La plupart des études qui visent à modéliser la performance d'un utilisateur ne tiennent compte que de la composante motrice de l'activité de l'utilisateur via le paramètre « Keystroke Saving » (KS : nombre de frappes économisées grâce à la prédiction de mot,

$$KS = \frac{\text{NombreDeCaractèresSaisis}}{\text{NombreTotalDeCaractères}}).$$

Koester et Levine rajoutent dans leurs travaux une composante cognitive par un modèle à deux paramètres t_k (« Keypress Time ») et t_s (« List Search Time »). Malgré l'économie de frappes réalisée mesurée par KS, la charge des activités perceptuelles et cognitives du sujet va augmenter du fait que la liste des mots change toujours et que le sujet doit parcourir toute cette liste pour retrouver le mot. Les expérimentations menées à partir de ce modèle leur permettent de conclure que le gain de temps qu'implique l'économie de frappes est en partie compensé par le temps perdu en activités cognitives et perceptives pour rechercher le mot adéquat dans la liste [KOES94b].

I.4.4. Le modèle basé sur le canal de transmission

Dans l'objectif d'optimiser la configuration d'un clavier de communication, Sanger et Henderson modélisent l'interaction humain-machine en évaluant la capacité du canal de transmission selon le paradigme issu de la théorie de l'information de Shannon [SANG07].

Ces auteurs se basent sur l'adaptation de l'interface de communication pour maximiser le taux de communication de chaque utilisateur. Ce taux est déterminé par 3 facteurs :

- Les capacités cognitives du sujet pour sélectionner une séquence significative ;
- La capacité sensorielle du sujet pour localiser et identifier les boutons sur l'interface ;
- La capacité motrice du sujet pour activer un choix.

Le taux optimal de communication est déterminé par la capacité du canal, qui est la valeur maximale du taux d'information des boutons du clavier ou de l'aménagement cellulaire de l'outil de communication. Les auteurs construisent un modèle mathématique pour calculer le taux d'information basé sur le rapport entre le temps de réaction et le nombre de boutons par écran, la dimension des boutons et la longueur d'une séquence de boutons qui doivent être appuyés. Leur méthode de travail consiste à chercher pour chaque sujet un modèle mathématique qui prédit un rapport entre la taille du bouton, l'espacement entre les boutons, le nombre d'appuis et le taux de communication.

Après avoir développé un tel modèle, les auteurs peuvent choisir les paramètres nécessaires pour le communicateur qui maximise le taux de communication pour chaque utilisateur en se basant sur sa vitesse d'action et les erreurs réalisées pour activer un bouton bien défini. Les paramètres utilisés pour la configuration de l'interface sont :

- La largeur du bouton : w ;
- Le nombre de bouton par écran : b ;
- La plus longue séquence (m) de boutons qui doivent être pressés pour communiquer avec un seul symbole ou un mot dans le vocabulaire.

La valeur du taux d'information de l'interface est donnée par la formule suivante :

$$R(w,b,m) = \frac{\text{entropy}(\text{vocabulary})}{TT(w,b,m)} \quad \text{où}$$

$\text{entropy}(\text{vocabulary})$ est la somme sur tous les éléments du vocabulaire de $p(x)\log(p(x))$, $p(x)$ étant la probabilité d'utiliser le mot x du vocabulaire ;

$TT(w,b,m)$ est le temps total moyen (temps du mouvement et temps de réaction) nécessaire pour sélectionner un élément du vocabulaire.

Les résultats expérimentaux montrent que l'utilisation des paramètres optimaux prédits pour chaque utilisateur pour programmer leurs aides à la communication a amélioré le taux de communication de cinq des dix sujets, comparée avec la programmation proposée par des professionnels. Les auteurs ont trouvé que leur modèle de prédiction donne des résultats proches des valeurs mesurées du taux d'information. Ils ont trouvé aussi que chaque sujet se comporte de manière différente, des valeurs personnalisées pour le nombre optimal de boutons ainsi que pour leurs dimensions s'avèrent donc nécessaires.

I.4.5. Le simulateur basé sur un modèle

Biswas décrit un simulateur visant à évaluer les aides à la communication [BISW07a]. Le simulateur prend la définition de la tâche et la localisation des différents objets dans l'interface comme des entrées. Il prédit le parcours nécessaire pour réaliser une tâche donnée et le temps d'achèvement de cette tâche, pour les différents dispositifs d'entrée (par exemple souris ou capteur de type Tout Ou Rien) et pour les différents niveaux de déficience des sujets.

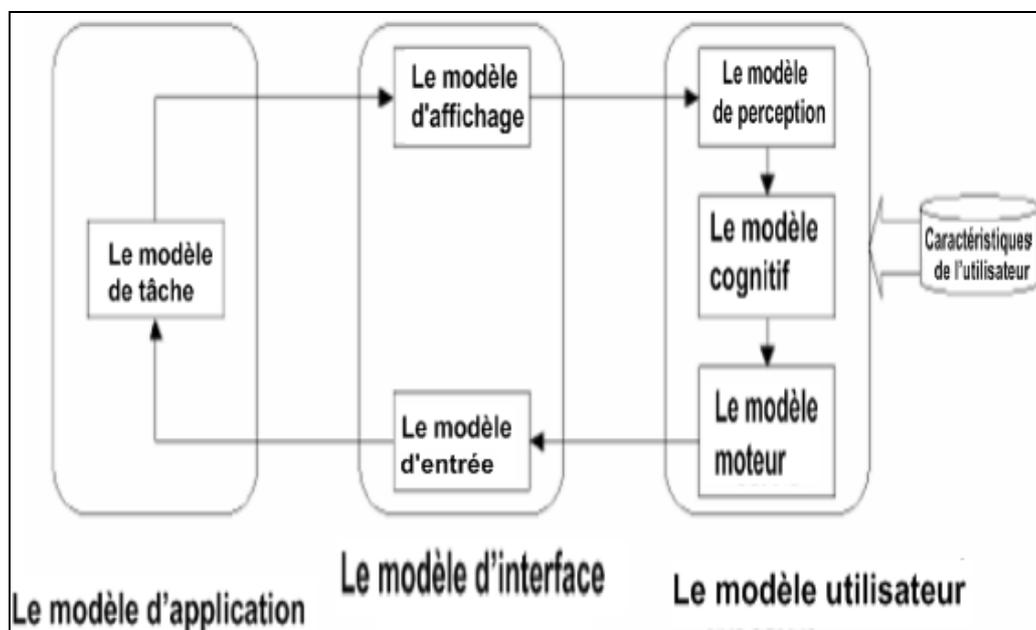


Figure I.22 : Architecture du simulateur

L'architecture du simulateur est présentée Figure I.22. Elle est composée de trois blocs :

- Le modèle d'application : présente la tâche courante en la décomposant en un ensemble de tâches simples.
- Le modèle d'interface : selon le type de déficience le modèle décide du type d'interface utilisée par le sujet et configure les paramètres adéquats.
- Le modèle utilisateur : simule le modèle de l'utilisateur pour une tâche donnée. Cette tâche est analysée par le modèle d'application sous la configuration établie par le modèle d'interface. L'utilisateur est modélisé à l'aide du modèle MHP.

I.5. Conclusion du chapitre 1

La réalisation d'une interface de communication suppose la connaissance précise du comportement de chacune des entités à relier. Dans le cas particulier où l'une des entités est un sujet handicapé, la définition de l'interface devient une tâche complexe, souvent teintée d'empirisme et d'arbitraire. C'est pour cette raison que plusieurs auteurs visent à optimiser les performances du système humain-machine en se basant sur la modélisation du comportement du sujet et l'ajustement de l'interface.

Ce chapitre a présenté les différents aides à la communication pour personnes handicapés moteurs. Nous avons tout d'abord classé ces aides en communications indirecte et directe, ensuite nous avons détaillé les différentes approches visant à accélérer la communication dans les systèmes à balayage. Deux approches sont proposées dans la conception de plusieurs logiciels pour augmenter le taux de saisie du texte. La première vise à optimiser le temps d'accès à la touche recherchée pour réduire le temps de défilement. La seconde approche vise à économiser le nombre de saisies, en complétant d'une manière automatique des parties du message en fonction de la saisie. Dans la deuxième partie du chapitre, nous avons cité les quelques études portant directement sur notre travail de thèse : adapter la vitesse de défilement automatiquement en cours d'utilisation du système. Les méthodes proposées se basent toujours sur deux paramètres, le temps de réaction et/ou les erreurs réalisées par la personne handicapée.

La troisième partie de ce chapitre a été consacrée à la modélisation de l'interaction entre la personne handicapée et l'aide technique à la communication. Nous avons présenté quelques modèles décrits dans la littérature visant cet objectif, et nous avons détaillé le modèle MHP

défini initialement pour des tâches de microinformatique mais bien adapté à ce contexte spécifique. Ce modèle décompose notre structure mentale en trois sous-systèmes interdépendants : les systèmes sensoriel, moteur et cognitif. Notre travail de thèse se base sur ce modèle car il apparaît bien adapté à notre contexte de travail où la tâche du sujet est de répondre à un stimulus visuel par une action motrice simple sur un contacteur.

II. CHAPITRE 2 :

CONCEPTION ET MODELISATION D'UN COMMUNICATEUR A DEFILEMENT

II.1. Introduction

Depuis le début des années 80, de nombreux systèmes ont été conçus visant à aider les personnes sévèrement handicapées à contrôler leur environnement. Dans les situations de déficiences motrices les plus sévères la sélection d'une commande repose toujours sur le même principe : un défilement lumineux contrôlé par le capteur qui permet de sélectionner un item. Le problème essentiel qui se pose lors de la conception d'une aide technique de ce type est la rapidité d'accès aux différents choix. Plusieurs paramètres interviennent dans la rapidité de communication :

- La vitesse de défilement ;
- Le type de balayage (défilement linéaire, ligne/colonne, ...) ;
- Le nombre d'items ;
- La forme de la matrice des items ;
- L'ordre des items.

La réalisation de tels dispositifs repose en général sur des méthodes empiriques. Leur performance est fonction de l'interface graphique, mais aussi des capacités physiques et cognitives de la personne. Notre objectif dans ce contexte est l'optimisation de l'interaction homme-machine pour permettre une communication ou un contrôle d'environnement, le plus rapide possible. Nous nous proposons pour cela d'analyser cette interaction en la modélisant.

Nous présentons tout d'abord dans ce chapitre le système EDiTH (Environnement Digital de Téléaction pour personnes Handicapées) développé au laboratoire LASC et sur lequel est basé la partie applicative de notre travail. Nous proposons ensuite une modélisation de l'interaction

humain-machine pour les communicateurs à balayage basée sur le modèle « MHP » ainsi qu'une méthode pour estimer ses paramètres : les temps moteur, de cognition et de perception.

II.2. Le système EDiTH (Environnement Digital de Téléaction pour personnes Handicapées)

Le système EDiTH (Environnement Digital de Téléaction pour personnes Handicapées) est une interface intégrant de multiples fonctionnalités d'aide à la communication et au contrôle d'un environnement multimédia dont la première version a été développée en 1996 pour une personne atteinte de sclérose latérale amyotrophique. Il s'agit d'un système à balayage, colonne/ligne ou séquentiel selon les écrans, contrôlé par un capteur de type tout-ou-rien adapté à l'utilisateur. EDiTH est à la fois une aide à la communication et au contrôle d'environnement et un outil de recherche permettant des évaluations en situation « écologique » c'est-à-dire dans le contexte réel de vie des utilisateurs.

II.2.1. L'objectif d'EDiTH

L'objectif du système EDiTH est de pallier la déficience interactionnelle progressive en concevant une interface compatible avec les processus cognitifs centraux et la (ou les) ressource(s) motrice(s) encore valides. En d'autres termes, le système doit remettre le sujet en interaction avec son environnement. La cible d'EDiTH englobe les personnes atteintes d'un lourd handicap moteur de type tétraplégie ou de syndromes évolutifs comme la sclérose latérale amyotrophique amenant progressivement à une paralysie complète de l'individu.

Le système EDiTH est défini comme une interface interactive adaptée aux exigences mentales de la personne handicapée et de son environnement pour améliorer sa vie quotidienne. Les caractéristiques de cette interface reposent tout d'abord sur une analyse empirique des besoins de l'utilisateur. Le but de notre interface est d'agir sur l'environnement physique et social du malade.

II.2.2. Etude critique de l'existant

Le projet EDiTH a commencé en 1996 [BRAN97]. Son développement, reposant sur le langage Visual Basic, n'a cessé depuis cette date, le logiciel recevant régulièrement de nouvelles fonctions. Son architecture se base sur l'utilisation d'un grand nombre de fenêtres et de fichiers rendant la maintenance du code informatique laborieuse. La gestion de l'espace repose sur un menu horizontal disposant de six fonctions ainsi que du bouton d'urgence (Figure II.1). Le point crucial d'EDiTH se situe donc logiquement au niveau de l'ergonomie, d'où la nécessité d'un changement du langage pour utiliser les nouvelles technologies de programmation et rendre l'interface plus interactive.



Figure II.1 : Menu principal d'EDiTH « version 2005 »

II.2.3. Choix du langage de programmation

La nouvelle version d'EDiTH développée dans le cadre de cette thèse est implémentée en langage C-Sharp (couramment noté C#) et son Framework « .NET ». Le choix de développer dans le langage C#.net sous Visual C# Express est dû au fait que ce langage se montre parfaitement adapté aux nouvelles technologies. C# est orienté objet et il est compilé comme du code managé. Il bénéficie des services du « Common Language Runtime » tels que l'interopérabilité des langages ou la sécurité améliorée. La communauté des développeurs est très active, offrant de nombreux exemples et solutions pour la plupart des problèmes rencontrés.

Le Framework .NET a été conçu dans un but de compatibilité inter-langage, ce qui signifie qu'un programme écrit en C# pourra utiliser une DLL d'un tout autre langage sans aucune problème. Cette interopérabilité s'étend même à l'héritage. Il est très étoffé et permet simplement l'intégration de composants ActiveX, dll et autres, apportant ainsi aisément les possibilités d'Internet Explorer ou encore de Windows Media Player.

II.2.4. Spécifications Techniques d'EDiTH

L'architecture matérielle d'EDiTH est celle d'un PC multimédia portable pour des raisons de commodité en milieu hospitalier. Le logiciel est composé de deux parties, une interface utilisateur et une interface de configuration que nous allons voir plus loin. Le dispositif physique d'interaction humain-machine est, par hypothèse, un capteur de type tout ou rien, adapté aux capacités motrices de l'utilisateur. Deux types de défilements sont utilisés, un défilement colonne/ligne dans les interfaces « clavier virtuel » et « communication verbale » et un défilement linéaire dans les autres fonctions. La validation d'un item a pour effet de faire reprendre le défilement au début.

Nous avons introduit un « clic long » qui sera activé pour les personnes qui peuvent maîtriser suffisamment la commande du capteur TOR en termes de durée. Sa durée et sa fonction sont paramétrables dans la partie configuration (Figure II.12). Le « clic long » consiste à appuyer de façon prolongée sur une touche quelconque jusqu'au changement de couleur. L'action associée peut correspondre par exemple au retour au niveau précédent ou à l'activation de la sonnette. L'activation de l'option « clic long » permet de minimiser le nombre de défilements selon l'action qui lui correspond.

L'analyse de l'utilisation du clavier virtuel d'EDiTH [BOUR02] a montré que le délai de sélection de la première case d'un écran est largement supérieur au temps d'action moyen mesuré pour les autres items. L'ajout d'un délai supplémentaire à chaque changement d'écran permet donc de diminuer le nombre de fois où on rate le premier item. Ce délai est réglable dans la partie de configuration.

II.2.5. Interface utilisateur

Nous nous sommes basés pour la conception d'EDiTH sur les principes ergonomiques suivants :

- Uniformisation de l'interface pour faciliter la compréhension des interactions items/fenêtres ;
- Utilisation du double encodage texte + icônes pour faciliter la mémorisation des fonctions ;
- Standardisation de la disposition des groupes de bouton pour optimiser l'espace ;

- Agencement des items dans chaque écran prévu pour optimiser l'interface ;
- Intégration de la lecture par synthèse vocale dans toutes les fonctionnalités présentant des informations écrites (Editeur, Mail) ;
- Utilisation d'items à doubles actions complémentaires (ex: pause/play) ;
- Positionnement de l'item « Quitter » dans le début de l'interface pour permettre un retour rapide au niveau précédent en cas d'erreur ;
- Possibilité d'activer l'émission d'un son lorsqu'une touche est validée afin d'éviter les doubles clics intempestifs (feedback sonore).

Dès qu'EDiTH est lancé, le défilement commence sur la partie gauche de l'écran sur les trois fonctions principales « Appel », « Communiquer » et « Loisir ». Au niveau de la présentation des informations, l'écran est segmenté en deux parties :

- La zone des fonctionnalités générales (ou menu principal) est positionnée à gauche de l'écran et recouvre les fonctionnalités générales du système ;
- Des aires secondaires d'affichage sont positionnées au centre de l'écran chaque fois qu'une fonctionnalité est activée. Au démarrage cette partie centrale présente la carte d'EDiTH, elle permet à l'utilisateur de connaître le chemin à suivre pour atteindre une fonction désirée comme le montre la Figure II.2.

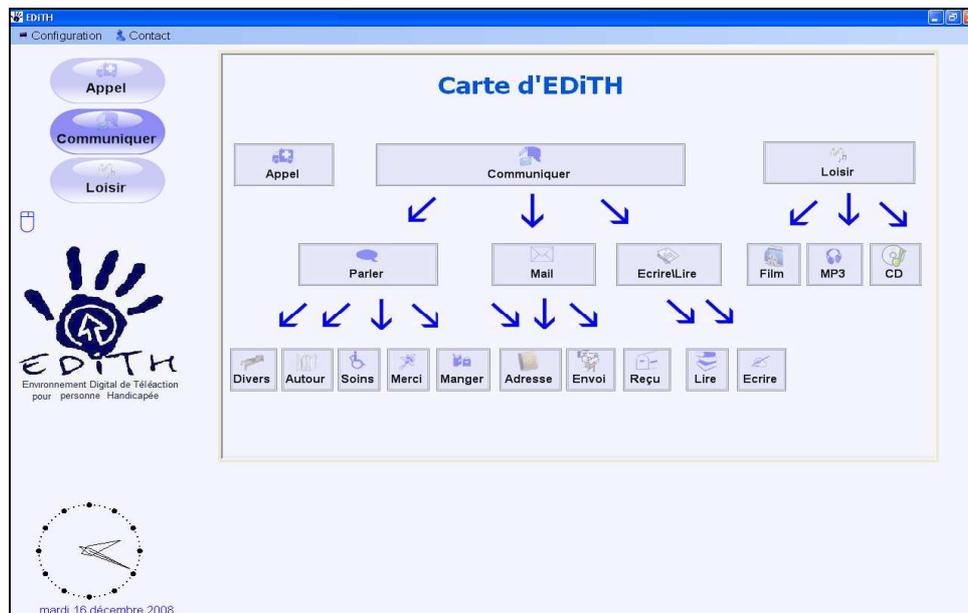


Figure II.2 : Interface principale d'EDiTH

Pour accéder aux menus *Configuration* et *Contact* dans la barre de menus (action réalisable par une personne pouvant manipuler une souris), il faut appuyer sur la touche « M » du clavier pour rendre la souris mobile. Pour rendre la main à l'utilisateur, il faut appuyer sur la touche « F » du clavier, la souris reviendra sur sa position fixe. Il n'est donc pas prévu à l'heure actuelle de laisser la possibilité à l'utilisateur de configurer lui-même son système.

II.2.6. Les fonctionnalités du système EDiTH

La fonction d'EDiTH est d'améliorer l'autonomie de sujets atteints d'un déficit moteur sévère. Ils ont ainsi la possibilité d'étendre leurs capacités opératoires et cognitives en pouvant par l'intermédiaire du système avertir, lire, écrire, parler, écouter et regarder.

II.2.6.1. Appeler le personnel soignant

Cette fonction permet d'avertir un proche ou le personnel soignant d'un problème ou d'un besoin particulier. La sélection de la commande « appel » déclenche une sonnerie.

II.2.6.2. Lire des textes

Cette fonction donne accès à la lecture de textes : nous pouvons rechercher un texte écrit ou un ouvrage, l'ouvrir, le lire, passer à la page suivante ou précédente. Un marque-page virtuel permet de quitter le texte à une position donnée pour y retourner plus tard (fonction automatique) (Figure II.3). La fonction « Lecture Audio » permet d'activer une synthèse vocale pour faire prononcer le texte à partir d'une position donnée.

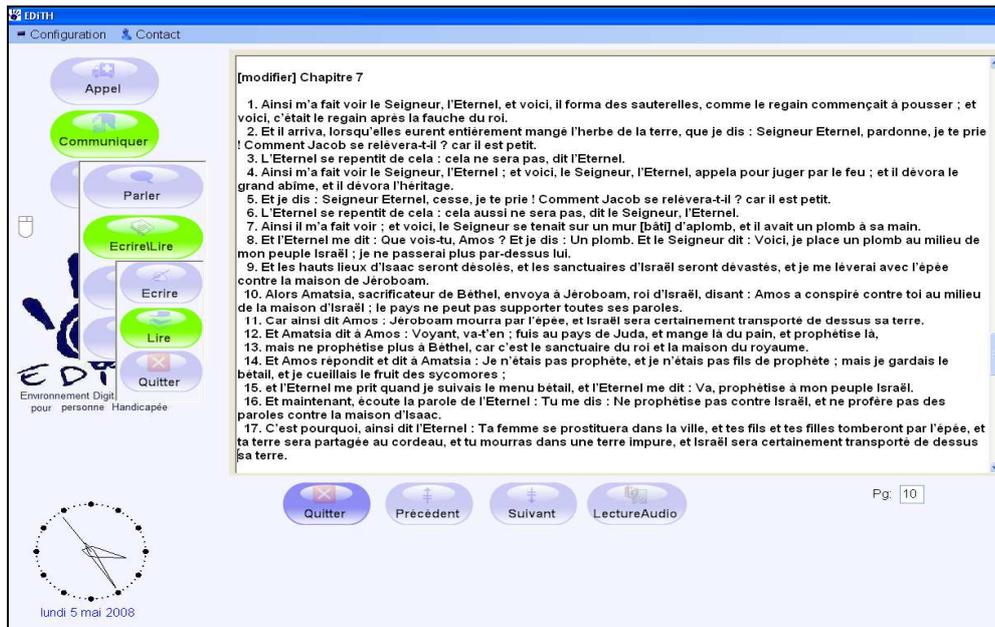


Figure II.3 : Interface pour lire du texte

II.2.6.3. Communiquer par écrit

L'utilisateur peut communiquer par écrit par l'intermédiaire d'un clavier virtuel statique (Figure II.4). L'interface est composée d'une zone d'édition et d'un clavier virtuel. Le balayage s'effectue de façon classique selon le mode « colonne-ligne ». Chaque item nécessite donc deux actions sur le capteur pour être sélectionné, à l'exception de la touche « espace » constituant la première colonne et validable en une seule action. Après chaque validation d'une ligne le défilement reprend au début. L'action sur la dernière case vide de chaque colonne permet en cas d'erreur de sélection de reprendre le défilement au début. C'est également le cas si nous laissons le défilement s'effectuer deux fois sans validation sur une même colonne.

L'agencement des lettres sur le clavier présenté en Figure II.4 résulte d'un compromis entre la fréquence des lettres dans la langue française et les préférences des utilisateurs. En sus des caractères de l'alphabet, le clavier virtuel est composé de touches de déplacement, de touches de ponctuation et d'une touche « 123 » d'accès à un pavé numérique. Lorsque nous sommes en mode pavé numérique, le défilement ne se poursuit que sur une partie du clavier (Figure II.5). La touche « ABC » permet de revenir au clavier initial.

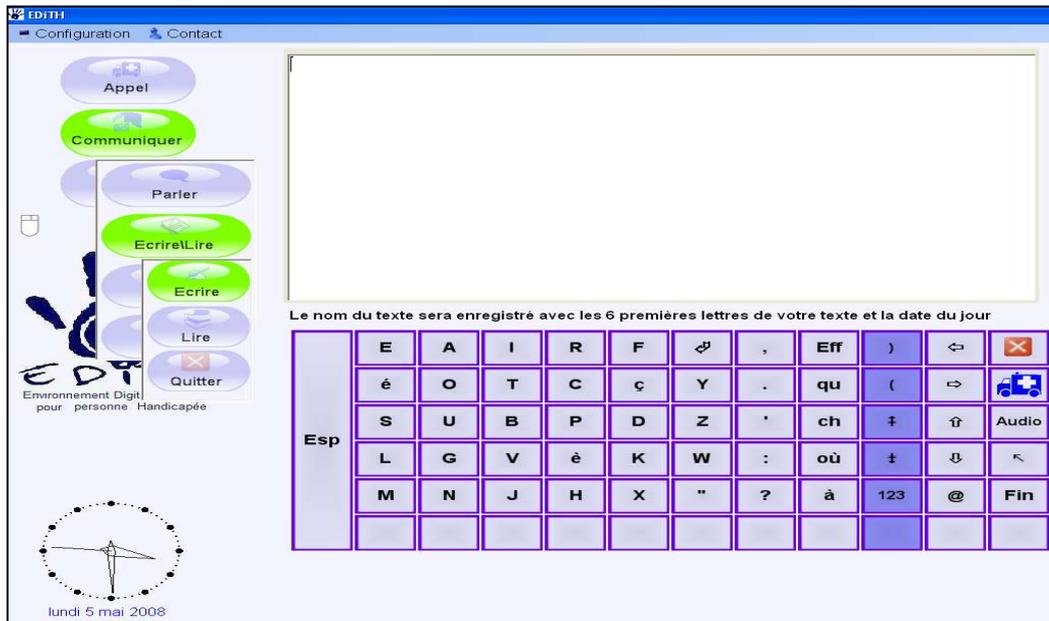


Figure II.4 : Interface « Editeur de texte » (défilement colonne/ligne)

Quelques touches sont spécifiques au communicateur :

- La touche « **Eff** » permet d'effacer un caractère ;
- La première touche de la dernière colonne fait quitter le mode édition. On peut alors enregistrer le texte, envoyer un mail ou retourner à l'éditeur ;
- La touche symbolisant une **ambulance** est un accès direct à la fonction « appel » sans retour au menu principal, elle active la sonnette d'alarme ;
- La touche « **Audio** » permet d'énoncer verbalement via une synthèse vocale le message écrit.

