



**HAL**  
open science

# Plasticité de l'interaction Homme-Machine : présentation à l'utilisateur, une question de compromis

Sara Bouzit

► **To cite this version:**

Sara Bouzit. Plasticité de l'interaction Homme-Machine : présentation à l'utilisateur, une question de compromis. Informatique. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAM023 . tel-01681499v2

**HAL Id: tel-01681499**

**<https://theses.hal.science/tel-01681499v2>**

Submitted on 12 Jan 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.

## THÈSE

Pour obtenir le grade de

### **DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES**

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

**Sara BOUZIT**

Thèse dirigée par **Gaëlle CALVARY, Professeur, Grenoble INP**,  
et codirigée par **Denis CHÊNE, Chercheur senior, Orange Labs**

préparée au sein du **Laboratoire d'Informatique de Grenoble –  
Orange Labs**  
dans l'**École Doctorale Mathématiques, Sciences et  
Technologies de l'Information, Informatique**

### **Plasticité de l'Interaction Homme-Machine : présentation à l'utilisateur, une question de compromis**

Thèse soutenue publiquement le **15 juin 2017**,  
devant le jury composé de :

**Mr Patrick GIRARD**

Professeur, Université de Poitiers (Rapporteur)

**Mr Christophe KOLSKI**

Professeur, Université de Valenciennes (Rapporteur)

**Mr Patrick REIGNIER**

Professeur, Grenoble INP (Président)

**Mr Jean VANDERDONCKT**

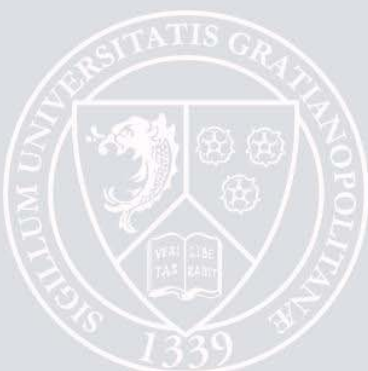
Professeur, Université catholique de Louvain, Belgique (Membre)

**Mr Gilles BAILLY**

Chargé de recherche, Université Pierre & Marie Curie (Membre)

**Mme Audrey SERNA**

Maître de Conférences, INSA de Lyon (Membre)







## Résumé

La thèse s'inscrit dans le domaine de l'ingénierie de l'interaction homme-machine (IHM). Elle s'intéresse à la problématique de l'adaptation des interfaces homme-machine au contexte d'usage. Le contexte est défini par le triplet < Utilisateur, Plate-forme, Environnement >. La question de recherche porte sur la présentation de l'adaptation : comment accompagner le changement pour maximiser le bénéfice de l'adaptation ? L'étude porte sur les petites surfaces d'affichage, telles que celles des téléphones. Ce choix se justifie, d'une part, par le succès des *smartphones* et des tablettes et, d'autre part, par la capacité limitée des petits écrans en termes d'affichage. Le temps de navigation et de recherche de cibles est pénalisé.

Dans l'étude, nous supposons qu'un algorithme d'adaptation peut anticiper les tâches de l'utilisateur par des prédictions. Nous considérons deux cas d'adaptation selon que l'algorithme de prédiction fournit des résultats corrects ou incorrects. Une prédiction est correcte (inversement incorrecte) lorsqu'elle correspond (inversement ne correspond pas) aux besoins de l'utilisateur. Le défi de la thèse est d'explorer de nouvelles techniques d'interaction et d'adaptation pour **accélérer** l'interaction lorsque la prédiction est correcte **sans la pénaliser** lorsque la prédiction est incorrecte.

Les contributions sont doubles : d'une part, un cadre théorique modélisant l'interaction adaptative et identifiant les critères qualité clés ; d'autre part, de nouvelles techniques d'adaptation issues du cadre théorique grâce à son pouvoir non seulement descriptif et comparatif, mais aussi génératif.

Nous proposons ainsi sept techniques : **l'adaptation éphémère avec disparition *In Context*** (1) vs ***Out of Context*** (2) **contrôlée** ou non (3) dissout la fenêtre de propositions au bout d'un certain temps, facilitant ainsi l'accès aux items recherchés en cas de prédiction incorrecte ; (4) le **menu *cloud*** qui permet d'accroître le nombre d'items prédits dans le but d'augmenter la probabilité de présenter des items corrects ; la navigation hiérarchique incarnée en deux techniques d'adaptation ***step-by-step*** (5) et ***shortcut*** (6) permettant de jouer sur le facteur de la distribution temporelle des prédictions ; le **menu *polymodal*** (7) qui présente la prédiction avec différentes modalités (graphique et vocale).

Nous appliquons ces techniques au cas des menus sur smartphone. Les évaluations expérimentales concluent que ces techniques permettent effectivement d'absorber les erreurs de prédiction pour in fine, au mieux, accélérer l'interaction mais sans jamais la pénaliser.

En conclusion, par sa contribution théorique, la thèse dimensionne l'espace d'exploration de l'interaction adaptative, en intégrant les mécanismes d'adaptation à la boucle interactionnelle. Elle positionne également les axes de présentation de l'adaptation et explicite de nouvelles pistes d'exploration. Clairement la présentation de l'adaptation est une affaire de compromis. Des recherches communes entre les communautés Intelligence Artificielle et Interaction Homme-Machine sont une perspective naturelle à cette thèse pour prolonger conjointement cette étude.

**Mots-clés** : Adaptation, plasticité, menu adaptatif, prédiction, variable visuelle, perception, cognition, action, décision, apprentissage, commande, petit écran, compromis.

## Abstract

This thesis contributes to the field of Human Computer Interaction Engineering. It deals with the problem of User Interfaces (UIs) adaptation to the context of use, the context being defined as  $\langle \text{User, Platform, Environment} \rangle$ . The research question is about presentation of adaptation: how to conduct the change for disturbing the user as less as possible, and thereby maximizing the adaptation benefit? The study targets small screen devices like smartphones for two reasons: in one hand, the large success of smartphones and tablets, and in the other hand, the major constraint imposed by small surfaces where all elements cannot be displayed on a single screen. Consequently, the navigation time and target research time increase.

In the thesis, we assume that an adaptation algorithm can anticipate user tasks by predictions. Prediction can be correct or incorrect with regard to real user's needs. The challenge of the thesis is to explore interaction and adaptation techniques suitable to **accelerate** interaction when prediction is correct **without penalizing** it when prediction is incorrect.

Contributions are two-fold: in one hand, a theoretical framework modeling adaptive interaction and identifying key quality criteria; in the other hand, new adaptation techniques derived from the theoretical framework thanks to its generative power in addition to being descriptive and comparative.

We propose four techniques: (1) the **ephemeral adaptation with disappearing *In Context*** (1) vs ***Out of Context*** (2) **controlled** or not (3) which makes the prediction window disappear after a certain delay, considering the prediction probably incorrect; (4) the **cloud menu** that increases the number of predicted items, thereby increasing the probability of displaying what the user is looking for; two variants of hierarchical navigation named **step-by-step** (5) and **shortcut** (6); the **polymodal menu** (7) that presents the prediction with different modalities (graphical and vocal).

We apply these techniques to menus on smartphone. Experimental evaluations show that they accelerate interaction while never penalizing it.

In conclusion, this thesis brings two kinds of contributions: on one hand, a theoretical framework that structures the design space of adaptive interaction; on the other hand, a series

of interaction and adaptation techniques that cover the design space and altogether confirm the importance of being aware of prediction confidence for making the right design decisions. Clearly, presentation of adaptation is a matter of trade off. This calls for joint research with Artificial Intelligence and Human-Computer Interaction.

**Keywords :** Adaptation, plasticity, adaptive menu, prediction, visual variable, perception, cognition, action, decision, learning, command, small screen, trade-off.



# Table des matières

Résumé.....	i
Abstract .....	iii
Dédicaces .....	xi
Remerciements.....	xii
INTRODUCTION .....	14
1. Motivations .....	14
2. Problématique .....	15
3. Objectifs et approche .....	17
4. Organisation du mémoire.....	18
5. Synthèse .....	19
PARTIE I : ETAT DE L'ART EN ADAPTATION D'IHM.....	21
Chapitre 1 : Concepts fondamentaux .....	22
1. Notions liées à l'adaptation.....	22
2. Processus d'adaptation.....	25
3. Conclusion .....	31
4. Synthèse .....	32
Chapitre 2 : Techniques d'adaptation .....	34
1. Techniques spatiales .....	34
2. Techniques graphiques.....	36
3. Techniques temporelles .....	37
4. Cas des petits écrans .....	38
5. Conclusion .....	41
6. Synthèse .....	43
PARTIE II : ESPACE DE CONCEPTION POUR L'INTERACTION ADAPTATIVE.....	45
Chapitre 3 : AIMS, modèle systémique pour la caractérisation de la qualité.....	46
1. Éléments fondateurs.....	47

2.	Modèle AIMS .....	48
3.	Critères de qualité .....	53
4.	Conclusion .....	63
5.	Synthèse .....	64
Chapitre 4 : PCA, modèle de présentation de l'adaptation.....		66
1.	Exigences .....	67
2.	Modèle PCA.....	68
2.1.	Objets de l'interface.....	68
2.2.	Perception .....	69
2.3.	Cognition.....	71
2.4.	Action.....	73
3.	Caractérisation de l'état de l'art.....	75
3.1.	Position .....	75
3.2.	Taille .....	77
3.3.	Couleur.....	78
3.4.	Mouvement .....	79
4.	Comparaison et génération de techniques.....	81
5.	Support logiciel pour le prototypage rapide.....	83
6.	Conclusion .....	86
7.	Synthèse .....	86
PARTIE III : EXPLORATION DE L'ESPACE DE CONCEPTION SUR PETITS ECRANS		88
.....		88
Chapitre 5 : Adaptation éphémère avec disparition, axes Elimination, Ajout et Mouvement..		89
1.	Techniques étudiées .....	91
1.1.	In Context Disappearing (ICD).....	91
1.2.	Out of Context Disappearing (OCD) .....	92
2.	Evaluation expérimentale.....	93
2.1.	Méthodologie .....	94
2.2.	Hypothèses .....	95
2.3.	Mesures qualitatives et quantitatives .....	96

2.4.	Tâche.....	97
2.5.	Matériels .....	97
2.6.	Participants.....	97
2.7.	Procédure .....	98
2.8.	Résultats.....	98
2.9.	Discussion .....	100
3.	Conclusion .....	102
4.	Synthèse .....	102
Chapitre 6 : Adaptation éphémère avec disparition contrôlée, axe Contrôle .....		105
1.	Techniques étudiées .....	107
1.1.	ICD en deux dimensions.....	107
1.2.	ICD avec contrôle utilisateur .....	108
1.3.	Contrôle utilisateur sans disparition.....	109
2.	Evaluation expérimentale.....	109
2.1.	Méthodologie .....	110
2.2.	Hypothèses .....	111
2.3.	Mesures quantitatives et qualitatives .....	113
2.4.	Tâche.....	113
2.5.	Matériels .....	114
2.6.	Participants.....	115
2.7.	Procédures.....	115
2.8.	Résultats.....	115
2.9.	Discussions .....	118
3.	Conclusion .....	120
4.	Synthèse .....	120
Chapitre 7 : Menu cloud, axes Quantité de prédiction et Forme .....		122
1.	Techniques étudiées .....	123
1.1.	Travaux connexes sur les Tag Cloud .....	123
1.2.	Conception du cloud .....	124
2.	Evaluation expérimentale.....	127

2.1.	Méthodologie .....	127
2.2.	Hypothèses .....	127
2.3.	Mesures quantitatives et qualitatives .....	128
2.4.	Tâche.....	129
2.5.	Matériels .....	130
2.6.	Participants.....	130
2.7.	Procédure .....	130
2.8.	Résultats.....	131
2.9.	Discussion.....	133
3.	Conclusion .....	134
4.	Synthèse .....	135
Chapitre 8 : Shortcut et step-by-step, axe Moment.....		137
1.	Techniques étudiées .....	140
1.1.	Travaux connexes sur les menus hiérarchiques .....	140
1.2.	Conception des prototypes.....	144
2.	Evaluation expérimentale.....	147
2.1.	Méthodologie .....	148
2.2.	Hypothèses .....	149
2.3.	Mesures quantitatives et qualitatives .....	150
2.4.	Tâche.....	150
2.5.	Matériels .....	151
2.6.	Participants.....	152
2.7.	Procédure .....	152
2.8.	Résultats.....	153
2.9.	Discussion.....	155
3.	Conclusion .....	156
4.	Synthèse .....	157
Chapitre 9 : Menu polymodal, axe Modalité .....		159
1.	Techniques étudiées .....	161
1.1.	Travaux connexes sur les menus multimodaux .....	161

1.2.	Conception du menu polymodal .....	162
2.	Evaluation expérimentale.....	164
2.1.	Méthodologie .....	165
2.2.	Hypothèses .....	165
2.3.	Mesures quantitatives et qualitatives .....	166
2.4.	Tâche.....	167
2.5.	Matériels .....	167
2.6.	Participants.....	167
2.7.	Procédure .....	167
2.8.	Résultats.....	168
2.9.	Discussion .....	170
3.	Conclusion .....	172
4.	Synthèse .....	173
CONCLUSION ET PERSPECTIVES.....		175
1.	Rappel des contributions.....	175
1.1.	Espace de conception.....	175
1.2.	Exploration de l'espace de conception .....	176
2.	Perspectives.....	179
2.1.	Environnement de conception pour l'exploration intensive .....	179
2.2.	Plasticité pour le meilleur compromis à l'exécution.....	180
Bibliographie.....		i
Liste des tableaux.....		i
Liste des figures .....		ii
Liste des acronymes.....		v
Annexe .....		vi



## **Dédicaces**

*A mes parents, à ma famille et à tous ceux qui me sont chers et proches*

## Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude ainsi que mes sincères remerciements à mes encadrants de thèse Gaëlle Calvary et Denis Chêne, qui ont été de véritables guides durant toute la thèse. Je les remercie pour m'avoir donné la chance d'effectuer cette thèse, pour m'avoir proposé ce sujet si passionnant, pour avoir éclairé mon cheminement de doctorante vers des voies aussi originales que constructives et pour m'avoir appris ce qu'est être un chercheur. Je leur adresse toute ma reconnaissance pour toute la confiance qu'ils m'ont accordé, leur investissement et leurs encouragements qui m'ont permis de mener à bien ce projet.

Je remercie Professeur Jean Vanderdonckt pour son aide et sa collaboration.

Je remercie les membres du jury pour m'avoir fait l'honneur d'accepter de participer à ma soutenance, pour avoir lu mes travaux, de m'avoir écouté et pour m'avoir fait de remarques constructives.

Je tiens ensuite à remercier les équipes TAP et XDLab à Orange Labs Meylan, et l'équipe IIHM au LIG d'avoir mis tous les moyens nécessaires à la réussite de ce partenariat en m'accueillant dans leurs laboratoires et me permettant d'accomplir au mieux mon travail de thèse.

Merci à tous mes amis et à tous mes collègues.

Merci à toutes les personnes qui ont accepté de participer aux différentes expériences utilisateurs.

Enfin, mes remerciements vont à mes parents et à toute ma famille, pour avoir cru en moi et pour m'avoir soutenu tout au long de ces années.

En dernier, mes remerciements aux personnes qui m'ont aidé de près ou de loin à réaliser cette thèse.





# INTRODUCTION

Cette thèse porte sur l'adaptation de l'interaction homme-machine. Nous développons les motivations à ce travail de recherche. Nous posons ensuite la problématique et décrivons le cap de la thèse en termes d'objectifs et d'approche. Nous présentons enfin l'organisation du mémoire et terminons par une synthèse comme ce sera le cas pour tout chapitre du mémoire.

## 1. Motivations

De nos jours, les plates-formes rivalisent de fonctionnalités, dépassant souvent ce dont l'utilisateur a besoin. Le problème est que cette prolifération fonctionnelle s'accompagne souvent d'une complexité d'utilisation (Debevc *et al.*, 1996, Gajos *et al.*, 2006, Horvitz *et al.*, 1998 Linton *et al.*, 2000) : interface visuellement encombrée, difficulté de compréhension, diminution de performance (Carroll, 1984), frustration (Mc Grenere et Moore, 2000). Cette explosion fonctionnelle peut être imputée à la diversité des utilisateurs et par là même à la diversité de leurs besoins (Sischer, 2000). Chaque individu a des compétences motrices et perceptives différentes. Il a également des capacités cognitives telles que la capacité d'apprentissage, les degrés d'intelligence, de résolution de problème, de prise de décision, d'attention qui diffèrent en de multiples points. Les diversités culturelles et les environnements sociaux (milieu professionnel ou privé...) sont d'autres facteurs. Enfin les environnements physiques et les dispositifs d'interaction sont d'autres sources de variation.

Dès lors, concevoir une interface utilisateur qui concilie la grande diversité des individus et de leurs contextes d'usage est une tâche complexe et, en conséquence, coûteuse. Pourtant la qualité est essentielle au risque sinon que ce soit à l'utilisateur de s'adapter au système.

Afin de résoudre cette problématique, une communauté de recherche (Oppermann, 1994, Shneiderman, 1994, Coutaz, 1999, 2006, McGrenere et al., 2002) a proposé d'accroître la flexibilité des interfaces et de fournir aux utilisateurs des interfaces qui s'adaptent individuellement à leurs besoins. L'adaptation concerne des parties ou l'ensemble de cette

interface (Mitchell et Shneiderman, 1989, Bederson, 2000, Kurtev *et al.*, 2002, Tsandilas et Schraefel, 2005, Sendín, 2008).

## 2. Problématique

L'adaptation d'une interface utilisateur doit être réalisée de manière précautionneuse (Hartmann, 2009). Le défi consiste à pousser l'adaptation adéquate au bon moment et au bon endroit pour qu'elle soit bénéfique à l'utilisateur. Sinon l'adaptation pourrait contrevenir aux bénéfices attendus (Lavie et Meyer, 2010). Plusieurs risques sont identifiés :

- **La perturbation cognitive** de l'utilisateur par une mauvaise adaptation (Gardiner et Christie, 1987, Bunt *et al.*, 2004, Lavie et Meyer, 2010). L'utilisateur aura des difficultés à comprendre l'interface dans la mesure où le système fournit rarement des explications à l'utilisateur sur comment ce processus d'adaptation prend place (Lim *et al.*, 2009, Myers *et al.*, 2006, Bunt *et al.*, 2009, 2012) ;
- **La perturbation actionnelle** de l'utilisateur par le manque de participation dans le processus d'adaptation (Norman, 1994, Weld *et al.*, 2003, Jameson, 2003, 2009, Wobbrock *et al.*, 2011) ;
- **La perturbation perceptuelle** de l'utilisateur par le manque de prévisibilité de l'adaptation : l'utilisateur ne sait pas quand et comment l'interface va être changée par le système (Gajos *et al.*, 2008, Findlater *et al.*, 2009) ;
- **La protection de la vie privée** de l'utilisateur dans la mesure où le système stocke des informations caractérisant ce dernier (Bellotti et Sellen, 1993, Holleis *et al.*, 2007).

La difficulté provient de la large diversité des contextes d'usage. Aujourd'hui, avec l'évolution des technologies, l'espace et les objets de la vie courante deviennent de plus en plus des entités d'interaction (Weiser 1993). La combinatoire augmente en conséquence. Or, l'adaptation dépend du type de dispositif et change selon les caractéristiques et les contraintes imposées par chaque dispositif.

Les techniques d'adaptation proposées dans la littérature ont été principalement destinées à des IHM de bureau (Sears et Shneiderman, 1994, Tsandilas et Schraefel, 2005, Ahlström *et al.*, 2010, Matejka *et al.*, 2013). Pourtant, selon (Findlater *et al.*, McGrenere *et al.*, 2007, 2009), les

IHM de petite taille bénéficient des avantages de l'adaptation plus que les IHM de bureau. Le spectre des études sur l'adaptation des IHM des « *smartphones* » est réduit, ce qui représente une opportunité pour l'exploration de nouvelles techniques d'interaction adaptative.

Nous centrons notre étude sur l'adaptation des interfaces de *smartphones*. En particulier, nous nous intéressons à la présentation de l'adaptation dans un contexte de taille d'écran réduite. L'objectif est d'accélérer l'accomplissement de la tâche en facilitant l'accès à la cible par réduction des temps de navigation et de recherche visuelle. le principe est d'anticiper la tâche de l'utilisateur en lui présentant des prédictions d'actions.

Du point de vue de la valorisation, cibler l'étude sur les *smartphones* est pertinent dans la mesure où ils sont des dispositifs multi-usages. Avec un *smartphone*, l'utilisateur peut interagir en tout lieu et à tout instant pour effectuer des actions diverses telles que communiquer, naviguer sur internet ou regarder des films et vidéos.

L'usage des *smartphones* n'est pas toujours simple. Les novices, les personnes âgées, les personnes avec des capacités réduites et parfois même les experts ont des difficultés pour chercher une cible ou atteindre un objectif facilement (Bailly *et al.*, 2008, Roudaut *et al.*, 2009, Francone *et al.*, 2010, Ahlström, 2010, Shin *et al.*, 2012). Malgré les différentes conceptions et optimisations faites pour améliorer ces dispositifs, ils souffrent encore de nombreux problèmes. L'usage en situation de mobilité n'est pas facile. Lorsque l'attention est partagée entre le déplacement et la manipulation du terminal, l'utilisateur doit se concentrer pour être précis et ainsi éviter les erreurs de pointage. De même, les variations de luminosité dans l'environnement entraînent des difficultés de lisibilité. La multiplicité des tâches pouvant être effectuées avec ces dispositifs varient elles-mêmes considérablement en fonction des horaires, des lieux, des historiques de chaque individu. Mais de façon générale, les interfaces de *smartphones* prennent peu en compte le changement de situations d'usage. Certes, la luminance de l'écran va changer en fonction des capteurs lumineux ; l'écran va s'éteindre lorsqu'il est porté à l'oreille lors d'un appel ; il passera en mode horizontal lorsqu'il sera tourné. Mais ces ajustements restent mineurs par rapport à l'ensemble des besoins d'adaptation. Par exemple, lorsque l'utilisateur est en situation de conduite de voiture, l'interface pourrait s'adapter à cette situation et de nouvelles modalités pourraient lui être proposées. Lorsqu'il entre dans des locaux, le GPS pourrait s'éteindre afin d'économiser la

batterie ; lorsqu'il est en position statique depuis un certain temps, des fonctionnalités de réunion pourraient automatiquement se mettre en place et se désactiver lorsque le terminal se déplace à nouveau ; ou encore, lorsque l'utilisateur a l'habitude d'effectuer telle ou telle action, à telle heure ou à tel jour, l'interface pourrait alors lui faciliter la tâche. Le défi est de faciliter l'usage.