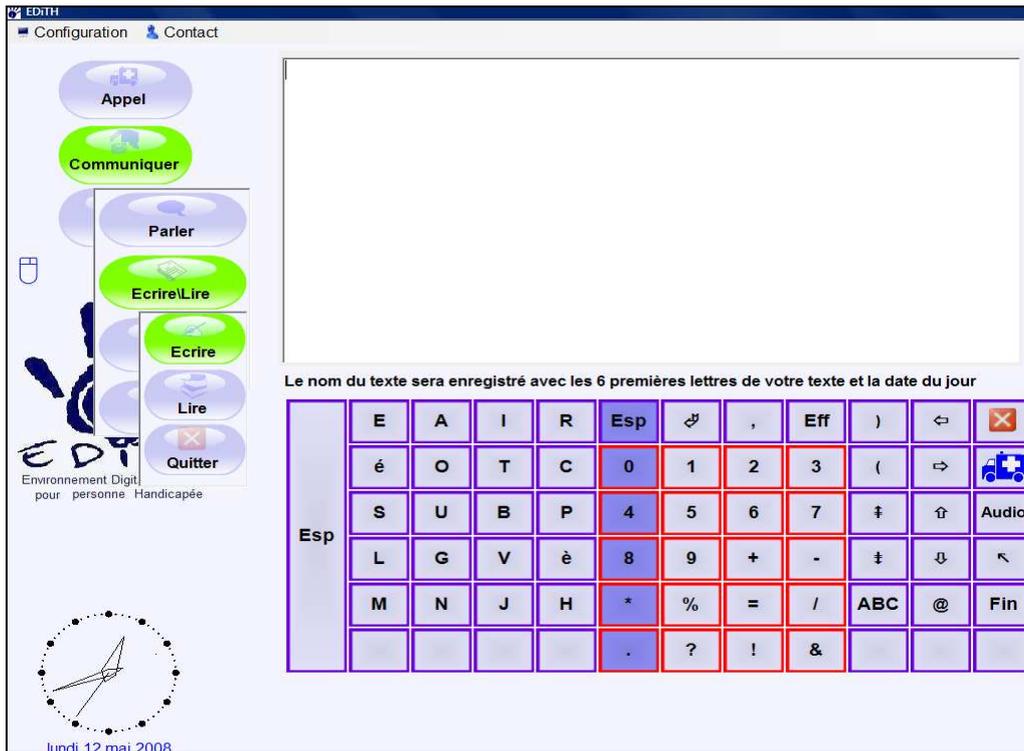


Figure II.5 : Interface « Editeur de texte » (mode pavé numérique)

II.2.6.4. Communiquer verbalement

Un certain nombre de phrases préenregistrées sont classées par thèmes (soins, politesse, besoins physiologiques, requêtes de confort,...). Leur sélection déclenche leur prononciation par le synthétiseur vocal. Cette fonctionnalité n'autorise donc qu'une communication standardisée, elle est utilisée en particulier pour des requêtes rapides. L'interface « soins » de la Figure II.6 permet notamment de sélectionner une zone du corps d'une femme ou d'un homme par balayage ligne/colonne puis la requête elle-même (masser, gratter, etc.).

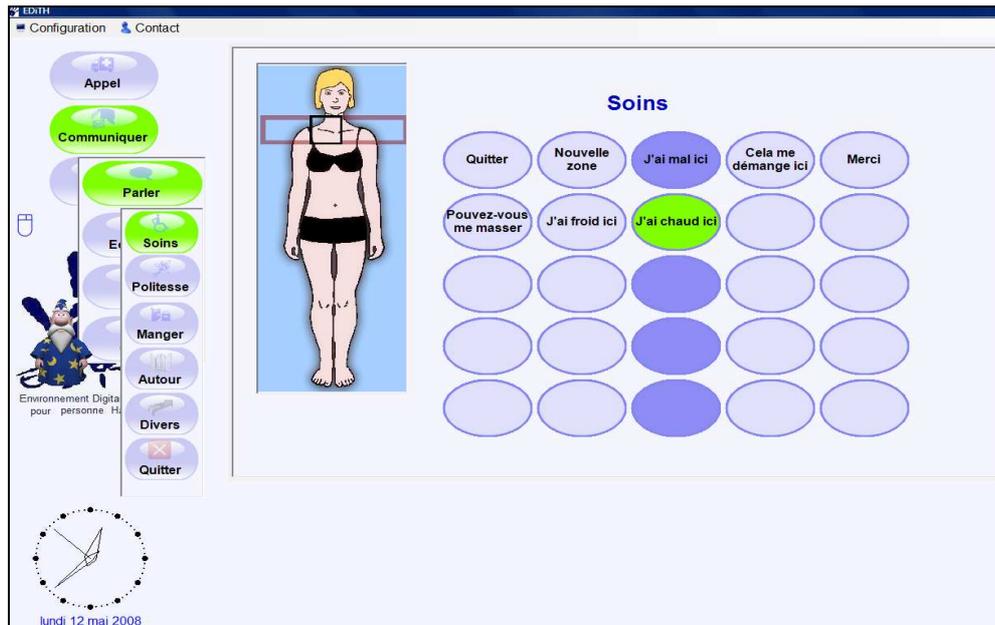


Figure II.6 : Interface « Soins »

II.2.6.5. Accéder à un environnement multimédia

a) *Ecouter de la musique*

L'utilisateur a la possibilité d'écouter de la musique (MP3, CD) et de regarder des films. Il peut choisir le morceau et régler le volume sonore (Figure II.7). Un mode de répétition permet de jouer en boucle de la musique.

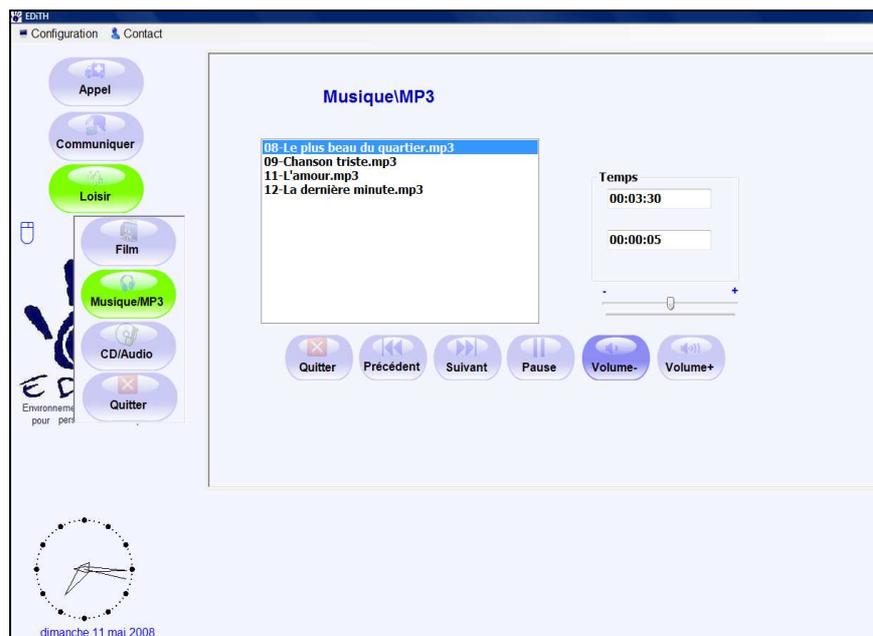


Figure II.7 : Interface « Musique\MP3»

Dans le cas où le sujet maintient son capteur de commande pour activer les boutons « Précédent » ou « Suivant », le défilement continue dans la liste de musique tant que le sujet n'a pas relâché le capteur. Cette méthode permet d'accélérer le choix dans la liste de musique. Nous trouvons le même scénario pour « Volume- » et « Volume+ », le maintien permettant de diminuer (ou augmenter) le volume d'une manière continue. Quand la musique est arrêtée la touche « Pause » se transforme en « Lecture » et inversement la touche se transforme en « Pause » quand on écoute la musique.

b) Visualiser les photos

Cette interface permet au sujet de regarder un diaporama de photos, d'augmenter ou diminuer la vitesse d'affichage et même d'agrandir l'image. Quand le diaporama est arrêté, la touche « Pause » se transforme en « Lecture », « Vitesse- » se transforme en « Précédent » et « Vitesse+ » se transforme en « Suivant » (Figure II.8).

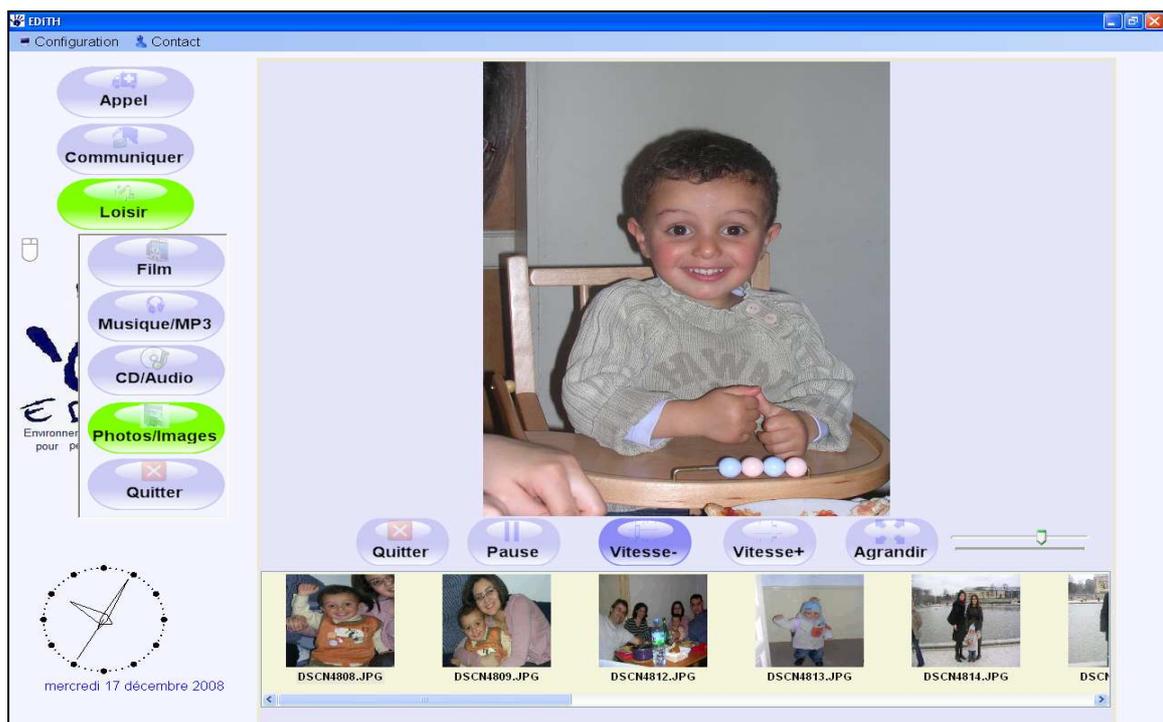


Figure II.8 : Interface « Photos/Images »

II.2.6.6. Courrier électronique

La fonction « courrier électronique » permet d'envoyer des emails et de lire les messages. Il faut auparavant configurer la fonctionnalité Mail dans le menu de configuration. Cette

fonctionnalité est très basique : elle ne permet pas de voir les pièces jointes ni de filtrer son courrier.

L'interface « E-mails envoyés » contient les mails envoyés (police normale) et non-envoyés (en gras). Un exemple est présenté Figure II.9. Tous les mails sont classés par date, les mails les plus récents se trouvant au début de la liste. On peut sélectionner le destinataire du mail dans un carnet d'adresses (Figure II.10). Chaque fois qu'on écrit à une nouvelle personne, son adresse s'ajoute automatiquement au carnet d'adresses.

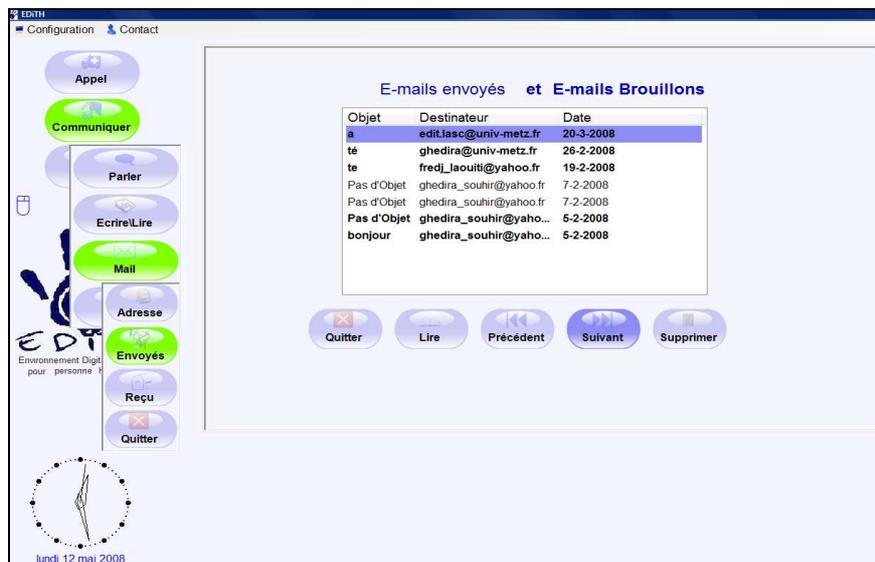


Figure II.9 : Interface « E-mails envoyés »

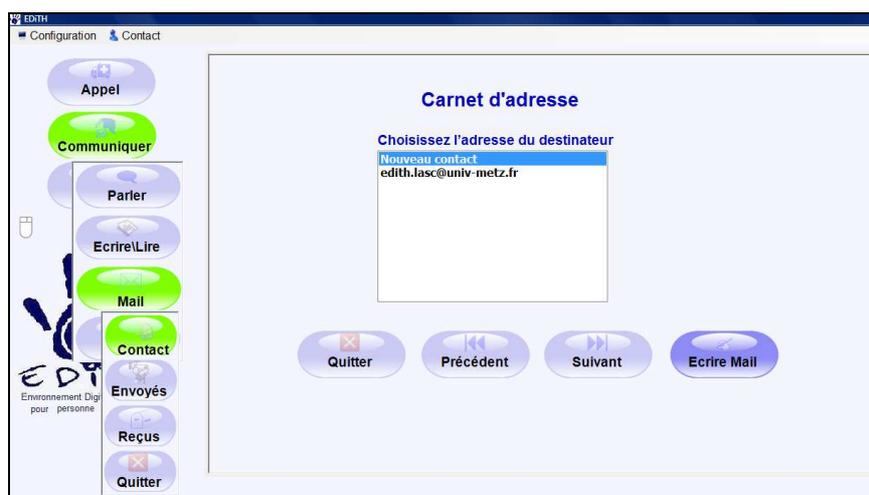


Figure II.10 : Interface « Carnet d'adresses »

II.2.7. Interface de configuration

L'interface de configuration n'est pas accessible à l'utilisateur, elle permet à son entourage médical ou familial de personnaliser l'interface opérationnelle d'EDiTH : choix des couleurs, ajout de livres, de dossiers MP3 ou de films, choix des phrases pour communiquer, configuration de la voix, modification de l'emplacement des touches du clavier virtuel (Figure II.11).

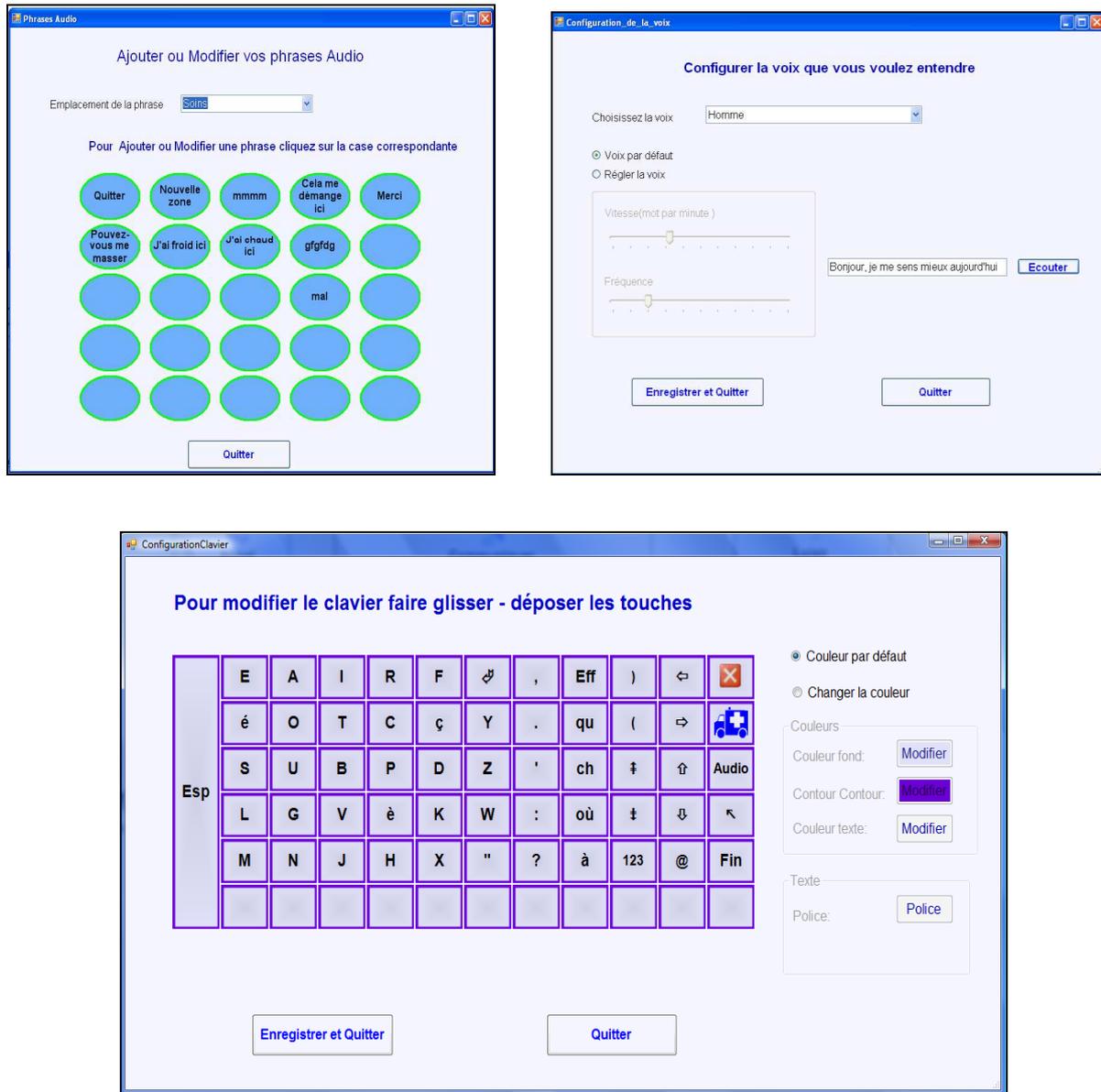


Figure II.11 : Configuration des phrases audio, de la voix et du clavier virtuel

La vitesse de défilement est configurable selon trois modes :

- Un mode fixe : la vitesse de défilement est fixée au début et reste la même tout au long de l'utilisation du système ;
- Un mode adaptatif selon l'algorithme des « anticipations/erreurs »: la vitesse de défilement se modifie en cours d'utilisation du système pour s'adapter à l'état de l'utilisateur. Plusieurs paramètres intervenant dans le calcul du temps de défilement adaptatif sont réglables (Figure II.12) et seront détaillés dans le chapitre 3 ;
- Un mode adaptatif selon l'algorithme de Lesher et al. [LESH00]. Ce mode nous servira d'élément de comparaison pour la validation de l'algorithme des « anticipations/erreurs » proposé dans ce mémoire.

Dans cette même interface nous avons également le choix d'activer ou pas le « clic long ». S'il est activé nous pouvons configurer le temps de pause et l'action associée. Nous avons aussi la possibilité d'ajouter un délai supplémentaire à chaque changement d'écran. Ce délai est réglable selon l'utilisateur.

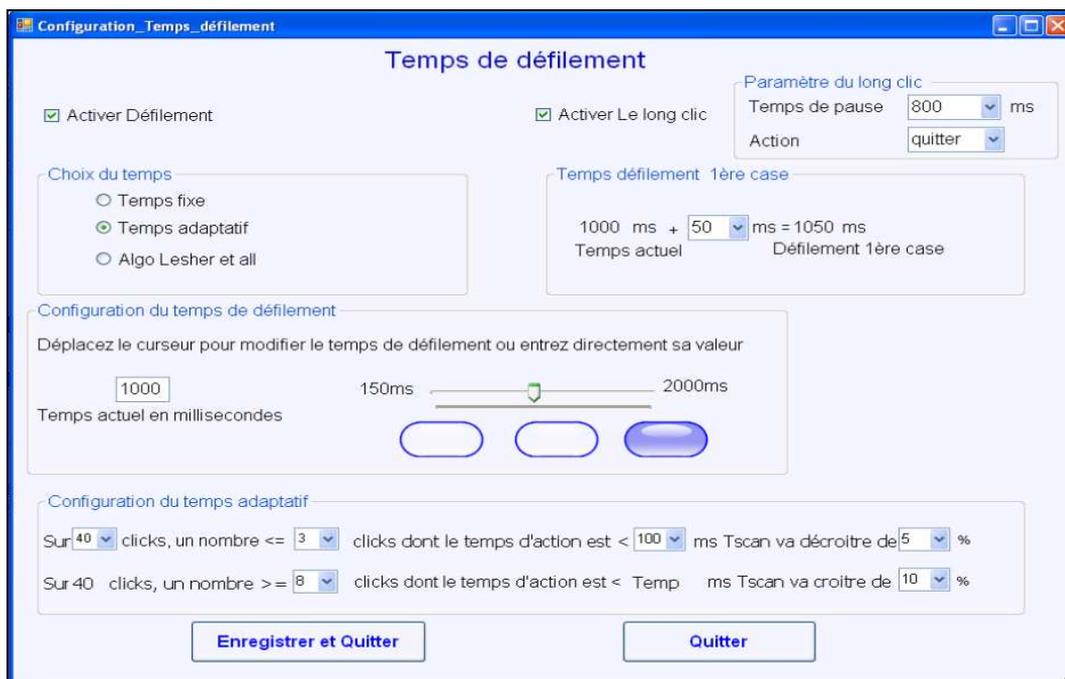


Figure II.12 : Configuration du temps de défilement et du « clic long »

Dans notre application interactive nous avons utilisé aussi des formulaires pour saisir les données de l'utilisateur ou pour configurer le serveur email (Figure II.13). L'avantage des formulaires est qu'ils sont auto-explicatifs, ils rendent explicites la sémantique et la syntaxe

du système, ils permettent un ajout facile et visible des données et une utilisation optimale de l'espace disponible à l'écran [BRAN00].

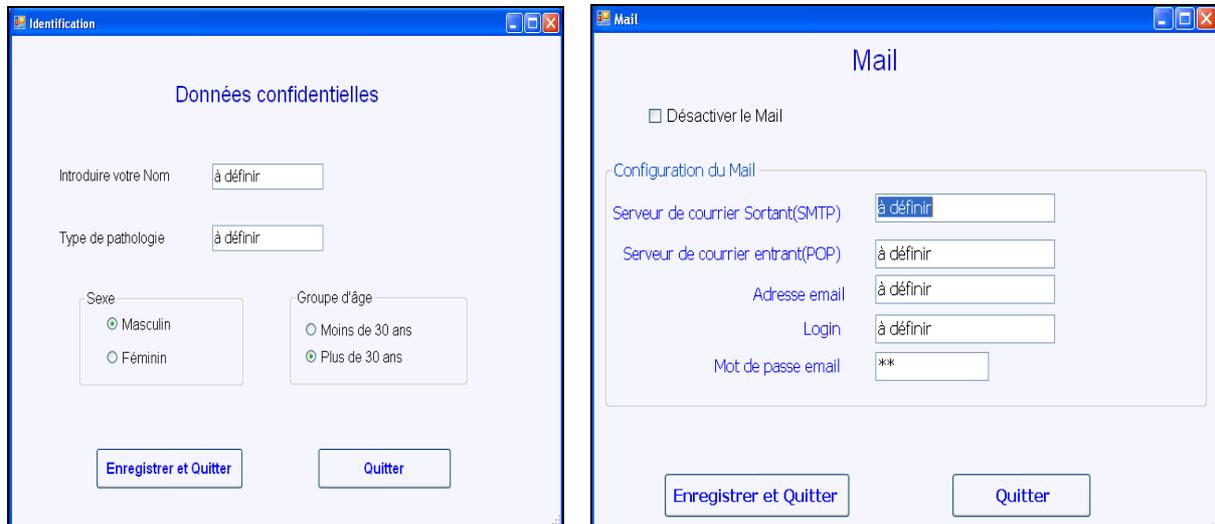


Figure II.13 : Configuration des données confidentielles et du mail

II.2.8. Conclusion

Nous avons présenté dans cette partie notre outil de travail le système EDiTH. La conception de ce logiciel poursuit deux objectifs :

- Le premier objectif, pratique, consiste à fournir aux utilisateurs potentiels un outil de communication et de contrôle d'environnement évolutif et intégrant les conclusions les plus récentes de nos expérimentations et de nos recherches. Il est téléchargeable sur le site du laboratoire, un espace de discussion étant prévu ;
- Le second objectif est de disposer d'un outil permettant une évaluation « écologique » de nos résultats de recherche.

II.3. Acquisition et traitement des données

La méthode de travail que nous avons adoptée pour recueillir des informations sur l'utilisation d'EDiTH est celle des « fichiers logs ». En informatique, le concept « d'historique des événements » ou de « logging » désigne l'enregistrement séquentiel dans un fichier ou une base de données de tous les événements affectant un processus particulier. Le fichier journal, ou plus simplement « log », désigne alors un fichier créé par un logiciel, une application, un serveur ou tout autre système informatique qui permet de garder une trace des requêtes qu'il

reçoit et des opérations qu'il effectue. Généralement toutes ces requêtes sont datées et classées par ordre chronologique. Recueillir des données dans ces fichiers permet d'analyser pas à pas l'activité interne du processus et ses interactions avec son environnement. Ces résultats peuvent être utiles pour l'évaluation ergonomique d'un système notamment pour inciter le concepteur à effectuer les modifications nécessaires pour l'améliorer.

II.3.1. Fichiers logs d'EDiTH

Les « fichiers logs » provenant de l'utilisation d'EDiTH sont issus d'expérimentations sur des personnes valides au laboratoire et sur des personnes handicapées à domicile ou en centre de réadaptation. Pour chaque session d'expérimentation, l'ensemble des données recueillies est enregistré dans un fichier en respectant un formalisme XML (Figure II.14). Dans ces fichiers nous enregistrons toutes les données issues de l'utilisation d'EDiTH en général et de son interface « Editeur de textes » plus particulièrement. L'analyse de ces fichiers permet d'une part de repérer les erreurs survenant dans l'utilisation de l'interface, d'autre part de mesurer les temps de réaction de l'utilisateur.

Le défilement est visualisé sur l'interface par un changement de couleur entre les différents items. Nous notons T_{scan} , le délai de défilement, c'est-à-dire le temps entre deux changements de couleur successifs de colonnes ou d'items. Le temps d'action T_{act} de l'utilisateur est mesuré entre le moment où un item ou un ensemble d'items est validable, noté par un changement de couleur sur l'interface, et le moment où il est validé par action sur le contacteur. Nous avons $0 < T_{act} < T_{scan}$, c'est-à-dire que pour une action trop tardive le temps réel de réaction de l'utilisateur est $T_{act} + T_{scan}$ avec en général dans cette situation $0 < T_{act} < 100ms$. La seule exception à cette règle concerne la 1^{ère} colonne (touche espace) : un délai supplémentaire ayant été ajouté en début de défilement, nous pouvons mesurer dans certains cas $T_{act} > T_{scan}$.

```
<?xml version="1.0" standalone="no" ?>
<essai15-5-9H14M.xml>
  <TempsAdaptatif TscanInitial="528" Adaptatif="True" DefilColZero="578" NbClickMax="40" NbClickInf="9" NbClickSup="8" TempsMin="100" TauxDecroitre="5"
  TauxCroitre="10" />
  <defilement>
    <BouttonT973843>Appel</BouttonT973843>
    <BouttonT974437>Communiquer</BouttonT974437>
    <Tempactiondownt974671>235</Tempactiondownt974671>
    <AjouterPileT975656>0</AjouterPileT975656>
    <ActionT975656>1</ActionT975656>
    <BouttonT976015>Parler</BouttonT976015>
    <BouttonT976593>Ecrire\Lire</BouttonT976593>
    <Tempactiondownt976828>227</Tempactiondownt976828>
    <AjouterPileT977406>0</AjouterPileT977406>
```

Figure II.14 : Exemple de fichier log en format XML

II.3.2. Analyse des données

Pour analyser les données recueillies, nous avons développé un logiciel qui permet de les représenter sous forme d'une table composée de plusieurs colonnes, chaque colonne correspondant à un champ spécifié du fichier log (Figure II.15), et sous forme graphique (courbes, histogrammes) (Figure II.16).

	A	B	C	D	E	F	G	H	I	J	K
1					Adaptatif:	True	Tscan:	300	TscanCol0:	350	
2											
3											
4		NbClickMax:			40	NbClickSup:	8	NbClickInf:	9	TempMin:	100
5											
6											
7		Phrase			NOTRE OBJECTIFM* EST DE Ré G LERgauchegauchegauche*gauche*Fin IN *TIALEMENT T D						
8											
9											
10	Boutton	TempsAction	Règle0_65	Règle0_65sar	AjouterPile	Action N:	TempsdeDefil	DefilColZero	Nombre d'act	Lettre	NbAction
11	Communiquer	207	208,846154	268,791209	0	1	300	350			
12	Ecrire\Lire	213	208,846154	268,791209	0	2	300	350			
13	Ecrire	133	208,846154	268,791209	0	3	300	350			
14	Col8	0	208,846154	268,791209	1	4	300	350			
15	Lig5	219	208,846154	268,791209	0	5	300	350			
16	Col2	110	208,846154	268,791209	0	6	300	350			
17	Lig4	137	208,846154	268,791209	0	7	300	350		N	
18	Col2	197	208,846154	268,791209	0	8	300	350			
19	Lig1	162	208,846154	268,791209	0	9	300	350		O	
20	Col3	219	208,846154	268,791209	0	10	300	350			
21	Lig1	191	208,846154	268,791209	0	11	300	350		T	
22	Col8	8	208,846154	268,791209	1	12	300	350			
23	Lig5	98	208,846154	268,791209	1	13	300	350			
24	Col4	202	208,846154	268,791209	0	14	300	350			
25	Col1	65	208,846154	268,791209	1	15	300	350		R	
26	Col1	89	208,846154	268,791209	1	16	300	350			
27	Col1	174	208,846154	268,791209	0	17	300	350		E	
28	Col1	184	208,846154	268,791209	0	18	300	350			
29	Lig5	191	208,846154	268,791209	0	19	300	350			
30	Col2	37	208,846154	268,791209	0	20	300	350		Esp	
31	Col2	116	208,846154	268,791209	0	21	300	350			
32	Lig1	152	208,846154	268,791209	0	22	300	350		O	
33	Col3	170	208,846154	268,791209	0	23	300	350			
34	Lig2	173	208,846154	268,791209	0	24	300	350		B	
35	Col3	103	208,846154	268,791209	0	25	300	350			
36	Lig4	173	208,846154	268,791209	0	26	300	350		J	
37	Col1	40	208,846154	268,791209	1	27	300	350			
38	Col4	49	208,846154	268,791209	0	28	300	350		E	
39	Col4	147	208,846154	268,791209	0	29	300	350			
40	Lig1	112	208,846154	268,791209	0	30	300	350		C	

Figure II.15 : Présentation des données sous forme d'un tableau sous Excel

La Figure II.16 montre un exemple de présentation des données sous forme d'histogramme. Nous représentons en abscisses les temps T_{act} regroupés par tranches de 5ms et en ordonnées le nombre d'occurrences de ces valeurs. Sur la Figure II.16 par exemple, l'utilisateur a réagi deux fois durant l'expérimentation avec un T_{act} de 200ms et 4 fois avec un T_{act} de 300ms.

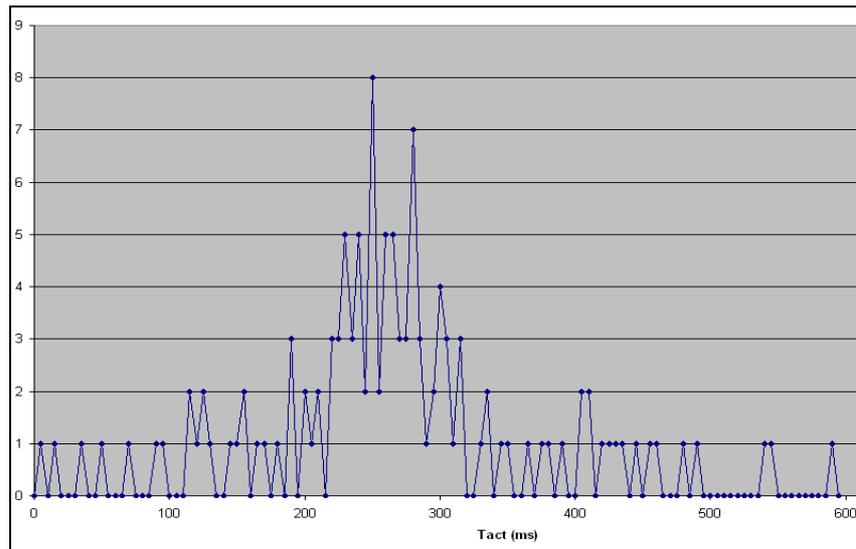


Figure II.16 : Histogramme des temps d'action d'un utilisateur pour un délai $T_{scan} = 600ms$

II.4. Modélisation de l'interaction humain-machine

Dans un communicateur à défilement en général et dans l'éditeur de texte du système EDiTH en particulier, la tâche essentielle de l'utilisateur consiste simplement à attendre que le balayage de l'écran atteigne l'item ou la colonne visée, ce qui est indiqué visuellement par un changement de couleur. La validation de la case s'effectue par l'action d'appui sur le contacteur. Il s'agit donc d'une réaction motrice simple à un stimulus visuel à l'écran. Le modèle du « processeur humain modèle » (MHP) décrit en I.4.2 semble alors bien adapté pour analyser l'interaction humain-machine dans ce contexte de travail. Nous allons l'utiliser pour identifier les facteurs humains qui influencent la performance du sujet.

D'après ce modèle la réaction motrice au stimulus devrait nécessiter un temps de perception, un temps de cognition et un temps moteur : $T_{act} = T_p + T_c + T_m$. Cette règle admet une exception en défilement colonne/ligne : pour valider la 1^{ère} ligne de chaque colonne nous supposons que la deuxième validation T_{act} correspond seulement à un temps moteur T_m comme avec un « double clic ». Pour valider expérimentalement cette règle nous mesurons en premier lieu les trois temps de cycle élémentaires T_m , T_p , et T_c pour chaque personne par des expériences simples décrite ci-dessous. Ensuite, nous demandons à chaque personne d'écrire une phrase en utilisant EDiTH et nous extrayons la valeur moyenne du temps d'action réalisé par le sujet. Enfin nous comparons cette valeur aux résultats obtenus par la mesure directe des temps de cycle élémentaires T_p , T_c et T_m .

Dans ce qui suit nous allons présenter les résultats obtenus dans cette série d'expérimentations.

II.4.1. Estimation des paramètres du modèle MHP

Pour estimer les paramètres T_m , T_p et T_c , nous nous sommes basés sur les travaux réalisés par Keates à l'université de Cambridge [KEAT00] [KEAT98] inspirés par les expérimentations décrites dans [CARD83]. Nous allons mesurer les trois temps de cycle élémentaires T_m , T_p , et T_c par des expérimentations indépendantes du système EDITH. Dans cet objectif nous avons développé un logiciel que nous allons décrire ci-dessous. Nous exposerons ensuite nos résultats obtenus sur un panel de 4 personnes valides.

II.4.1.1. Méthode

a) Estimation du temps de perception

L'expérience consiste à fixer le regard sur un curseur rouge en mouvement circulaire sur un anneau de couleur bleue (Figure II.17). Le mouvement du curseur est échantillonné avec un délai initial de balayage fixé par exemple à 200ms, délai que l'on diminue graduellement. La tâche du sujet consiste à cliquer sur la roue lorsque le mouvement du curseur lui apparaît continu et non plus haché. Dès que le sujet actionne le capteur, le délai de balayage actuel sera enregistré comme une estimation du temps de perception du sujet. Le délai initial ainsi que le taux de diminution peuvent varier selon la personne ; ces deux valeurs sont fixées dans la partie configuration.

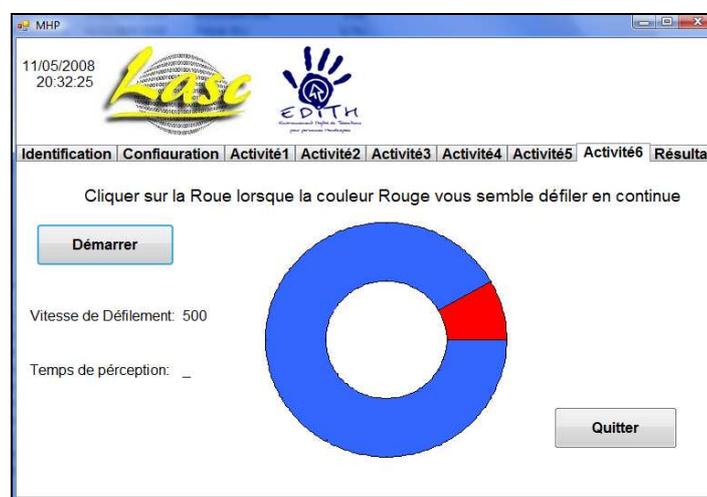


Figure II.17 : Expérience pour mesurer le temps de perception T_p

b) Estimation du temps de cognition

Le temps de cognition T_c peut être évalué en comparant les temps de réponse à deux différents types de stimuli :

- La première expérience consiste à demander à l'opérateur d'actionner le contacteur dès qu'un rectangle rouge apparaît à l'écran. D'après le modèle MHP, le temps de réaction correspond à un cycle de perception pour percevoir le stimulus, un cycle cognitif pour l'associer à l'action motrice et un cycle moteur pour réaliser l'action : $T_1 = T_p + T_c + T_m$;
- Dans la deuxième expérience le rectangle peut être bleu ou rouge de façon aléatoire. Le sujet ne doit cliquer que si la couleur rouge apparaît. Par rapport à l'expérience précédente un deuxième temps de cycle de cognition est nécessaire pour reconnaître la couleur. Le temps de réaction est alors : $T_2 = T_p + 2T_c + T_m$.

La différence $T_2 - T_1$ est donc une estimation du temps de cycle élémentaire de cognition. Dans les expériences réalisées, l'apparition du stimulus se fait soit d'une manière aléatoire (Test1), soit à des intervalles de temps fixes (Test2). Nous avons prévu ces deux types d'expérimentations pour tester une éventuelle influence de la répétitivité du stimulus sur la valeur du temps cognitif. Le Test2 reflète cependant mieux le contexte d'un communicateur à balayage.

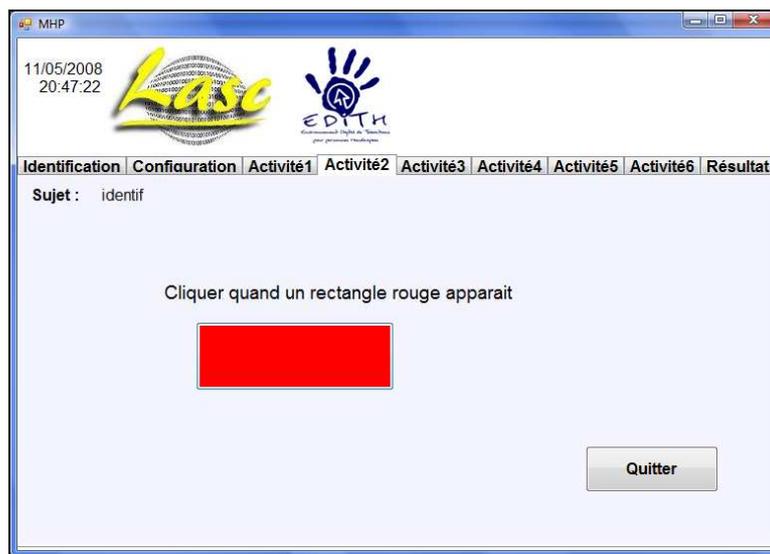


Figure II.18 : Expérience pour mesurer le temps de cognition T_c

c) Estimation du temps moteur

La tâche de l'opérateur consiste à appuyer rapidement sur le contacteur pour valider le bouton OK qui est affiché sur l'écran (Figure II.19). Il s'efface alors puis réapparaît. Cette séquence est répétée un grand nombre de fois. D'après le modèle MHP ces mouvements répétitifs ne nécessitent ni temps de perception, ni temps de cognition. Le temps de réaction pour chaque séquence comprend donc uniquement un temps de cycle moteur pour l'appui sur le contacteur et d'un autre pour son relâchement soit $2T_m$. On obtient alors une estimation de T_m en faisant la moyenne des temps de réaction correspondant à chaque séquence.

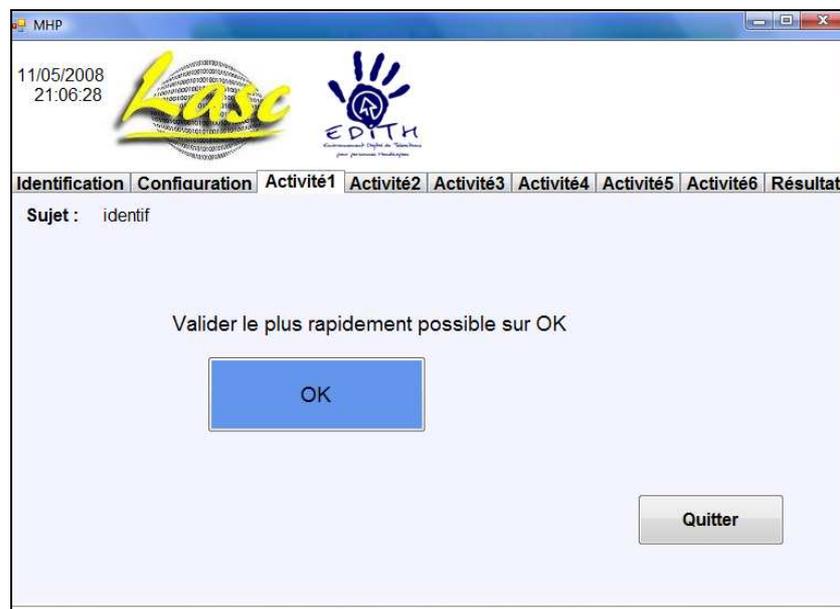


Figure II.19: Expérience pour mesurer le temps moteur T_m

d) Interface de configuration

L'interface de configuration (Figure II.20) permet de choisir les différents paramètres de l'interface de calcul des temps de cycle T_p , T_c et T_m : le nombre d'actions pour le temps moteur, le délai de défilement pour le temps de cognition,

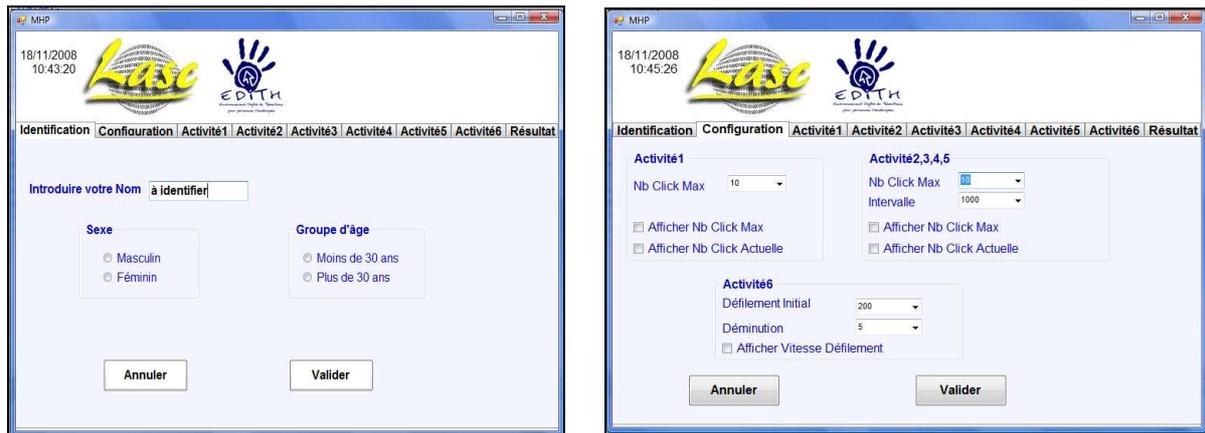


Figure II.20 : Interface d'identification et de configuration

II.4.1.2. Protocole expérimental

Nous présentons dans cette partie les résultats d'une expérimentation que nous avons menée sur 4 sujets valides (2 femmes, 2 hommes). Nous avons répété les tests chaque jour et à des horaires différents pendant 3 jours sur chaque personne. Nous avons fixé le nombre d'appuis pour chaque expérience à 15. Le sujet doit être concentré pour réaliser la suite de 15 appuis sans erreur, dans le cas contraire il recommence.

Pour calculer le temps de cognition, nous avons effectué deux séries de tests (Test1 et Test2). Pendant le Test1, le stimulus change de couleur à des délais variables entre 1000ms et 3000ms. Pour le Test2, le stimulus change de couleur à un intervalle de temps fixe égal à 1500ms. Chaque test comprend deux expériences comme nous l'avons indiqué plus haut (cf. II.4.1.1) : Expe1 où l'icône à valider est toujours rouge et Expe2 où l'icône peut être bleue ou rouge.

II.4.1.3. Résultats

a) Temps de Cognition

Pour chaque série de tests, nous avons trouvé des valeurs aberrantes non significatives soit trop grandes soit trop petites (valeurs écrites en gras Tableau II.1) dues à une mauvaise concentration ou de la fatigue du sujet par exemple. Ce phénomène nous l'avons trouvé pour tous les sujets et à n'importe quel moment de la journée. Pour enlever ces valeurs, nous avons procédé comme suit : pour chaque série d'expériences, nous avons calculé la moyenne « Moy » et l'écart type « Ect », puis nous avons décidé de garder les valeurs qui appartiennent

à l'intervalle [Moyenne-Ecart type, Moyenne+ Ecart type]. Les résultats détaillés de ces expériences sont présentés Tableau II.1.

On peut noter que dans la plupart des cas le temps de cognition mesuré avec un délai de défilement fixe est plus grand que celui trouvé lorsque ce délai est variable. Il semblerait que le fait d'avoir un délai d'apparition du stimulus aléatoire augmente la concentration du sujet et diminue son temps de cognition. Pour calculer le temps de cognition de chaque personne, nous avons fait la moyenne des différentes valeurs obtenues sur les 3 jours, les valeurs aberrantes ayant été supprimées (Tableau II.2).

		Test1				Test2			
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2	
1	1100	269	1500	339	1	1500	316	349	
2	1200	290	2500	334	2	1500	249	400	
3	1200	268	1200	94	3	1500	231	295	
4	2200	271	1300	312	4	1500	268	382	
5	1700	265	2200	336	5	1500	299	462	
6	2000	266	2200	378	6	1500	345	392	
7	2900	265	1700	31	7	1500	304	341	
8	2500	293	2200	303	8	1500	343	413	
9	2300	271	1900	168	9	1500	303	302	
10	2600	254	1000	295	10	1500	279	309	
11	1900	244	2400	355	11	1500	276	323	
12	1400	270	1500	303	12	1500	250	322	
13	2800	283	2400	265	13	1500	303	280	
14	2000	260	1700	331	14	1500	295	342	
15	1700	334	2100	359	15	1500	263	350	
	Moy	275		280		Moy	288	351	
	Ect	21		102		Ect	33	50	
	Moy+Ect	296		382		Moy+Ect	321	401	
	Moy-Ect	254		178		Moy-Ect	255	300	
	Moy1	273				Moy1	290		
	Moy2	326				Moy2	346		
	T_c	53				T_c	56		

Sujet 1 : 1^{er} jour

	Test1					Test2		
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2
1	1400	246	1300	155	1	1500	274	360
2	2400	271	1400	373	2	1500	260	296
3	2700	298	1500	368	3	1500	286	429
4	2700	301	2300	330	4	1500	261	602
5	2500	279	2100	333	5	1500	240	316
6	1200	248	2200	300	6	1500	602	341
7	2100	339	1300	354	7	1500	252	366
8	1500	233	1900	329	8	1500	297	303
9	1800	332	2300	312	9	1500	266	300
10	1400	269	1500	259	10	1500	292	323
11	2500	284	1000	303	11	1500	262	313
12	2600	270	2000	317	12	1500	236	316
13	2800	249	1400	303	13	1500	264	358
14	2700	246	2700	363	14	1500	259	442
15	2900	249	2700	100	15	1500	270	372
	Moy	274		299		Moy	281	362
	Ect	31		76		Ect	63	80
	Moy+Ect	306		376		Moy+Ect	344	442
	Moy-Ect	242		223		Moy-Ect	218	283
	Moy1	268				Moy1	265	
	Moy2	326				Moy2	345	
	T_e	59				T_e	80	

Sujet1 : 2^{ème} jour

	Test1					Test2		
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2
1	2300	316	1600	292	1	1500	367	378
2	1200	268	1900	364	2	1500	266	359
3	1300	291	1300	349	3	1500	290	363
4	1100	266	1900	144	4	1500	271	335
5	1100	321	1900	349	5	1500	251	365
6	2600	269	1700	334	6	1500	259	283
7	1100	261	2700	308	7	1500	287	290
8	2300	298	1700	371	8	1500	243	330
9	2700	259	2500	273	9	1500	234	360
10	2500	325	2200	262	10	1500	251	357
11	1400	275	1600	308	11	1500	273	319
12	2700	332	1100	382	12	1500	272	338
13	2200	325	2300	984	13	1500	269	355
14	2500	284	1400	566	14	1500	450	360
15	2100	262	1200	287	15	1500	252	358
	Moy	289		372		Moy	282	343
	Ect	28		191		Ect	56	28
	Moy+Ect	317		562		Moy+Ect	338	371
	Moy-Ect	262		181		Moy-Ect	227	316
	Moy1	279				Moy1	263	348
	Moy2	342				Moy2	348	
	T_e	63				T_e	85	

Sujet1 : 3^{ème} jour

		Test1			Test2			
	T(ms)	Expe1	T(ms)	Expe 2		T(ms)	Expe1	Expe2
1	1700	337	1400	325	1	1500	286	343
2	1400	290	2100	314	2	1500	300	317
3	1100	301	2300	443	3	1500	287	308
4	1400	233	1500	356	4	1500	350	344
5	1600	259	2600	327	5	1500	291	322
6	2700	221	1300	90	6	1500	298	303
7	1100	467	2600	258	7	1500	303	312
8	2500	233	1700	285	8	1500	325	373
9	1000	265	1200	274	9	1500	316	338
10	2200	217	1100	360	10	1500	264	409
11	2100	249	2500	223	11	1500	290	410
12	2000	192	2400	283	12	1500	253	333
13	1400	237	2900	343	13	1500	245	276
14	2000	208	1100	287	14	1500	318	327
15	2700	251	1600	322	15	1500	264	343
	Moy	264		299		Moy	293	337
	Ect	68		78		Ect	28	37
	Moy+Ect	332		377		Moy+Ect	321	374
	Moy-Ect	196		222		Moy-Ect	264	300
	Moy1	247				Moy1	292	
	Moy2	304				Moy2	330	
	T_e	57				T_e	38	

Sujet 2 : 1^{er} jour

		Test1			Test2			
	T(ms)	Expe1		Expe2		T(ms)	Expe1	Expe2
1	2400	519	1700	297	1	1500	433	340
2	1800	250	1900	325	2	1500	245	335
3	2200	223	2800	435	3	1500	242	354
4	2900	235	2500	308	4	1500	223	354
5	2300	315	1100	281	5	1500	204	299
6	1100	247	2500	333	6	1500	210	325
7	1900	249	1200	402	7	1500	193	375
8	1200	244	2400	441	8	1500	280	343
9	2900	258	1600	74	9	1500	228	316
10	2400	234	1100	308	10	1500	253	325
11	2500	246	1600	168	11	1500	182	319
12	1000	306	2300	333	12	1500	213	303
13	2400	270	1500	377	13	1500	192	318
14	1500	319	1500	349	14	1500	190	302
15	2400	267	2400	337	15	1500	204	284
	Moy	279		318		Moy	233	326
	Ect	73		95		Ect	62	24
	Moy+Ect	352		412		Moy+Ect	294	351
	Moy-Ect	206		223		Moy-Ect	171	302
	Moy1	262				Moy1	218	
	Moy2	332				Moy2	328	
	T_e	70				T_e	109	