Dans le reste de ce mémoire, par prédiction nous entendons **toute information ou action inférée par le système comme étant pertinente pour l'utilisateur dans son contexte d'usage.**

Lorsque l'algorithme d'adaptation fournit de bonnes prédictions, c'est-à-dire qui correspondent à ce que l'utilisateur cherche, l'interaction peut être accélérée et le nombre d'erreurs réduit. Le problème vient du fait que les prédictions fournies par les algorithmes d'adaptation sont rarement certaines à 100% En conséquence, lorsque les prédictions sont incorrectes, l'interaction risque d'être ralentie ou entachée d'erreurs.

### **3. Objectifs et approche**

Nous supposons qu'il est possible de prédire les applications et fonctionnalités d'intérêt pour un utilisateur dans une situation donnée. En nous basant sur cette hypothèse, nous considérons deux cas de prédiction selon qu'elle est correcte ou incorrecte. L'objectif est d'explorer de nouvelles techniques d'adaptation permettant de présenter la prédiction dans un **contexte contraint** (sur *smartphone*). **Notre défi est d'accélérer l'interaction de l'utilisateur lorsque la prédiction est correcte, sans la pénaliser lorsque la prédiction est incorrecte tout en minimisant le nombre d'erreurs.**

Notre approche consiste à :

1. Définir un modèle qui décrit le phénomène d'interaction adaptative ;
2. Définir les propriétés de qualité à respecter et à prendre en compte pour l'adaptation ;
3. Définir les aspects de présentation de l'adaptation tout en respectant les propriétés de qualité, et fournir les outils pour explorer de nouvelles techniques d'adaptation.

## 4. Organisation du mémoire

Le manuscrit est organisé en trois parties. La première est relative à l'état de l'art ; la seconde aux contributions théoriques ; et la troisième aux contributions expérimentales.

La partie I « État de l'art en adaptation d'IHM », se structure en deux chapitres.

Le chapitre 1 vise à définir les concepts fondamentaux de l'adaptation. Nous présentons un état de l'art sur les différents espaces de conception de l'adaptation ainsi que les propriétés essentielles dans la conception des interfaces adaptatives.

Le chapitre 2 est dédié à la présentation des techniques d'interaction adaptatives.

La partie II « Espace de conception pour l'interaction adaptative » est structurée en deux chapitres.

Le chapitre 3 présente le modèle d'interaction/adaptation AIMS. Nous définissons le modèle pour chaque entité (utilisateur et système). Nous exprimons les propriétés de qualité pour l'interaction adaptative au sein du modèle et au niveau de chaque entité.

Dans le chapitre 4 nous présentons un modèle pour la présentation de l'adaptation.

La partie III « Exploration de l'espace de conception sur petits écrans » est structurée en cinq chapitres.

Le chapitre 5 discute les contraintes imposées par les dispositifs de petite taille. Ensuite, nous présentons le concept d'adaptation éphémère avec disparition sur *smartphone*. Nous présentons ensuite l'évaluation de ce concept.

Le chapitre 6 présente le concept de l'adaptation éphémère avec disparition contrôlée par l'utilisateur. Nous présentons l'expérimentation menée pour tester ce concept et les résultats obtenus.

Le chapitre 7 présente le concept des menus Cloud. Nous présentons en particulier une étude expérimentale ainsi que les résultats de l'évaluation des menus Cloud.

Le chapitre 8 traite la problématique de complexité de navigation dans des arborescences sur des dispositifs de petite taille. Nous rapportons l'expérimentation menée dans ce contexte ainsi que les résultats obtenus.

Le chapitre 9 se consacre aux menus polymodaux. Nous évaluons expérimentalement différentes modalités de rendu et de commande de l'adaptation et nous interprétons les résultats obtenus.

Enfin, nous concluons le travail mené dans cette thèse et présentons des perspectives à ces travaux de recherche.

## **5. Synthèse**

Cette thèse étudie les IHM graphiques sur petits écrans. L'ensemble des éléments de l'interface ne pouvant être affiché sur un seul écran, les temps de navigation et de recherche visuelle s'en trouvent pénalisés. La solution explorée est l'adaptation de l'IHM par anticipation. Le défi est d'accélérer l'interaction lorsque la prédiction est correcte mais sans la ralentir lorsqu'elle est incorrecte.





## **PARTIE I : ETAT DE L'ART EN ADAPTATION D'IHM**

Par une analyse de l'état de l'art, nous montrons que les erreurs de prédiction sont insuffisamment considérées du point de vue de l'interaction homme-machine tant en terme d'espaces de conception (Chapitre 1) qu'en terme de techniques d'interaction (Chapitre 2).

# Chapitre 1 : Concepts fondamentaux

Ce chapitre définit les concepts fondamentaux liés à l'adaptation des interfaces homme-machine. Nous définissons l'adaptation, expliquons comment elle fonctionne et peut être conçue.

La section 1 est dédiée à la définition des différents termes liés à l'adaptation. Dans la section 2, nous analysons les différents cycles et processus de conception d'adaptation.

## 1. Notions liées à l'adaptation

L'adaptation n'est pas un phénomène nouveau. L'Intelligence Artificielle (IA) s'est la première intéressée à l'adaptation des systèmes (Winograd, 2006, Grudin, 2009, Kolski *et al.*, 2014). Depuis trois décennies, un champ de recherche très large est dédié à l'étude de l'adaptation des systèmes interactifs adaptatifs (Mackinlay, 1986, Tendjaoui *et al.*, 1991, Schneider-Hufschmidt *et al.*, 1993, Yu, 1997, Mark, 1998, Hook, 2000, McKinley, *et al.*, 2004).

M. Beaudouin Lafon (1997) définit un système interactif comme une application informatique capable, au cours de son exécution, de prendre en compte des informations communiquées par le ou les utilisateurs du système, et qui produit au cours de son exécution une représentation perceptible de son état interne.

Norcico et Stanley indiquent qu'un système informatique idéal est un système qui devrait être capable de s'adapter à son utilisateur en compensant les faiblesses, en fournissant une aide appropriée au contexte et en diminuant la charge de travail mentale et physique d'un utilisateur particulier. Cette définition ajoute à celle de M. Beaudouin Lafon la notion d'adaptation et d'assistance. Les « informations communiquées par l'utilisateur » doivent non seulement lui permettre d'interagir avec le système interactif mais aussi permettre en retour de lui faciliter l'accomplissement de sa tâche. Il doit donc être adaptable.

L'**adaptabilité** est la capacité du système à s'adapter aux personnalisations demandées par l'utilisateur. En d'autres termes, l'utilisateur change (avec parfois l'aide du système) les

fonctionnalités du système (Benyon et Murray, 1993, Moisuc et Fischer, Kobsa *et al.*, 1993, 2001). Trigg *et al.*, (1987) ont défini trois façons d'exposer une adaptabilité système :

- Un système **flexible** est capable de proposer des objets génériques et des comportements qui peuvent être interprétés et utilisés différemment par différents utilisateurs au cours de différentes tâches.
- Un système **paramétrable** (*parameterised*) offre un éventail de comportements alternatifs parmi lesquels les utilisateurs peuvent choisir.
- Un système **intégrable** (*integratable*) est interfacé et intégré avec d'autres dispositifs au sein de son environnement ou bien de manière distante.

La **personnalisation** est une forme d'adaptation (Greenberg et Witten, 1985, Cremene *et al.*, 2004) définie par McBurney *et al.*, (2007) comme une opération d'adaptation du comportement et des fonctionnalités du système. L'objectif est que le système réagisse de manière adaptée au contexte utilisateur ainsi qu'aux préférences de l'utilisateur.

Papadopoulou *et al.*, (2008) définissent la personnalisation comme un processus destiné à créer, maintenir, et appliquer les préférences utilisateur pour la prise de décision. La personnalisation doit se faire de manière dynamique en accord avec l'évolution des préférences utilisateur et selon l'état de son environnement.

Dans la littérature, les deux termes **adaptation** et **adaptativité** sont très souvent employés comme synonymes. Un système adaptatif (*adaptive*) est un système capable de modifier de manière automatique ses caractéristiques en fonction des besoins de l'utilisateur (Browne *et al.* 1986).

Selon Moisuc et Fischer, l'adaptativité désigne la capacité du système à répondre aux besoins de l'utilisateur sans que ce dernier n'intervienne explicitement. C'est une adaptation par le système à la tâche en cours et à l'utilisateur.

Stephanidis *et al.* (1998) définissent l'adaptativité par le fait que les adaptations se produisent en cours d'exécution du système. Les auteurs ne tiennent pas compte du fait que ces adaptations sont réalisées par le système ou par l'utilisateur (elles se produisent de manière dynamique).

La **plasticité** est une forme d'adaptation introduite en (1999) par Thevenin et Coutaz. Une interface est dite plastique lorsqu'elle est capable de s'adapter à son contexte d'utilisation dans

le respect de sa qualité pour l'utilisateur. Le contexte est défini comme le triplet « Utilisateur, Plate-forme, Environnement ».

Pour résumer, une interface adaptative est une interface ayant la capacité de s'adapter avec ou sans l'intervention de l'utilisateur.

Mais l'objet même de l'adaptation est à considérer. La variabilité des utilisateurs, des environnements, des dispositifs eux-mêmes génère de nombreux **contextes d'usage** (Elrod, Hall *et al.* (1993), Adams *et al.* (1994), Cooperstock, Tanikoshi *et al.*, (1995), Salber (1996), Fickas, Kortuem *et al.*, 1997, Hull, Neaves *et al.*, 1997, Kortuem, Segall *et al.*, 1998, Ayatsuka *et al.*, (1998), Abowd *et al.*, (1998)). Dans le milieu des années quatre-vingt-dix, ce terme fait référence à un ensemble de conditions d'utilisation perçues par le système qui génère en réaction des informations et des services adaptés (Cheverset *et al.*, (2002), Chaari *et al.*, (2004, 2009), Dey (2001), Ryan *et al.*, (1998), Brown *et al.*, 1986, Schilit *et al.*, 1994).

Rey *et al.*, (2002), ont donné une définition du contexte en se basant sur la notion temporelle. Ils définissent un contexte d'interaction comme étant un instant  $t$  qui est la composition des situations (circonstances qui entourent l'action) entre les instants  $t_0$  et  $t$  lors de la réalisation de l'activité (A) par l'utilisateur (U). Le contexte d'usage évolue dans le temps et est à la croisée de l'activité et des capacités de l'utilisateur.

Calvary *et al.*, (2003) Samaan *et al.*, (2004), Vanderdonckt *et al.*, (2005) décrivent le contexte par le triplet <Utilisateur, Plate-forme, Environnement>. Ces trois dimensions couvrent l'ensemble des informations périphériques à la tâche en cours et susceptibles de l'influencer :

- L'utilisateur est décrit par ses capacités perceptuelles, motrices et cognitives. Il est également décrit par ses préférences et ses caractéristiques culturelles comme sa langue naturelle ;
- La plate-forme ou dispositif d'interaction se décompose en deux catégories : physique représentant le matériel et logique représentant le logiciel ;
- L'environnement est l'environnement physique dans lequel se passe l'interaction. Prenons l'exemple de la luminosité, le bruit, la localisation géographique.

Ce triplet recouvre bien les différents niveaux de contexte, qu'ils soient propres à l'individu, au dispositif qu'il utilise, ou bien au milieu dans lequel il l'utilise. Mais contrairement à Rey *et al.*, la dimension temporelle n'y est pas soulignée. Or l'exécution de l'activité (A) est un

phénomène qui s'étale dans le temps. L'unité temporelle pouvant d'ailleurs s'étaler de la micro-seconde (feedbacks lors d'une manipulation) à des comportements récurrents mensuels, voire annuels. Cette question temporelle est cruciale dans un cadre adaptatif car pour s'ajuster à l'activité de l'utilisateur il est nécessaire **d'anticiper et de prédire** les besoins de ce dernier selon le **contexte d'usage**. En d'autres termes, le système est capable de décider, d'initier une adaptation (s'auto-adapter), voire de s'adapter à la demande directe de l'utilisateur en fonction de ses capacités prédictives sur ledit comportement.

La qualité de l'adaptation dépendra ainsi de la qualité des prédictions fournies par l'algorithme d'adaptation. Ces prédictions ne sont pas toujours correctes exposant, en conséquence, l'adaptation à être inappropriée. Cela peut entraîner des conséquences négatives sur l'interaction. Par exemple, l'interface peut être surchargée visuellement par des informations inutiles, risquant de ralentir ce dernier, ou bien le conduire sur de fausses pistes d'actions. Les conséquences négatives d'une adaptation inappropriée doivent être prises en compte dès la phase de conception afin d'éviter de telles conséquences.

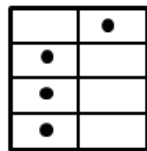
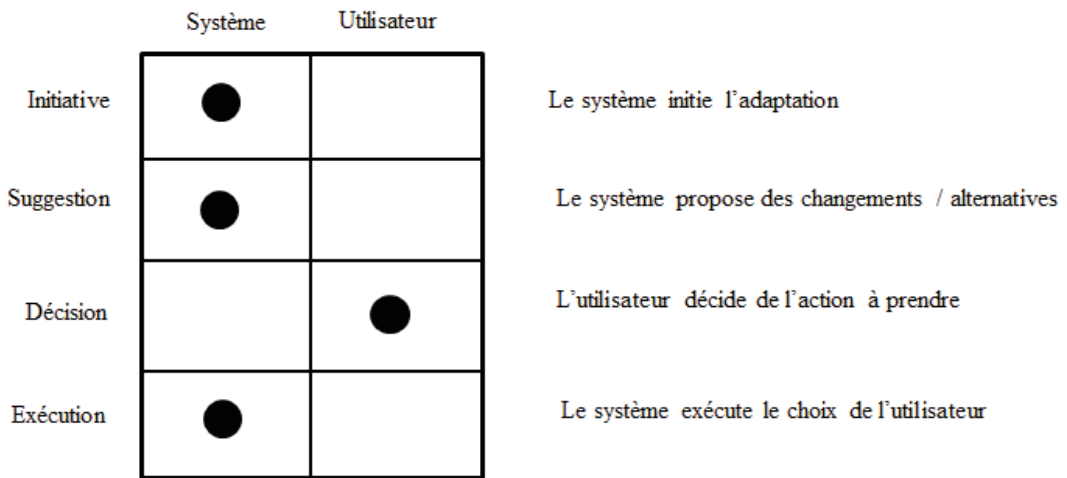
## 2. Processus d'adaptation

L'adaptation est un processus basé principalement sur les connaissances de l'utilisateur, le schéma d'interaction, la tâche et le système (Norcio et Stanley, 1989). Pour qu'une adaptation puisse se mettre en œuvre, plusieurs étapes sont nécessaires.

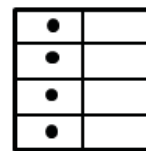
Dieterich *et al.*, ont défini quatre étapes principales dans les processus d'adaptation, ces étapes pouvant être prises en charge par le système et/ou l'utilisateur :

- l'**initiative** où la décision de proposer une adaptation est prise ;
- la **suggestion** où des propositions d'adaptation sont présentées ;
- la **décision** où une des alternatives est retenue ;
- l'**exécution** qui consiste à mettre en œuvre l'adaptation sélectionnée.

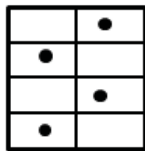
La figure 1 décrit ces quatre principales étapes permettant la mise en œuvre de l'adaptation.



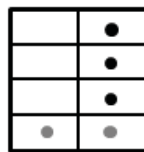
Initiative utilisateur et auto-adaptation système



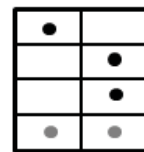
Auto-adaptation totale du système



Adaptation assistée par le système



Adaptation manuelle par l'utilisateur



Initiative système et adaptation utilisateur

Figure 1. Une configuration d'adaptation selon Dieterich.

Ces quatre étapes permettent de séquencer le processus adaptatif. Chaque étape peut être réalisée avec plus ou moins de succès et, bien entendu, un écueil à l'une ou l'autre des phases peut venir contrefaire les bénéfices attendus. Le modèle de conception de Dieterich souligne des étapes importantes dans la conception de l'adaptation. Toutefois la (non) qualité des suggestions n'est pas considérée.

En 1986, Norman a proposé une théorie d'action, hors processus adaptatif, basée sur le cycle Exécution – Évaluation présenté sur la figure 2. Ce cycle est composé de sept étapes essentielles. Dans la phase d'exécution, l'utilisateur commence par établir un objectif et

former une intention. Ensuite, l'utilisateur spécifie puis exécute une séquence d'actions. Dans la phase d'évaluation, l'utilisateur perçoit l'état du système (Perception). Il interprète l'état du système perçu et enfin évalue cet état système par rapport à ses objectifs et intentions. Cependant, le modèle de Norman ne décrit que la phase d'interaction du point de vue de l'utilisateur. Comment le système est capable de percevoir le contexte d'usage et d'initier un dialogue interactionnel ne sont pas décrits.

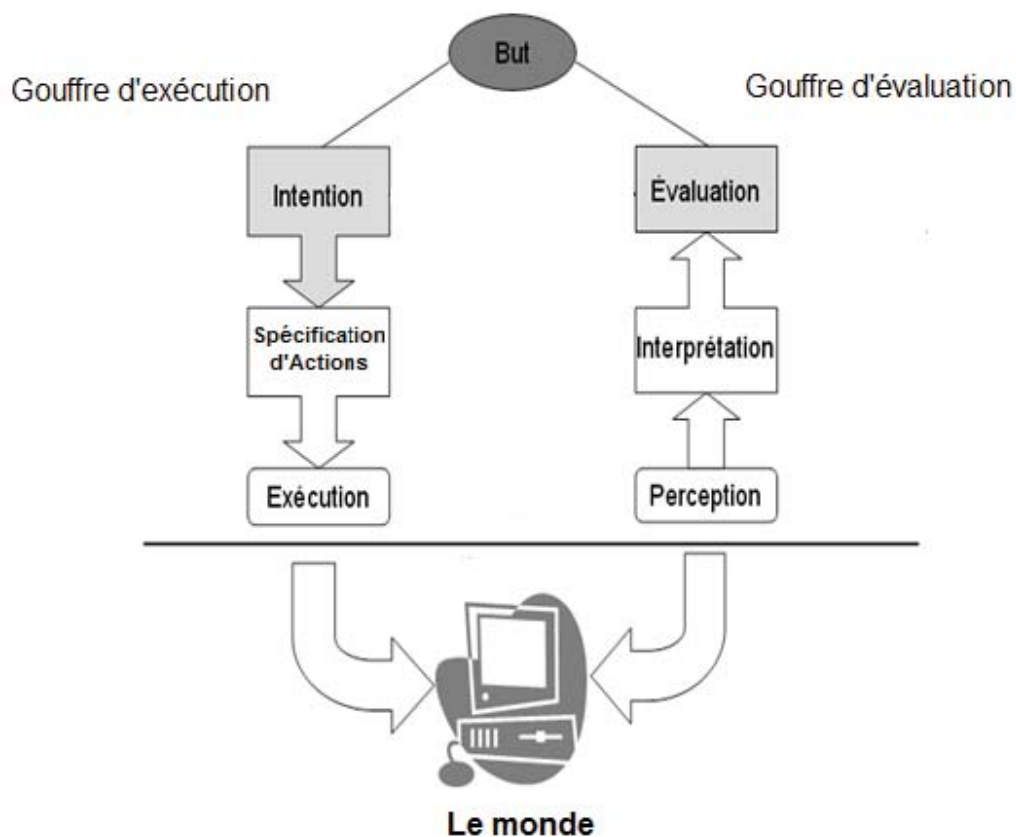


Figure 2. Étapes d'exécution et d'évaluation selon Norman.

En étendant la taxonomie de Dieterich et en spécialisant la théorie de l'action de Norman, (Lopez-Jaquero *et al.*, 2007) ont proposé le framework ISATINE pour l'adaptation des IHM. Ce cadre structurant décompose le cycle de vie de l'adaptation en sept étapes selon le modèle de Norman, identifiant ainsi : le gouffre d'exécution de l'adaptation et le gouffre d'évaluation de l'adaptation. Les différentes étapes peuvent être réalisées par l'utilisateur, le système et/ou toute tierce partie telle qu'un modérateur. Le framework ISATINE est décrit sur la figure 3.