Sujet 2 : 2^{ème} jour

	Test1					Test2			
	T(ms)	Expe1		Expe2		T(ms)	Expe1	Expe2	
1	2700	293	2200	513	1	1500	748	366	
2	2400	249	2100	495	2	1500	285	421	
3	1800	226	1200	156	3	1500	248	407	
4	2000	407	1600	288	4	1500	212	378	
5	1100	238	1900	407	5	1500	273	418	
6	2000	232	2600	309	6	1500	271	346	
7	1700	226	1000	264	7	1500	228	454	
8	2300	215	2100	32	8	1500	247	343	
9	2000	252	2100	180	9	1500	270	346	
10	2000	407	2200	400	10	1500	268	407	
11	2000	284	2700	473	11	1500	276	352	
12	2900	265	1200	238	12	1500	264	359	
13	1100	232	1600	470	13	1500	249	283	
14	2800	317	1700	122	14	1500	304	294	
15	1000	372	2900	310	15	1500	241	469	
	Moy	281		310		Moy	292	376	
	Ect	66		148		Ect	128	53	
	Moy+Ect	347		458		Moy+Ect	420	429	
	Moy-Ect	215		163		Moy-Ect	164	323	
	Moy1	252				Moy1	260		
	Moy2	319				Moy2	377		
	T_e	66				T_e	117		

Sujet 2 : 3^{ème} jour

	Test1					Test2			
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2	
1	1100	405	2700	372	1	1500	546	256	
2	1300	368	2800	319	2	1500	340	349	
3	1400	309	1100	377	3	1500	292	416	
4	1100	369	2000	324	4	1500	306	378	
5	2100	311	1600	381	5	1500	265	466	
6	1200	446	1000	449	6	1500	286	361	
7	2900	280	1000	142	7	1500	283	416	
8	1000	353	1300	525	8	1500	281	415	
9	1400	253	2300	339	9	1500	238	520	
10	1600	244	1500	466	10	1500	219	439	
11	1000	309	2500	389	11	1500	247	408	
12	1300	246	1600	405	12	1500	268	385	
13	2800	281	2100	151	13	1500	202	426	
14	1900	272	2800	144	14	1500	238	340	
15	1600	256	2900	144	15	1500	250	452	
	Moy	313		328		Moy	284	402	
	Ect	62		126		Ect	80	62	
	Moy-Ect	251		202		Moy+Ect	365	464	
	Moy+Ect	375		455		Moy-Ect	204	340	
	Moy1	306				Moy1	265		
	Moy2	373				Moy2	388		
	T_e	67				T_e	122		

Sujet 3 : 1^{er} jour

	Test1					Test2		
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2
1	2500	475	1900	325	1	1500	462	987
2	1100	327	2000	385	2	1500	339	472
3	1500	284	2900	598	3	1500	304	389
4	2600	338	1100	386	4	1500	281	344
5	1500	421	1800	211	5	1500	278	410
6	2500	241	2600	544	6	1500	336	429
7	2500	226	2100	360	7	1500	281	344
8	2700	208	1400	341	8	1500	246	400
9	1800	259	2100	448	9	1500	261	443
10	1600	245	1800	144	10	1500	280	467
11	2300	303	2800	500	11	1500	255	417
12	1100	225	1200	407	12	1500	297	400
13	2200	365	1400	399	13	1500	271	504
14	2100	326	2100	538	14	1500	353	387
15	1500	248	2900	421	15	1500	242	359
	Moy	299		400		Moy	299	450
	Ect	77		121		Ect	56	155
	Moy+Ect	376		521		Moy+Ect	355	606
	Moy-Ect	222		280		Moy-Ect	243	295
	Moy1	287				Moy	287	412
	Moy2	397						
	T_e	110				T_e	124	

Sujet 3 : 2^{ème} jour

	Test1					Test2		
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2
1	2300	464	1500	395	1	1500	371	601
2	2800	317	2600	144	2	1500	468	458
3	2000	231	1400	682	3	1500	463	561
4	2700	440	2900	195	4	1500	491	524
5	2200	261	2900	620	5	1500	539	475
6	2100	281	1700	89	6	1500	535	535
7	2100	455	1400	503	7	1500	615	389
8	1600	228	2500	427	8	1500	465	522
9	2000	278	1000	548	9	1500	357	523
10	2200	382	1700	553	10	1500	366	520
11	2800	418	2300	580	11	1500	411	471
12	1500	320	2800	584	12	1500	391	485
13	2100	319	2300	467	13	1500	380	448
14	2700	282	2900	73	14	1500	286	460
15	2800	578	1400	381	15	1500	329	425
	Moy	350		416		Moy	431	493
	Ect	101		200		Ect	89,76	55
	Moy+Ect	452		617		Moy+Ect	521	548
	Moy-Ect	249		215		Moy-Ect	341	438
	Moy1	333				Moy1	408	
	Moy2	477				Moy2	493	
	T_e	144				T_e	84	

Sujet 3 : 3^{ème} jour

		Test1					Test2		
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2	
1	2900	211	2100	438	1	1500	1374	384	
2	2800	294	2400	381	2	1500	544	494	
3	1200	244	1700	460	3	1500	258	459	
4	1100	298	1400	398	4	1500	479	391	
5	1900	316	1300	346	5	1500	294	339	
6	1900	244	2300	382	6	1500	269	363	
7	1700	285	2700	51	7	1500	252	410	
8	2200	276	1700	389	8	1500	243	421	
9	2500	223	1600	321	9	1500	301	433	
10	2600	293	2100	536	10	1500	417	596	
11	1000	304	1300	98	11	1500	254	374	
12	1300	304	1900	292	12	1500	324	513	
13	2700	282	1600	377	13	1500	295	474	
14	1300	300	1400	441	14	1500	360	377	
15	2500	283	2000	351	15	1500	458	423	
	Moy	277		351		Moy	408	430	
	Ect	32		127		Ect	283	68	
	Moy+Ect	309		478		Moy+Ect	691	498	
	Moy-Ect	245		224		Moy-Ect	125	362	
	Moy1	284				Moy1	339	416	
	Moy2	381				Moy2	417		
	T_e	97				T_e	78		

Sujet 4 : 1^{er} jour

		Test1					Test2		
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2	
1	2400	514	1400	325	1	1500	713	362	
2	2600	644	2600	387	2	1500	329	440	
3	1700	453	1600	248	3	1500	288	423	
4	1100	267	2500	368	4	1500	270	456	
5	1500	297	2300	413	5	1500	271	411	
6	2700	413	1000	510	6	1500	294	456	
7	1500	279	1100	414	7	1500	324	397	
8	1400	459	2600	444	8	1500	307	393	
9	2700	332	1300	554	9	1500	325	481	
10	2400	386	2800	314	10	1500	344	419	
11	2200	503	1800	224	11	1500	372	470	
12	2000	346	2900	295	12	1500	281	385	
13	1200	298	2800	444	13	1500	334	418	
14	2500	360	1700	382	14	1500	295	714	
15	1700	362	2600	513	15	1500	266	357	
	Moy	394		389		Moy	334	439	
	Ect	105		97		Ect	109	85	
	Moy+Ect	499		486		Moy+Ect	443	523	
	Moy-Ect	289		292		Moy-Ect	225	354	
	Moy1	371				Moy1	307		
	Moy2	388				Moy2	419		
	T_e	17,28				T_e	112		

Sujet 4 : 2^{ème} jour

Test1					Test2			
	T(ms)	Expe1	T(ms)	Expe2		T(ms)	Expe1	Expe2
1	1600	444	2800	0	1	1500	1454	323
2	2600	339	1200	343	2	1500	538	376
3	1000	498	1100	357	3	1500	324	403
4	2900	274	1400	332	4	1500	309	411
5	2700	318	1400	366	5	1500	273	354
6	2700	340	2800	422	6	1500	295	438
7	2700	281	1400	199	7	1500	329	479
8	2800	372	1600	317	8	1500	310	497
9	2300	325	2200	488	9	1500	412	400
10	2000	267	2600	534	10	1500	313	350
11	1900	285	1200	376	11	1500	284	314
12	1600	323	2100	921	12	1500	411	415
13	1600	318	2500	523	13	1500	314	399
14	2400	409	2500	132	14	1500	318	423
15	1400	338	2000	343	15	1500	309	425
	Moy	342		376		Moy	412	400
	Ect	65		206		Ect	295	51
	Moy+Ect	407,08		583		Moy+Ect	708	451
	Moy-Ect	277		170		Moy-Ect	116	348
	Moy1	326				Moy1	338	
	Moy2	383				Moy2	387	
	T _c	56				T _c	48	

Sujet 4 : 3^{ème} jour

Tableau II.1 : Résultats pour la mesure de T_c (panel de 4 sujets valides)

Sujet	Jour1 : T _c (ms)		Jour 2 : T _c (ms)		Jour 3 : T _c (ms)		Moyenne T _c (ms)	
	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2
1	53	56	59	80	63	85	58	74
2	57	38	70	109	66	117	64	88
3	67	122	110	124	144	84	107	110
4	97	77	17	112	56	48	76	79

Tableau II.2 : Tableau récapitulatif des T_c pour les quatre sujets

Keates a mesuré dans [KEAT00] un temps de cognition moyen de 110ms pour un panel de 5 personnes handicapés moteurs de déficiences motrices diverses et de 93ms pour un panel de 3 sujets valides. Rappelons que Card [CARD83] situe ce temps pour des personnes valides dans un intervalle [25ms-170ms]. Les valeurs trouvées ici sont donc en conformité avec les résultats cités dans la littérature à l'exception d'une valeur aberrante (jour 2, 4^{ème} sujet).

b) Temps moteur et Temps de perception

La mesure du temps de cycle moteur est effectuée avec un nombre d'appuis fixé à 15. Pour le temps de perception, nous avons fixé la valeur initiale du délai de défilement du curseur à 200ms. Il diminue ensuite par paliers de 10ms. Le Tableau II.3 présente les résultats trouvés pour chaque sujet.

Sujet	T _m (ms)	T _p (ms)
1	80	80
2	54	90
3	70	100
4	87	90

Tableau II.3 : T_m et T_p pour les quatre sujets valides

Keates a mesuré dans [KEAT00] un temps de perception moyen de 100ms pour un panel de 5 personnes handicapés moteurs de déficiences motrices diverses et de 80ms pour un panel de 3 sujets valides. Rappelons que Card [CARD83] évalue ce temps pour des personnes valides dans un intervalle [50ms-200ms]. De même pour le temps moteur les mesures de Keates varient de 110ms à 300ms pour le panel de personnes handicapées et sont de 70ms en moyenne pour le panel de sujets valides. Card estime ce temps pour des personnes valides dans un intervalle [30ms-100ms]. Comme pour le temps de cognition, nos résultats sont donc conformes à ceux évoqués dans la littérature.

II.4.2. Validation de l'hypothèse MHP

II.4.2.1. Méthode

Les résultats que nous allons détailler portent sur le même groupe d'expérimentateurs valides que dans la partie précédente. Après une phase d'apprentissage il est demandé à chacune de ces personnes d'écrire une phrase quelconque nécessitant 120 sélections par le biais de

l'interface « Editeur » du système EDiTH. Le mode de défilement est fixe pour une valeur de T_{scan} égale à 1500ms.

II.4.2.2. Résultats

A partir des données enregistrées dans les « fichiers logs » nous avons extrait les « temps d'action » T_{act} correspondant au délai entre le moment où une case ou une colonne est validable et le moment où elle est sélectionnée. Ces temps sont représentés dans la Figure II.21 sous forme d'histogrammes pour les quatre expérimentateurs. L'histogramme présente en ordonnées le nombre d'occurrences des temps T_{act} regroupés toutes les 5ms.

Sur chaque histogramme apparaît un mode entre 150ms et 300ms environ. Pour interpréter ce mode nous pouvons nous référer aux résultats obtenus par la mesure directe des temps de cycles élémentaires T_p , T_c et T_m (Tableaux II.2 et II.3). La somme de ces trois temps que prévoit la théorie (validation sur l'appui du capteur tout-ou-rien) peut être comparée avec la valeur moyenne de T_{act} (moyenne pondérée des actions réalisées dans la plage où l'on situe visuellement le pic de l'histogramme soit en général [200ms-350ms] Figure III.21). Les valeurs de T_{act} ainsi obtenues sont présentées dans le Tableau II.4. Pour les quatre sujets, le T_{act} trouvé par l'utilisation d'EDiTH est plus grand que le T_{act} trouvé par la décomposition MHP. Le T_{act} trouvé dans le Test2, qui reflète mieux l'utilisation d'un communicateur à balayage, s'approche de T_{act} trouvé par EDiTH avec un écart qui varie de 10ms à 30ms. Ces résultats semblent donc valider l'hypothèse MHP : le temps nécessaire pour valider une case est la somme des 3 temps de perception T_p , de cognition T_c et moteur T_m .

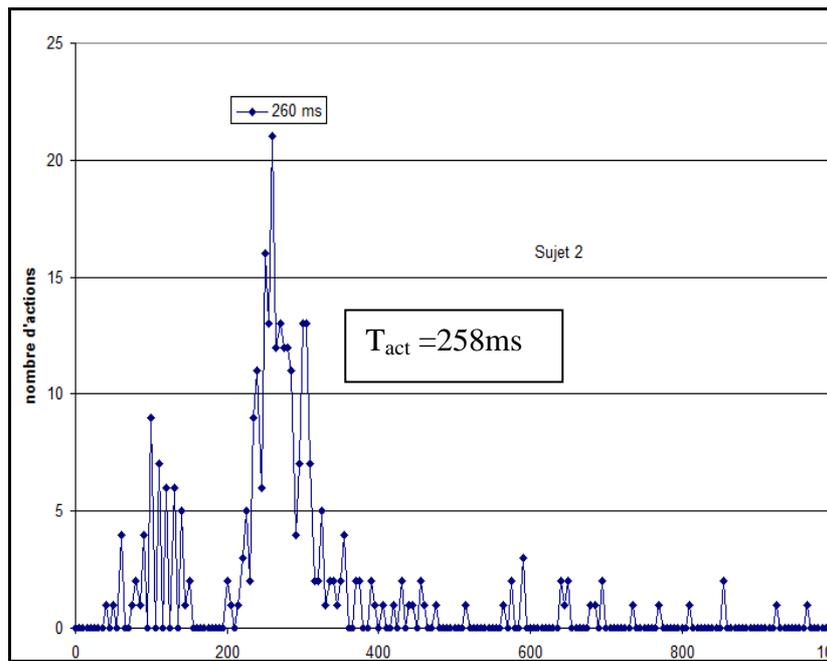
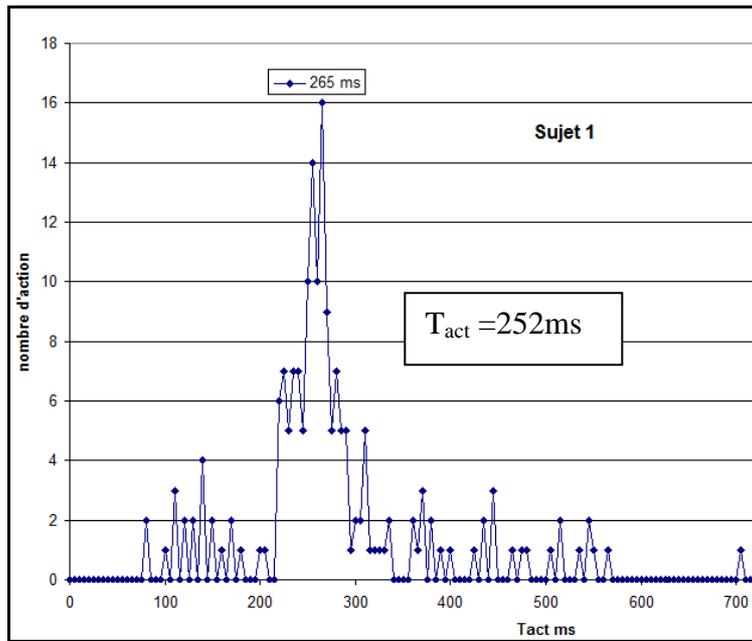


Figure II.21.a : Histogrammes des temps T_{act} de sujet1 et sujet2

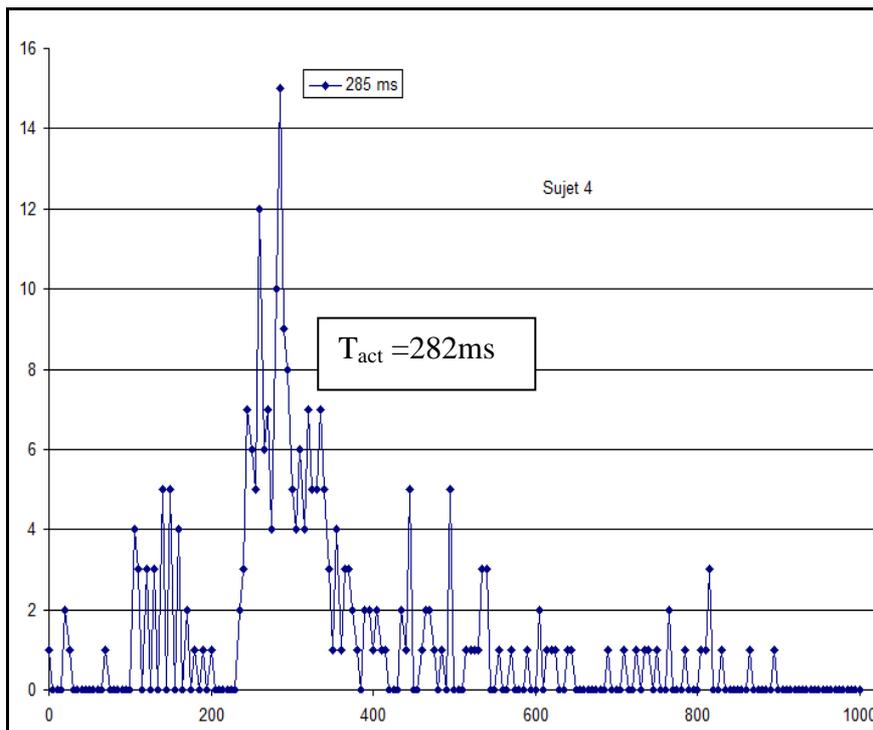
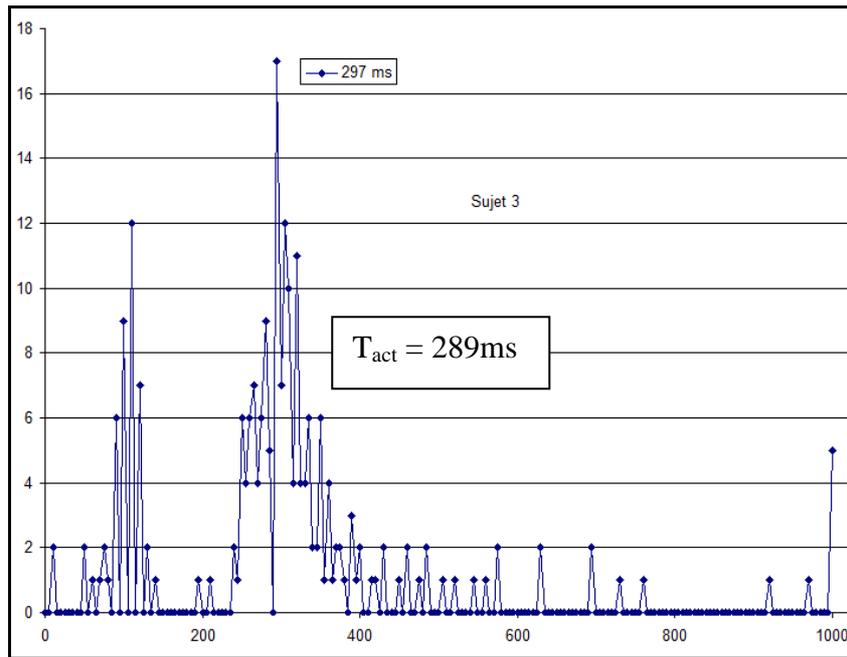


Figure II.21.b : Histogrammes des temps T_{act} de sujet3 et sujet4

Sujet	T_{act} (ms) trouvé par MHP (Temps aléatoire)	T_{act} (ms) trouvé par MHP (Temps fixe)	T_{act} (ms) trouvé par EDiTH avec T_{scan}=1500ms
1	218	233	252
2	209	232	258
3	277	280	289
4	254	256	282

Tableau II.4 : Mesures de T_{act} par le modèle MHP et par EDiTH

Dans chaque histogramme nous trouvons un pic qui correspond approximativement au modèle MHP mais nous trouvons aussi de nombreuses occurrences de T_{act} qui sont inférieures à 100ms. En analysant les fichiers logs, nous constatons que ces valeurs faibles sont révélatrices soit d'erreurs (validation trop tardive de l'item précédent), soit d'un phénomène d'anticipation (la décision de l'action motrice est prise avant que l'item ne soit validable). On note aussi sur les histogrammes des valeurs de T_{act} trop grandes (supérieures à 400ms environ) révélatrices cette fois d'un manque de concentration du sujet.

II.4.2.3. Conclusion

Nous avons commenté dans cette partie quelques résultats expérimentaux obtenus à partir du logiciel MHP. L'analyse des fichiers logs issus de l'utilisation du système EDITH nous a permis de valider l'hypothèse que l'activation d'une case nécessite généralement trois temps de cycle élémentaire T_p+T_c+T_m. Cette validation a cependant été faite pour un délai de défilement T_{scan} fixé à 1500 ms. Il est donc important de vérifier si le mode d'interaction du sujet avec le communicateur ne varie pas avec T_{scan}. C'est ce qui est fait dans ce qui suit.

II.4.3. Modèle de comportement à 3 zones

II.4.3.1. Protocole expérimental

Des expérimentations plus poussées ont dû être menées pour compléter la modélisation décrite ci-dessus [BOUR02a], [BOUR01] et [BOUR04]. Elles sont de deux types : scénario

libre et scénario imposé. En scénario imposé il est demandé de sélectionner la phrase suivante à l'aide de l'éditeur de texte : "pourriez-vous me passer de la pommade sur les épaules ?". Les éventuelles erreurs de sélection doivent être corrigées à l'aide de la touche "DEL" du clavier virtuel, ceci est réalisé pour différents délais de défilement. T_{scan} est relativement long au départ (2s) et nous le diminuons jusqu'à arriver aux limites des possibilités de la personne (1.5s, 1s, 800ms, 600ms, etc.). Un même sujet réalise toute l'expérimentation dans une même séance. Cependant, avant chaque sélection de phrase (nouvelle valeur de T_{scan}), une pause conséquente est prévue pour que la fatigue n'influe pas sur les résultats.

Cette expérimentation a tout d'abord concerné un panel de six personnes valides au laboratoire pour établir un ensemble de données de référence. Elle a ensuite porté sur une dizaine de personnes handicapées : trois personnes tétraplégiques, deux personnes avec une infirmité motrice cérébrale, deux personnes atteinte du syndrome d'enfermement (LIS), une personne traumatisé crânien, une personne atteinte de myopathie et une personne souffrant du syndrome latéral amyotrophique (SLA).

Dans le second type d'expérimentations, l'utilisation du système EDiTH est libre : des personnes atteintes de SLA l'utilisent en tant qu'aide à la communication à domicile et en centre de réadaptation. Il est laissé à ces personnes la possibilité de communiquer en mode confidentiel (fichiers logs non-utilisés) quand elles le souhaitent. Le délai de défilement est cette fois ajusté initialement de façon empirique par l'ergothérapeute en charge de ces personnes.

II.4.3.2. Résultats en scénario imposé

Les résultats de l'analyse des temps d'action issus de l'utilisation d'EDiTH en scénarios imposés sur le panel de personnes valides sont très similaires pour les six personnes (Figure II.22) et les histogrammes des temps d'action T_{act} peuvent être découpés en trois zones :

- Nous constatons tout d'abord systématiquement la présence d'un pic entre 100ms et 400ms environ. Ce pic peut être interprété à l'aide du modèle MHP : $T_{act} = T_p + T_c + T_m \in [100ms-400ms]$ comme vu précédemment ;
- En analysant les « fichiers logs » nous nous apercevons également que le nombre d'occurrences de T_{act} inférieures à 100ms augmente pour des délais de balayage faible. Un

temps T_{act} faible peut être simplement du à un retard dans la sélection de l'icône précédente, donc à une erreur de sélection. Il peut être également, pour des valeurs identiques, révélateur d'un phénomène d'anticipation : l'opérateur, familier du rythme du défilement lumineux, décide d'actionner le capteur avant que l'icône à sélectionner ne soit validable ;

- Nous pouvons raisonner de même pour des temps T_{act} supérieurs à 400ms environ, quand T_{scan} est supérieur à cette valeur. A la lumière de l'analyse des « fichiers logs », ces valeurs correspondent soit à des actions retardées pour diverses raisons comme une perte de concentration ou une icône repérée trop tardivement sur l'écran, soit à des erreurs.

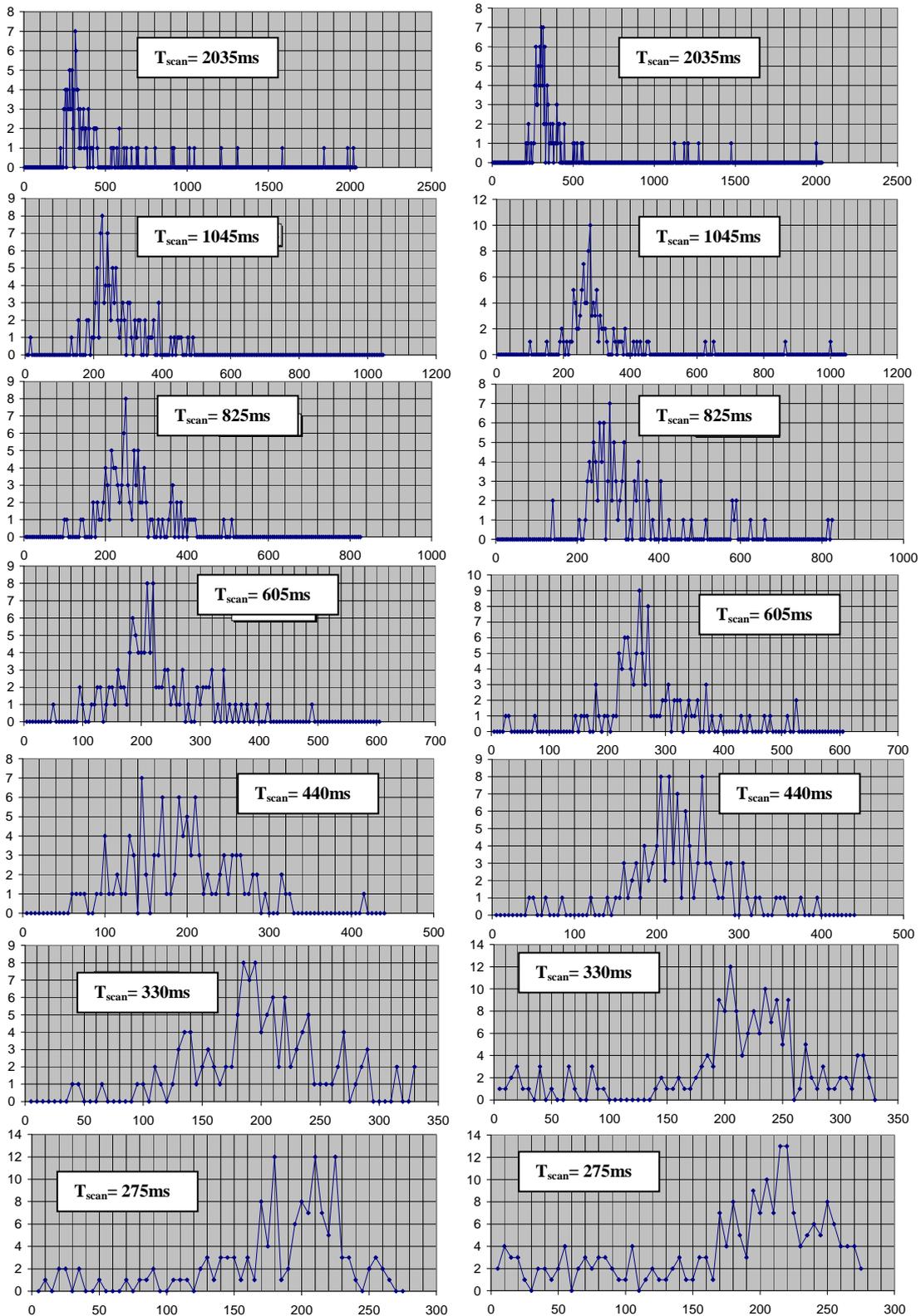


Figure II.22 : Histogrammes de T_{act} pour deux personnes parmi les six expérimentateurs valides en scénario imposé

Nous retrouvons ce « modèle de comportement à trois zones » pour des personnes handicapées contrôlant sans difficulté une aide à la communication par balayage. Pour certaines personnes « expertes », le délai élémentaire de défilement peut être réglé très court. Un exemple est donné en Figure II.23 qui concerne une personne tétraplégique : le délai de défilement T_{scan} étant fixé à 200ms les temps T_{act} sont pour moitié des erreurs, pour moitié des anticipations. Ces erreurs ou anticipations apparaissent d'autant plus fréquemment que T_{scan} est faible.

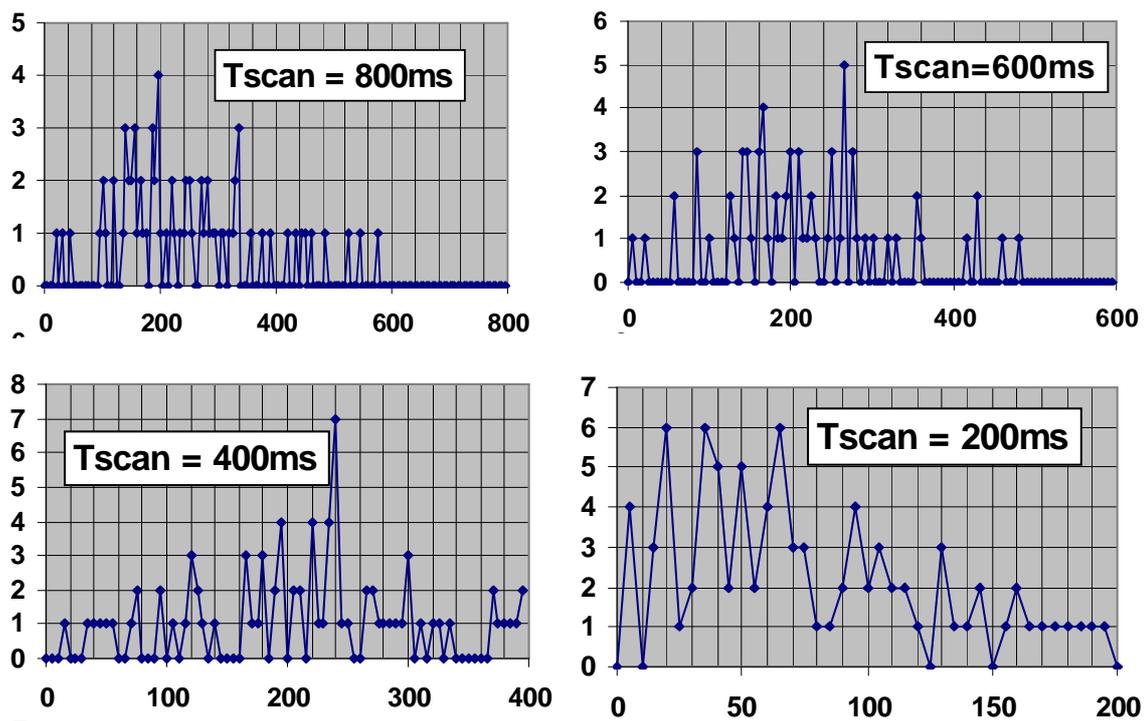


Figure II.23 : Histogramme des temps d'action T_{act} pour une personne tétraplégique pour différentes valeurs de T_{scan} en scénarios imposés

II.4.3.3. Résultats en scénario libre

Les histogrammes représentés Figure II.24 et obtenus en utilisation libre par une personne SLA semblent être en accord avec le « modèle de comportement à trois zones » constaté en scénarios imposés. Leurs examens amènent par ailleurs à faire diverses remarques d'ordre qualitatives :

- Pour cet utilisateur, le temps T_{scan} fixé empiriquement à 1200ms à l'initialisation du système n'était manifestement pas bien adapté. D'autre part la valeur que nous pourrions estimer optimale à la vue des histogrammes a varié avec l'apprentissage de l'utilisateur et

également, si nous analysons plus finement les fichiers logs, avec son état de fatigue ou avec la longueur de la phrase à composer ;

- Nous pouvons noter qu'en mode de communication directe les erreurs de sélection ne sont rectifiées que si cela s'avère indispensable pour la compréhension de la phrase. Il est donc difficile de repérer automatiquement ces erreurs et par voie de conséquence d'utiliser ce critère pour un réglage adaptatif de T_{scan} ;
- Nous constatons que le nombre d'anticipations (T_{scan} inférieur à 100ms) augmente avec l'apprentissage.

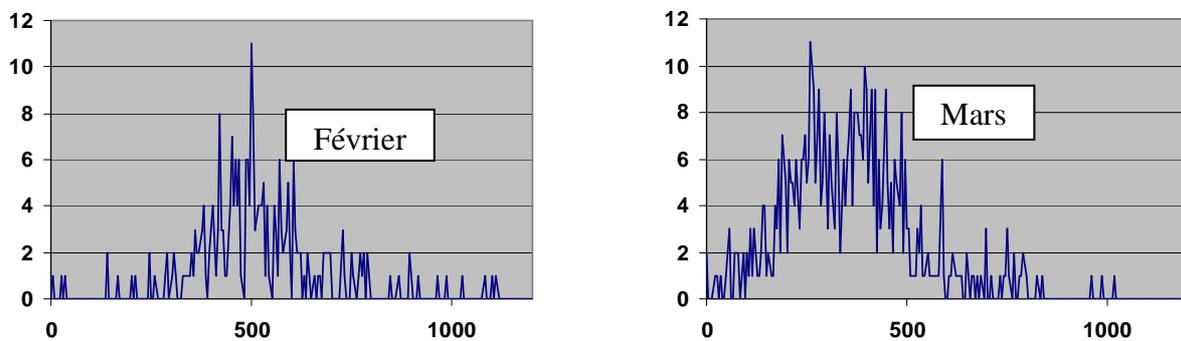


Figure II.24 : Histogrammes des temps d'action T_{act} d'une personne SLA en utilisation libre sur une période d'une journée en février (à gauche) et en mars (à droite)

II.4.3.4. Discussion

Ainsi, l'application du modèle comportemental MHP sur le système EDiTH a permis de dégager trois zones dans l'intervalle du temps de défilement (Figure II.25) :

- Une « zone avance » avec un T_{act} inférieur à 100ms (erreurs ou anticipations) ;
- Une « zone centrale » avec un $T_{act} \in [100\text{ms} - 400\text{ms}]$ conforme au modèle MHP ;
- Une « zone retard » avec un T_{act} supérieur à 400ms.

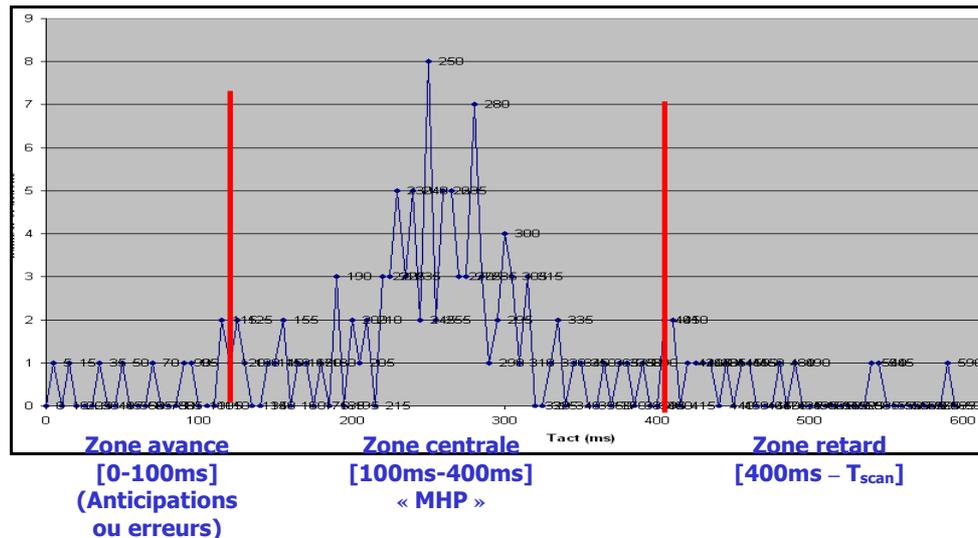


Figure II.25 : Histogramme des temps d'action T_{act} d'un opérateur valide en scénario imposé pour $T_{scan}=600ms$

La «zone avance » semble corrélée avec l'évolution du délai de défilement, elle reflète les difficultés de l'opérateur : le nombre de temps d'actions T_{act} situé dans cette zone augmente fortement quand T_{scan} diminue. En analysant les fichiers logs nous nous apercevons qu'il s'agit à la fois d'une augmentation des erreurs, ce qui était largement prévisible, et du nombre d'anticipations. Ce phénomène d'anticipation apparaît d'autant plus fréquemment que T_{scan} est faible et que le sujet est expérimenté. Nous pouvons alors envisager de fixer un seuil d'occurrences dans cette zone pour en déduire un temps T_{scan} optimal pour une personne et une situation donnée.

Dans la «zone centrale », la réaction de l'opérateur est conforme au modèle MHP. La position de cette zone n'étant pas nettement corrélée avec le temps de défilement elle n'apporte pas d'information particulière pour le réglage automatique de ce temps. Elle peut par contre permettre son réglage initial hors-ligne en fonction de la position du pic sur l'histogramme des temps d'action.

La zone retard, correspondant à T_{act} supérieur à 400ms environ, n'est bien sûr pas présente pour des délais de défilement faibles. Dans cette zone le comportement de l'opérateur n'est pas optimal : sa réaction est trop lente du fait d'un manque de concentration.

Notons d'autre part que certains utilisateurs potentiels d'une aide à la communication ne suivent pas le modèle de comportement précédemment décrit, les temps T_{act} sont beaucoup

plus distribués sur tout l'intervalle de temps possible, ceci pour des raisons motrices (difficultés pour réaliser l'action sur le contacteur pour certaines personnes infirmes moteurs cérébraux par exemple) ou cognitives (difficultés de concentration pour certaines personnes traumatisés crâniens) (Figure II.26).

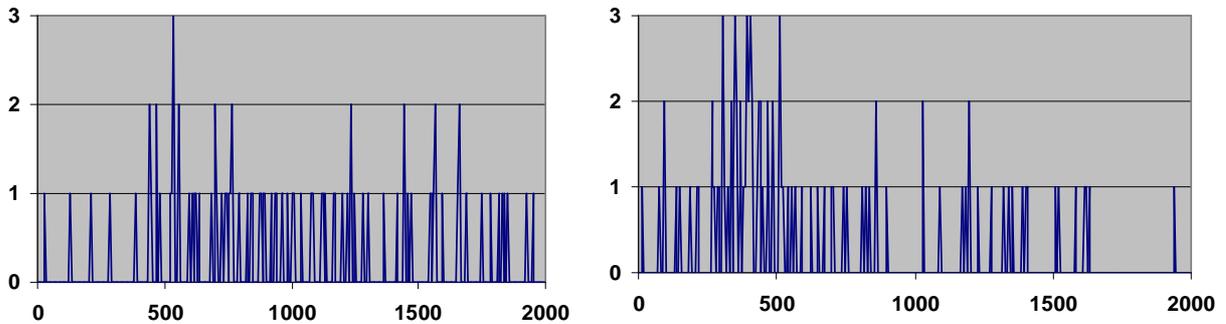


Figure II.26 : Exemples d'histogrammes des temps d'action T_{act} pour une personne infirme moteur cérébral (à gauche) et pour une personne traumatisée crânienn (à droite)

II.4.3.5. Conclusion

Pour interpréter les délais de réaction des utilisateurs du communicateur à défilement nous nous sommes basés sur le modèle du « Processeur Humain Modèle ». Ceci nous a permis de définir un « modèle de comportement à 3 zones » qui s'avère vérifié pour des personnes handicapées n'ayant pas de difficultés motrices ou cognitives majeures pour contrôler leur capteur d'interface TOR.

Plusieurs informations importantes découlent également de cette analyse, notamment la nécessité d'adapter le délai de défilement de façon automatique, sa valeur optimale variant au cours du temps. Pour cela un paramètre semble intéressant à utiliser : le nombre d'occurrences dans la zone « avance » du modèle ($T_{act} < 100ms$) semble bien corrélé à ce délai T_{scan} .

II.5. Conclusion du chapitre2

Dans ce chapitre, nous avons présenté en première partie les systèmes que nous avons développés : EDiTH, le logiciel MHP et le logiciel de traitement des données. Dans la

deuxième partie, nous avons détaillé nos hypothèses, notre méthodologie de travail et les résultats concernant la modélisation de l'interaction entre la personne handicapée et le communicateur à balayage.

La comparaison des temps d'action T_{act} extraits des fichiers logs avec ceux trouvés par le modèle MHP a permis de valider l'hypothèse que le temps nécessaire pour valider une case déjà repérée nécessite généralement un temps $T_p + T_c + T_m$. Le modèle MHP peut notamment ainsi donner une idée sur la valeur initiale optimale du temps de défilement du communicateur.

Une analyse plus fine de nos résultats expérimentaux, toujours à la lumière du modèle du « Processeur Humain Modèle », nous a permis de construire un « modèle de comportement à 3 zones ». Les résultats issus de l'utilisation du système EDiTH en scénarios imposés ou en utilisation libre ont permis de le valider pour des sujets valides et des sujets handicapés n'ayant pas de difficultés cognitives ou motrices pour contrôler leur capteur d'interface. L'analyse de ces résultats montre également la possibilité d'utiliser le nombre d'occurrences des temps d'action T_{act} inférieurs à 100ms pour induire une méthode d'adaptation automatique du délai de défilement.

III. CHAPITRE 3 :

DELAÏ DE DEFILEMENT ADAPTATIF

III.1. Introduction

La vitesse de défilement T_{scan} des interfaces de communication à défilement est généralement définie de façon empirique, et garde sa valeur tout au long de l'utilisation du système. Notre objectif est de régler cette vitesse automatiquement et de l'adapter en cours d'utilisation en fonction de l'état physique et cognitif de la personne. Dans ce chapitre nous allons tout d'abord détailler notre méthode pour régler le temps de défilement ainsi que les différents paramètres qui interviennent dans notre algorithme.

Nous exposerons et discuterons ensuite les résultats obtenus lors de nos expérimentations sur des sujets valides et des sujets handicapés. Une phase de tests sur des sujets valides est en effet indispensable, des expérimentations en scénarios imposés, répétitives et fatigantes, étant très difficiles à mettre en œuvre avec des personnes sévèrement handicapées. Il s'agit d'une part d'affiner ou de régler les paramètres de notre méthode, d'autre part de voir si le temps de défilement s'adapte bien au comportement du sujet.

Notre « algorithme des anticipations/erreurs » sera expérimenté tout d'abord en laboratoire pour régler ses paramètres. Une première phase expérimentale en situation écologique concernant une personne atteinte de SLA permettra ensuite de valider le principe de l'algorithme. Elle sera suivie d'une seconde phase de tests en laboratoire pour comparer ses performances à celles d'autres algorithmes décrits dans la littérature. Il fera enfin l'objet d'évaluations sur de longues périodes en situation de vie courante par des personnes handicapées. Les premiers résultats de ces évaluations concluront ce chapitre.

III.2. Méthodologie

III.2.1. Justification

Pour optimiser la communication il faut bien choisir le délai élémentaire de balayage de chaque utilisateur. Le temps de défilement optimal peut varier au cours du temps en fonction de l'attention, de l'apprentissage ou de la fatigue du sujet. Il est toujours réglable sur les aides à la communication mais ce réglage s'effectue de façon empirique et garde toujours la même valeur. Un réglage trop rapide rend l'utilisation de l'interface pénible et stressante et augmente le taux d'erreurs de sélection. Un réglage trop lent augmente le temps d'attente pour sélectionner un item et diminue donc la rapidité de la communication.

D'après la revue de la littérature du chapitre I, deux types de paramètres peuvent permettre de régler en temps réel le délai de défilement : une évaluation des erreurs de sélection et les temps de réaction de l'utilisateur. Ces deux paramètres interviennent également dans la définition et l'interprétation de la « zone avance » du modèle à trois zones que nous avons défini dans le chapitre II. C'est pourquoi nous nous sommes penchés plus particulièrement sur l'analyse de cette zone pour sélectionner un paramètre permettant de régler automatiquement le temps de défilement T_{scan} .

En faisant varier les valeurs de T_{scan} en scénario imposé, nous nous sommes aperçu en particulier que le nombre d'occurrences de T_{act} inférieures à 100ms augmentait fortement près de la valeur qui subjectivement pour l'utilisateur semble être le délai de défilement optimal. A titre d'exemple, nous présentons dans le Tableau III.1, pour différentes valeurs de T_{scan} , le nombre d'occurrences de T_{act} inférieures à 100ms, ceci pour trois utilisateurs expérimentés des communicateurs à balayage (deux personnes myopathes, une personne tétraplégique) lors de tests en scénarios imposés. Les cases colorées correspondent au T_{scan} jugé optimal par les expérimentateurs.

Nous allons donc développer un algorithme de calcul du délai de défilement basé sur le repérage en temps réel de cette zone. Le nombre N_{100} d'actions T_{act} dans cette zone est révélateur d'un délai T_{scan} trop petit ou trop grand. Dans le premier cas, du fait d'un rythme de défilement trop rapide, l'utilisateur augmente son taux d'erreurs (items sélectionnés trop tard)

et d'anticipations et donc N_{100} . Inversement, si T_{scan} est trop grand, il a moins tendance à anticiper et fait moins d'erreurs, N_{100} diminue donc.

T_{scan}	2s	1s	800ms	600ms	400ms	300ms	250ms	200ms
Exper. A			8	8	11	20	30	53
Exper. B	3	3		7	12	125		
Exper. C			4	8	11	30	58	60

Tableau III.1 : Variations du nombre N_{100} d'occurrences de T_{act} inférieures à 100ms

III.2.2. Algorithme de réglage adaptatif de T_{scan}

Le réglage adaptatif de T_{scan} se fait en ligne en se basant sur les variations du nombre d'occurrences N_{100} des actions réalisées dont les T_{act} sont inférieures à 100ms. La mesure de N_{100} se fait sur les *nb_clic_max* dernières actions. Nous considérons que si N_{100} est faible (peu d'anticipations/erreurs) nous pouvons diminuer T_{scan} et inversement. Deux seuils sont prévus pour ne pas imposer des variations permanentes de T_{scan} .

Nous implémentons alors les deux règles suivantes (« algorithme des anticipations/erreurs ») :

- Règle A : Pour un nombre d'actions sur le contacteur *nb_clic_max* donné, si le nombre d'actions N_{100} dont le T_{act} est inférieur à 100ms est supérieur à *nb_clic_sup* alors T_{scan} va croître à $T_{scan} = \text{taux2} * T_{scan}$.
- Règle B : Pour *nb_clic_max* donné, si N_{100} est inférieur à *nb_clic_inf* alors T_{scan} va décroître à $T_{scan} = \text{taux1} * T_{scan}$.

III.2.3. Réglage des paramètres

Le problème principal des deux règles ci-dessus est de fixer les valeurs des paramètres : *nb_clic_max*, *nb_clic_inf*, *taux1*, *nb_clic_sup* et *taux2*. Nous avons pour cela débuté nos expérimentations par des tests sur un panel de quatre personnes valides en scénario imposé en faisant varier les paramètres des deux règles pour analyser le comportement de chaque sujet. Cette phase est indispensable avant d'entamer des tests en situation écologique. Le sujet doit au préalable passer par une phase d'apprentissage de l'utilisation d'EDITH. On lui demande ensuite d'écrire la phrase suivante sans fautes de frappe : «pourriez vous me passer de la

pommade sur les épaules merci beaucoup» pour différentes valeurs des paramètres. On enregistre toutes les données d'interaction dans les fichiers « logs ».

Le paramètre *nb_clic_max* ne doit être ni trop petit, car T_{scan} va changer trop rapidement et rendre impossible l'acquisition d'automatismes, ni trop grand pour que le sujet n'attende pas trop longtemps avant que T_{scan} change. Nous avons fixé *nb_clic_max* à une valeur de compromis égale à 40 actions soit 20 sélections de caractères. Nous avons également fixé *taux1* à 0,9 et *taux2* à 1,3.

Un exemple de résultats concernant un sujet valide (sujet 4) peu familier de l'interface est présenté Figure III.1 et Figure III.2. Ce graphique et ce tableau permettent l'analyse du nombre d'erreurs et d'anticipations que nous associons à l'avis subjectif des sujets. Dans le tableau nous avons séparé les anticipations, les corrections de saisie (retour chariot) et les erreurs réalisées. La Figure III.1 présente une partie des résultats groupés après chaque 40 actions :

- Dans la « phrase écrite » nous trouvons la phrase telle qu'elle a été écrite par le sujet, erreurs et corrections comprises. Le symbole «<>» représente l'action « effacer » ;
- La colonne « colonne/ligne » analyse l'action réalisée par le sujet : temps d'action, erreur lettre, erreur colonne, anticipation lettre, etc. ;
- La colonne « Anticipations » présente le nombre d'anticipations réalisées après 40 actions ;
- La colonne « Erreurs » présente le nombre d'erreurs réalisées après 40 actions.