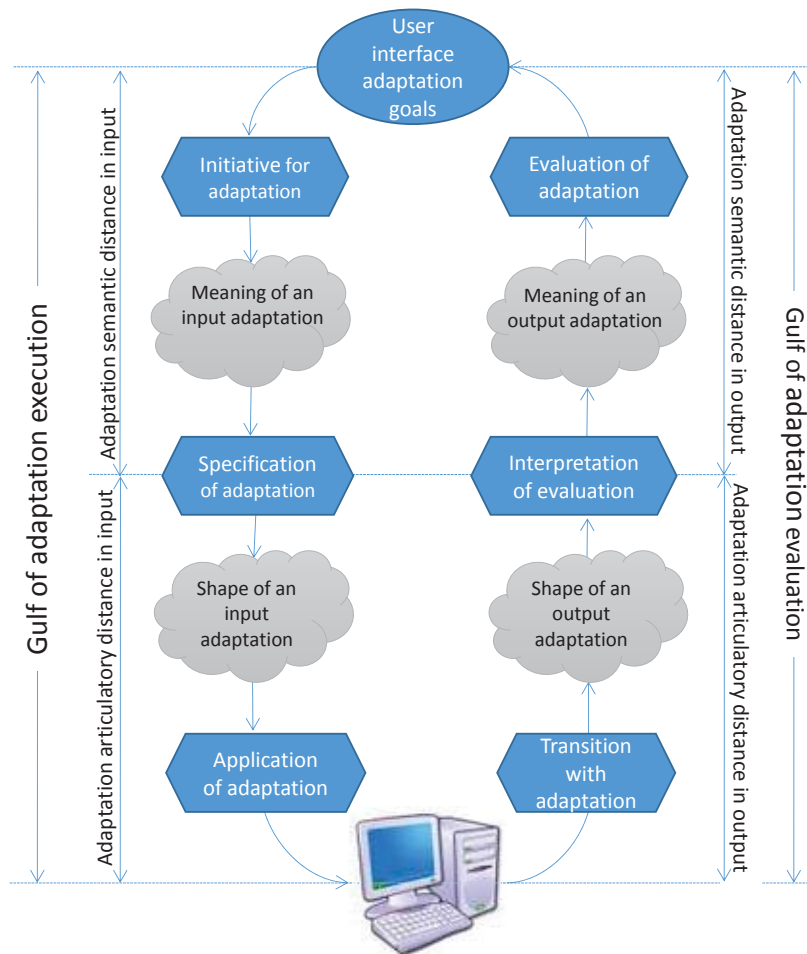


Figure 3. *Framework ISATINE pour l'adaptation des IHM (extrait de Lopez-Jaquero et al., 2007).*

Le framework ISATINE permet de décrire les potentiels écueils de l'adaptation, à savoir les gouffres d'évaluation et d'exécution de l'adaptation. Cependant, bien qu'inspiré d'un modèle d'interaction, son articulation fine avec le processus d'interaction n'est pas évoquée et notamment, par exemple, les aspects concernant le contrôle utilisateur et l'impact de mauvaises prédictions.

Oppermann (1994) souligne l'importance du contrôle donné à l'utilisateur en ce qui concerne le temps et le contenu de l'adaptation. Il a proposé un prototype combinant l'adaptativité et l'adaptabilité. Oppermann affirme que l'utilisation conjointe de l'adaptation automatique et de l'adaptation contrôlée par l'utilisateur est plus prometteuse qu'un usage alternatif de ces



éléments. Selon cette étude, l'adaptation devrait être effectuée en considérant à la fois l'introduction de sa raison d'être à l'utilisateur (par un tutoriel par exemple); la définition des différentes possibilités d'adaptation ; la présentation d'une vue d'ensemble des adaptations effectuées ; et la possibilité de modifications ultérieures des adaptations conclues.

En 2000, Oppermann, comme Lorenz, a divisé le processus d'adaptation en trois étapes :

- Afférence (*Afferential*) pour la collecte des informations de contexte ;
- Inférence (*Inferential*) pour le traitement et l'inférence des informations ;
- Efférence pour la mise en œuvre de l'adaptation.

À l'inverse de Dieterich ces étapes sont davantage centrées sur le point de vue système. La collecte, le traitement et la mise en œuvre relèvent d'un découpage générique mais ne permettent pas de qualifier précisément ce qui est à l'initiative de l'adaptation (l'utilisateur ou bien le système), ni même la qualité des suggestions proposées, ni même les critères décisionnels permettant de conserver une ou plusieurs suggestions, que ce soit par l'utilisateur ou bien par le système. Nous pensons que cette articulation système-utilisateur et que la balance des bénéfices attendus vis-à-vis des gênes occasionnées sont autant de points cruciaux à prendre en compte dans la description des phénomènes adaptatifs.

Le type d'élément adapté entre également en compte. Dès 1996, Brusilovsky, traite des hypermédias adaptatifs. En 2001 il présente une taxonomie qui classe ces systèmes selon les types de présentations (Kobsa *et al.*, 2001) qu'elles soient multimédia, ou faites de texte ou encore présentées selon différentes modalités. Concernant la navigation, il distingue le guidage direct, le tri de lien, le masquage de lien, l'annotation de lien et la génération de lien. Cette taxonomie supporte la classification de certains systèmes adaptatifs, mais des extensions sont nécessaires pour couvrir les systèmes adaptatifs actuels.

De ce fait, en 2006, Hanish *et al.*, ont étendu la taxonomie proposée par (Brusilovsky, 2001). Alors que les technologies web évoluent (Gómez et Tran, 2009, Rosaci *et al.*, 2006), non seulement des fragments de texte ou des liens peuvent être adaptés, mais aussi des clips vidéo, des graphiques 3D, etc. Les techniques et méthodes d'adaptation ont été étendues et également classées par type de composant. La taxonomie de Brusilovsky a été augmentée avec les types

de composants suivants : modèles, vues, contrôleurs, widgets, éléments, graphiques, clips, scripts et stratégies. Mais tout cela reste spécifique aux technologies web.

Décrire les processus de l'adaptation permet de qualifier ce que pourrait être un espace de conception d'une interface adaptative. Nous avons identifié que le but de l'adaptation pouvait concerner plusieurs facteurs de diversité (utilisateur, environnement, contexte de la tâche...), selon plusieurs échelles temporelles (de la micro-seconde à l'année), et qu'une adaptation pouvait être plus ou moins bien mise en œuvre en fonction de suggestions et décisions plus ou moins appropriées.

Arhippainen (2009) définit un espace de conception pour l'adaptation en fonction du triplet : cible, moyens et temps. Le temps distingue une adaptation statique ou dynamique, sans aller aussi loin que ce que nous avons décrit ci-avant en termes d'échelles temporelles. La cible précise le bénéficiaire de l'adaptation, par exemple, l'utilisateur, l'environnement ou la plateforme. Mais il est discutable d'attribuer les bénéfices de l'adaptation à d'autre que l'utilisateur ? Par exemple, l'adaptation à une trop forte luminosité ambiante est mise en œuvre pour le bien de l'utilisateur, pas d'une variation de l'environnement. Le critère environnemental est utilisé certes, mais pour une adaptation au bénéfice de l'utilisateur. Nous voyons que c'est non pas la cible proprement dite qui est utilisée dans cette adaptation, mais la cible au sein d'une boucle interactionnelle entre le système et l'utilisateur. Nous reviendrons sur ce point lors de la description de notre modèle. Les moyens réfèrent aux composants logiciels concernés tels que les modèles de tâche ou des techniques de rendu.

Cockton (1987) souligne que ces moyens de mise en œuvre de l'adaptation doivent être rendus souples dès le départ de la conception. Il propose le concept de paramétrisation de l'interface. où tout élément de l'interface doit pouvoir être paramétré. Il propose de préparer plusieurs chemins d'interaction au sein des boîtes de dialogue. Différentes voies peuvent ainsi être choisies en activant ou désactivant les différentes options qui mènent à ces chemins, en fonction des changements de contexte. Ainsi, le moteur d'adaptation passe d'une voie à une autre selon le contexte. Deux étapes du processus d'adaptation sont distinguées : la première est celle du diagnostic des besoins et des compétences de l'utilisateur en fonction des données enregistrées. C'est la phase de collecte des informations, de proposition de suggestions et de décision de les mettre en œuvre. La deuxième consiste à la mise en œuvre proprement dite.

Ces travaux sont une avancée cruciale sur l'opérationnalisation de ce qu'est effectivement l'adaptation. Nous retrouvons ici les éléments clés de Dieterich.

(Vanderdonckt *et al.*, 2005) ont proposé un espace de conception de l'adaptation qui prend en compte les questions suivantes : Qui, Pour qui, Quoi, Quand et Au regard de quoi ? Cet espace de conception permet de décrire et de prendre en compte les différents axes de l'adaptation.

L'intérêt de ce modèle est qu'il récapitule l'ensemble des éléments qui interviennent lors d'une adaptation. Le Qui précise qui du système ou de l'utilisateur est à l'initiative de l'adaptation ; le Pour qui précise le bénéficiaire (même comme vu précédemment, il nous paraît difficile de considérer un autre bénéficiaire que l'utilisateur), le Quoi précise l'objet de l'adaptation, fenêtre, fonction ou autre objet d'interface ; le Quand précise la dimension temporelle ou le moment pour qu'une adaptation soit mise en œuvre ; et le Au regard de quoi précise les types de variations de contexte nécessitant une adaptation (tâche, utilisateur, dispositif, etc.).

La critique qui peut en être faite est qu'il s'agit d'un modèle très générique et que la dimension précisant la façon dont l'adaptation va être mise en œuvre le (Comment) reste manquante. Notons à ce sujet que le (Comment) a été traité dans Kolski *et al.* (1992), dans le cadre d'une approche prenant en compte le (Quoi), le (Quand) et le (Comment). Par contre le (Qui) n'y a pas été traité, et le (Au regard de quoi) a été intégré dans le (Comment) sous forme de justifications.

### **3. Conclusion**

Dans ce chapitre, nous avons présenté un état de l'art sur les notions liées à l'adaptation et au contexte d'usage. Nous retenons comme définition d'un système adaptatif : un système capable, selon une échelle temporelle et un périmètre de contextes plus ou moins grands, de capturer le contexte d'usage, d'anticiper et de prédire les besoins de l'utilisateur, puis de changer l'interface afin de répondre à ces besoins, avec ou sans l'intervention de l'utilisateur. Le contexte d'usage est défini comme le triplet <Utilisateur, Plate-forme, Environnement>.

Nous avons également présenté un état de l'art sur les espaces de conception de l'adaptation. L'analyse de l'état de l'art démontre l'intérêt porté à l'adaptation des interfaces utilisateur depuis des décennies. Cependant, l'adaptation ne se limite pas au périmètre du système :

l'utilisateur s'adapte également, évolue et change sa stratégie d'interaction. Ces facteurs de coévolution sont peu traités et militent pour l'étude d'un système plus large incluant l'humain.

Enfin, les espaces de conception supposent tous implicitement que les suggestions d'adaptation sont correctes ainsi que leur présentation pertinente, ce qui n'est pas forcément le cas.

#### **4. Synthèse**

Le spectre des études sur l'adaptation des IHM est large. Néanmoins, la coévolution entre l'utilisateur et le système est peu considérée. Pourtant, c'est lors d'un processus interactif homme-machine que l'adaptation se met en œuvre. De fait, ce n'est pas uniquement le système qui nécessite d'être adapté, mais bien la situation d'interaction. Par ailleurs, les adaptations système peuvent ne pas être pertinentes et devenir un frein à l'interaction. Ces points de couplage interaction-adaptation sont peu considérés. Ils sont au cœur de cette thèse.



## Chapitre 2 : Techniques d'adaptation

L'adaptation des interfaces homme-machine pose plusieurs questions fondamentales (Totterdell *et al.*, 1990, Rekimoto *et al.*, 1995, Norman, 1996, Langley, 1999, Bérard *et al.*, 2000, Beaudouin-Lafon, 2000, Calvary *et al.*, 2001, Constantine, 2001, Coutaz, 2007, Weibelzahl, 2002, Pignotti *et al.*, 2004, Limbourg *et al.*, 2005, Quan et Jianmin, 2006). Les premiers travaux depuis les années 90 furent principalement axés sur les questions : Quoi, Quand, Pourquoi et Comment adapter (Karagiannidi *et al.*, 1996, « *Adaptive User Interfaces* » de (Browne *et al.*, 1990 et « *Adaptive User Interfaces : Principles and Practice* » de Schneider-Hufschmidt *et al.*, 1993). Dans ce chapitre nous nous intéressons à comment présenter l'adaptation.

Les techniques d'adaptation existantes peuvent être classées en trois catégories principales : spatiales, graphiques et temporelles. Nous passons en revue ces techniques et soulignons leur impact sur l'interaction lorsque la prédiction est correcte versus incorrecte. Les sections 1, 2 et 3 traitent respectivement des techniques d'adaptation spatiales, graphiques et temporelles. La section 4 s'intéresse aux cas des petits écrans.

### 1. Techniques spatiales

Les techniques d'adaptation spatiales consistent à changer l'ordre et la position spatiale des éléments dans l'interface. Parmi les techniques d'adaptation les plus utilisées et qui ont reçu l'attention d'un grand nombre de travaux de recherche, on trouve le menu *split* (adaptatif) proposé par Sears and Shneiderman (1994). Le menu *split* est une interface divisée en deux parties. Les items les plus fréquemment utilisés par l'utilisateur sont déplacés dans la partie supérieure du menu au-dessus d'une ligne de séparation. La partie inférieure contient les items non prédits. Une évaluation avec 38 participants a montré que les menus *split* sont plus rapides que les menus statiques traditionnels. Cependant, les menus *split* ont été critiqués principalement à cause de leur instabilité spatiale. La position des items change fréquemment ce qui ne permet pas à l'utilisateur de créer un modèle mémoire de l'interface et impacte les

performances, en particulier, lorsque la prédiction ne correspond pas à ce que l'utilisateur cherche.

Le principe des menus *split* a été amélioré par le menu *split* avec réplique. Dans cette interface, les items prédits sont affichés dans la partie supérieure adaptative mais sont conservés dans la partie inférieure. Findlater et McGrenere, 2004, ont effectué une étude expérimentale avec 27 participants dans le but de comparer le menu *split* aux menus statique et *split* avec réplique. Les résultats ont montré que lorsque le menu *split* apparaît en premier dans l'ordre de présentation des conditions expérimentales, les utilisateurs sont moins rapides qu'avec le menu statique et le menu *split* répliqué.

Dans une étude avec 16 participants menée par Gajos *et al.*, 2004, les auteurs ont montré que les utilisateurs avaient une forte préférence pour le menu *split* avec réplique, en comparaison du menu statique : les utilisateurs préfèrent avoir des items dupliqués plutôt que des items déplacés dans la section adaptative de l'interface. Cela augmente globalement la stabilité de l'interface et laisse l'utilisateur libre d'utiliser ou non l'adaptation.

Les menus de Microsoft Office dans les versions 2000 jusqu'à XP sont un bon exemple d'interfaces adaptatives commerciales. Une grande majorité des gens se sont familiarisés avec ces menus adaptatifs. À l'ouverture du menu, le *Smart* menu offre à l'utilisateur une petite version du menu complet qui contient uniquement les items que le système détecte comme les plus fréquemment ou récemment utilisés. Pour accéder à un item qui n'est pas présent dans ce petit menu, l'utilisateur doit étendre le menu en cliquant sur une petite flèche en bas pour afficher le contenu complet. Lorsque le menu est étendu, la position des items n'est pas maintenue. Les nouveaux items visibles sont intercalés avec les anciens. Les utilisateurs doivent alors ré-analyser le menu entier pour trouver l'item désiré, plutôt que de simplement analyser l'ensemble des nouveaux éléments visibles. Le *Smart* menu peut être considéré comme un menu *split* qui est réparti temporellement.

Une évaluation expérimentale réalisée par Fischer et Schwan en (2008) comparant le *Smart* menu et le menu statique a montré que le *Smart* menu est moins efficace que le menu statique traditionnel.

## 2. Techniques graphiques

Les approches d'adaptation dites graphiques consistent à introduire des effets visuels afin de mettre en évidence les éléments prédits. Contrairement aux techniques spatiales, les éléments ne changent pas de position. Cette stabilité spatiale permet aux utilisateurs de s'appuyer sur leur mémoire.

Comme exemples de techniques d'adaptation graphiques, nous pouvons citer le menu *morphing* proposé par Cockburn *et al.*, 2007. Ce menu consiste à augmenter dynamiquement la taille des items qui sont fréquemment utilisés ou sélectionnés par l'utilisateur.

Dans ce même cadre, nous trouvons les interfaces dites « *walking UI* » (Shaun *et al.*, 2008). Ce type d'interface consiste à augmenter la taille des éléments de l'interface afin de faciliter la lecture et la manipulation en situation de marche.

Les menus *fish-eye* proposés par Bederson, 2000, consistent à afficher les items avec une taille de police qui augmente ou diminue en fonction de la distance de la position du curseur : plus le curseur est proche de l'élément, plus la taille de ce dernier est grande. Cependant, cette technique nécessite un haut niveau de précision pour sélectionner la cible.

Le système SUPPLE proposé par Gajos *et al.*, (2004) permet d'ajuster la disposition, la taille et la structure des éléments dans l'interface. L'objectif est de s'adapter à la fois au dispositif utilisé et aux caractéristiques et habitudes de l'utilisateur. Les utilisateurs peuvent indiquer explicitement leurs préférences mais le système peut aussi détecter automatiquement leurs capacités motrices (Grossman *et al.* 2007) et s'ajuster automatiquement. Les objets de l'interface changent automatiquement, par exemple : une liste déroulante devient une barre de réglage avec deux boutons (+/-). La première évaluation du système montre que le système produit des variantes raisonnables par rapport à celles créées par des concepteurs expérimentés (Gajos *et al.*, 2004, 2005), (Weld *et al.*, 2003). Notons que dans ce système il est clairement assumé que les techniques de présentations sont étroitement liées aux capacités de manipulation.

La technique dite *highlighting* a été étudiée par plusieurs chercheurs (Gajos *et al.*, 2004, 2006, Tsandilas et Schraefel, 2005). Elle consiste à utiliser une couleur de fond différente afin de mettre en évidence les items prédits. L'objectif de cette technique est de réduire le temps de



recherche visuelle. Cependant, les évaluations n'ont pas montré un grand bénéfice de cette technique par rapport à une interface statique non adaptative.

Le menu *Bolding* proposé par Park *et al.*, 2007 est une autre technique d'adaptation graphique. Ce type de menu consiste à mettre en gras les items prédits. Une évaluation avec 32 participants n'a pas montré de bénéfice par rapport à la condition contrôle statique. Le gain de présentation semble être trop faible pour pouvoir être mis en évidence statistiquement.

Le système Patina (Matejka *et al.*, 2013) propose d'utiliser des *heap map* colorés pour mettre l'accent sur les fonctions à utiliser. Ce système permet l'introduction d'effets visuels indiquant la fréquence d'usage des items de l'interface. Cependant, cela génère de nombreux effets graphiques et surcharge l'interface visuellement, ce qui au final ne semble pas être un gain.

Gong et Salvendy, 1995, ont proposé une interface adaptative où les utilisateurs sont progressivement amenés à utiliser de nouvelles commandes. Les utilisateurs deviennent plus familiers avec les noms de commandes au fur et à mesure de l'usage. Après un certain nombre d'utilisations d'un item du menu via une interface de pointage, le système met en évidence cet item et propose à l'utilisateur d'utiliser plutôt la ligne de commande en lui indiquant le nom de celle-ci. Le système masque alors l'item du menu principal. L'évaluation de cette technique a montré qu'elle a le potentiel d'améliorer les performances de l'utilisateur.

Malinowski, (1993) a proposé une approche consistant à surligner en vert les paramètres que l'utilisateur manipule fréquemment et fane les paramètres que l'utilisateur a tendance à ne pas utiliser.

Un autre exemple d'interfaces utilisateur graphiques adaptatives peut être trouvé dans l'architecture des barres d'outils adaptatives de Miah *et al.*, 1997. Les auteurs ont proposé une interface adaptative qui consiste à ajouter et enlever automatiquement les items ou éléments de barre d'outils en fonction de la fréquence et récence d'utilisation. Cependant, ces ajustements automatiques sont problématiques quand ils ne correspondent pas à ce que l'utilisateur veut faire à un moment donné.

### **3. Techniques temporelles**

Les techniques d'adaptation temporelle reposent sur l'introduction de la dimension temporelle lors de la présentation de l'adaptation. Lee et Yoon (2004) ont proposé le menu dit temporel (*Temporal menu*). Le menu temporel consiste à afficher les items en deux étapes. A

l'ouverture du menu, seuls les items prédits sont affichés à l'utilisateur. Après un délai de 170ms, les items non prédits apparaissent brusquement. Le délai de 170ms a été considéré comme suit : 100ms utilisés pour la perception et 70ms utilisés pour la cognition. L'apparition brusque des items non-prédits a été critiquée car c'est un facteur pouvant perturber les utilisateurs.

À la suite de ce travail, Findlater *et al.*, 2009, ont introduit l'adaptation éphémère comme une nouvelle technique pour l'adaptation des interfaces graphiques. Cette technique a pour objectif de maintenir la stabilité spatiale tout en jouant sur le caractère temporel, mais sans perturber l'utilisateur. Le principe de l'adaptation éphémère est d'utiliser l'apparition progressive des items non prédits. A l'ouverture du menu, seuls les items prédits sont affichés. Les items non-prédits apparaissent progressivement jusqu'à ce qu'ils soient affichés complètement dans l'interface. L'objectif est d'attirer l'attention de l'utilisateur sur les items prédits afin de lui permettre d'atteindre la cible plus rapidement.

Pour montrer les bénéfices de l'approche éphémère, une étude expérimentale a été effectuée par Findlater *et al.*, (2009). D'une part, l'objectif est de vérifier que le menu adaptatif éphémère est plus rapide que le menu statique lorsque la prédiction est correcte et n'est pas pire que le menu statique lorsque la prédiction est incorrecte. D'autre part, l'objectif est de tester la performance du menu éphémère par rapport au menu *highlighting*. Le choix de l'approche *Highlighting* est justifié par le fait que cette dernière maintient la stabilité spatiale tout comme l'approche éphémère. La performance a été mesurée en termes de rapidité. Les résultats ont montré que les utilisateurs sont plus rapides avec le menu éphémère qu'avec le menu statique et le menu *Highlighting* dans le cas d'une bonne prédiction. Lorsque la prédiction est incorrecte, les performances sont affectées. Le menu statique était plus rapide que les deux autres menus éphémères et *Highlighting*. Aucune différence significative n'a été observée entre le menu éphémère et le menu *Highlighting*.