Après chaque 40 actions, le système calcule N_{100} le nombre d'actions réalisées avec $T_{act} < 100ms$. Si $N_{100} < 3$ alors T_{scan} va diminuer, si $N_{100} > 8$ alors T_{scan} va augmenter, sinon T_{scan} garde la même valeur. Ainsi, dans l'exemple Figures III.1 et III.2, après 160 actions, le sujet a fait deux erreurs. T_{scan} a alors diminué et pris la valeur de 356ms. Cependant, aux dires du sujet, ce temps est trop rapide et par conséquent il fait plus d'erreurs. Après les 40 clics suivants T_{scan} va augmenter et prend la valeur de 473ms.

		nb clic max = 40			
		nb clic inf = 3	nb clic sup = 8		
phrase écrite:	pouc<ri<riez vous mé<e passéh<<é<er <de la pomade < sur lép<h<<esépo<aules mé<erci <beaucoup				
colonne/ligne	Temps d'action (ms)	Défilement (ms)	Nbre de Clics	Anticipations	Erreurs
Ant col4	63	400	3		
Err col	146	400	39		
a	219	400	40	1 anticipations	4 erreurs
Err col	168	400	1		
col	333	400	39		
r	309	400	40	0 anticipations	6 erreurs
Err col	341	400	1		
Err col	392	400	39		
b	260	400	40	0 anticipations	5 erreurs
col	285	400	1		
e	323	400	39		
Err col	284	356	40	0 anticipations	2 erreurs
col	292	356	1		
Erreur	15	356	39		
Err col	270	476	40	1 anticipations	9 erreurs
Err col	0	476	1		

Figure III.1 : Analyse anticipations/erreurs (sujet 4)

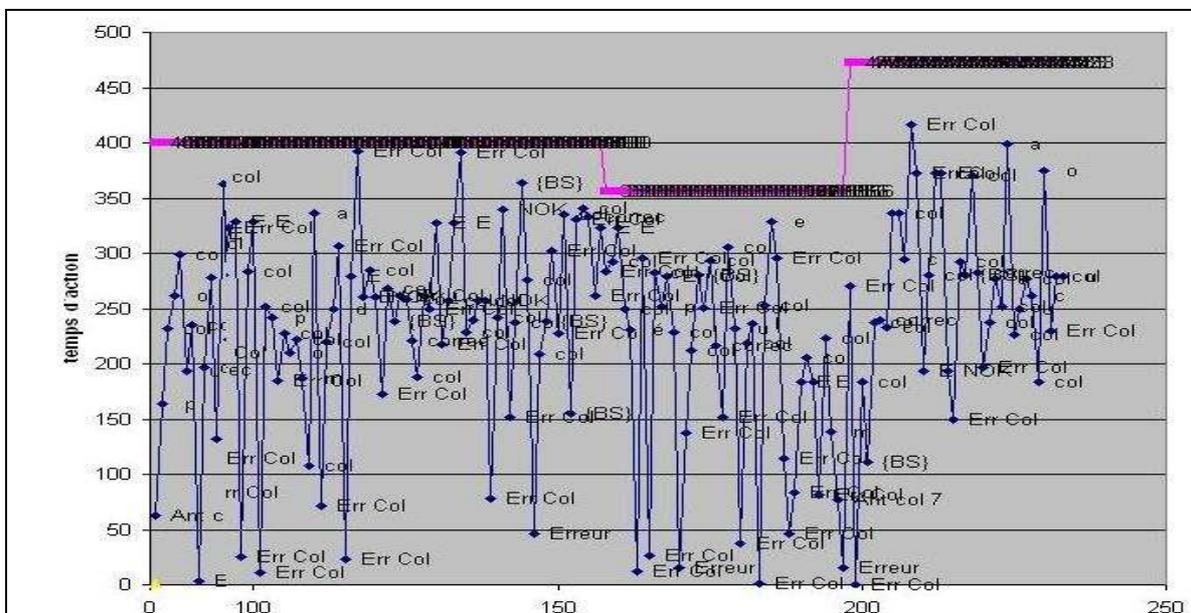


Figure III.2 : Analyse des actions et évolution de T_{scan} (traits horizontaux) du sujet 4 ; en abscisses : chronologie des actions ; en ordonnées : T_{act} en ms

Toujours dans le but de régler les valeurs pour nb_clic_inf et nb_clic_sup , nous avons réalisé le même test avec un sujet qui connaît bien l'interface (sujet 1), ceci avec les mêmes valeurs

(3,8) de nb_clic_inf , nb_clic_sup . La figure III.3 montre les résultats obtenus pour ce sujet. La valeur de T_{scan} n'a pas changé lorsque le sujet a écrit la phrase demandée, alors que le nombre d'anticipations a augmenté après un certain nombre de clics (sur les 40 derniers clics le nombre d'anticipations a brusquement augmenté, et prend la valeur de 7, le défilement a gardé la même valeur).

		nb clic max = 40			
		nb clic inf = 3	nb clic sup = 8		
phrase écrite:	pourriez vous me passer de la pommade sur les épaules				
	ma<erci beag<ucoup				
colonne/ligne	Temps d'action (ms)	Défilement (ms)	Nbre de Clics	Anticipations	Erreurs
Ant col4	94	300	3		
p	125	300	4		
col	160	300	38		
s	145	300	39		
col	137	300	40	4 anticipations	0 erreurs
s	174	300	1		
col	162	300	2		
l	198	300	38		
col	195	300	39		
e	90	300	40	4 anticipations	2 erreurs
col	186	300	1		
s	174	300	2		
col	161	300	38		
a	52	300	39		
col	163	300	40	7 anticipations	0 erreurs
Erreur	10	300	1		

Figure III.3 : Analyse anticipations/erreurs (sujet 1)

Après plusieurs tests, nous pouvons conclure que la valeur initiale de nb_clic_inf doit être au début la plus petite possible : le sujet ne connaît pas bien l'interface et a tendance à faire plus d'erreurs que d'anticipations. Après la phase d'apprentissage, le sujet connaît bien l'interface, mémorise bien son clavier, donc d'autres valeurs pour les deux règles seraient nécessaires pour suivre l'évolution de la connaissance de l'interface par l'utilisateur. Les paramètres nb_clic_inf et nb_clic_sup permettant de calculer T_{scan} doivent notamment dépendre des possibilités physiques de la personne puis évoluer en fonction du niveau d'apprentissage, le nombre d'anticipations étant fonction de ce niveau.

III.2.4. Conclusion

Nous avons présenté dans cette partie, notre algorithme « anticipations/erreurs » pour adapter la vitesse de défilement en temps réel ainsi que les tests réalisés pour régler les paramètres de

notre algorithme. Cette première phase d'expérimentation sur des sujets valides nous a permis de conclure que chaque personne se comporte de manière différente avec l'interface. En particulier les valeurs optimales de notre algorithme dépendent principalement de l'état physique et du niveau d'apprentissage de la personne.

III.3. Résultats comparatifs

III.3.1. Introduction

Les essais précédents nous ont permis de tester le comportement de sujets valides lors de l'utilisation d'EDiTH avec un réglage adaptatif du temps de défilement. Nous allons présenter dans ce qui suit une phase expérimentale qui va comparer notre algorithme avec ceux proposés par Lesher et al. et par Simpson et Koester (cf. I.3.2). Pour ces expérimentations nous avons utilisé trois couples de valeurs pour *nb clic inf*, *nb clic sup* : (3, 8), (15, 20) et (20, 25).

III.3.2. Evolution de T_{scan} pour une personne SLA en situation écologique

L'algorithme des « anticipations/erreurs » a été expérimenté pendant deux mois par une personne atteinte de Sclérose Latérale Amyotrophique. Cette expérimentation a eu lieu en situation « écologique », le sujet utilisant le système EDiTH en tant qu'aide à la communication et au contrôle d'environnement à son domicile et en centre de réadaptation. On a pu noter une évolution de T_{scan} au cours de la journée fonction de l'état physique et cognitif de la personne.

Un exemple d'évolution de T_{scan} sur une journée est présenté dans la Figure III.4. Sur cette figure est également notée l'évolution fictive de T_{scan} calculé selon la « règle des 0.65 » proposé par Simpson et Koester [SIMP06a], [SIMP06b] : on calcule la moyenne des temps d'action sur les 50 dernières sélections et on la divise par le facteur 0.65. Il s'agit d'un calcul a posteriori, l'algorithme « anticipations/erreurs » étant en place et le sujet réagissant donc par rapport à lui. On peut néanmoins en tirer quelques indications. En particulier on note que les valeurs de T_{scan} obtenues par ce calcul sont nettement plus petites que les valeurs réelles obtenues par notre algorithme. Si on les compare aux valeurs de T_{act} correspondantes elles sont manifestement trop faibles et donc génératrices d'erreurs de sélection. Ceci s'explique

par la présence de nombreux T_{act} inférieurs à 100ms révélateurs, comme nous l'avons montré, d'erreurs ou d'anticipations.

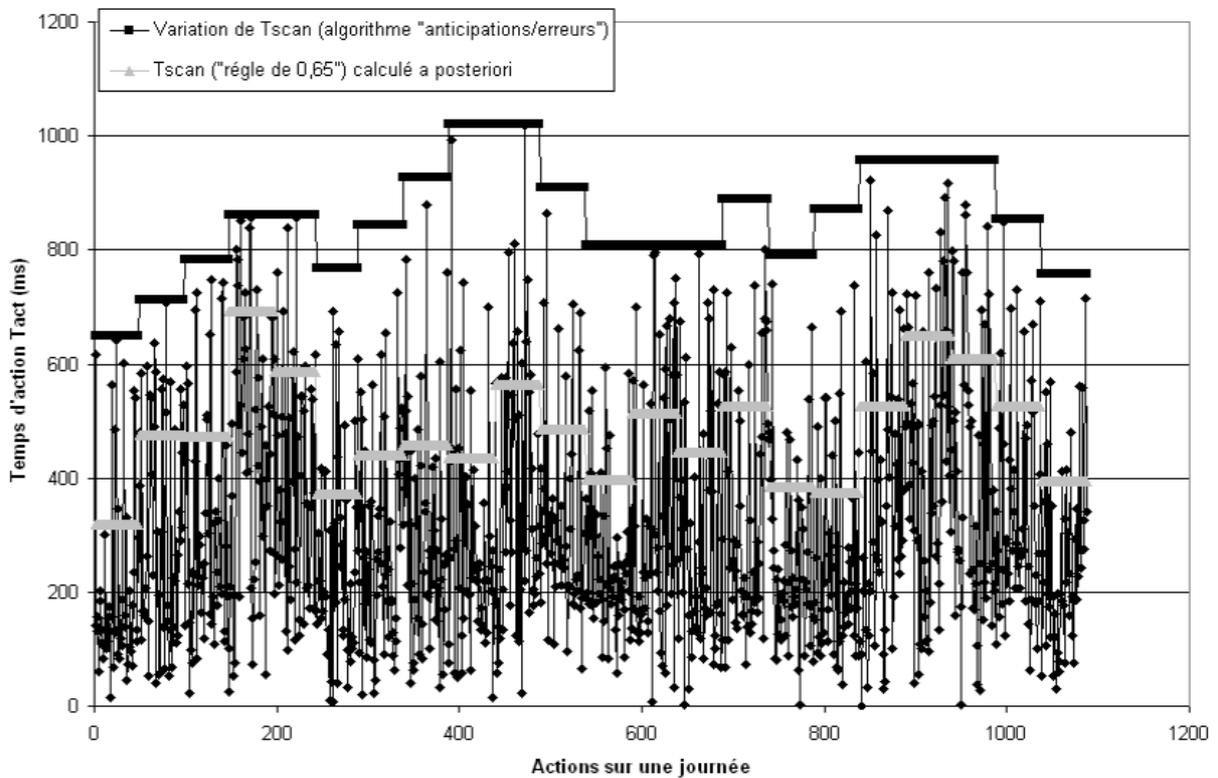


Figure III.4 : Evolution de T_{scan} sur une journée pour une personne SLA avec l'algorithme « anticipations/erreurs »; en abscisses : chronologie des actions; en ordonnée : T_{act}

III.3.3. Expérimentations en laboratoire (Sujets Valides)

III.3.3.1. Introduction

L'expérimentation précédente en situation écologique a montré que l'algorithme des « anticipations/erreurs » est une solution plausible pour le calcul adaptatif de T_{scan} . Pour compléter cette analyse nous avons organisé une nouvelle série de tests en laboratoire sur un panel de trois personnes valides. L'objectif est cette fois de comparer notre algorithme avec les autres méthodes décrites dans la littérature, plus particulièrement avec l'algorithme de Leshner et al. [LESH00b] évoqué dans le chapitre 1 (cf. 1.3.2.4). Nous l'avons implémenté de la manière suivante:

- On analyse les 40 dernières sélections d'items (le choix d'un caractère nécessite 2 sélections) ;

- Si l'une au moins des trois conditions ci-dessous est vérifiée, on augmente T_{scan} à $1.05T_{scan}$:
 - L'annulation du défilement des colonnes (dernière case de la colonne) a été sélectionnée 3 fois ou plus ;
 - La touche « annulation de caractère » a été sélectionnée trois fois ou plus ;
 - Un défilement sur une ligne ou une colonne complète sans sélection d'item et isolé (pas deux défilements consécutifs de ce type) a été sélectionné trois fois ou plus.
- Si on n'est pas dans le cas ci-dessus et que la moyenne des temps d'action T_{act} est inférieure à $0.65 * T_{scan}$, on diminue T_{scan} à $0.95 * T_{scan}$.

III.3.3.2. Méthodologie

La tâche des sujets consiste à écrire un long texte (1200 caractères environ) avec l'algorithme de Leshner et al. puis avec l'algorithme des « anticipations/erreurs ». Ils sont passés au préalable par une phase d'apprentissage conséquente (2 écritures de texte de plus de 1000 caractères par l'algorithme des « anticipations/erreurs ») pour stabiliser la valeur des paramètres *nb clic inf* et *nb clic sup*. Des pauses importantes sont prévues entre deux écritures de texte. Il est également possible de se reposer à la demande pendant l'écriture d'un texte de façon à ce que le paramètre « fatigue » n'intervienne pas dans l'interprétation des résultats. La valeur initiale de T_{scan} est fixée arbitrairement à 300ms.

III.3.3.3. Résultats

Les Figures III.6 à III.8 présentent les résultats des trois sujets sous forme de chronogrammes des temps d'action T_{act} . Sur le même diagramme est tracée l'évolution de T_{scan} ainsi que le taux d'erreurs pendant les 40 dernières sélections (erreurs de sélection d'une colonne ou d'une case). On note également sur le diagramme, toutes les 40 sélections, le calcul de la valeur moyenne des temps d'action T_{act} divisé par le facteur 0.65. Il s'agit cette fois encore d'un calcul a posteriori donné à titre indicatif.

Le sujet 1 (Figure III.6.a) peut être considéré comme expert : il utilise un délai de balayage faible avec un taux d'erreurs négligeable. Les paramètres (*nb clic inf*, *nb clic sup*) ont été fixés dans son cas à (20, 25) ce qui correspond en pratique à réaliser environ une anticipation

toutes les deux sélections. Un exemple de phrase écrite par le sujet est présenté Figure III.5, le symbole « * » représentant l'action « effacer » réalisée par le sujet. Dans ce cas le sujet ne fait pas beaucoup d'erreurs.

Les deux algorithmes donnent ici des résultats similaires avec une valeur de T_{scan} qui se stabilise autour de 180ms environ. Sur ce même sujet nous avons refait l'expérimentation en fixant les paramètres *taux1* et *taux2* à 1.05 et 0.95 respectivement pour qu'ils concordent avec les valeurs de Leshner et al. sans que cela ne change notablement le résultat (Figure III.6.b). Pour les autres expérimentations nous avons conservé les valeurs fixées initialement à (*taux1*, *taux2*) = (0.9, 1.3).

Notre objectif est de régler initialement T_{scan} de façon objective puis de l'adapter automatiquement en cours d'utilisation en fonction de l'état physique de la personne, quelques rares travaux en ce sens sont évoqués dans la littérature, Dans Cronk et Schubert utilisent les erreurs de sélection comme facteur de réglage du temps de défilement, ces erreurs sont réparées de diverses manières utilisation de la touche backspace sélection d'une ligne sans sélection d'une colonne, Leshner et al utilisent le même principe basé sur les erreurs de sélection pour augmenter T_{scan} ,

Figure III.5 : Texte écrit par le sujet 1

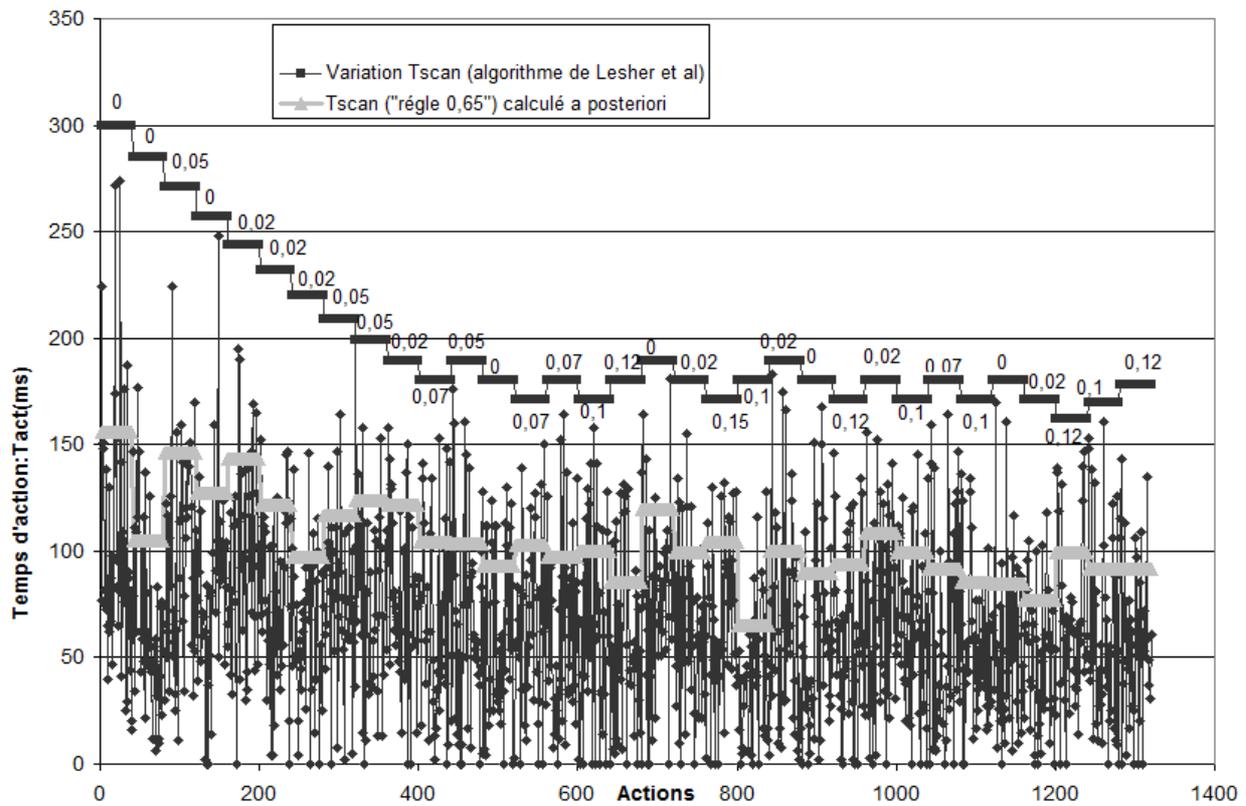
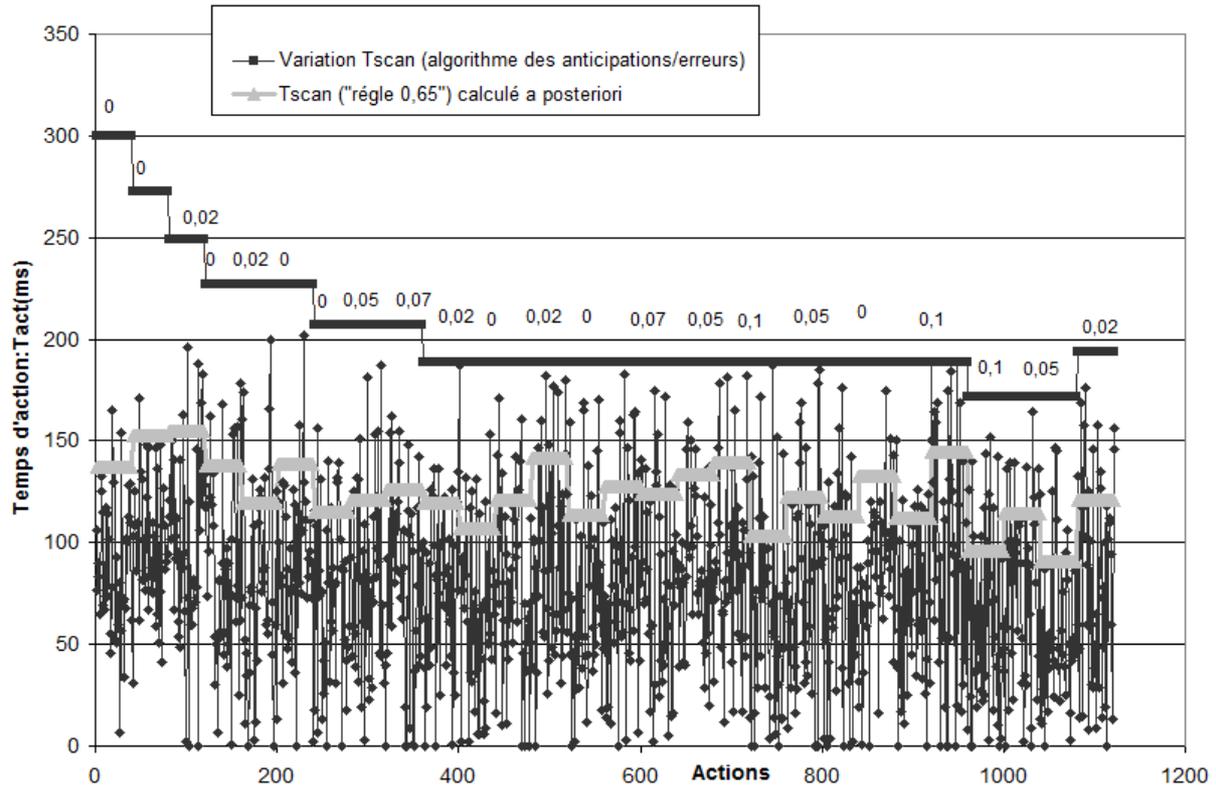


Figure III.6.a : Chronogrammes du sujet 1. En haut : algorithme « anticipations/erreurs » avec $(nb_click_inf, nb_click_sup) = (20, 25)$ et $(taux1, taux2) = (0,9, 1,3)$. En bas : algorithme de Lesher et al.

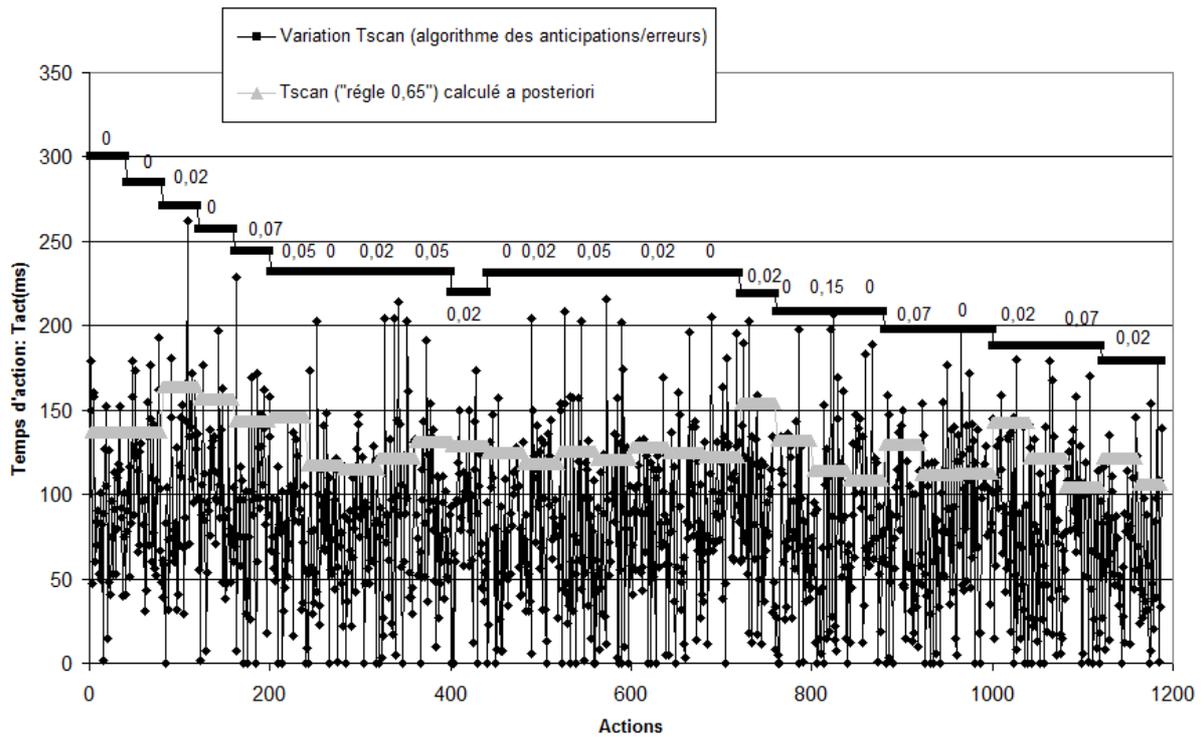


Figure III.6.b : Algorithme « anticipations/erreurs » (sujet 1) avec $(taux1, taux2) = (1.05, 0.95)$

Le sujet 2 (Figure III.7) présente également de bonnes performances : T_{scan} faible, taux d'erreurs faible. L'algorithme « anticipations/erreurs » présente de bons résultats avec une stabilisation de T_{scan} autour de 220ms environ. L'algorithme de Leshner et al. s'est stabilisé à une valeur, jugée subjectivement moins confortable, de 160ms environ sans que le taux d'erreurs n'augmente notablement.

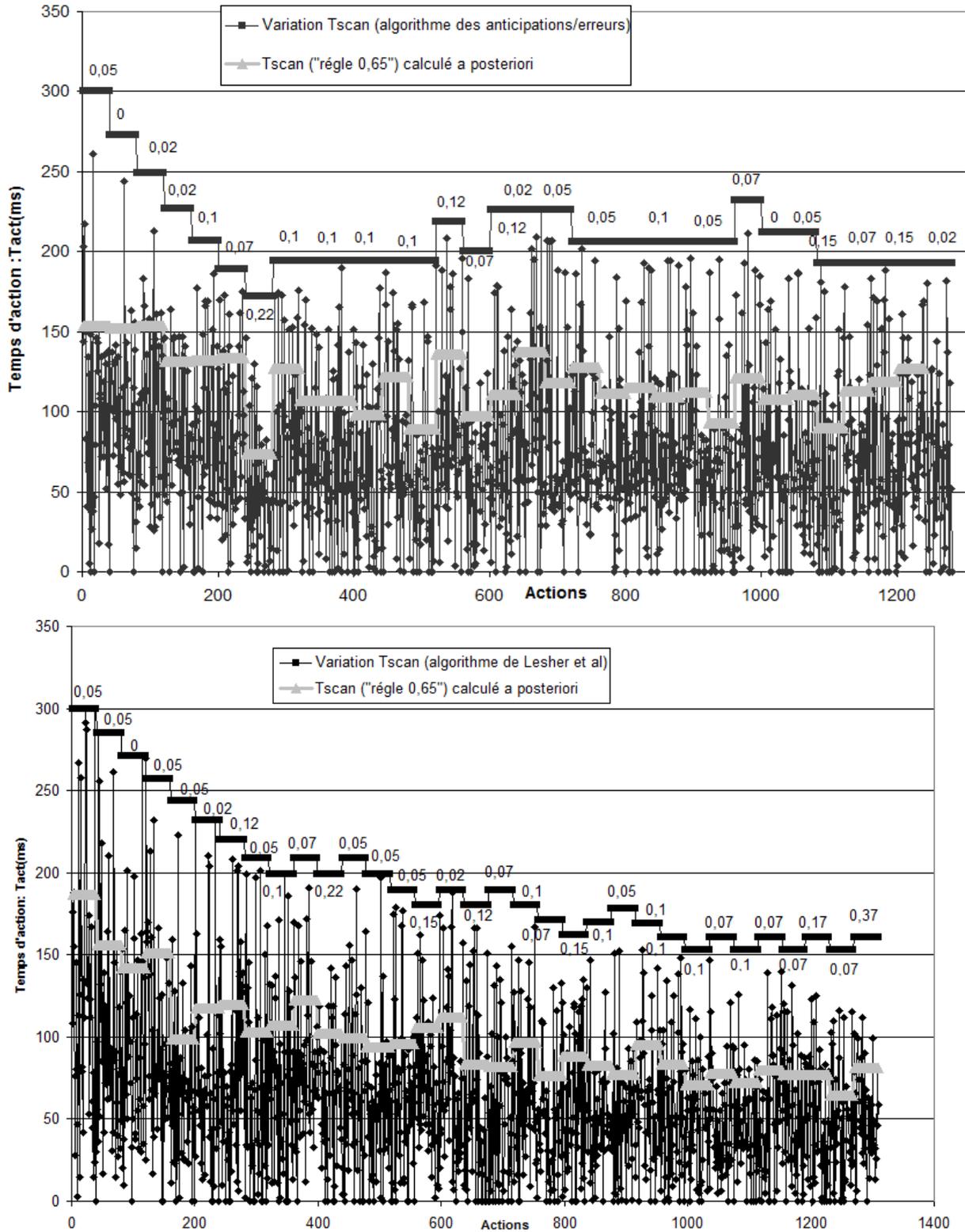


Figure III.7 : Chronogrammes du sujet 2. En haut : algorithme anticipations/erreurs avec (nb_click_inf, nb_click_sup)= (20,25). En bas : algorithme de Lesher et al.

Le sujet 3 (Figure III.8) est un peu moins habile que les deux précédents au sens où il anticipe moins. Un couple de valeurs (15, 20) pour (*nb_clic_inf*, *nb_clic_sup*) s'est révélé optimal dans son cas. L'algorithme « anticipations/erreurs » s'est stabilisé autour de 250ms environ pour T_{scan} et l'algorithme de Lesher et al. à une valeur légèrement plus faible de 230ms. Les taux d'erreurs et la sensation subjective de confort sont similaires avec les deux algorithmes.

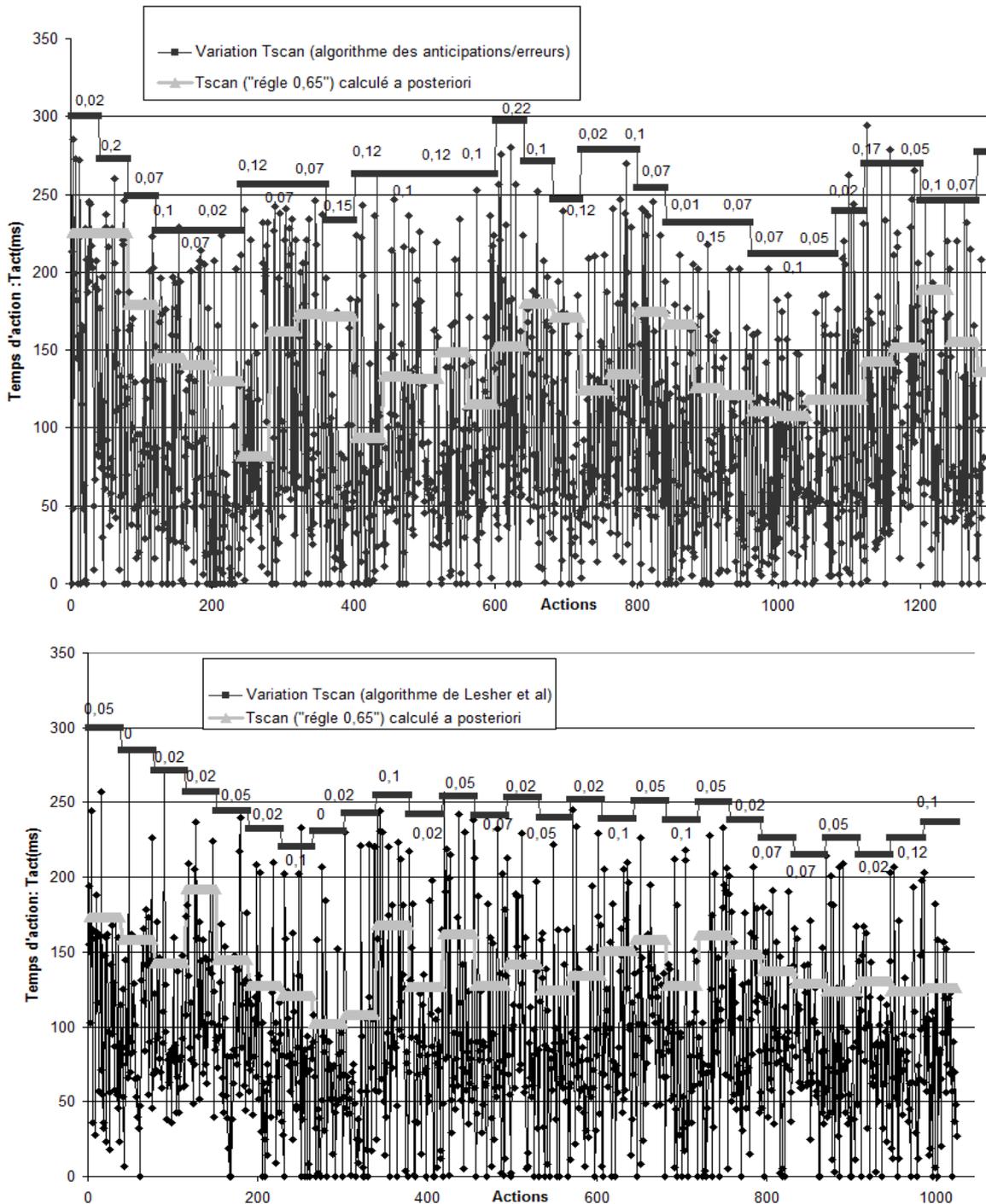


Figure III.8 : Chronogrammes du sujet 3. En haut : algorithme « anticipations/erreurs » avec (*nb_clic_inf*, *nb_clic_sup*)= (15,20). En bas : algorithme de Lesher et al.

Lors des nombreuses utilisations du système EDiTH par des personnes handicapées en situations réelles de communication, nous nous sommes aperçus que beaucoup d'entre elles, pour gagner du temps, ne corrigeait pas leurs fautes de frappe tant que le texte restait intelligible. Pour tester cette situation nous avons recommencé la sélection d'un texte avec l'algorithme de Lesher et al. en enlevant le comptage des annulations de caractère comme critère de remontée de T_{scan} . Cela n'a rien changé pour le sujet 1 qui présentait un taux d'erreurs faible à des valeurs très basses de T_{scan} (Figure III.9). Par contre, pour le sujet 2, l'algorithme de Lesher et al. s'est alors trouvé en défaut (Figure III.10): à un moment donné T_{scan} est devenu trop rapide et le sujet s'est trouvé dans l'impossibilité de faire remonter ce temps du fait de sa stratégie d'utilisation du communicateur. Aucune condition ne permettant de faire remonter T_{scan} ne se trouvait validée. Pour le sujet 3 (Figure III.11), T_{scan} s'est stabilisé mais à une valeur plus faible qu'avec l'algorithme « anticipations/erreurs » (150ms au lieu de 250ms) augmentant son taux d'erreurs et, selon ses dires, son inconfort.

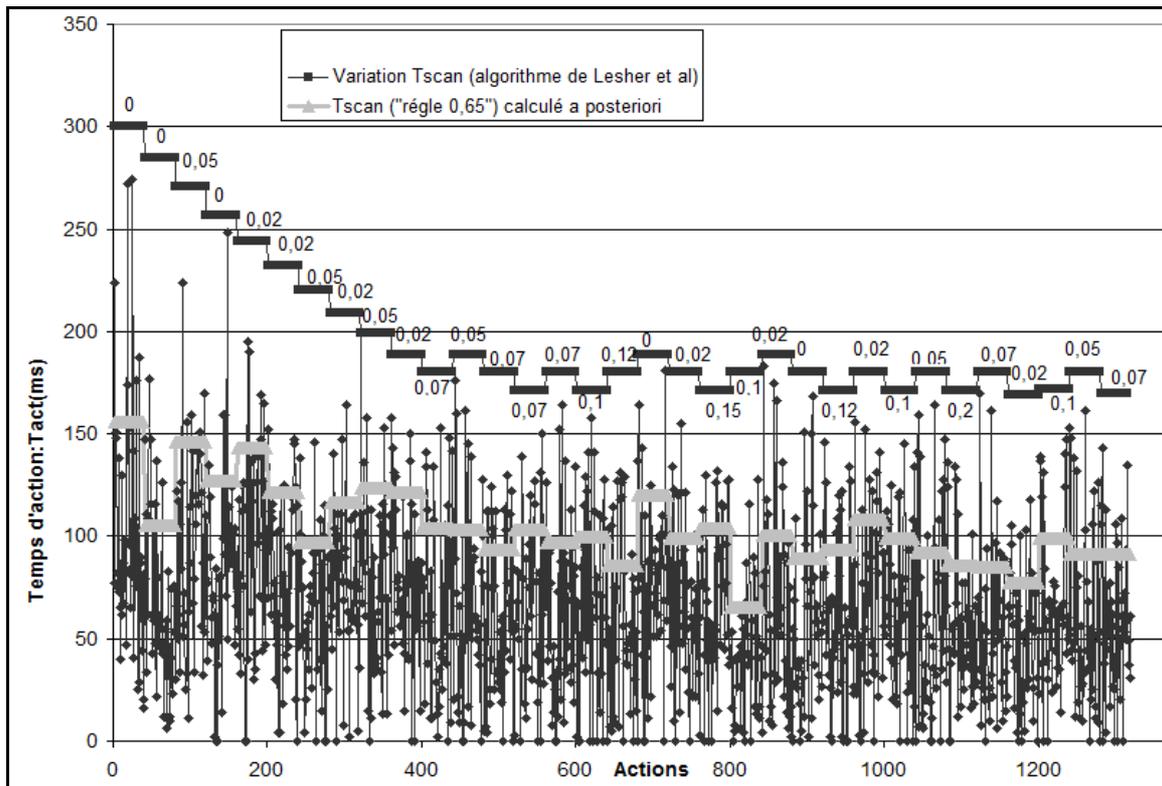


Figure III.9 : Chronogramme du sujet 1. Algorithme de Lesher et al. sans annulation de caractère

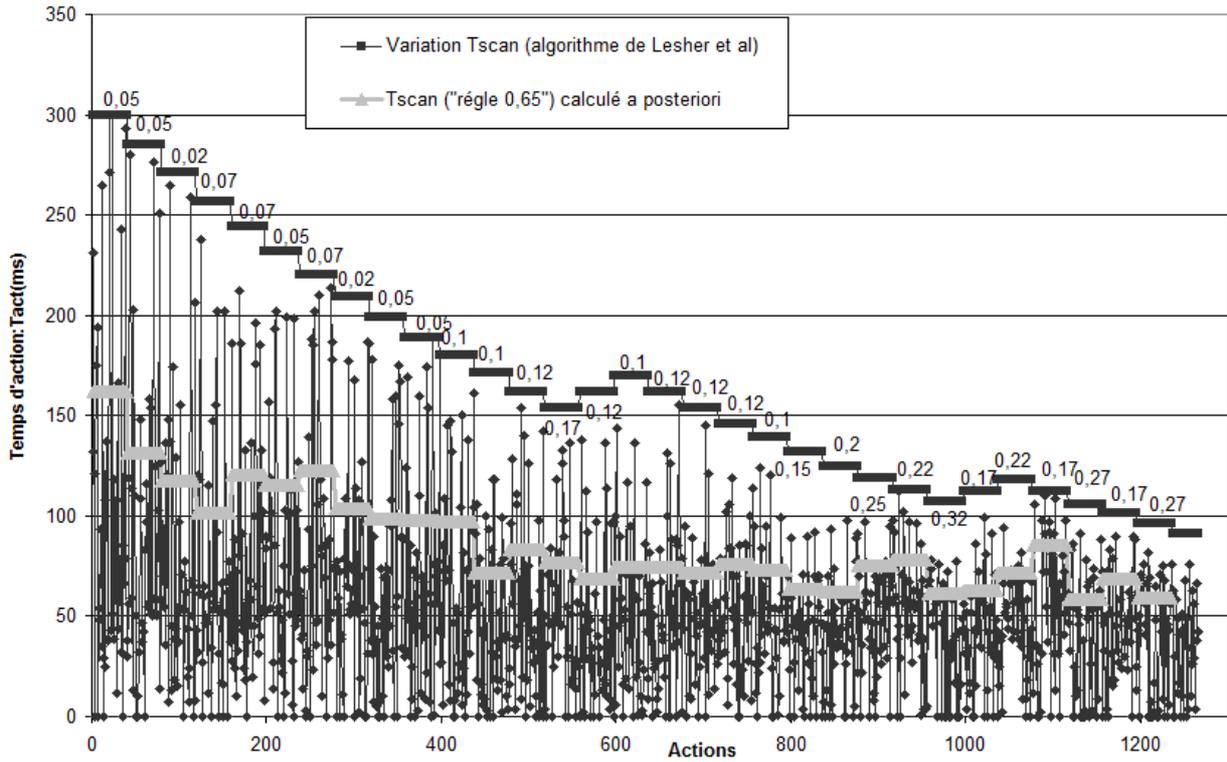


Figure III.10 : Chronogramme du sujet 2. Algorithme de Leshet et al. sans annulation de caractère

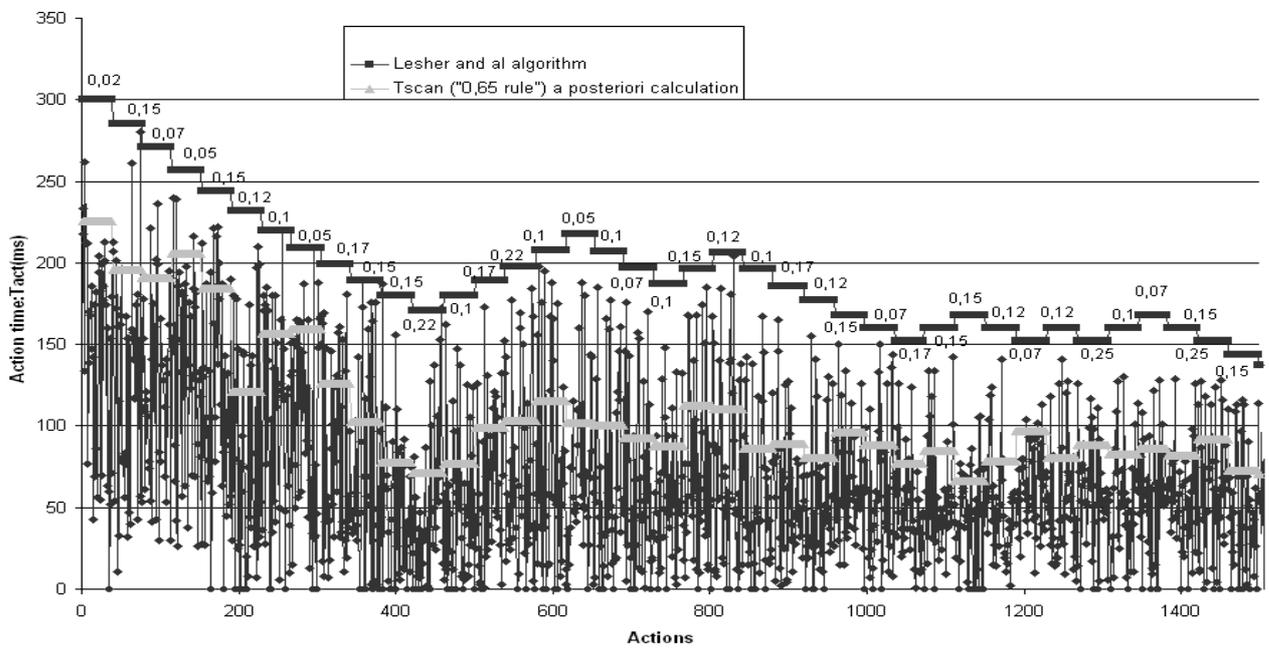


Figure III.11 : Chronogramme du sujet 3. Algorithme de Leshet et al. sans annulation de caractère.

III.3.4. Discussion

L'ensemble des résultats présentés ci-dessus semble valider l'algorithme « anticipations/erreurs » proposé pour le calcul de T_{scan} adaptatif. Il faut cependant que l'utilisateur se conforme au modèle de comportement à trois zones défini plus haut. Ce n'est pas le cas, nous l'avons vu dans le chapitre 2 (Figure II.26), pour certaines personnes infirmes moteurs cérébraux par exemple. L'intérêt de cet algorithme est son indépendance par rapport au type d'interface proposé puisqu'il ne s'intéresse qu'aux temps d'action et non au repérage des erreurs de sélection. Il apparaît donc potentiellement plus facilement transportable sur tout type de communicateur. Son problème majeur par contre concerne le réglage des deux paramètres *nb_clic_inf* et *nb_clic_sup*. S'il est relativement simple de leur donner empiriquement une valeur optimale en conditions de laboratoire, il est plus compliqué de les faire varier de façon adaptative en fonction de l'utilisateur et de son niveau d'apprentissage. Nos travaux futurs vont porter sur ce point essentiel.

Il ressort d'autre part des expérimentations relatées en III.3.3 que l'algorithme de Leshner et al. propose une valeur de T_{scan} « optimale » plus faible que celle proposée par l'algorithme « anticipations/erreurs ». Ces tests portant sur un panel réduit et composé uniquement de personnes valides, il n'est pas possible de généraliser cette constatation. On peut cependant noter que l'application de la règle des 0.65 lorsque les temps d'actions sont faibles, par exemple un temps d'action moyen de 80ms, amène à proposer des délais de défilement, 123ms en l'occurrence, bien trop rapides puisque de l'ordre de grandeur du temps moyen du cycle de perception dans le modèle MHP.

III.3.5. Expérimentations en situation écologique (sujet handicapé)

Dans le domaine des aides techniques pour personnes handicapées une innovation ne peut, à notre sens, être validée que par une utilisation en situation réelle. Nous avons donc entrepris de tester notre « algorithme des anticipations/erreurs » en situation écologique sur de longues périodes. Dans le cadre de ce mémoire nous allons simplement illustrer cette démarche en présentant les premiers résultats concernant une personne de sexe masculin atteinte de Sclérose Latérale Amyotrophique. Cette personne n'a aucune connaissance de l'utilisation d'un ordinateur.

Pour la première utilisation du système EDiTH et après une phase de démonstration de son interface et de toutes ses fonctionnalités, le sujet semble bien motivé pour utiliser le système. C'est là une condition initiale indispensable : l'acceptation de l'aide technique n'est pas une évidence pour des personnes atteintes d'une maladie dégénérative comme la SLA. Après quelques essais nous avons fixé la valeur initiale de T_{scan} à 1400ms, valeur qui nous semble bien adaptée au sujet, puis nous avons activé le mode adaptatif. Nous avons fixé les paramètres $nb_clic_inf = 3$ et $nb_clic_sup = 8$.

La Figure III.12 présente l'histogramme des temps d'action T_{act} de cette personne en utilisation libre sur une période d'une journée avec T_{scan} initial égal à 1400ms et en mode adaptatif. Nous pouvons noter que l'histogramme semble en accord avec le « modèle de comportement à trois zones » constaté en scénarios imposés pour les sujets valides. Nous pouvons constater que sur les 230 actions réalisées, nous trouvons 19 actions avec $T_{act} < 100ms$, 129 actions avec $T_{act} \in [100ms-400ms]$ et 85 actions avec $T_{act} > 400ms$. Pour examiner la zone « avance », nous nous sommes basés sur les fichiers logs pour analyser les actions réalisées par le sujet.

A la lumière de l'analyse des fichiers logs, nous avons trouvé que le sujet s'est intéressé principalement à la communication écrite « clavier virtuel » et à la communication verbale « Soins, Politesse, Autour ». Sur les 40 premières actions (Figure III.12), il a travaillé un peu sur le clavier virtuel, puis il a visité les fonctions simples d'EDiTH comme « Loisir » et enfin la partie communication verbale « Politesse ». Il n'a réalisé aucune action avec $T_{act} < 100ms$ donc T_{scan} a diminué. Sur les 40 actions suivantes, le sujet a réalisé 2 occurrences avec $T_{act} < 100ms$ (la première est réalisée avec $T_{act} = 99ms$ et correspond à une anticipation, la deuxième correspond à une erreur de choix) ; par conséquent T_{scan} diminue encore. Le délai de défilement T_{scan} garde la même valeur sur les 80 actions suivantes : le nombre d'erreurs est égal à 1 et le nombre d'anticipations est de 3 sur les 40 premières actions. Le même phénomène nous le trouvons sur les 40 actions qui suivent (3anticipations, 1 erreur). A la fin de cette journée d'utilisation T_{scan} a encore diminué (une seule anticipation réalisée par le sujet lors de défilements séquentiels).

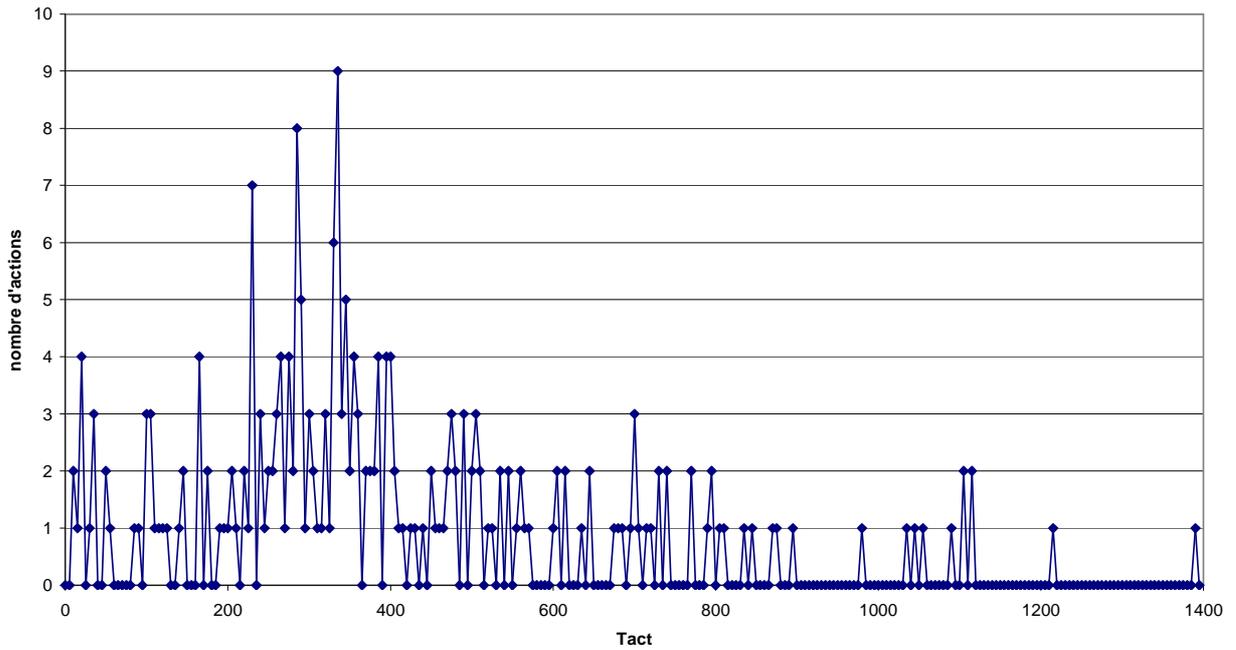


Figure III.12 : Histogrammes des temps d'action T_{act} du sujet SLA en utilisation libre sur une période d'une journée

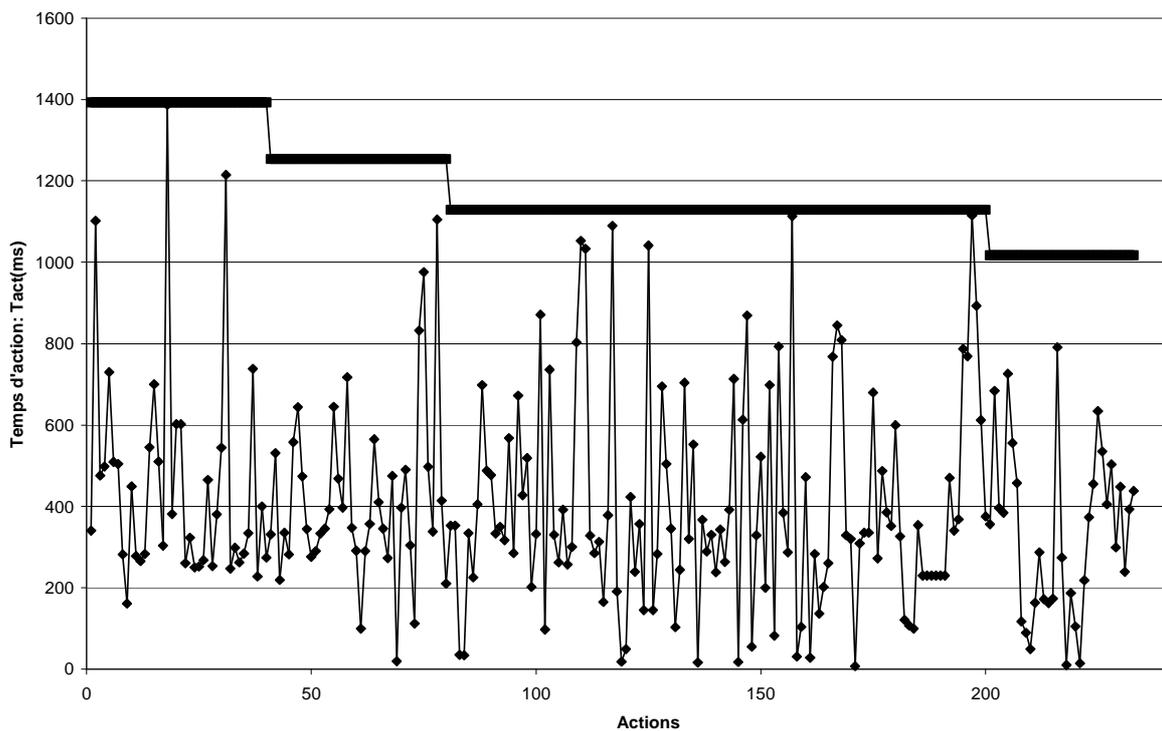


Figure III.13 : Chronogrammes du sujet SLA : variations de T_{scan} selon l'algorithme « anticipations/erreurs »

L'analyse de ses fichiers, nous a permis d'extraire plusieurs actions réalisées par la personne handicapée avec un temps d'action inférieur à 100ms dans les interfaces à défilement

séquentiels. L'objectif de cette expérimentation est d'observer le fonctionnement de l'algorithme des « anticipations/erreurs » en situation écologique sur de longues périodes. Ces premiers résultats ne sont donc là que pour illustrer notre propos. On peut toutefois déjà noter qu'ils ne sont en rien contradictoires avec les autres expérimentations sur personnes valides ou sur personnes handicapées présentées dans ce mémoire.

III.3.6. Conclusion

Nous avons présenté dans cette partie les différents essais réalisés par des sujets valides et handicapés lors de l'utilisation du système EDiTH avec un réglage adaptatif de T_{scan} . Les résultats issus de ces expérimentations ont montré que chaque personne se comporte de manière différente avec l'interface et que l'algorithme « anticipations/erreurs » semble être adapté aux sujets à condition de trouver les bonnes valeurs des paramètres *nb_clic_inf* et *nb_clic_sup*.