En termes de satisfaction utilisateur, le menu éphémère et le menu *Highlighting* étaient préférés au menu statique mais aucune différence significative n'a été trouvée entre les deux.

#### **4. Cas des petits écrans**

La taille de l'écran du dispositif d'interaction joue un rôle important en interaction homme-machine (Raptis *et al.*, 2013). Une interface de taille réduite contraint l'interaction, surtout

lorsqu'elle est tactile car la position de la main entre l'écran et l'œil réduit encore plus le champ de vision.

Comme les éléments de l'interface ne peuvent pas être affichés en une seule fois lorsqu'ils sont nombreux, cela se traduit par davantage de navigation. Findlater et McGrenere, 2008, ont montré que l'adaptation est davantage bénéfique pour les IHM de petite taille plutôt que celles de grande taille. Les auteurs ont mené une étude expérimentale avec 36 sujets. Les résultats ont montré que lorsque la prédiction est correcte, les menus adaptatifs sont plus efficaces que les non adaptatifs à la fois sur petit et sur grand écran. Les utilisateurs sont même plus rapides sur petit écran que sur grand écran.

Patel *et al.* (2009), Sherman *et al.* (2003), Jones *et al.* (1999) ont étudié l'interaction web sur des dispositifs de petite taille. Une étude expérimentale sur 20 sujets a été menée dans le but de quantifier et de qualifier l'effet d'un petit espace d'affichage sur l'achèvement d'une tâche web. Les résultats du questionnaire ont montré que la taille de l'écran empêche l'exécution des tâches. 80% des utilisateurs ont indiqué être contraints par la taille réduite de l'écran. Les auteurs ont proposé des adaptations web pour résoudre cela. Leurs suggestions sont les suivantes :

- Fournir des catégories orientées selon les buts de l'utilisateur. Par exemple, structurer les informations pour fournir une navigation ciblée en présentant à l'utilisateur une liste des objectifs pouvant être réalisés. Pour ce faire, les agents d'adaptation (humains ou automatiques) doivent tenir compte de la raison pour laquelle un utilisateur peut accéder au site ou une page en particulier (Theng, 1996, López-Jaquero *et al.*, 2009).
- Réduire le défilement en mettant en place des fonctions de navigation (menu, barre, etc.) près du sommet de pages dans un endroit stable (par exemple, des menus fixes sur le côté gauche de chaque page).
- Mettre en place des informations clés en haut des pages (chemin de faire, consigne, alerte, etc.).
- Réduire la quantité d'informations sur la page. Nielson (1997) suggère que cette règle soit applicable à tous les sites web, en faisant valoir que les utilisateurs ne lisent pas les pages web de la même façon. En effet, 79% des utilisateurs balayent le texte tandis que 16% procèdent par une lecture mot à mot.

Dans le cadre de la manipulation de fonctions sur petits écrans, Bridle et McCreath (2005) ont proposé des raccourcis permettant de remplacer des séquences d'actions effectuées fréquemment. Les utilisateurs de téléphones mobiles répètent souvent certaines tâches, par exemple, l'envoi du même message à un ami quotidiennement. Cette tâche peut être automatisée par un raccourci de commande permettant d'améliorer les performances.

Les raccourcis qu'ils soient des macro-commandes ou des accès directs à une catégorie ou une fonction sont du même ordre : ils permettent d'accélérer l'exécution de la tâche. Ils peuvent être calculés à partir de la tâche globale et nécessitent alors une modélisation de l'utilisateur. Mais ils peuvent aussi être générés plus simplement à partir des sous-tâches voire de l'usage des fonctions elles-mêmes. Les techniques d'adaptation que nous avons présentées dans ce chapitre sont toutes liées à ces notions de raccourcis. Cependant, ces techniques sont appliquées sur grand écran et ne peuvent pas être transposées facilement sur petit écran.

Les techniques de rendus graphiques sont problématiques. Par exemple, le menu *Morphing* qui permet d'agrandir la taille de certains items prédits ne peut pas être transposé à des interfaces de petite taille. Dans le cas où la liste complète des items est longue, ils vont être présentés sur plusieurs écrans. Si un item prédit se trouve en bas de liste, il n'est pas visible immédiatement et l'utilisateur doit faire un défilement pour l'atteindre. Non seulement la prédiction est cachée en bas de liste, et la retrouver demande un effort cognitif de la part de l'utilisateur. De plus, cette prédiction est peu utile puisque peu visible et difficilement accessible.

La même problématique se pose avec les techniques *Highlighting*, *Bolding*, *Temporal* et *Ephemeral*, lorsque la prédiction se situe en fin de liste, et donc potentiellement sur des écrans nécessitant un défilement. De plus, pour *Ephemeral*, l'utilisation du grisé progressif contraint fortement la lisibilité. Néanmoins, *Temporal* et *Ephemeral* possèdent un atout qui est de ne pas rester bloqué sur les items prédits en cas de prédictions non pertinentes.

Quant aux techniques spatiales, elles ne sont que partiellement applicables aux petits écrans. Par exemple, les menus *Split*, *Smart* et *Frequency-based* sont envisageables sur petit écran mais ont cependant été largement critiqués car ils ne maintiennent pas la stabilité spatiale. Cet effet de désordre dans l'interface est frustrant pour les utilisateurs et ne leur permet pas de créer un modèle mémoire de leur interface. Le menu *Split* avec réplique maintient pourtant

la stabilité spatiale en dupliquant les items prédits dans la zone adaptative. Cependant, cette technique « spatiophage » ne convient pas sur petit écran. Lorsque la prédiction ne correspond pas aux besoins de l'utilisateur, les coûts de navigation et de défilement sont augmentés.

## **5. Conclusion**

Dans ce chapitre, nous avons présenté différentes techniques d'adaptation tout en soulignant leurs avantages et inconvénients. Dans la plupart des techniques d'adaptation, les variables position et format sont explorées de manières différentes. Par exemple, le déplacement des items prédits en haut d'interface pour les mettre en avant (techniques d'adaptation spatiales) ou le changement de format des items prédits (techniques d'adaptation graphiques). Les différentes techniques sont illustrées dans la figure 4.

Mais faire varier les paramètres de rendu peut se faire à 'l'infini' selon la créativité des graphistes (Samp *et al.*, 2010). Or, il n'existe pas de cadre permettant de mettre à plat l'ensemble des variables pouvant être utilisées pour transmettre un changement dans l'interface de manière efficace afin d'accélérer l'interaction lorsque la prédiction est correcte, mais sans la pénaliser lorsqu'elle est incorrecte. Pour cette raison, nous proposons un modèle de présentation de l'adaptation que nous définissons dans la partie suivante.

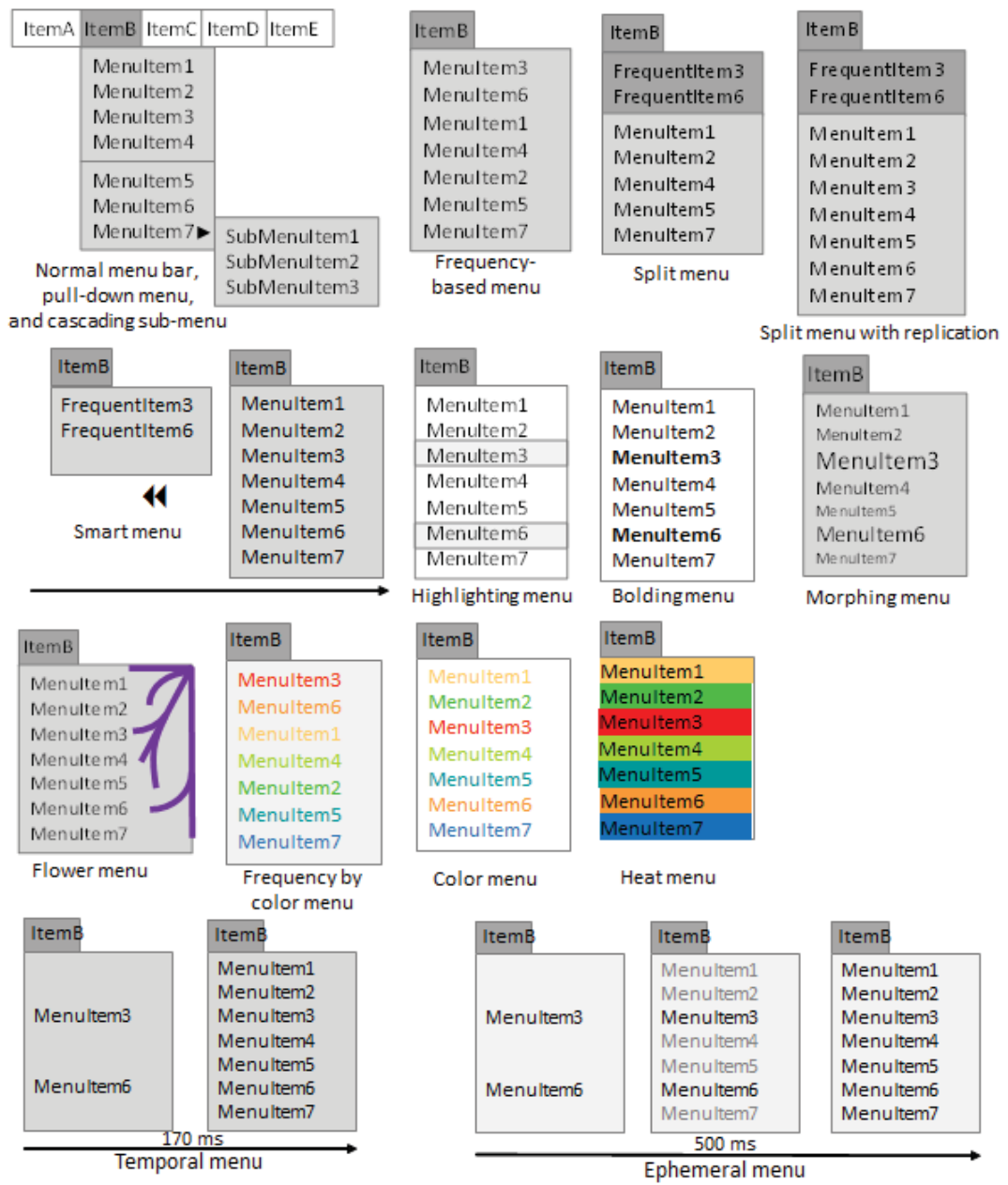


Figure 4. Exemples d'interfaces adaptatives.

## 6. Synthèse

Il existe trois grandes classes de techniques d'adaptation :

- **Spatiales** qui consistent à changer l'ordre et la position des éléments dans l'interface : nous retenons que les techniques d'adaptation spatiales permettent d'accélérer l'interaction lorsque la prédiction est correcte car elles mettent en avant les items prédits et facilitent l'accès à ces items. Cependant, ces techniques ne permettent pas à l'utilisateur de s'appuyer sur leur mémoire du fait de l'instabilité spatiale. Par conséquent, lorsque la prédiction est incorrecte, l'accès à la cible devient difficile ;

- **Graphiques** qui introduisent des effets visuels afin de mettre en évidence les éléments prédits : nous retenons que les techniques d'adaptation graphiques visent à améliorer la perception en réduisant le temps de recherche visuelle. Cependant, les évaluations n'ont pas montré un grand bénéfice de ces techniques par rapport aux interfaces statiques (non adaptatives). En outre, lorsque la prédiction est incorrecte, l'interface est surchargée par des effets visuels, ce qui dégrade l'interaction ;

- **Temporelles** qui introduisent la notion de temps dans la présentation de l'adaptation : nous retenons que les techniques d'adaptation temporelles permettent de réduire le temps de navigation et de recherche visuelle lorsque la prédiction est correcte. Dans le cas contraire (prédiction incorrecte), la tâche de l'utilisateur peut être ralentie. L'utilisateur doit en effet attendre l'apparition de sa cible.

En conclusion, dans tous les cas, lorsque la prédiction est incorrecte, l'interaction est affectée : les techniques d'adaptation spatiales ne permettent pas à l'utilisateur de s'appuyer sur sa mémoire ; les techniques graphiques surchargent l'interface; les techniques temporelles peuvent ralentir l'utilisateur dans sa tâche. Par ailleurs, aucune de ces techniques ne se transposent aisément sur petit écrans.





## **PARTIE II : ESPACE DE CONCEPTION POUR L'INTERACTION ADAPTATIVE**

La partie I a montré que de nombreuses études ont été menées sur les espaces de conception d'adaptation. La conception de l'adaptation est souvent centrée principalement sur le système. Les modèles d'adaptation proposés dans la littérature (Arhipainen, 2009, Norcio et Stanley, Benaboud *et al.*, 2006, Brusilovsky, 1996, Cote-Munoz, 1993, MacLean *et al.*, 1991) ne gèrent pas, ou peu, la coévolution entre l'utilisateur et le système. C'est la situation d'interaction qui nécessite d'être adaptée et non seulement le système.

Dans cette partie nous présentons nos contributions théoriques qui se répartissent sur deux chapitres. Le chapitre 3 présente AIMS, un modèle pour l'interaction adaptative. Le chapitre 4 présente PCA, un modèle de présentation de l'adaptation.

## **Chapitre 3 : AIMS, modèle systémique pour la caractérisation de la qualité**

Ce chapitre a fait l'objet d'une publication en conférence internationale :

**Bouzit, S., Calvary, G, Coutaz, J., Chêne, D., Petit, E., Vanderdonckt, J. (2017).** The PDA-LDP Design Space for User Interface Adaptation. The eleventh IEEE International Conference on Research Challenges in Information Science (RCIS), 10-12 May 2017, Brighton (UK).

Dans ce chapitre, nous présentons un modèle d'adaptation pour l'interaction appelé AIMS (Adaptive Interaction Model with Symmetry). Il adopte une approche systémique et décrit l'adaptation côté utilisateur et système dans le cadre d'une interaction.

La structuration de ce chapitre est la suivante : nous présentons les éléments fondateurs du modèle puis décrivons le modèle. Enfin nous exprimons à travers ce modèle les critères de qualité à considérer en conception d'interaction adaptative.

## **1. Éléments fondateurs**

Selon la théorie d'action proposée par Norman (1986), l'interaction se fait en sept étapes comme évoqué dans l'état de l'art. Norman s'appuie sur les études de psychologie du comportement, mais aussi sur la théorie de la décision humaine (Bratman, 1987, Kellen, 2003). Dans cette dernière, sont décrites les trois activités essentielles qui sont : Perception, Décision, Action. L'activité de Perception correspond à la phase de capture et d'interprétation des éléments de l'interface. L'activité de Décision correspond à la phase de choix entre plusieurs options possibles. L'activité Action correspond à la phase de mise en œuvre de la décision prise. La dimension Perception-Décision-Action (PDA) est illustrée sur la figure 5a. L'utilisateur va effectuer plusieurs itérations sur cette dimension jusqu'à ce que sa tâche soit accomplie. La dimension PDA ne fonctionne pas de manière indépendante : elle s'inscrit dans le cadre d'une interaction où des éléments nouveaux viennent alimenter sa perception et sont produits suite à ses actions (flèches rouges figure 5b).

Cependant, symétriquement le système doit percevoir ce que l'utilisateur est en train de faire par le biais de capteurs ou d'événements générés par l'utilisateur. Le système doit également décider quelle action va être prise (par l'intermédiaire d'un contrôleur de dialogue par exemple). Le système exécute les actions nécessaires. Par conséquent, les activités essentielles côté système sont aussi représentées sur une dimension PDA. De même, le système subit les modifications du contexte et génère des modifications (flèches rouges figure 5b).

Dans le cadre de l'interaction homme-machine, nous avons donc deux dimensions qui interagissent comme décrit dans la figure 5b. La membrane d'échange entre ces deux dimensions est l'interface.

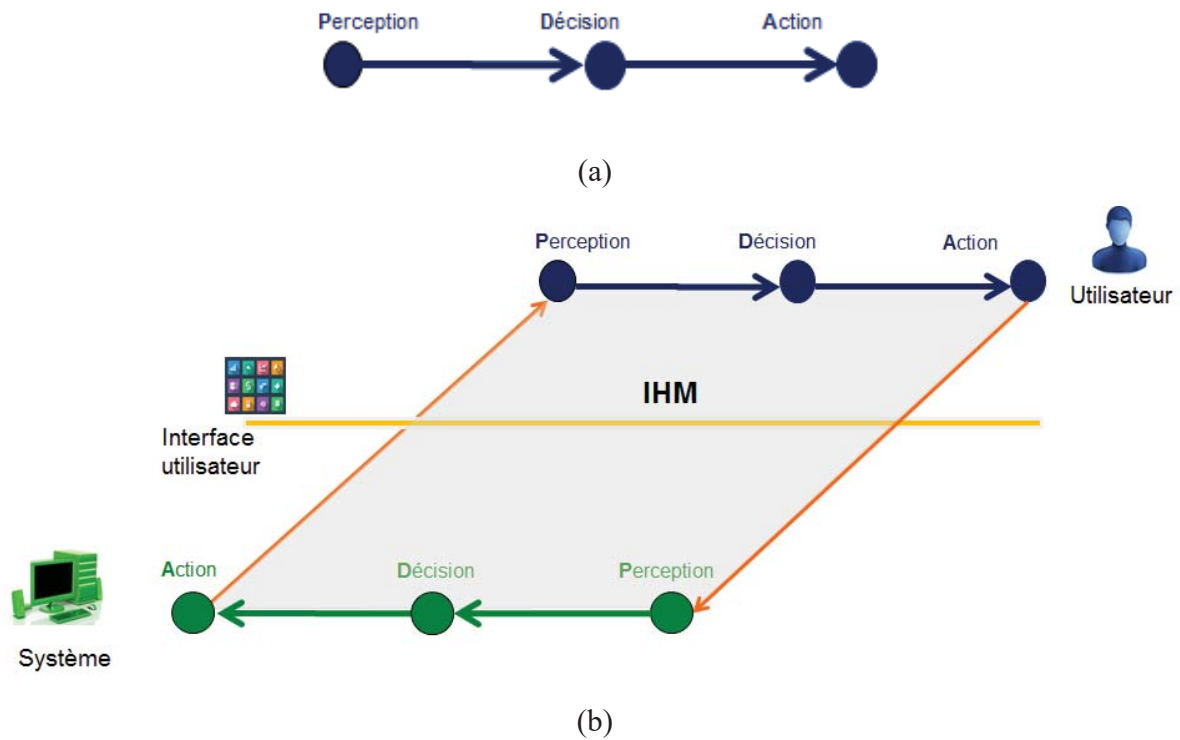


Figure 5. (a) Dimension Perception-Décision-Action, (b) Boucle d'interaction.

## 2. Modèle AIMS

En plus de la dimension (Perception-Décision-Action) permettant de décrire l'activité, l'originalité du modèle est d'intégrer une dimension dédiée à l'adaptation. Le mécanisme d'adaptation interfère avec la dimension PDA aussi bien côté utilisateur que côté système.

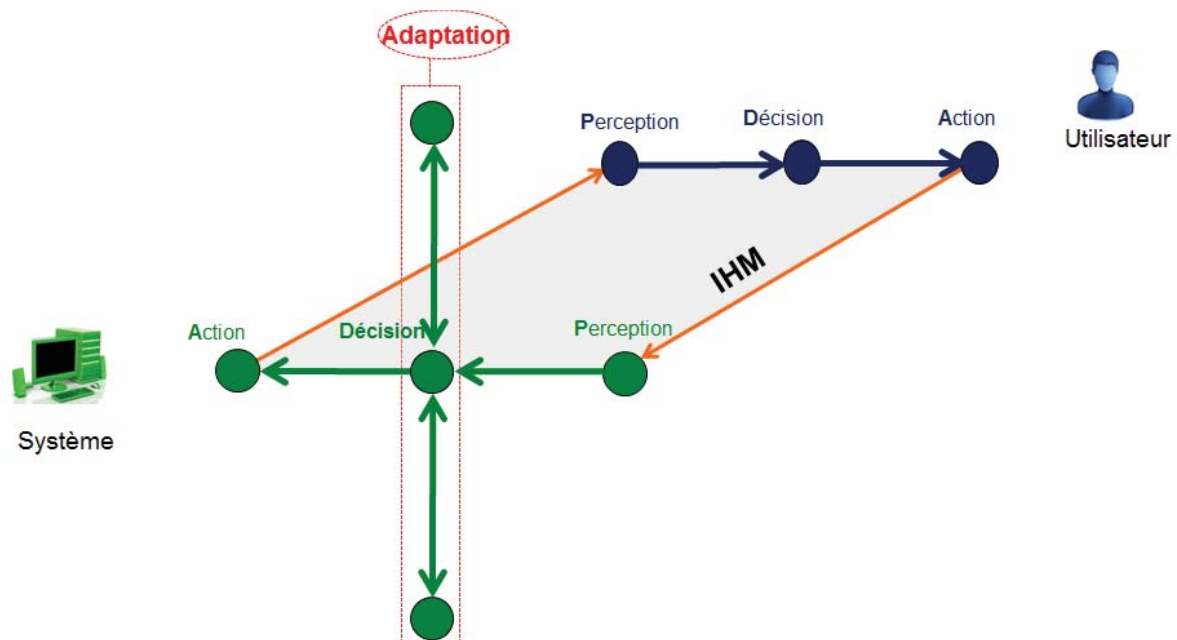


Figure 6. Modèle d'adaptation du système pour l'interaction.

L'adaptation est l'ajustement d'une entité d'interaction (système ou utilisateur) au contexte d'usage pour un objectif donné. Pour ce faire, au-delà de la perception et des actions sur le contexte (PDA), il est nécessaire de stocker les états du contexte, de prévoir les états futurs par rapport à l'objectif visé et éventuellement de déclencher les ajustements nécessaires. Ainsi arrive la dimension d'adaptation appelée ADP faisant référence aux notions : Apprentissage-Décision-Prédiction.