III.4. Conclusion du chapitre3

La dernière partie de ce mémoire s'est attachée à la présentation de notre méthode de travail pour adapter la vitesse de défilement des interfaces à balayage utilisées par de nombreuses personnes handicapées moteur souffrant de troubles de la communication verbale. Cette méthode est dérivée d'une modélisation de l'interaction humain – machine basée sur le modèle MHP adapté à ce contexte d'application.

Nous avons présenté notre « algorithme des anticipations/erreurs » qui se base sur le calcul de N_{100} le nombre d'actions réalisées avec un temps T_{act} inférieur à 100ms. Ensuite nous avons décrit les expérimentations réalisées en laboratoire sur des sujets valides et en situation écologique sur des personnes handicapées et nous avons comparé notre méthode d'adaptation du délai de défilement T_{scan} à d'autres méthodes décrites dans la littérature. L'analyse des fichiers logs issus de l'utilisation d'EDiTH a permis de valider notre algorithme.

CONCLUSION ET PERSPECTIVES

Notre travail visait à concevoir et valider un algorithme d'adaptation automatique du temps de défilement des aides à la communication à balayage. Notre méthode de travail est basée sur une analyse de l'interaction humain-ordinateur en utilisant le modèle MHP développé initialement par Card.

Nous avons tout d'abord établi un état de l'art sur les différents systèmes d'aide à la communication par défilement. Nous avons décrit ensuite les différentes techniques d'accélération pour améliorer les performances en termes de rapidité de communication. Ces performances sont fonctions des possibilités physiques et cognitives de l'utilisateur, en particulier de son temps moyen de réaction qui induit le temps de défilement du balayage de l'interface graphique ou alphabétique. Ensuite nous avons cité les quelques études qui se sont intéressées à adapter en temps réel le temps de défilement et nous avons constaté que deux paramètres sont utilisés dans ce but : le temps de réaction du sujet et les erreurs de sélection.

Le deuxième chapitre a porté sur la présentation du système EDiTH que nous avons développé au LASC, ce système a été conçu dans le but d'être à la fois un système de communication et de contrôle de l'environnement utilisable dans la vie courante du sujet et d'être aussi un outil de recherche dans notre travail. Nous avons commencé par décrire les différentes interfaces et les fonctionnalités d'EDiTH. Ensuite, nous avons présenté notre méthode de travail qui est basée sur une analyse des fichiers logs issus de l'utilisation du système EDiTH. Nous avons aussi décrit les expériences réalisées avec le modèle MHP pour valider l'hypothèse que le temps nécessaire pour valider une case déjà repérée nécessite généralement un temps $T_p + T_c + T_m$. Nous en avons déduit un « modèle à 3 zones » pour des sujets valides ou des sujets handicapés contrôlant bien leur capteur et sans difficultés cognitives. Nous avons conclu ce chapitre par une description des premiers résultats issus de l'utilisation d'EDiTH qui montrent la possibilité d'utiliser le nombre d'occurrences des temps d'action T_{act} inférieurs à 100ms pour régler le temps de défilement.

Le dernier chapitre s'est intéressé au délai de défilement adaptatif. Nous avons tout d'abord décrit notre méthode pour régler le temps de défilement ainsi que les différents paramètres utilisés dans notre « algorithme des anticipations/erreurs ». Les expérimentations en

laboratoire et en situation écologique ont montré que notre méthode est une solution plausible pour le calcul adaptatif de T_{scan} , et que nous devons trouver les bons paramètres pour chaque utilisateur. Pour évaluer la performance de notre méthode, nous avons comparé notre algorithme avec une autre méthode décrite dans la littérature, l'algorithme proposé par Lesh et al. Enfin nous avons présenté les premiers résultats d'une évaluation en situation écologique. L'avantage de notre méthode de travail vient du fait que notre algorithme est indépendant du type d'interface puisqu'il ne s'intéresse qu'aux temps d'action et non au repérage des erreurs de sélection.

D'autre part, outre la poursuite de nos expérimentations en situations écologiques, quelques perspectives ressortent de notre étude. A court terme il serait intéressant d'étudier le comportement de « l'algorithme des anticipations/erreurs » sur plusieurs types de claviers (statiques ou dynamiques) et sur des écrans multiples intégrant des listes de prédictions. Le type de défilement, séquentiel ou ligne/colonne notamment, pourrait être également intéressant à prendre en compte pour le réglage de l'algorithme. Il nous faudra aussi poursuivre nos expérimentations pour définir des solutions spécifiques à des situations où notre « modèle de comportement à trois zones » n'est plus valide du fait de troubles moteurs (infirmités motrices-cérébrales) ou cognitifs (traumatismes crâniens) par exemple.

Enfin une perspective importante est l'intégration de notre algorithme dans d'autres communicateurs que le système EDiTH. Du fait qu'il ne tient compte que des temps d'action de l'utilisateur il est en effet potentiellement transposable sur tout type d'appareil d'aide à la communication par défilement.

BIBLIOGRAPHIE

- [ABAS04] J. Abscal, L. Gardeazaba, N. Garay. « *Optimisation of the selection features for scanning text input* ». ICCHP, Paris (France), 788-795, July 7-9 2004.
- [ABLE] <http://www.ablenetinc.com/Default.aspx?tabid=361>.
- [ALM92] N. Alm, J.L. Arnott, A.F. Newell. « *Prediction and conversational momentum in an augmentative communication system* ». Communications of the ACM, New York (ETATS-UNIS), vol. 35, n°5, p. 47-57, 1992.
- [ANSO93] D. An son. « *The effect of word prediction on typing speed* ». American Journal of Occupational Therapy, 47 (11), p. 1039-1042, 1993.
- [ANTO07] J.Y. Antoine, D. Maurel. « *Aide à la communication pour personnes handicapées et prédiction de texte : Problématique, état des lieux et retour sur trente ans de recherche en communication augmentée* ». Traitement Automatiques des Langues, vol. 48, n° 2, 2007.
- [AXEL] <http://www.axelia.com/>.
- [BELL04a] E. Bellengier, S. Rauzy, P. Blache. « *La Plateforme de Communication Alternative : Un système d'aide à la communication pour personnes handicapées* ». Travaux Interdisciplinaires du Laboratoire Parole et Langage d'Aix-en-Provence (TIPA), vol. 23, 2004.
- [BELL04b] E. Bellengier, P. Blache, S. Rauzy. « *PCA : Un système de communication alternative évolutif et réversible* ». In actes Conférence ISAAC, Neuchâtel (Suisse), p. 78-85, 6-8 Mai 2004.
- [BELL06] E. Bellengier, S. Rauzy, J. Marty. « *Système de communication iconique : Reformulation avancée* », IFRATH, Handicap 2006, France, 2006.
- [BELA07] Belatar M., Poirier F., « *UniGlyph : une méthode universelle pour la saisie de texte sur dispositifs mobiles* », Actes IHM'2007, Paris (France), p. 111-118, 2007.
- [BIRD07] N. Biard. « *Les outils technologiques de Communication Améliorée et Alternative* », Plate- Forme Nouvelles Technologies, Juillet 2007.
- [BISW07a] P. Biswas. « *Simulating HCI for special needs* ». ACM SIGACCESS Accessibility and Computing, issue 89, 7-10, Septembre 2007.
- [BISW07b] P. Biswas, P. Robinson. « *Performance comparison of different scanning systems using a simulator* ». In Proceedings of the 9th European Conference for the Advancement of the Assistive Technologies in Europe AAATE, p. 873-877, 2007.
- [BOIS00] PH. Boissière. « *VITIPI : Un système d'aide à l'écriture basé sur un principe d'autoapprentissage et adapté à tous les handicaps moteurs* ». Handicap 2000, Paris, p. 81- 86, 2000.
- [BOIS01] Ph. Boissière, D. Dours. « *Comment VITIPI un système d'assistance à l'écriture pour les personnes handicapées peut offrir des propriétés intéressantes pour le TALN ?* ». Atelier Thématique TALN 2001, Tours (France), p. 183- 192, 2-5 Juillet 2001.
- [BOIS03] Ph. Boissières. « *An overview of existing writing assistance systems* ». Workshop International Franco-Espagnol organisé par l'IFRATH, Paris, 16-17 octobre 2003.
- [BOUR86] G. Bourhis, J.M. André, E. Yvroud. « *Désigneur polyvalent pour grands handicapés moteurs présentant des troubles de la communication verbale* ».

- Innovation Technologique en Biologie et Médecine, vol.7, n°4, p. 474-481, 1986.
- [BOUR01] G. Bourhis, P. Pino, A. Leal-Olmedo. « *Temporal analysis of the use of an augmentative communication device* ». In proceedings of the 7th European Conference for the Advancement of Assistive Technology, Ljubljana (Slovenia), p. 139-143, Septembre 2001.
- [BOUR02] G. Bourhis, P. Pino, A. Leal-Olmedo. « *Aide à la communication et au contrôle d'environnement pour personnes handicapées : modélisation de l'interaction homme-machine* ». Revue d'Interaction Homme-Machine, Vol. 3, n°2, p.135-148, 2002.
- [BOUR02a] G. Bourhis, P. Pino, A. Leal-Olmedo. « *Communication devices for persons with motor disabilities* ». Human-machine interaction modelling in proceedings of IEEE SMC, Hammamet (Tunisia), 2002.
- [BOUR04] G. Bourhis, P. Pino, C. Dumas, N. Biard, F. Stoll. « *Modelisation of "the person with disabilities – aid to communication" system: some experimental results* ». AMSE Periodicals, Modelling C, vol. 65, n° 7/8, p. 121-130, 2004.
- [BRAN97] E. Brangier, P. Pino. « *Métaphores de l'action et conception d'une interface Pour grands handicapés moteurs : De la description de la sclérose latérale amyotrophique à la conception d'une prothèse interactionnelle* ». IHM Poitiers, 1997.
- [BRAN00] E. Brangier, P. Pino. « *Psychodesign of interaction : communication system design for patient with motor impairments* ». Science and technique of design, Décembre 2000.
- [BRUG96] F. Brugnara, M. Federico. « *Techniques for approximating a trigram language model* ». In Proceeding of International Conference on Spoken Language Processing, Philadelphia, 1996.
- [CARD83] S.K. Card, T.P. Moran, A. Newell. « *The Psychology of Human-Computer Interaction* ». Hillsdale, 1983. NJ (USA): Lawrence Erlbaum Associates.
- [CATA98] E. Cataix-negre. « *L'utilisation des codes alphabétiques de communication* ». Conférence ISAAC, Paris, 27 novembre 1998.
- [CANT00] B. Cantegrit, J.M. Toulotte. « *Communication enrichie et palliative : application au handicap moteur* ». European Journal of Automation, vol. 34, n° 6-7, p. 859-882, 2000.
- [CANT01] B. Cantegrit, J.M. Toulotte. « *Réflexions sur l'aide à la communication des personnes présentant un handicap moteur* », actes atelier Ingénierie de la langue et handicap, TALN, Tours (France). p. 193-202, 2001.
- [CANT03] B. Cantegrit, J.M. Toulotte. « *Communication enrichie et palliative : handicap moteur* ». Assistance technique au handicap, Hermès, chap.5, p. 305-357, 2003.
- [COUT90] J.Coutaz. 1990, livre « *Interfaces Homme-Ordinateur* » Edité par BORDAS.
- [CREM99] J. Cremelie, B. Noë, « *Mind Express* ». Assistive Technology on the Threshold of the New Millennium, IOS Press, p. 197-201, 1999.
- [CRON87] S.R. Cronk, R.W. Schubert. « *Development of a real-time expert system for automatic adaptation of scanning rates* ». In Proceedings of the RESNA 10th Annual Conference, San Jose (California), p. 109-111, 1987.
- [DEMA92] P.W. Demasco, K.F. McCoy. « *Generating text from compressed input : an intelligent interface for people with severe motor impairments* ». Communications of the ACM, vol. 35, n° 5, p. 68-78, 1992.
- [EVER04] G.E. Evreinov, R. Raisamo. « *Optimizing Menu Selection Process for single-Switch Manipulation* ». ICCHP : computers helping people with special needs Conference N°9, Paris (France), p. 836-844, 7-9 July 2004.

- [FELZ06] T. Felzer, R. Nordmann. « *Alternative text entry using different input methods* ». ACM SIGACCESS Conference on Computers and Accessibility, ASSETS'06. Portland, OR, P. 10-17, 2006.
- [FST] <http://www.fst.ch>.
- [GHED06] S. Ghedira, P. Pino, G. Bourhis. « *Interaction personne handicapée/système à balayage : vers un réglage automatique de la vitesse de défilement adapté à l'utilisateur* ». Colloque Ergo-IA, Biarritz, octobre 2006.
- [GHED08] S. Ghedira, P. Pino, G. Bourhis. « *Interaction between a disabled person and a scanning communication aid : towards an automatic adjustment of the scanning rate* ». 11th International Conference on Computers Helping People with special needs, ICCHP 2008, Linz (Autriche), p. 1204-1207, 9-11 July 2008.
- [GHED09] S. Ghedira, P. Pino, G. Bourhis. « *Conception and experimentation of a communication device with adaptative scanning* ». ACM Transactions on Accessible Computing special issue on Augmentative and Alternative Communication, vol.1, n°3, article n°14, 2009.
- [GUEN92] F. Guenther, K. Krüger-Thielmann, R. Pasero, P. Sabatier. « *Communications Aids for ALS Patients* », Proc. Third International Conference on Computers for Handicapped Persons, ICCHP, P. 303-307, 1992.
- [HORS94] K.H. Horstmann, S.P. Levine. « *Modeling the speed of text entry with word prediction interface* », IEEE Transactions on rehabilitation engineering, 2(3). p. 177-187, 1994.
- [HMC] <http://www.hmc-nv.be/index.php?id=676>
- [JACQ] <http://jacquet.stephan.free.fr/communication.pdf>
- [KEAT98] S. Keates, P.J. Clarkson and P. Robinson. « *Developing a methodology for the design of accessible interfaces* ». Proceedings of the 4th ERCIM Workshop, p. 1-15, 1998.
- [KEAT00] S. Keates, P.J. Clarkson and P. Robinson. « *Investigating the applicability of user models for motion-impaired users* ». In Proceeding of the fourth international ACM Conference on Assistive Technologies, Virginia (United States), p. 129-136, 2000.
- [KOES94a] H. Koester, S. Levine. « *Learning and performance of able-bodied individuals using scanning systems with and without word prediction* ». *Assistive Technology*, p. 42-53, 1994.
- [KOES94b] H. Koester, S. Levine. « *Modeling the speed of text entry with a word prediction interface* ». IEEE Transactions on rehabilitation engineering, vol. 2, n° 3, p. 177-187, Sept 1994.
- [KOES00] H. Koester and R. Simpson. « *Effect of system configuration on user performance with word prediction: Results for users with disabilities* ». In Proceedings of the RESNA Annual Conference, Orlando ,2000.
- [KOES05] H. Koester, E. F. LoPresti, and R. C. Simpson. « *Toward Goldilocks' pointing device: Determining a "just right" gain setting for users with physical impairments* ». in ACM SIGACCESS Conference on Assistive Technologies (ASSETS), Baltimore, ACM Press, 2005.
- [KUHN01] M. Kühn, J. Garbe. « *Predictive and highly ambiguous typing for a severely speech and motion impaired user* ». In Stephanidis C. (Ed.) *Universal access in Human-Computer Interaction*, p. 933-937, 2001.
- [LEBL] <http://www.leblatphone.fr/>

- [LEPE97] B. Le Pévédic. « *Prédiction morphosyntaxique évolutive dans un système d'aide à la saisie de textes pour les personnes handicapées physiques* ». thèse de doctorat Université de Nantes (France), 1997.
- [LEPE98] B. Le Pévédic. « *Les niveaux syntaxiques dans le système HandiAS* ». Actes TALN'1998, Paris (France), p. 132-141, 1998.
- [LESH98] G.W. Leshler, B.J. Moulton, D.J. Higginbotham. « *Techniques for augmenting scanning communication* ». *Augmentative and Alternative Communication. Decker Periodicals Inc., Canada*, 14 (2), p. 81-101, 1998.
- [LESH98a] G.W.Leshler, B.J.Moulton, D.J. Higginbotham, « *Techniques for augmenting scanning communication* ». *Augmentative and Alternative Communication, Decker Periodicals Inc., Hamilton, Ontario (Canada)*, p. 81-101, 1998.
- [LESH98b] G.W. Leshler, B.J. Moulton, D.J. Higginbotham. « *Optimal character arrangements for ambiguous keyboards* ». *IEEE Transactions on Rehabilitation Engineering*, vol. 6, p. 415-423, 1998.
- [LESH00] G.W. Leshler, B.J. Moulton, D.J. Higginbotham. « *Techniques for automatically updating scanning delays* ». In *Proceedings of the RESNA Annual Conference*, p. 85-87, 2000.
- [LESH00a] G.W. Leshler, B.J. Moulton. « *A method for optimizing single-finger keyboards* ». *RESNA Annual Conference*, p. 91-93, 2000.
- [LESH00b] G.W.Leshler, D.J. Higginbotham, B.J. Moulton. « *Techniques for automatically updating scanning delays* ». In *Proceedings of the RESNA Annual Conference*, p. 85-87, 2000.
- [LESH02] G.W.Leshler, B. J. Moulton, D. J. Higginbotham, and B. Alsofrom. « *Acquisition of Scanning Skills: The use of an adaptive scanning delay algorithm across four scanning displays* ». In *Proceedings of the Annual Conference of RESNA*, 2002.
- [LEVI90] S.H. Levine, C. Goodenough-Trepagnier. « *Customised text entry devices for motorimpaired users*». *Applied Ergonomics*, vol. 21, p. 55–62, 1990.
- [MACK99] I.S. MacKenzie, S.Z. Zhang. « *The design and evaluation of a high performance soft keyboard* ». *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*. New-York, p. 25-31, 1999.
- [MAUR00] D. Maurel, B.Fourche, S. Briffault. « *HandiAS : Aider la communication en facilitant la saisie rapide de textes* ». In *actes de Handicap 2000*.
- [MAUR01] D. Maurel, N. Rossi, N.Thibault. « *HandiAS : un système multilingue pour l'aide à la communication des personnes handicapées* ». *TALN, Tours*, vol 2, p. 203-212, 2001.
- [MENI01] G. Ménier, F. Poirier. « *Système adaptatif de prédiction de texte* ». *TALN, Tours*, vol. 2, p. 213-222, 2001.
- [MURR91] I. Murray, J. Arnott, N. Alm, A. Newell. « *Communication System for the Disabled with Emotional Synthetic Speech Produced by Rule* ». In *Proceedings of Eurospeech 91, 2nd European Conference on Speech Communications and Technology*, p. 311-314, 1991.
- [PINO00] P. Pino. « *EDITH : Adaptation automatique du temps de défilement aux caractéristiques et intentions de l'utilisateur* ». *Handicap2000, Paris*, p. 125-130, 2000.
- [PLOS06] O. Plos, S. Buisine. « *Décomposition Multimodale de l'Activité : Vers un outil d'aide à la conception* ». *Conférence Ergo IA*, p7, 2006.
- [RAUZ06] S. Rauzy & A. Bianco. « *Prise en charge des troubles de la communication par un système de communication alternative* », in *revue Rééducation Orthophonique, numéro spécial*, 2006.

- [RAYN05a] M. Raynal, N. Vigouroux. « *Genetic Algorithm to Generate Optimized Soft Keyboard* ». Proc. Human Factors in Computing Systems Conference, CHI, USA, p. 1729-1732, 2005.
- [RAYN05b] M. Raynal, N. Vigouroux. « *KeyGlasses : semi-transparent keys to optimize text input on a virtual keyboard* ». Proc. AAATE, Lille (France), p. 713-717, 2005.
- [RAYN05c] M. Raynal. « *Systèmes de saisie de textes pour les personnes handicapées moteur : optimisation, interaction et mesure de l'utilisabilité* ». Thèse de doctorat de l'université de Toulouse III Paul Sabatier, 5 décembre 2005.
- [RAYN07] M. Raynal. « *Le système KeyGlass Système d'ajout dynamique de touches sur clavier logiciel* ». TAL. vol. 48, n° 2, p. 97-121, 2007.
- [RICH98] N. Richardet. « *Composition de phrase assistée – Un système d'aide à la communication pour handicapés* ». Thèse de doctorat, Université de la Méditerranée, 1998.
- [RICC01a] X. Ricco, T. Dutoit. « *Vers un logiciel multilingue et gratuit pour l'aide aux personnes handicapées de la parole : HOOK* ». Actes TALN, atelier Ingénierie des Langues et Handicap, Tours (France), vol 2, p. 223-232, 2001.
- [SANG07] T.D. Sanger, J. Henderson. « *Optimizing assisted communication devices for children with motor impairments using a model of information rate and channel capacity* ». IEEE Transactions on neural systems and rehabilitation engineering, vol. 15, n° 3, p. 458-468, 2007.
- [SCHA01] I. Schadle, B. Le Pévédic, J.Y. Antoine, F., Poirier. « *SibyLettre Système de prédiction de lettre pour l'aide à la saisie de texte* », TALN, Tours, juillet 2001.
- [SCHA03] I. Schadle. « *Sibylle : système linguistique d'aide à la communication pour les personnes handicapées* ». Thèse de doctorat de l'université de BRETAGNE SUD, 18 décembre 2003.
- [SCHA04] I. Schadle, J.Y. Antoine, B. Le Pévédic, F. Poirier. « *Sibyl - AAC system using NLP techniques* ». (Eds): ICCHP 2004, Paris (France), p. 1009-1015, 7-9 July 2004.
- [SITB07] L. Sitbon, P. Bellot, Ph. Blache. « *Traitements phrastiques phonétiques pour la réécriture de phrases dysorthographies* », Actes TALN, Toulouse. vol. 1, p. 263-272, 2007.
- [SIMP99] R.C. Simpson, H.H. Koester. « *Adaptive one-switch row-column scanning* ». IEEE Transactions on Rehabilitation Engineering, 7(4), p. 464-473, 1999.
- [SIMP00] R.C. Simpson, H.H. Koester. « *An Adaptive Word Prediction Interface* ». Proceedings of the AAAI Spring Symposium on Adaptive User Interfaces, Palo Alto, CA, 2000.
- [SIMP06a] R.C. Simpson, H.H. Koester, E. Lopresti. « *Evaluation of an adaptative row/column scanning system* ». *Technology and Disability*, vol. 18, p. 127-138, 2006.
- [SIMP06b] R.C. Simpson, H.H. Koester, E. Lopresti. « *Selecting an appropriate scan rate: the "065 rule"* ». In Proceedings of the RESNA Annual Conference, 2006.
- [SUPP] <http://www.suppleance.fr/>
- [STEW00] H. Stewart, A. Wilcock. « *Improving the communication rate for symbol based, scanning voice output device users* », *Technology and Disability*, IOS Press, vol. 13, n° 3, P. 141-150, 2000.
- [TRNK06] K. Trnka, D. Yarrington, K.F. McCoy, C. Pennington. « *Topic Modeling in Fringe word prediction for AAC* ». Proc. International Conference on Intelligent User Interfaces, Sydney, Australie, p. 276-278, 2006.

- [TRNK07] K. Trnka, K.F. McCoy. « *Corpus studies in word prediction* ». Proc. International ACM Conference on Assistive Technologies, ASSETS, Phoenix, AZ. P. 195-202, 2007.
- [UTIL] <http://utilware.com>
- [VAIL97] P. Vaillant. « *PVI : Système de traduction d'icônes en langue – Interaction entre modalités sémiotiques : de l'icône à la langue* ». Thèse de doctorat en Sciences Cognitives, Université Paris-XI, Orsay, 1997.
- [VERG00] J. Vergne. « *Étude et modélisation de la syntaxe des langues à l'aide de l'ordinateur – Analyse syntaxique automatique non combinatoire* ». Thèse d'habilitation, Université de Caen, 2000.
- [WAND06] T. Wandmacher, J.-Y. Antoine. « *Training language models without appropriate language resources : experiments with an AAC system for disabled people* ». Proc. 5th European Conference on Language Resource and Evaluation, LREC, Genova (Italie), 2006.
- [WAND07] T. Wandmacher, J.-Y. Antoine, F. Poirier. « *Sibylle : a system for alternative communication adapting to the context and its user* ». Proc. International ACM Conference on Assistive Technologies, ASSETS, Phoenix, p. 203-210, 2007.
- [ZAGL99] W.L. Zagler. « *Word prediction and word completion for the German language* », Assistive Technology on the Threshold of the New Millennium, IOS Press, p. 191-196, 1999.
- [ZHAI00] S. Zhai, M. Hunter, B.A. Smith. « *The Metropolis Keyboard – An Exploration of Quantitative Techniques for Virtual Keyboard Design* ». In Proceedings of ACM Symposium on User Interface Software and Technology, San Diego, California, p. 119-128, November 2000.

Résumé

Le problème majeur des communicateurs à balayage pour personnes handicapés moteurs est la lenteur de la communication qu'ils induisent. Un moyen d'améliorer la vitesse de sélection des messages consiste à optimiser en temps réel le délai élémentaire de défilement. Notre méthode de travail dans cet objectif est basée sur l'analyse des données enregistrées dans des « fichiers logs » lors de l'utilisation du système de communication et de contrôle de l'environnement multimédia EDiTH développé au laboratoire. De cette analyse, en s'appuyant sur le modèle du « Processeur Humain modèle » (MHP), nous déduisons un modèle de l'interaction humain-machine dans le contexte de l'étude puis un algorithme de calcul adaptatif du délai de défilement, « l'algorithme des anticipations/erreurs ». Nous avons validé notre propos en se basant sur les résultats obtenus en scénarios imposés sur des personnes valides et des personnes handicapées, et en situation écologique sur des personnes handicapées.

Mots clés: communicateur à balayage, interaction humain-machine, modèle MHP, temps de défilement adaptatif.

Abstract

The main problem with the scanning systems is that the communication process tends to be exceedingly slow. One way of raising the speed of message selection is to optimise the elementary scanning delay in real time. Our method of work in this objective is based on an analysis of the data recorded in « log files » while applying EDiTH, the system of communication and control of the multimedia environment developed in our laboratory. This analysis makes it possible to develop a human-machine interaction model specific to the study based on the « Model Human Processor » (MHP), and then to establish an adaptive algorithm for the calculation of the scanning delay: the « anticipations / errors algorithm ». We validate our work based on the results obtained in imposed scenarios with persons with and without disability, and then in ecological situation with people with disabilities.

Keywords: scanning system, human-machine interaction, Model Human Processor, adaptive scanning delay.