De son côté le système adaptatif perçoit l'utilisateur et stocke les états du contexte d'interaction passés et en cours. Afin de simplifier le traitement de données, le système peut créer une base de connaissances. Il y collecte toutes les données issues des perceptions système, des décisions prises et des actions mises en œuvre. En d'autres termes, la base de connaissances est implémentée et enrichie à partir de la dimension PDA. C'est à partir de ces données enregistrées que le système passe en phase d'apprentissage (A de ADP). Sur les connaissances accumulées à partir de l'activité de l'utilisateur, le système peut créer des règles d'adaptation.

Les règles d'adaptation créées par le système sont appliquées lors d'un changement du contexte. Ce changement peut être déclenché par une décision prise par le système : c'est le cas de **l'adaptativité**. Au contraire, quand c'est le système qui s'adapte à la demande de

l'utilisateur, nous sommes dans la situation de **l'adaptabilité**. Lorsque c'est à la fois l'utilisateur et le système qui sont à l'initiative, nous parlons d'une **initiative mixte** (Horvitz, 1999). Dans ce cas, le système résout l'adaptation dans un cadre de dialogue : le système propose, l'utilisateur dispose et l'adaptation est appliquée ou non. C'est la décision de mise en œuvre de l'adaptation que représente le D de ADP.

L'adaptation peut se faire par rapport à une analyse de l'état courant (détection d'une situation de blocage) et décider d'une action immédiate (proposition d'assistance) sans prise en compte d'un objectif particulier. Mais elle peut également se faire en fonction d'un état futur que le système estime en correspondance avec les objectifs visés par l'utilisateur. Dans ce cas, l'adaptation consiste à proposer une ou plusieurs prédictions (c'est le P de ADP).

Finalement le système décide de mettre en œuvre (ou non) ce choix d'adaptation en l'exécutant au niveau de l'interface utilisateur. Le modèle AIMS côté système est décrit dans la figure 7.

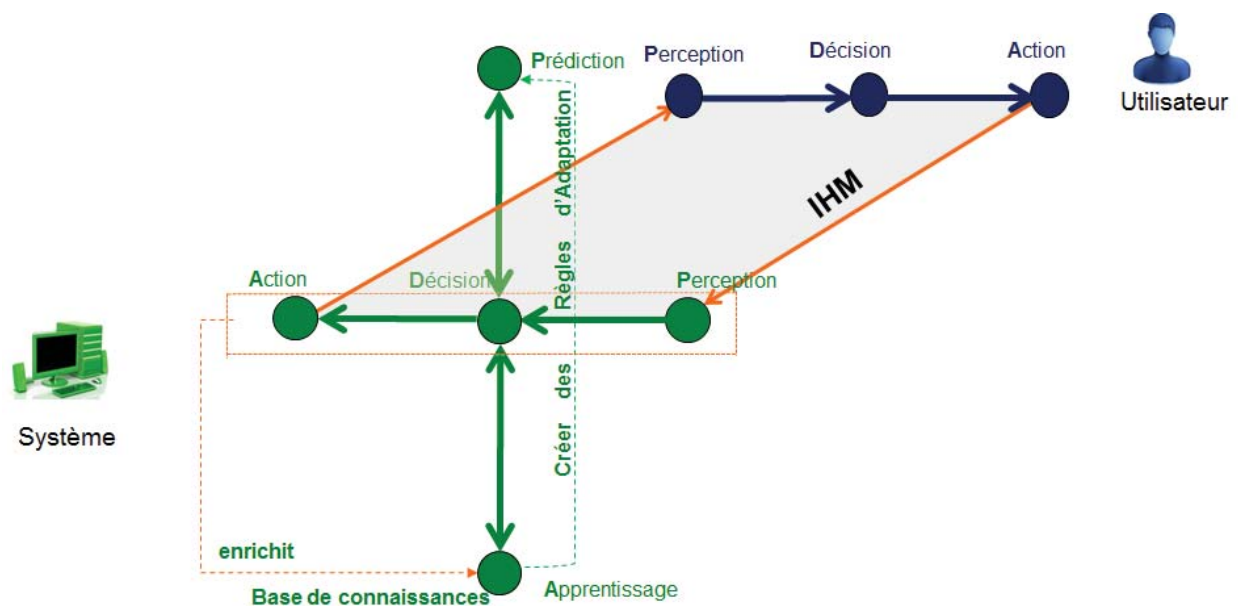


Figure 7. Modèle AIMS côté système.

Habituellement en présence d'un système non-adaptatif, seul l'utilisateur s'adapte (figure 8). En présence d'un système adaptatif, le même type d'apprentissage sera mis en œuvre côté utilisateur. À chaque interaction, l'utilisateur se construit un modèle du système avec lequel il interagit. Ce modèle créé par l'utilisateur peut être sous forme d'une base de connaissances de

ses expériences et dans laquelle sont stockées les fonctions du système et les chemins permettant d'atteindre son objectif. A partir de cette base de connaissances, l'utilisateur peut se créer des règles d'adaptation afin d'améliorer ses façons de faire. Il est alors capable de prédire le comportement du système adaptatif. Sur cette base, l'utilisateur définit sa stratégie d'interaction.

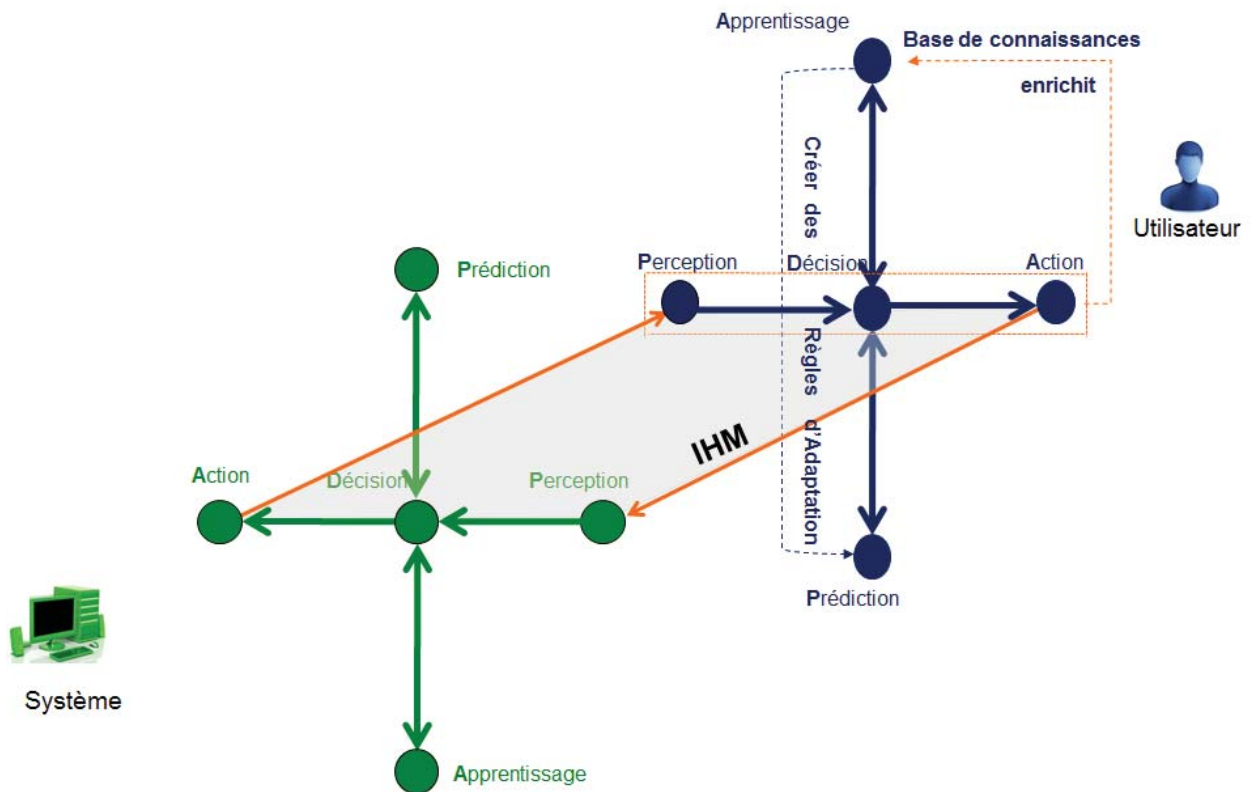


Figure 8. Modèle AIMS côté utilisateur.

Le modèle de l'interaction adaptative permet donc d'exprimer explicitement et symétriquement les activités des deux entités, utilisateur et système, tout en séparant le plan de la boucle interactionnelle du plan de l'adaptation. Ces deux plans sont illustrés dans la figure 9.

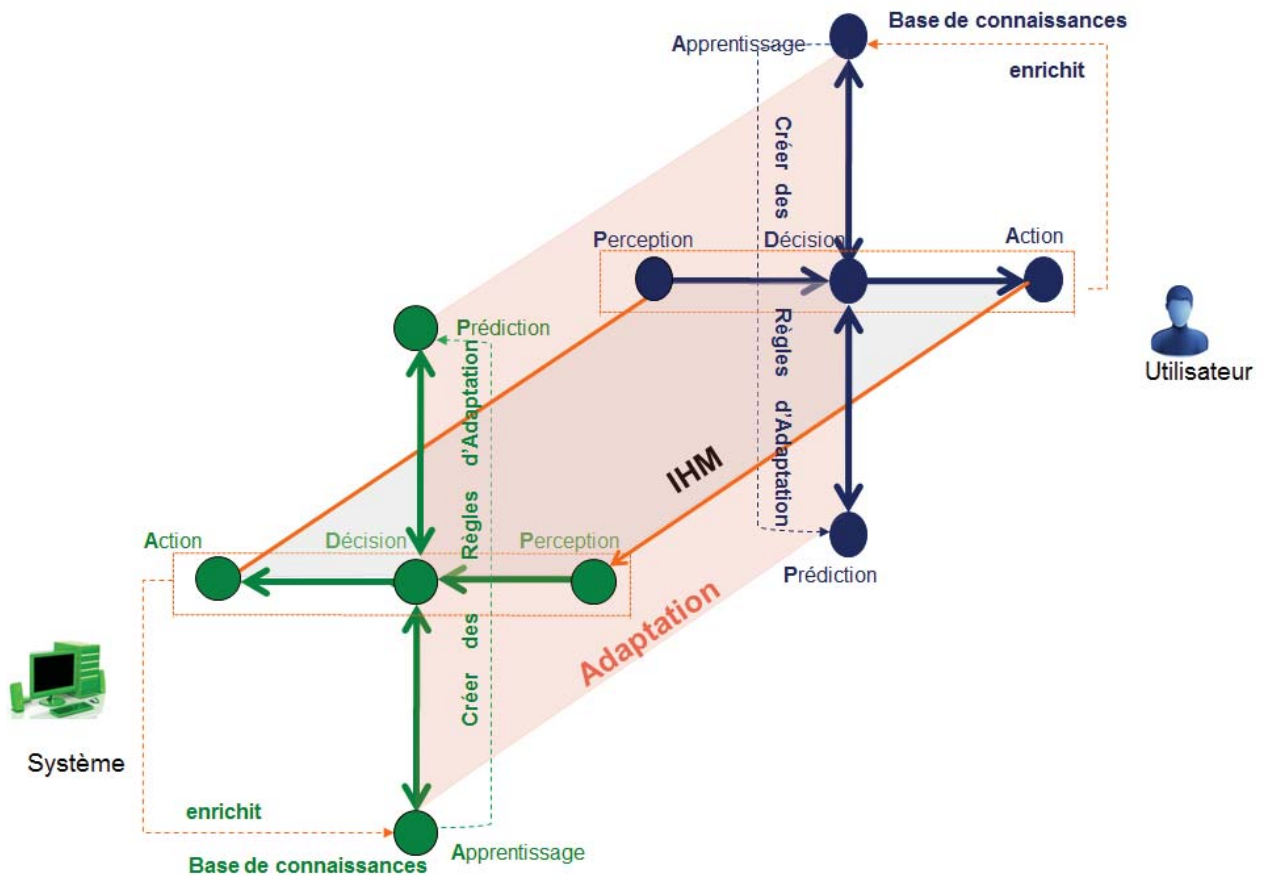


Figure 9. Les plans Interaction et Adaptation du modèle AIMS.

Illustrons ce modèle à partir du *Smart* menu. Le système affiche plusieurs items de menu au sein de la barre de menus (A de la dimension PDA système). L'utilisateur perçoit l'item Fichier au sein de la barre de menu (P de la dimension PDA utilisateur). Il décide de l'utiliser (D de la dimension PDA utilisateur) et clique dessus (A de la dimension PDA utilisateur). Le système reçoit ce clic (P de la dimension PDA système) puis calcule et décide en fonction de l'expérience passée (D à l'intersection des dimensions ADP et PDA) quels sont les deux items prédits à afficher (A de la dimension PDA système). L'utilisateur voit ces deux items (P de la dimension PDA utilisateur) et décide de ne pas les utiliser (D) et attend (A). Le système capture cette attente (P de la dimension PDA système), stocke l'information que les deux items prédits n'ont pas été utilisés (A de la dimension ADP système), décide de ne pas



proposer de prédictions et de déployer automatiquement toute la liste des items (D de l'intersection PDA-ADP système), puis affiche cette dernière (A de PDA système).

### 3. Critères de qualité

L'interaction adaptative se situe avant tout dans le cadre d'une boucle interactionnelle. L'utilisateur s'ajuste aux limitations du système en apprenant les limites et possibilités de ce dernier et en prédisant ses comportements à venir. Dans notre modèle, la qualité de cette boucle est exprimée par l'ajustement entre les dimensions PDA, rendu possible par les apprentissages successifs effectués côté utilisateur. Pour définir une interaction adaptative optimale, nous proposons donc d'identifier dans un premier temps les critères de qualité de l'interaction (ajustement des dimensions PDA) puis d'en déduire les critères de qualité de l'adaptation au regard de la littérature sur le sujet (ajustement des dimensions ADP). Les critères permettront de concevoir le meilleur ADP système (figure 10).

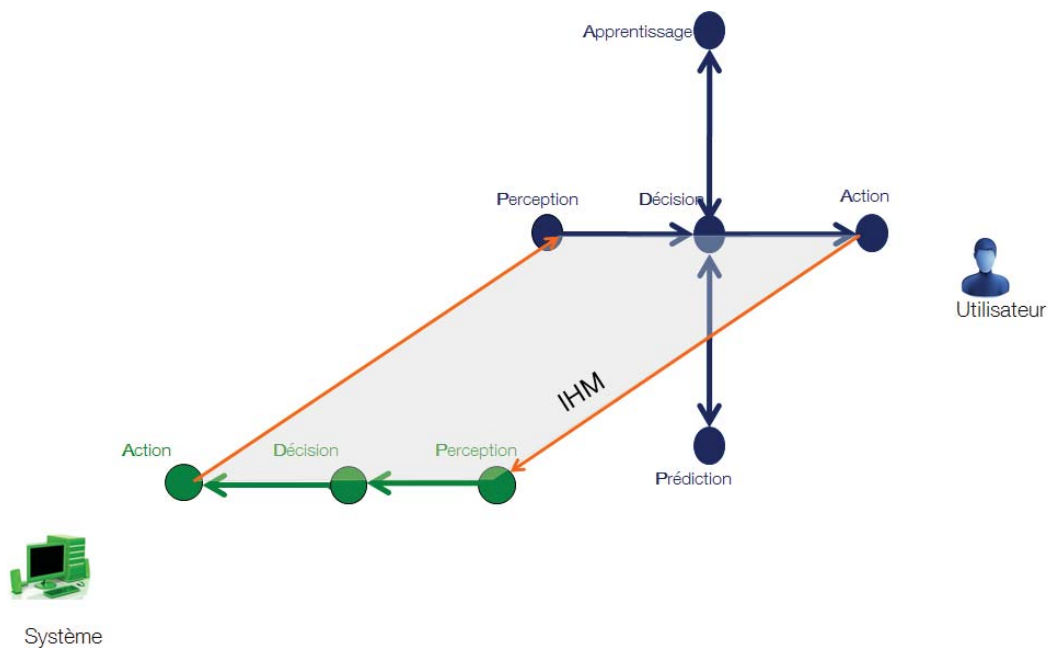


Figure 10. Ajustement des dimensions PDA système et PDA utilisateur.

Dans la norme ISO 9241-110, un ensemble de critères de qualité de l'interaction a été défini. Cette norme retient sept principes :

Le critère d'adaptation à la tâche signifie que le dialogue permet de réaliser la tâche de façon efficace et efficiente. Une boucle interactionnelle de qualité permettra à l'utilisateur d'agir sur

les actions proposées par le système afin d’accomplir ses objectifs sans trop avoir à jouer sur sa dimension ADP (figure 11).

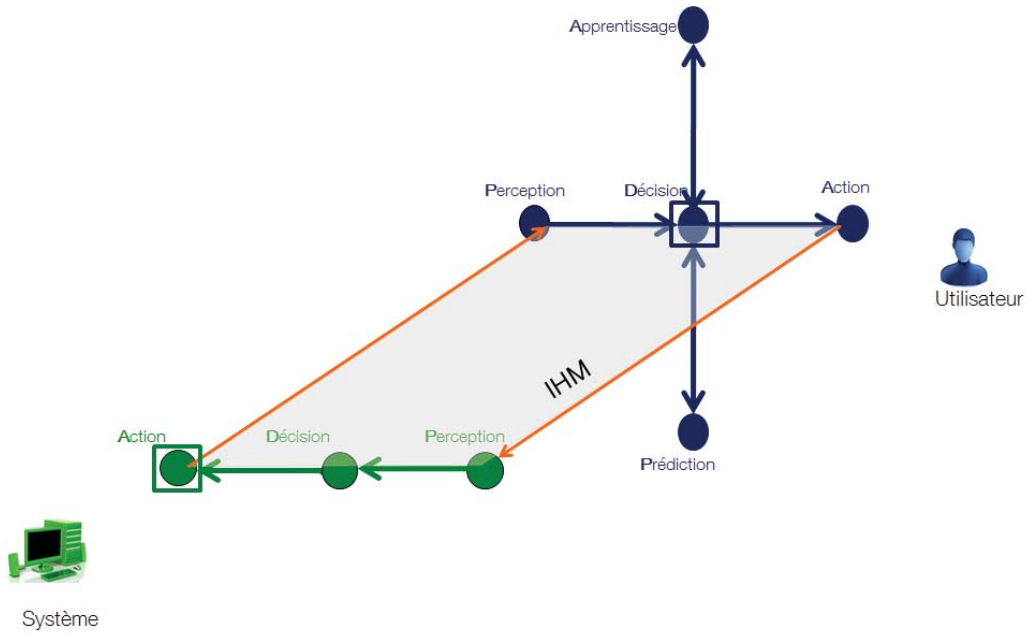


Figure 11. Ajustement à la tâche.

Le caractère auto-descriptif signifie que le dialogue est immédiatement compréhensible grâce au retour d’information du système ou expliqué à la demande de l’utilisateur (figure 12).

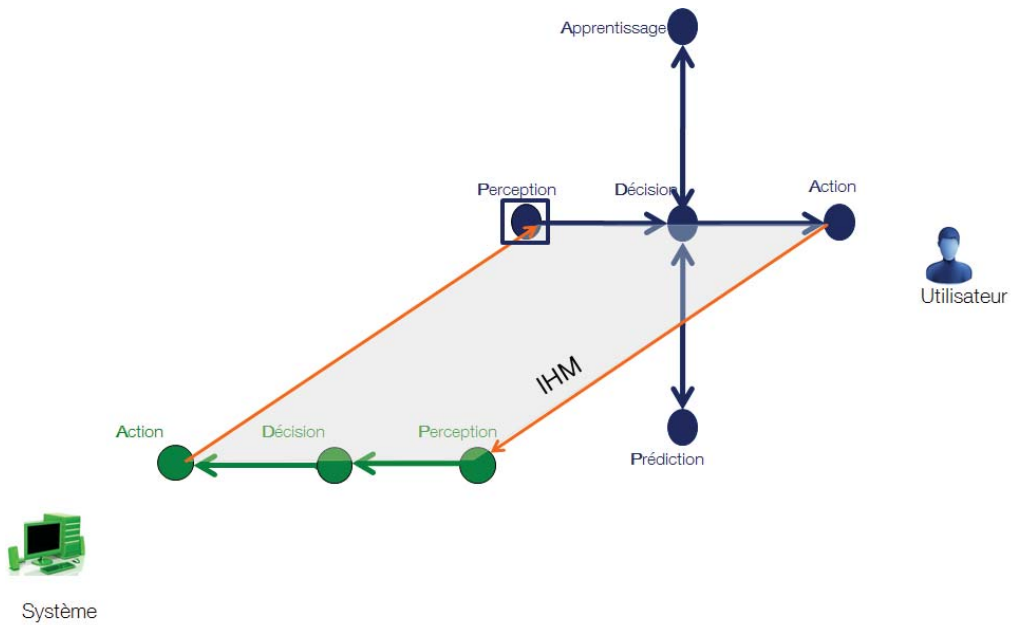


Figure 12. Caractère auto-descriptif.

Le contrôle utilisateur permet à l'utilisateur d'initier et de contrôler l'interaction (figure 13).

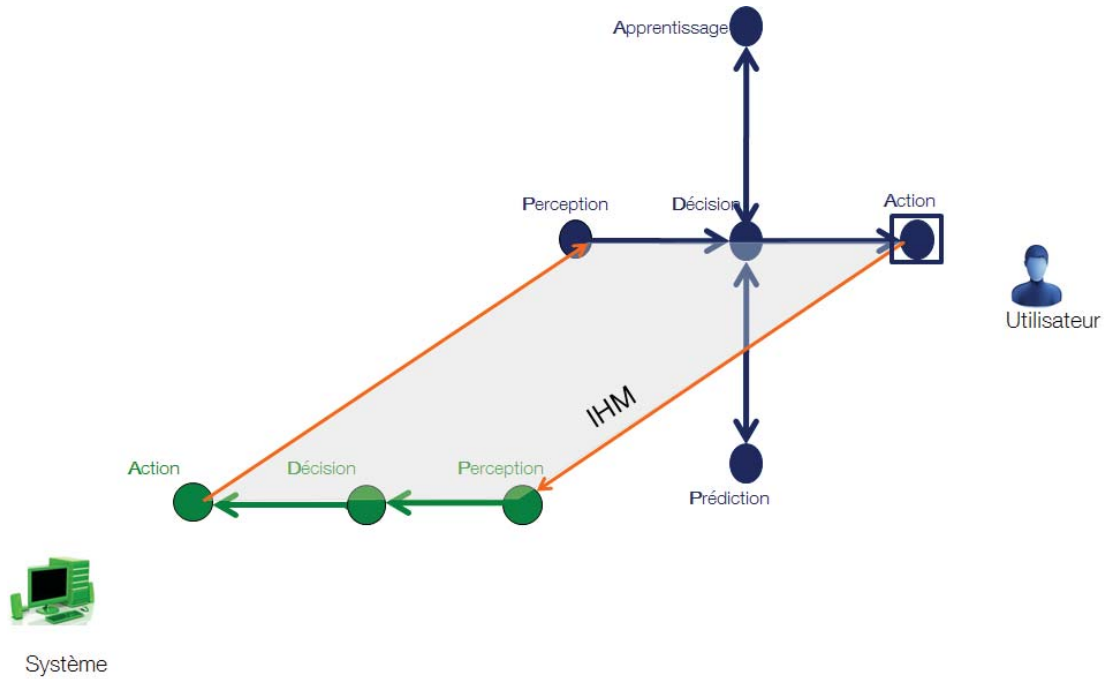


Figure 13. Contrôle utilisateur.

La conformité aux attentes utilisateurs signifie que le dialogue est cohérent et correspond aux caractéristiques des utilisateurs (figure 14).

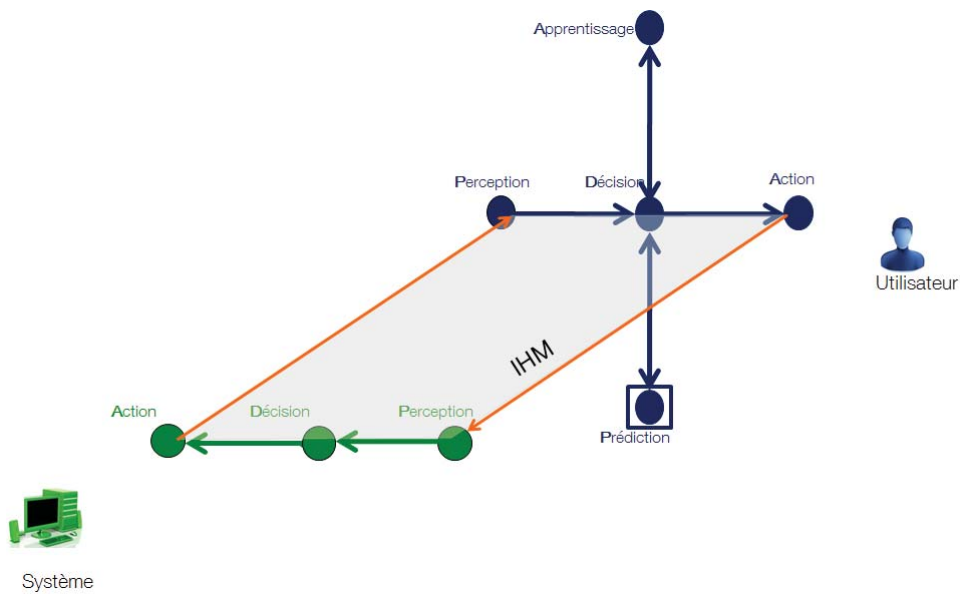


Figure 14. Conformité aux attentes de l'utilisateur.

La tolérance aux erreurs signifie que le résultat prévu peut être obtenu avec ou sans actions correctrices (figure 15). L'utilisateur produit une action imprécise et le système est en mesure de pallier cette imprécision (par exemple en séparant bien les zones d'activation des boutons).

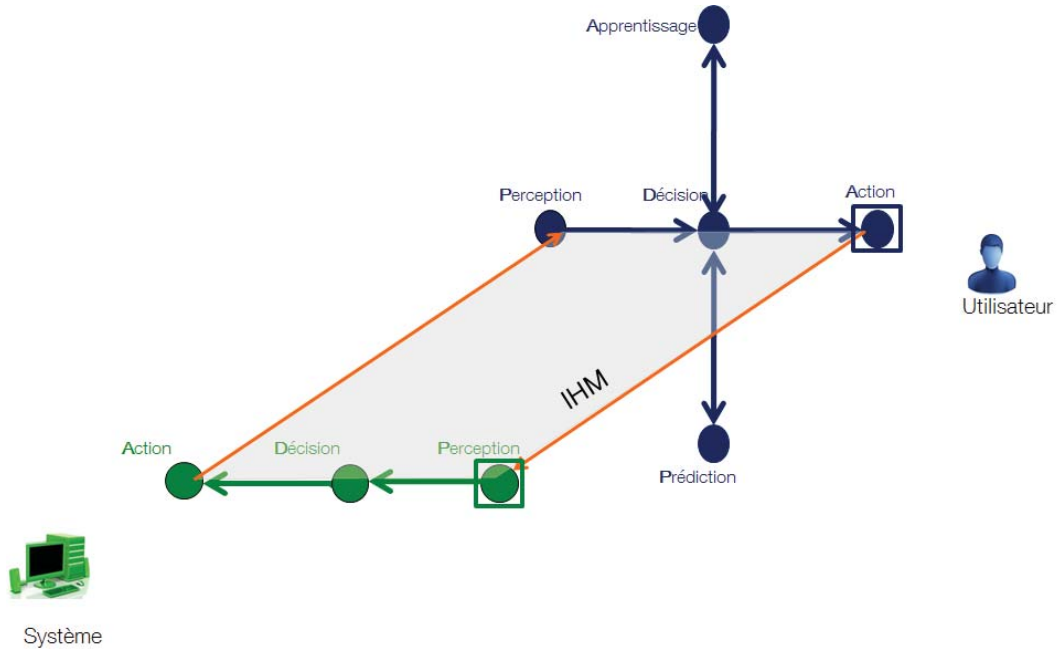


Figure 15. Tolérance aux erreurs.

Aptitude à l'individualisation signifie que l'interface peut être modifiée pour s'adapter aux besoins de la tâche ou aux préférences ou compétences des utilisateurs (figure 16).

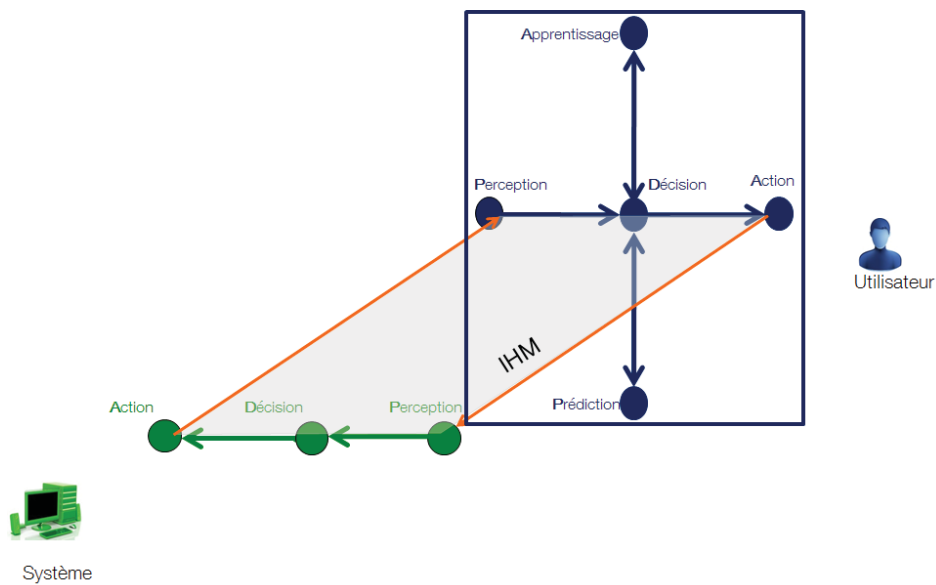


Figure 16. Aptitude à l'individualisation.

Facilité d'apprentissage signifie que le dialogue soutient et guide l'utilisateur dans l'apprentissage du système (figure 17).

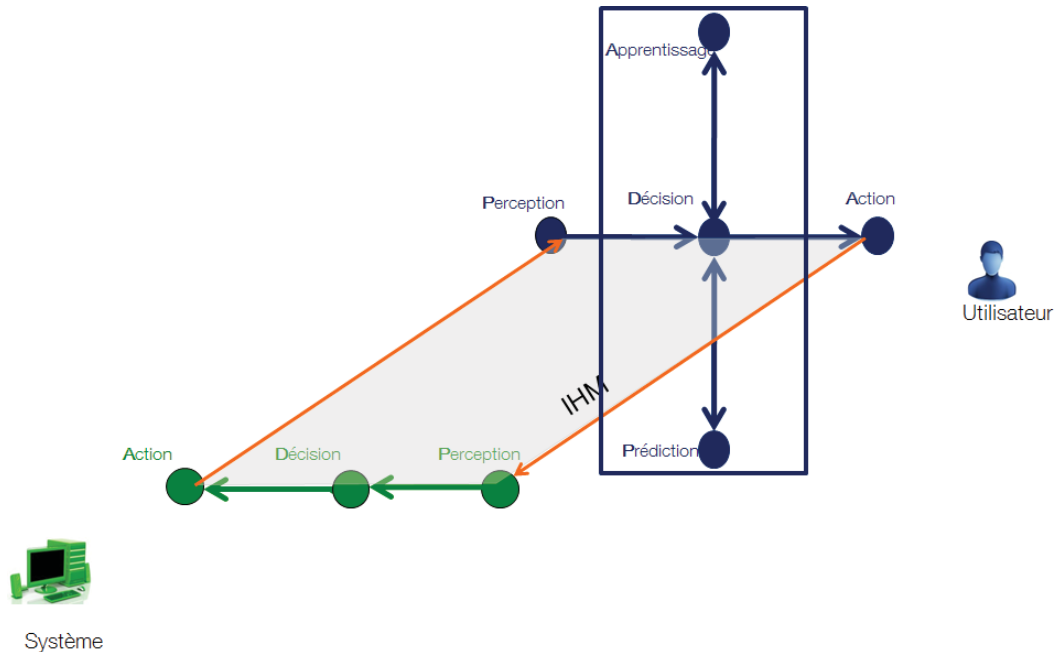


Figure 17. Facilité d'apprentissage.

Du côté du système, la qualité de l'adaptation est relative à la conformité de l'adaptation vis-à-vis des besoins de l'utilisateur dans son contexte d'usage (López-Jaquero *et al.*, 2008). À l'inverse, lorsque le système fournit de mauvaises prédictions (figure 18, bulle 1.), cela peut avoir des conséquences négatives sur l'interaction voire sur la suite des adaptations proposées. (figure 18, bulles 2 et 3) L'utilisateur peut en effet être surchargé par des prédictions incorrectes (figure 18, bulle 4.), ce qui peut générer une surcharge cognitive (figure 18, bulle 5.) et des difficultés de manipulation entraînant des erreurs de pointage (figure 18, bulle 6.). Dans cette situation, l'utilisateur n'arrivera pas à prédire le comportement du système. Son propre mécanisme d'adaptation sera faussé. Suite à cette situation, les erreurs de pointage et de manipulation de l'utilisateur peuvent générer des conséquences négatives sur les phases d'apprentissage côté système (figure 18, bulles 7, 2 et 8.). Ces données erronées vont être utilisées à tort lors de la prédiction suivante.

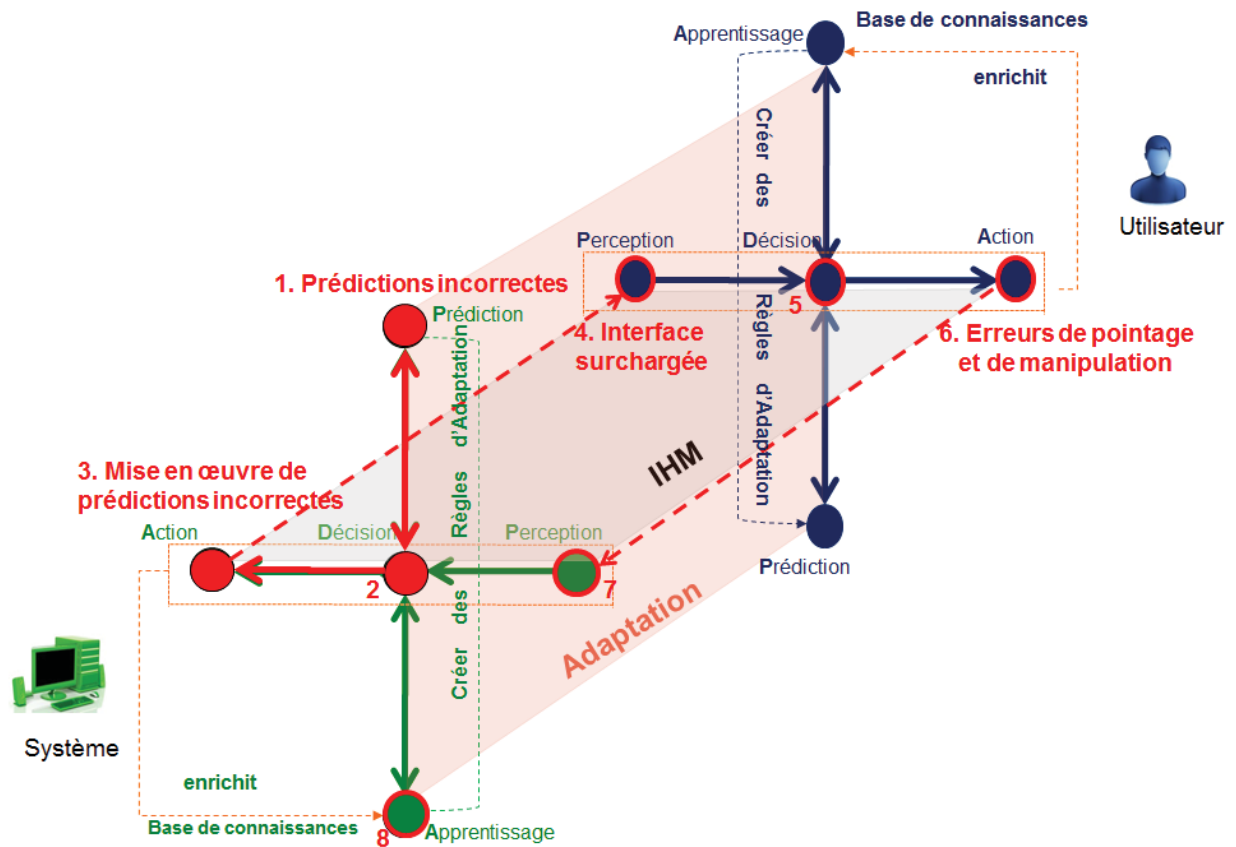


Figure 18. Description des conséquences générées par une adaptation inappropriée.

La non correspondance des prédictions du système vis-à-vis de l'objectif de l'utilisateur peut avoir d'autres origines telles que le changement d'habitudes de ce dernier. En effet, les systèmes savent apprendre à partir des habitudes mais ne savent pas désapprendre. Cela impose que le système soit capable de toujours mettre à jour sa base de connaissances et ses règles d'adaptation, mais cela présuppose également que la boucle interactionnelle ne génère pas elle-même de biais sur les adaptations proprement dites. En effet, les erreurs de pointage ou de manipulation citées ci-dessus se situent au niveau de l'articulation entre les deux dimensions horizontales PDA. De la qualité de cette boucle interactionnelle dépendra la qualité des adaptations réciproques. C'est en effet par cette boucle que pourront s'ajuster les dimensions ADP respectives.

Nous avons vu l'ensemble des critères permettant d'assurer la boucle interactionnelle, passons en revue maintenant ceux qui permettent de qualifier une interaction adaptative (figure 19). Dans le cas de la boucle interactionnelle, nous avons vu que c'est l'ajustement entre les dimensions PDA qui permet de définir la qualité. Dans le cas d'une adaptation réciproque, ce sera de même l'ajustement entre les dimensions ADP. Néanmoins, il n'existe pas de lien direct entre ces dimensions. Leur ajustement s'effectuera donc systématiquement en passant par la boucle interactionnelle. Ceci est exprimé par les pointillés de la figure 19.

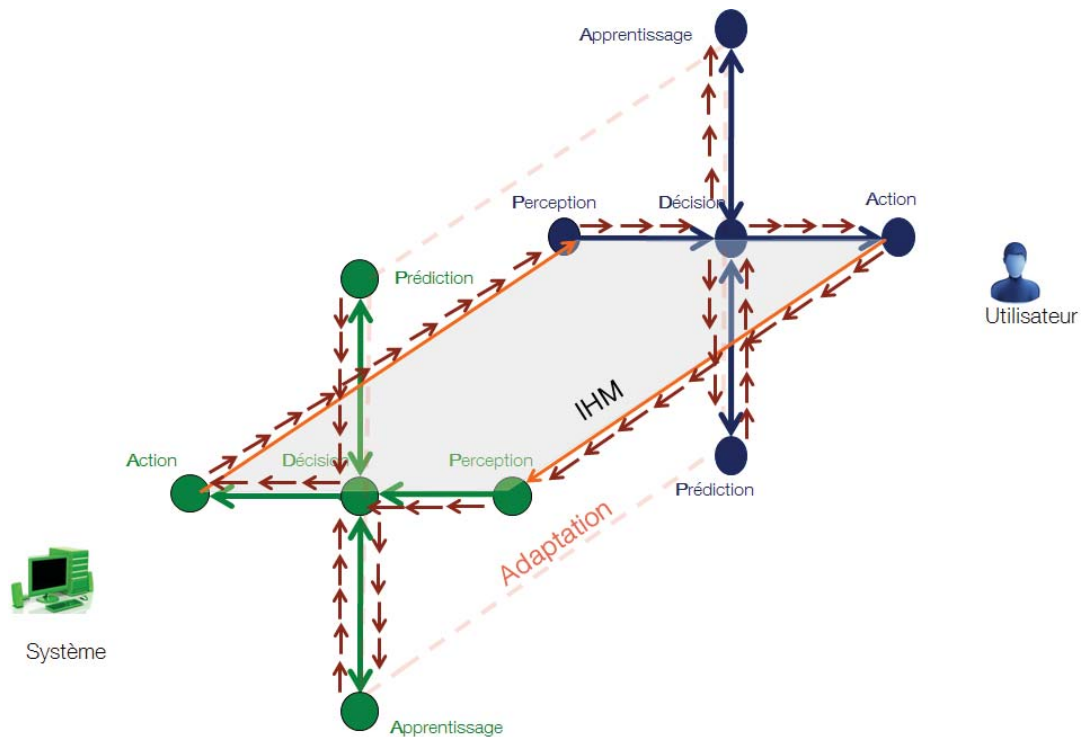


Figure 19. Ajustement des dimensions ADP système et ADP utilisateur.

Les critères de qualité d'une interaction adaptative optimale sont les suivants.

**L'Intelligibilité système** correspond à la capacité du système à expliquer et à communiquer à l'utilisateur comment les processus d'interaction et/ou d'adaptation sont menés de manière significative et représentative (Bellotti et Edwards, 2001, Brian et al., 2009, Vermeulen, 2010). Il s'agit pour le système de rendre les phases d'apprentissage, de création de règles d'adaptation et de prédictions compréhensibles par l'utilisateur (dimension ADP). En d'autres termes, le système explique comment il est capable de percevoir son utilisateur, quelles sont

les données utilisées pour l'adaptation et comment le choix de technique d'adaptation est mis en œuvre.

Ce critère qualité de l'adaptation peut faciliter l'apprentissage et ainsi faciliter et améliorer l'interaction de l'utilisateur.

**L'Intelligibilité utilisateur** correspond à la capacité de l'utilisateur de se rendre explicitement expressif. Dans ce cas, il explique comment il se construit son modèle mémoire du système et comment il préfère que le système réagisse et s'adapte (dimension ADP). L'utilisateur peut fournir au système ses préférences pour l'adaptation (dimension PDA). Cette expressivité permet au système d'améliorer la qualité des prédictions et des adaptations.

**La Stabilité** représente le fait que le système maintient l'ordre et la position des éléments dans l'interface, et par conséquent, réduit les effets de confusion et de désorientation chez l'utilisateur. L'interférence de la dimension ADP système avec la dimension PDA utilisateur ne devrait pas causer beaucoup de changements au niveau de la perception de l'utilisateur. Le système doit assurer des mesures d'adaptation cohérentes. Avec la stabilité spatiale l'utilisateur est capable de comprendre facilement l'interface et de prédire le comportement du système adaptatif. Au contraire dans une interface qui change fréquemment, l'utilisateur n'arrive pas à créer un modèle mémoire ce qui complique sa compréhension.

**La Transparence** du système et du processus d'adaptation consiste à rendre perceptibles à l'utilisateur les mécanismes sous-jacents du système (Dessart *et al.*, 2011; Huhtala *et al.*, 2009; Huhtala *et al.*, 2010; Schlienger, 2007). Donner à l'utilisateur un système transparent peut faciliter l'intelligibilité et par conséquent réduire la charge cognitive chez l'utilisateur.

**La Continuité** concerne la cohérence de la transition entre l'interface avant et après adaptation. Elle réduit tout changement appliqué de manière brutale. La continuité est également une propriété qui peut nourrir l'intelligibilité et ainsi aide l'utilisateur à comprendre les changements dans l'interface et augmente la prévisibilité du côté utilisateur. Le système doit assurer une continuité logique entre les différentes adaptations. L'utilisateur de son côté a également besoin de percevoir, au niveau de l'interface, une continuité avant et après adaptation.

**La Pertinence de la prédiction (Accuracy)** se réfère à la capacité du système à prédire correctement les intentions et futures actions de l'utilisateur en se basant sur sa base de connaissance et sur les précédentes actions de l'utilisateur (dimension PDA) (Wiebe *et al.*,



2016). La prédiction peut être correcte et dans ce cas susceptible d'améliorer l'activité de l'utilisateur, ou incorrecte et alors sujette à freiner l'activité de l'utilisateur (perturbation au niveau de la dimension PDA utilisateur). Dans ce cas l'utilisateur ne peut pas prévoir le comportement adaptatif et fera moins confiance au système (perturbation au niveau de la dimension ADP utilisateur).

**L'Autonomie** correspond à la capacité du système de décider (l'activité de décision) des actions d'adaptation. Lorsque le système est autonome, le contrôle est complètement interne. Toutes les actions sont effectuées par le système de manière autonome sans aucune intervention de la part de l'utilisateur.

**L'Auto-gestion système** est la capacité du système à faire face aux différents problèmes pouvant avoir lieu et de se protéger des différentes attaques et menaces sans recourir à une aide externe de la part de l'utilisateur ou d'une autre entité.

**L'Auto-optimisation** signifie que le système peut revisiter ses règles d'adaptation en prenant en compte de nouvelles exigences afin d'améliorer la qualité de ses prédictions.

**L'Auto-mise à jour** correspond à la capacité du système à toujours réviser les données stockées dans sa base de connaissance afin de l'enrichir avec de nouvelles données et éliminer des données obsolètes.

**La Décidabilité** se réfère à la capacité du système à décider quelle adaptation et comment la mettre en œuvre. Ces choix se feront en fonction soit des micro-tâches (manipulation de l'interface), soit des macro-tâches (objectif de l'utilisateur).

**Le retour d'information système (*System Feedback*)** fournit des informations à l'utilisateur suite à une action exécutée. Par exemple, le système informe l'utilisateur que son action a bien été prise en compte.

**Le retour d'information utilisateur (*User Feedback*)** est un retour d'information issu de l'utilisateur sur la qualité de l'adaptation. Il permet au système d'enrichir son adaptation. Par exemple, après une adaptation système, l'utilisateur l'évalue et indique son niveau de satisfaction (Cowie et al., 2001, Eisenstein et Puerta, 2000, 2001).

**La Prévisibilité (*Predictability*)** correspond à la capacité de l'utilisateur à prédire le comportement du système. L'anticipation du comportement du système par l'utilisateur est une propriété qui peut être renforcée par plusieurs facteurs ou propriétés système. En effet, l'utilisateur comprendra mieux le système si ce dernier est intelligible, transparent et stable.

Lorsque le système fournit des prédictions incorrectes, la prévisibilité utilisateur sera affectée.

**La Conscientisation (*Awareness*)** correspond à la capacité de l'utilisateur à prendre conscience du processus d'adaptation. Cela consiste pour l'utilisateur à être informé de l'ensemble des fonctionnalités proposées par le système. Cette prise de conscience peut être renforcée par l'intelligibilité du système et sa transparence.

**La Contrôlabilité** est la capacité de l'utilisateur à contrôler la dimension de l'interaction système et/ou la dimension d'adaptation système (Jameson, 2009, Wobbrock et al. 2011, Norman, 1991, Sischer, 2000, Weld *et al.*, 2003). Lorsque l'utilisateur n'a aucun contrôle, l'interaction et/ou l'adaptation sont entièrement pilotées par le système. La contrôlabilité de la dimension d'adaptation est essentielle car elle permet à l'utilisateur de participer au processus d'adaptation : en Apprentissage (pour spécifier au système ce qui est autorisé à capturer, interpréter et apprendre de l'utilisateur afin d'assurer la confidentialité), en Prédiction (pour contrôler les paramètres utilisés pour prédire l'adaptation, par exemple, via un apprentissage système), et en Décision (pour évaluer les propositions d'adaptation, accepter ou refuser une adaptation).

L'ensemble de ces critères de qualité de l'adaptation placé en sein du modèle est illustré dans la figure 20.

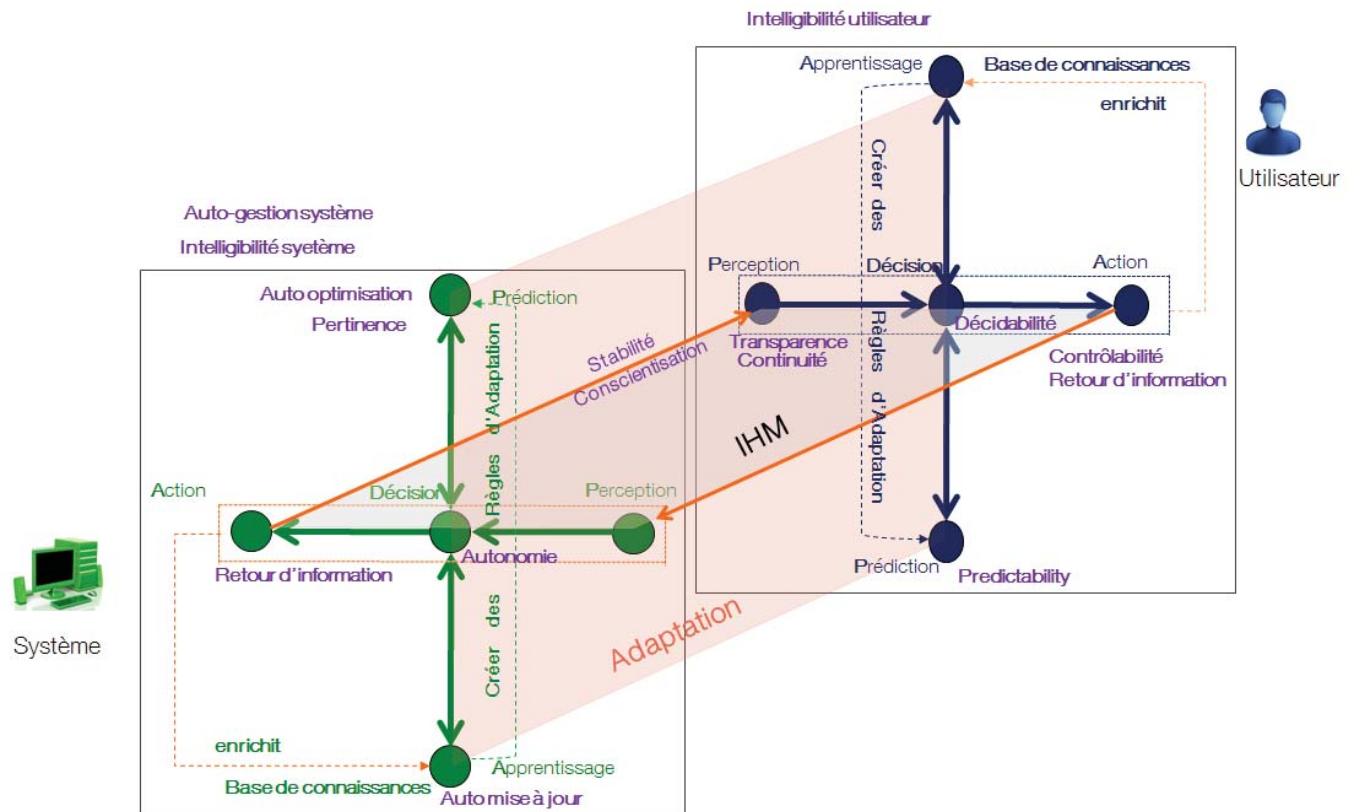


Figure 20. Critères de qualité au sein du modèle de l'interaction adaptative AIMS.

## 4. Conclusion

Dans ce chapitre, nous avons proposé un modèle pour l'interaction adaptative composé de deux dimensions essentielles : la dimension d'interaction et la dimension d'adaptation. Ce modèle décrit l'interaction et l'adaptation au niveau de l'entité utilisateur et de l'entité système. Nous avons placé sur cet espace les critères permettant d'assurer une boucle interactionnelle de qualité, c'est-à-dire une boucle facilitant l'ajustement entre les dimensions PDA système et PDA utilisateur. Nous avons également positionné sur cet espace les critères permettant d'assurer une adaptation de qualité, c'est-à-dire l'ajustement, via la boucle interactionnelle, entre les dimensions ADP système et ADP utilisateur.

## 5. Synthèse

Nous proposons **un modèle pour l'interaction adaptative appelé AIMS pour Adaptive Interaction Model with Symmetry**. Ce modèle suit une approche systémique : il intègre et unifie les deux entités d'interaction à savoir l'utilisateur et le système. Il décrit la dimension d'interaction **PDA** (Perception-Décision-Action) ainsi que la dimension d'adaptation **ADP** (Apprentissage-Décision-Prédiction) non seulement **au niveau du système** mais aussi de **l'utilisateur**.

Nous identifions au sein de ce modèle les éléments garants d'une boucle interactionnelle de qualité, permettant l'ajustement entre les dimensions PDA côté utilisateur et côté système, ainsi que les critères de qualité à mettre en œuvre au niveau de la boucle interactionnelle afin de faciliter l'ajustement des dimensions ADP côté utilisateur et système. Ces **critères de qualité** devront sous-tendre toute mise en œuvre de l'adaptation au sein d'une interface.

Indépendamment des propriétés ici présentées, leur caractérisation sur le modèle AIMS est intéressante, elle met en évidence des processus et phases peu considérés par exemple, l'observabilité système (P de la dimension PDA système) mériterait peut-être d'être intégrée.



## Chapitre 4 : PCA, modèle de présentation de l'adaptation

Ce chapitre a fait l'objet d'une publication en conférence internationale :

**Bouzit, S., Calvary, G, Chêne, D., Vanderdonckt, J. (2016).** A design Space for Engineering Graphical Adaptive Menus. Proceedings of the 8<sup>th</sup> ACM SIGCHI Symposium on Engineering Interactive Computing Systems. EICS'16, New York, NY, USA, 2016, ISBN: 978-1-4503-4322-0, pp 239-244.

L'adaptation est un processus de changement d'un état à un autre. Elle s'évalue par une **mesure d'écart** entre ces deux états : **avant** et **après** adaptation. La mise en œuvre de l'adaptation au niveau de l'interface doit respecter les critères de qualité vus dans le chapitre précédent. Ces derniers doivent réduire la charge perceptuelle, cognitive et motrice de l'utilisateur. La littérature ne propose pas de cadre structuré permettant de mettre en évidence toutes les variables possibles pour la présentation de l'adaptation.

Pour ce faire, nous nous basons sur les critères définis au sein du modèle AIMS (chapitre 3). Nous proposons un modèle constitué des axes relatifs aux variables visuelles utilisées pour présenter des objets d'interface, aux variables facilitant la compréhension et aux variables facilitant la manipulation. D'autres axes pourraient être ajoutés.

Ce chapitre est structuré comme suit : dans la section suivante nous définissons les exigences pour un modèle de présentation d'adaptation. Ensuite nous définissons le modèle PCA (Perception-Cognition-Action). Dans la section 3, nous représentons les techniques d'adaptation existantes au sein du modèle afin de mettre en évidence les axes de recherche inexplorés. Dans la section 4, nous proposons un outil (une application web) qui permet de générer des menus prenant en compte les variables visuelles du modèle.

## 1. Exigences

Le modèle de présentation des adaptations est construit selon les exigences suivantes :

Il doit être descriptif, comparatif et génératif. Nous détaillons ces trois exigences ci-après :

1. Le pouvoir descriptif doit permettre de décrire toute interface adaptative et un positionnement à toute approche existante ;
2. Le pouvoir comparatif doit permettre de mener une analyse critique comparative d'interfaces adaptatives et faire ressortir les aspects positifs et négatifs des approches existantes ;
3. Le pouvoir génératif doit permettre d'identifier les parties sous-explorées restant à étudier. Ainsi nous pourrions faire émerger de nouvelles opportunités.

## 2. Modèle PCA

Tout critère de qualité de l'adaptation (intelligibilité, continuité, transparence, etc.) va être mis en œuvre au niveau de l'interface à travers un ensemble de variables de rendu perceptibles, compréhensibles et manipulables par l'utilisateur. Nous étudions ici les différents aspects à considérer.

### 2.1. Objets de l'interface

L'axe « objets de l'interface » fait référence aux éléments sur lesquels l'adaptation peut être portée et mise en évidence. Les objets de l'interface peuvent être réduits à un ensemble de contenus et de conteneurs. Les contenus sont les items de commandes et les informations fournies à l'utilisateur. Les conteneurs sont les zones contenant ces contenus (fenêtre, cadre, palette). Lorsque l'adaptation est mise en œuvre, elle peut jouer soit sur les contenus (ex : ajout d'un raccourci pour les actions fréquentes) soit sur les conteneurs (ex : rajouter une palette de raccourcis des actions fréquentes).

L'adaptation des contenus consiste à mettre en avant les items susceptibles d'être intéressants pour l'utilisateur (**items prédits**), ou bien de mettre en arrière-plan les items moins intéressants pour l'utilisateur (**items non-prédits**). Dans ce cas par exemple on peut rétrécir la taille de ces items.

L'adaptation des conteneurs consiste à créer ou mettre en avant les zones contenant les items prédits (**zones prédites**), ou bien à cantonner dans des zones les items non-prédits (**zones non-prédites**).

Au-delà de la nature de ces objets, ce sont surtout les propriétés de rendus qui vont restituer à l'utilisateur la mise en œuvre de l'adaptation.

Nous explorons un ensemble de variables pouvant être utilisées afin de produire une présentation de l'adaptation qui couvre les propriétés clés. Nous observons que certains critères de qualité tels que la transparence, la continuité, et la stabilité permettent d'améliorer principalement la perception. La transparence permet de rendre tout changement des objets de l'interface transparent visuellement. De même la continuité évite que le changement soit brutal en donnant la possibilité à l'utilisateur de percevoir le processus de changement. En outre, les



propriétés telles que l'intelligibilité, la conscientisation (*awareness*) et la prévisibilité visent à réduire l'effort cognitif et à faciliter la compréhension de l'utilisateur. Or, les propriétés de feedback, contrôlabilité, etc., permettent de faciliter la manipulation de l'interface. Cela implique que les variables de présentation de l'adaptation doivent être capables d'exprimer l'adaptation sur les trois niveaux : perception, cognition et manipulation.

## 2.2. Perception

En 1967, Bertin a défini huit variables visuelles permettant de présenter de manière efficace un changement dans l'interface : elles concernent la variation de la localisation, de tailles, du format, de la saturation des couleurs (par exemple, passer du clair au foncé), d'orientation (alignement, angle), de couleurs, de textures, de mouvements (transitions animées, Florins et Vanderdonckt, 2004).

Ces huit variables visuelles sont des variables de référence dans toute conception visuelle, en visualisation d'information et en interaction homme-machine.

Nous avons classé ces variables en quatre catégories. Les variables qui jouent sur l'aspect **graphique**, celles qui concernent l'aspect spatial avec des positions différentes, l'aspect spatial avec orientation et celles qui jouent sur l'aspect temporel. Les variables taille, forme, valeur, couleur et texture permettent de mettre en évidence graphiquement les objets de l'interface. Les variables position, orientation jouent sur la **spatialisation** et permettent de changer la position des objets dans l'interface. La variable position joue sur la répartition en x,y,z des contenus et conteneurs, alors que la variable orientation joue sur l'agencement pour un même x,y,z de ces derniers. Les mouvements introduisent la notion du **temps** et peuvent jouer sur les objets de l'interface à la fois graphiquement et spatialement. Par conséquent, nous définissons quatre axes qui permettent de décrire visuellement un changement graphique, temporel, spatial avec des positions différentes, ou spatial avec une position constante mais des orientations différentes. Ces axes sont nommés Forme, Mouvement, Position et Orientation. Nous les décrivons ci-dessous.

### L'axe Forme

L'axe Forme représente l'écart entre la forme des objets de l'interface (items prédits, items non-prédits, zone prédite, zone non-prédite) avant et après l'adaptation. Le changement de la

forme de certains éléments de l'interface a pour objectif de les mettre en évidence et d'attirer l'attention de l'utilisateur à ces éléments afin de réduire le temps de recherche visuelle. L'axe forme se compose de plusieurs variables.

- Couleur : comme définie par Bertin, cette variable consiste à modifier la couleur du texte et/ou du fond des éléments. La couleur peut donc être appliquée sur le texte de l'élément prédit et/ou sur la zone de l'élément prédit. Une ou plusieurs couleurs peuvent être définies. Par exemple dans la technique d'adaptation *highlighting* (Tsandilas *et al.*, 2005), une couleur de fond différente a été utilisée permettant de mettre en évidence les éléments prédits. Dans le menu *Heatmap*, plusieurs couleurs ont été utilisées en fonction de la fréquence d'utilisation des items.
- Gras : il permet de mettre en évidence certains éléments intéressants afin d'attirer l'attention de l'utilisateur sur ces éléments. Différents niveaux de gras peuvent être utilisés.
- Texture : plusieurs types de textures peuvent être utilisés.
- Encadrement : il consiste à encadrer les éléments prédits et/ou non prédits.
- Taille : la variable taille peut être décomposée en quatre sous-variables en fonction du nombre de dimensions utilisées pour le changement de taille : ligne (pour un changement à une dimension), surface (pour un changement en deux dimensions), multi surfaces et volume (pour un changement en trois dimensions). Le changement de taille peut être appliqué aussi bien sur la police des items prédits et/ou non prédits, que sur la zone prédite et/ou non prédite.
- Police : par exemple il est possible d'utiliser une police moins lisible pour les éléments non-prédits.
- Format : cela consiste à changer le format de la zone prédite et/ou non prédite, par exemple, passer d'une forme rectangulaire à une forme circulaire.

### **L'axe Mouvement**

L'axe Mouvement concerne l'animation des objets de l'interface (Huhtala *et al.*, 2009, 2010, Ponsard *et al.*, 2015). Avec le mouvement ou animation plusieurs critères de qualité peuvent être satisfaits comme la continuité ou la transparence ce qui améliore l'intelligibilité. La transition ou animation peut prendre plusieurs formes (Rogers et Iba, 2000) :

- Continue : lorsque le processus d'animation des éléments est continu dans le temps.
- Discontinue : lorsque la transition se fait de manière brusque après un certain délai.
- Avec répétition : lorsque le processus de transition ou d'animation se fait de manière répétitive.
- Sans répétition : lorsque le processus d'animation ne se répète pas dans le temps.

La variable mouvement peut être encore décomposée en sous-variables, telles que la direction, la vitesse, la fréquence, le rythme et le style (Carpendale, 2003).

### **L'axe Position**

L'axe Position dénote l'écart entre la position des objets de l'interface avant et après adaptation. Les objets peuvent changer de position. Par exemple, déplacer certains éléments en haut de page, déplacer les éléments en bas, décaler les éléments vers la gauche ou vers la droite, etc. La variable position est explorée dans le menu basé sur la fréquence (*frequency based*, Mitchell, 1989). Les items prédits sont déplacés en haut du menu. Réduire le changement de position des objets de l'interface permet d'augmenter la stabilité. En effet, le critère de prévisibilité va être augmenté dans la mesure où une interface stable permet à l'utilisateur d'anticiper plus facilement le comportement adaptatif.

### **L'axe Orientation**

L'axe Orientation consiste à changer l'alignement de certains objets, contenus ou conteneurs. Plusieurs types d'orientations peuvent être définis tels que horizontale, verticale, profondeur, etc.

## **2.3. Cognition**

Une adaptation doit être compréhensible par l'utilisateur et le changement ne doit pas induire un effort cognitif. Nous explorons un ensemble de variables cognitives facilitant la compréhension et l'apprentissage de l'adaptation. Elles permettent d'augmenter certains critères de qualité tels que l'intelligibilité, la prévisibilité et l'awareness. Ces variables sont distribuées sur les axes suivants : Ajout, Elimination, Explication et Moment.

### **L'axe Ajout**

L'axe Ajout consiste à ajouter des éléments dans l'interface. Il peut s'agir d'items prédits et/ou d'items non prédits mais aussi de zones contenant ces items (zones prédites et/ou non-prédites).

L'ajout peut se faire avec répllication, ce qui consiste à répliquer certains éléments pour les mettre en avant et souligner leur importance. Par exemple, l'interface *split* avec répllication (Gajos, 2006) où les items prédits se trouvent à la fois dans la partie adaptative et la partie statique.

L'ajout peut se faire sans répllication ce qui consiste à ajouter des éléments dans l'interface sans les répéter.

### **L'axe Elimination**

L'axe élimination consiste à rendre certains éléments de l'interface inaccessibles. Cela permet d'augmenter la perception et faciliter la manipulation car l'interface est moins encombrée. L'élimination peut être utilisée dans les interfaces à fonctionnalités réduites. Ces interfaces sont optimales notamment pour les novices et facilitent l'apprentissage (Caroll et Carrithers, 1984, Kang *et al.*, 2010). L'interface s'enrichit alors au fur et à mesure en fonction de l'avancement et du niveau d'expertise atteint par l'utilisateur. L'élimination des objets de l'interface peut être faite de différentes manières :

Le masquage consiste à conserver les éléments en les rendant inactifs.

La suppression consiste à les enlever.

### **L'axe Explication**

L'axe Explication permet d'informer l'utilisateur sur les adaptations faites au niveau de l'interface. Les explications peuvent être fournies textuellement, par des images et graphiques, par des sons et vidéos, ou par une combinaison de possibilités (texte, graphique, son, etc.). Expliquer l'interface et le processus d'adaptation permet à l'utilisateur de mieux comprendre le système et ses changements et d'anticiper son comportement.

### **L'axe Moment**

L'axe Moment détermine à quel instant le changement adaptatif est effectué (Schlee et Vanderdonckt, 2004). L'axe temporel peut être considéré selon plusieurs échelles : L'année, le mois, la journée, l'heure, voire l'instant présent.

Le moment est très important car il faut pousser la bonne adaptation au bon moment. Par exemple dans le cas des novices utilisant une interface à fonctionnalités réduites, il faut choisir le moment opportun et s'assurer que l'utilisateur ait atteint un niveau d'expertise suffisant pour enrichir l'interface avec de nouvelles fonctionnalités. De plus, cela doit être fait en évitant que l'activité de l'utilisateur soit interrompue ou perturbée.

Proposer une adaptation au début de la tâche consiste à présenter le changement au lancement d'une application, d'un service.

Proposer une adaptation en cours de tâche consiste à changer ou adapter l'interface durant l'usage.

Proposer une adaptation en fin de tâche consiste à adapter l'interface à l'issue de l'interaction.

## **2.4. Action**

Au-delà de la perception et de l'aspect cognition, la manipulation est une caractéristique importante de l'interaction. Simplifier la manipulation et faciliter l'accès aux objets de l'interface sont des sources d'adaptation pouvant être utiles pour l'utilisateur. Par exemple, présenter des raccourcis d'action, proposer de piloter l'interface par différentes modalités d'interaction vont permettre à l'utilisateur de fiabiliser et d'accélérer l'action.

### **L'axe Quantité de prédiction**

L'axe Quantité de prédiction est le nombre des éléments prédits présentés dans l'interface. Dans la plupart des techniques d'adaptation, cette quantité est inférieure ou égale à trois prédictions. Selon la pertinence de la prédiction (*accuracy*), plus ou moins d'items prédits seront présentés à l'utilisateur.

### **L'axe Modalité**

L'axe Modalité représente la manière dont les objets de l'interface vont être présentés et manipulés. La modalité de présentation permet de mettre en évidence les éléments soit

graphiquement en changeant leur forme, soit vocalement par des messages vocaux ou des sons, ou bien par une combinaison de plusieurs modalités.

Les modalités de commande peuvent également être adaptées à l'utilisateur en fonction du contexte d'usage, de ses préférences et de ses capacités physiques. Présenter à l'utilisateur une bonne modalité de commande lui permet de mieux contrôler l'interface et de faciliter l'interaction.

### **L'axe Contrôle**

L'axe Contrôle donne à l'utilisateur les moyens de maîtriser le comportement adaptatif lui-même ou sa présentation.

Au niveau du rendu, l'utilisateur peut contrôler la présentation des objets de l'interface (format, position, orientation, transition).

Au-delà du rendu, l'utilisateur a la possibilité de contrôler le contenu de prédiction et d'adaptation. Par exemple, afficher ou faire disparaître une prédiction, déclarer un élément prédit comme non-pertinent, augmenter ou réduire la quantité des données prédites, spécifier le moment de présentation de l'adaptation, aider le système à améliorer la qualité des prédictions.

Au niveau de la manipulation, l'utilisateur a la possibilité de contrôler le mode de présentation et d'exécution de la prédiction ainsi que la modalité de commande d'adaptation.

Le modèle PCA est constitué de plusieurs axes et variables permettant de décrire la présentation de l'adaptation à différents niveaux. Au niveau de la *Perception* où les variables visuelles de Bertin sont explorées, *au niveau Cognitif et au niveau de la manipulation ou Action*. Les variables explorées permettent d'assurer les critères de qualité pour l'interaction et l'adaptation. Ce modèle peut être étendu et augmenté par de nouveaux axes où chaque axe peut également être augmenté par de nouvelles variables. Nous décrivons le modèle dans la figure 21.

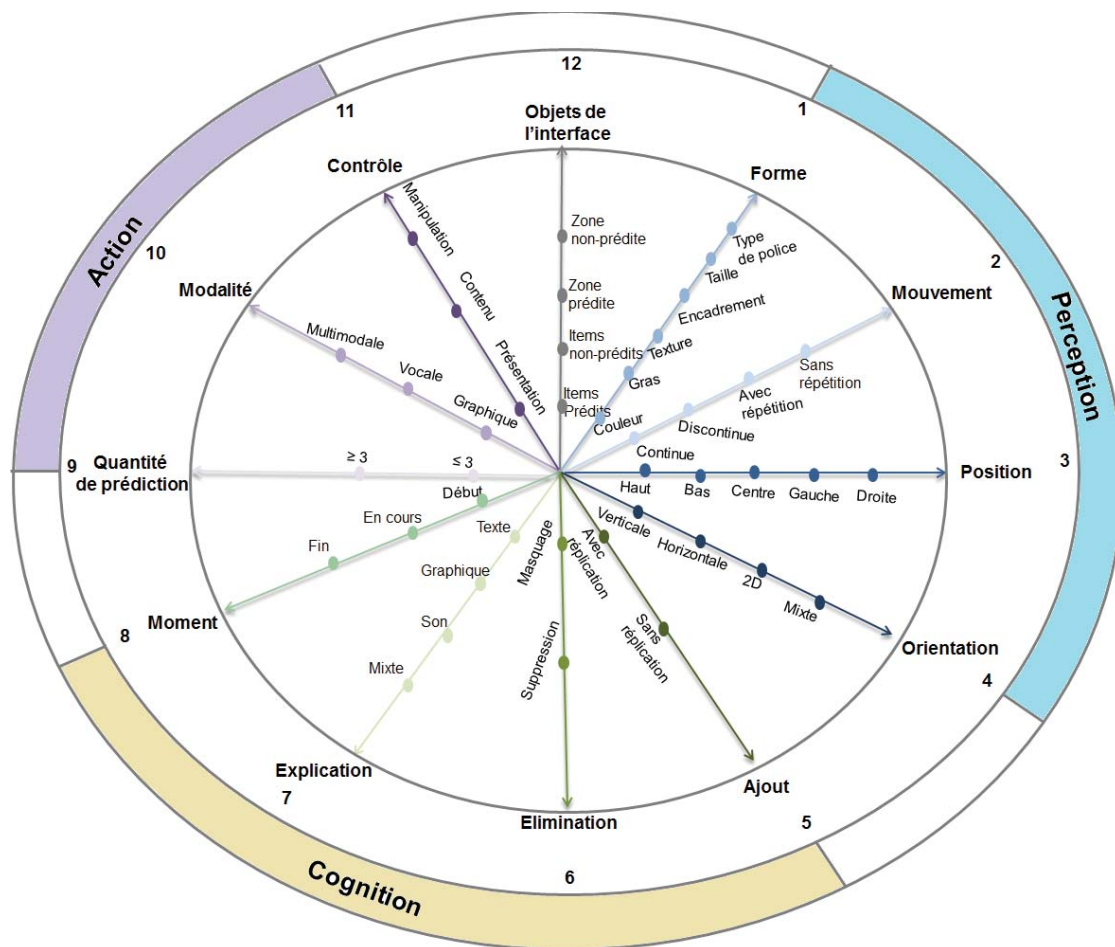


Figure 21. PCA modèle de présentation de l'adaptation.

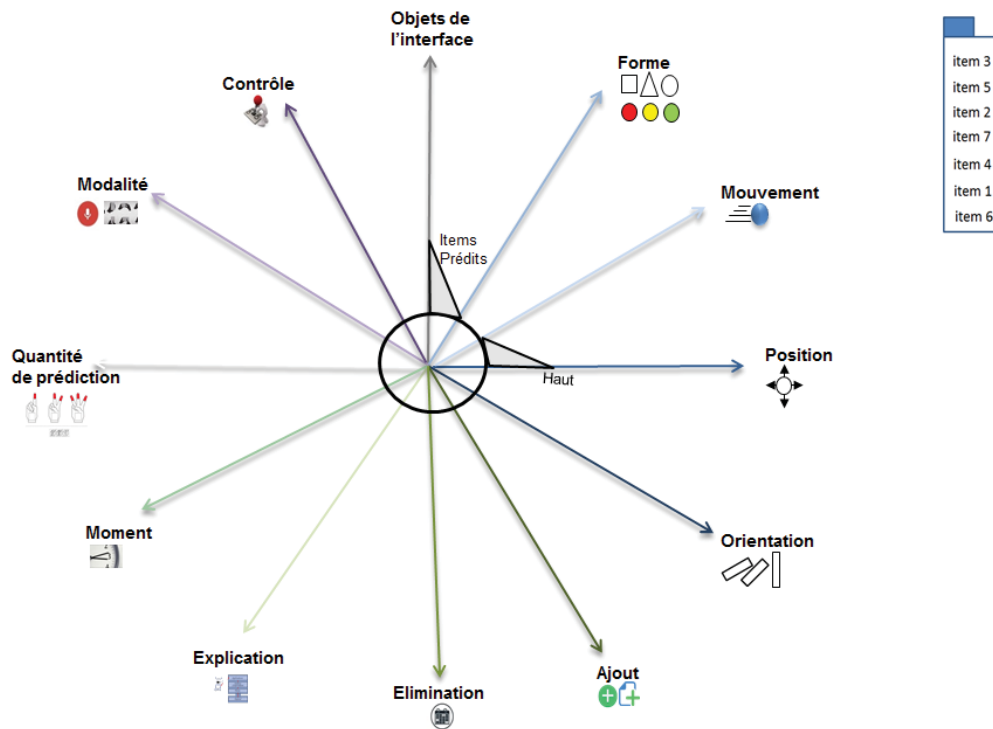
### 3. Caractérisation de l'état de l'art

Dans cette section, nous caractérisons quelques exemples d'interfaces adaptatives au sein de PCA. L'objectif est de montrer où se situent les techniques existantes et de mettre en évidence les axes peu explorés.

#### 3.1. Position

Dans la technique *Frequency-based menu* (Mitchell et Shneiderman, 1989), l'axe position a été exploré. Les items les plus fréquents sont présentés en haut du menu, tandis que les items

les moins fréquents sont affichés en bas du menu. Les auteurs se sont basés sur l'hypothèse que les items les plus fréquents vont être sélectionnés plus rapidement que les items les moins fréquents. Dans ce menu, la variable position change fréquemment et cette technique ne maintient pas la stabilité spatiale. La caractérisation PCA du menu basé sur la fréquence est indiquée dans la figure 22.



Les menus *split* (Sears et Shneiderman, 1994) séparent graphiquement le menu en deux parties. La fenêtre de prédiction (la zone en haut du menu contient une petite liste d'items prédits, typiquement 2 ou 3) et une partie statique qui est la seconde zone contenant les items non-prédits. Cette séparation ne conserve pas la stabilité spatiale du menu. La position des items au sein de la partie statique change. Les menus *split* ont été fortement critiqués et une variante a été créée. Les menus *split* avec réplique (Cockburn, 2007) ont rétabli la stabilité spatiale en conservant au sein de partie statique tous les items, ce qui amène à dupliquer ceux qui sont également prédits. L'axe Ajout de notre modèle nous permet d'exprimer cette technique.



La caractérisation PCA du menu *split* avec réplification est décrite dans la figure 23.

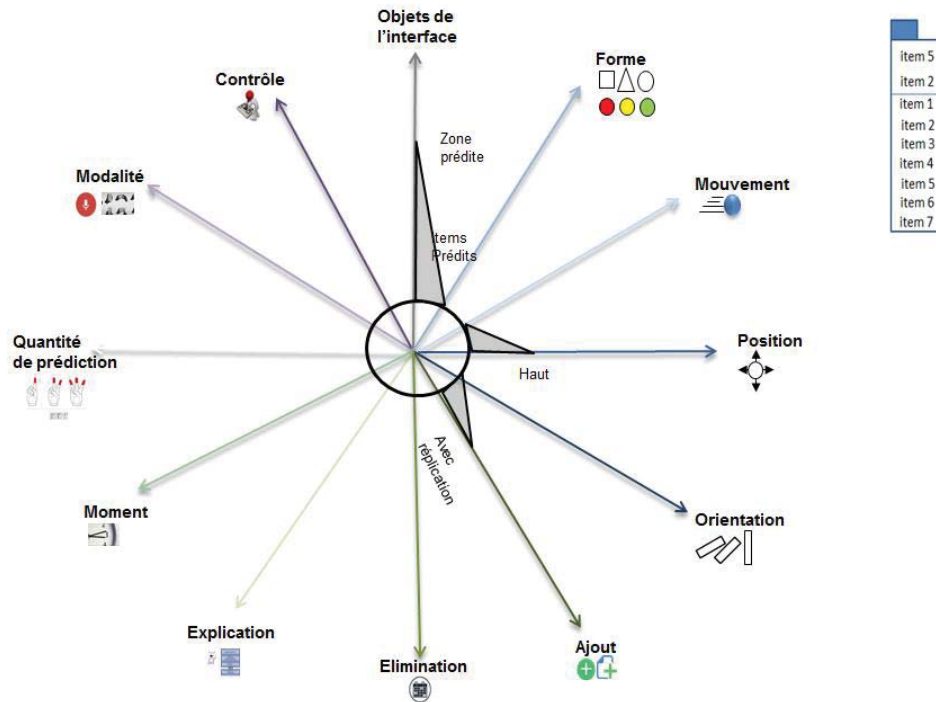


Figure 23. Caractérisation PCA de la technique Split avec réplification.

### 3.2. Taille

Le menu *Morphing* (Cockburn, 2007) joue sur la variable visuelle de la taille des polices. La taille de police est augmentée ou inversement réduite si le niveau de précision de prédiction est élevé ou respectivement faible. Cette variable peut être exprimée sur l'axe Forme de notre modèle. La même chose s'applique pour le menu *fish-eye* (Bederson, 2000) où la taille des éléments dépend de la position du curseur. Toute la zone entourant l'item est également agrandie. Ainsi seule la position absolue de l'item est modifiée mais pas l'ordre des items. Le principe est le même pour *Adaptive Activation Area Menu* (Tanvir et al., 2008) où la variable taille a été également utilisée pour la zone des items prédits. Cette zone est agrandie pour faciliter le pointage. La caractérisation PCA des techniques basées sur le changement de taille est illustrée dans la figure 24.

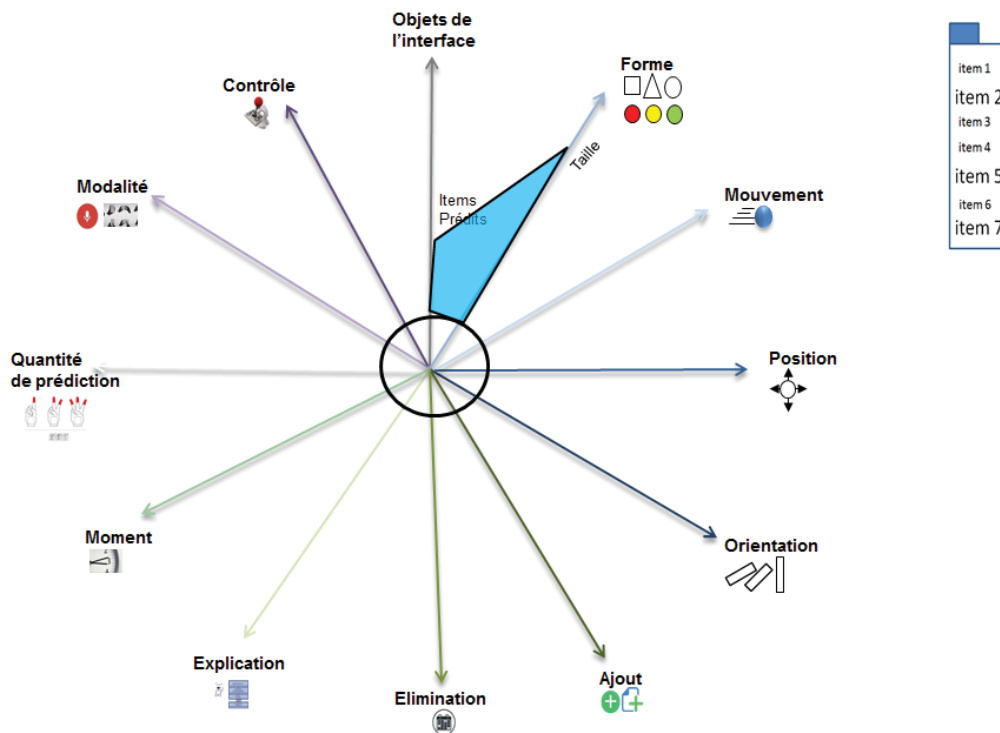


Figure 24. Caractérisation PCA des techniques basées sur le changement de taille.

### 3.3. Couleur

Le menu *Highlighting* (Tsadilas et Schraefel, 2005) utilise la variable couleur. Une couleur différente a été appliquée au fond de la zone des items prédits pour les mettre en valeur. Cette variable peut être exprimée sur l'axe Forme de notre modèle.

Les variables explorées dans le menu *highlighting* sont décrites dans la figure 25.

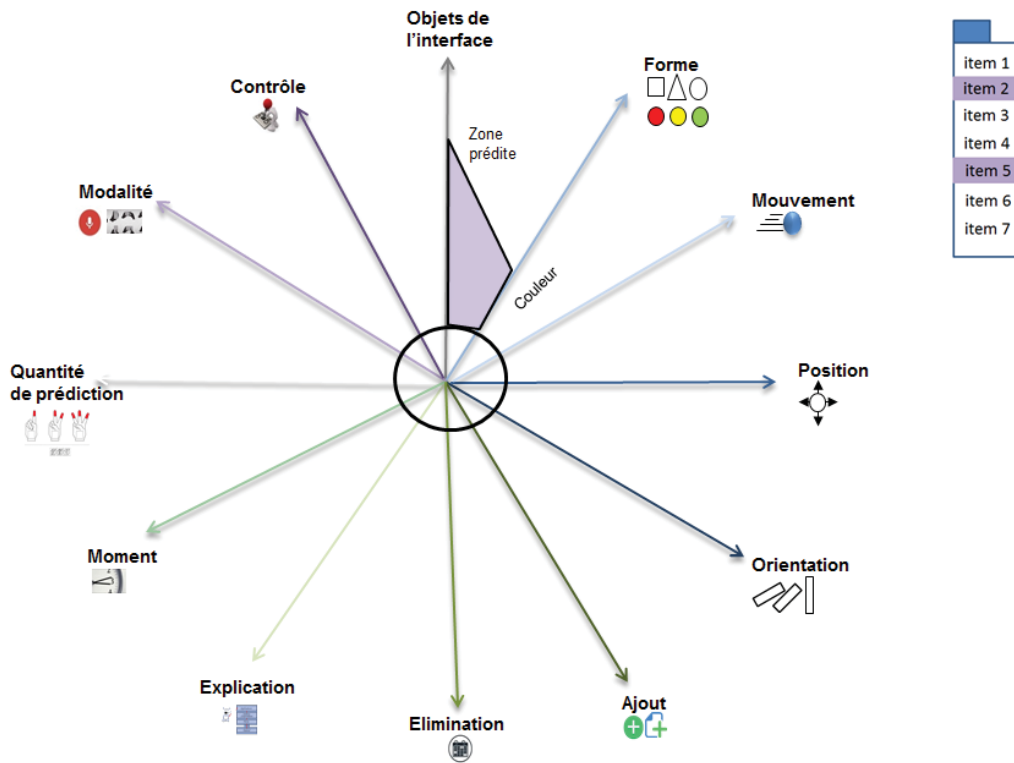


Figure 25. Caractérisation PCA de la technique Highlighting.

### 3.4. Mouvement

Le *temporal* menu (Lee et Yoon, 2004) se caractérise sur les axes Mouvement et Ajout. C'est par un mouvement discontinu que s'ajoutent brusquement les items non-prédits après un délai de 170ms. Les variables explorées dans le menu temporel sont décrites dans la figure 26.

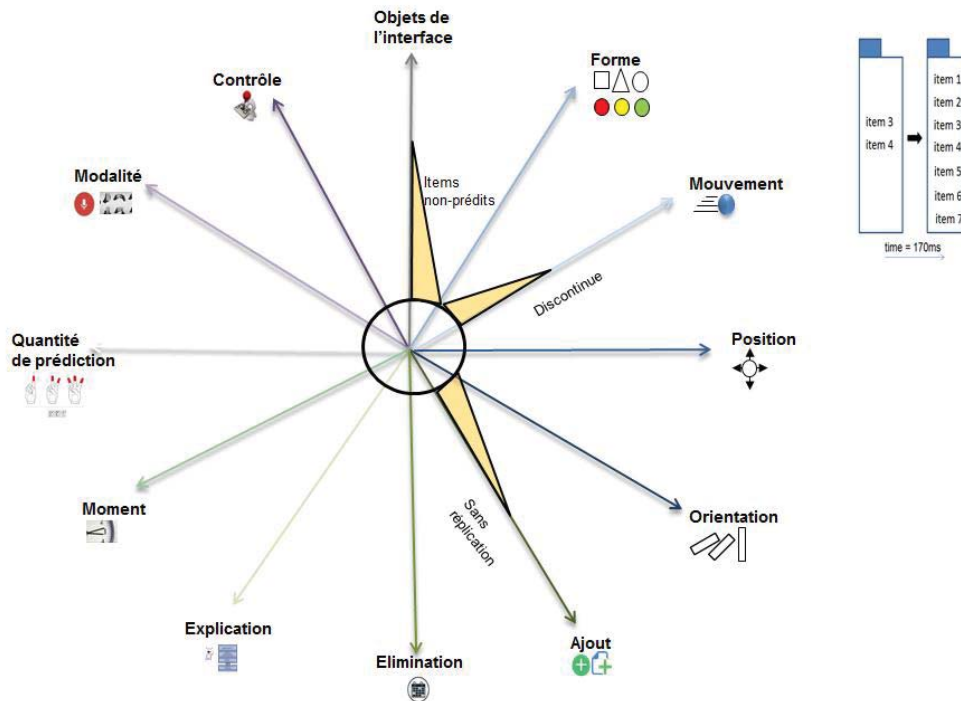


Figure 26. Caractérisation PCA de la technique du menu temporel.

Dans le menu éphémère (Findlater *et al.*, 2009) les axes Mouvement (avec une animation continue) et Ajout (sans répliation) ont été utilisés pour afficher les items non-prédits. Les variables explorées dans le menu éphémère sont illustrées dans la figure 27.

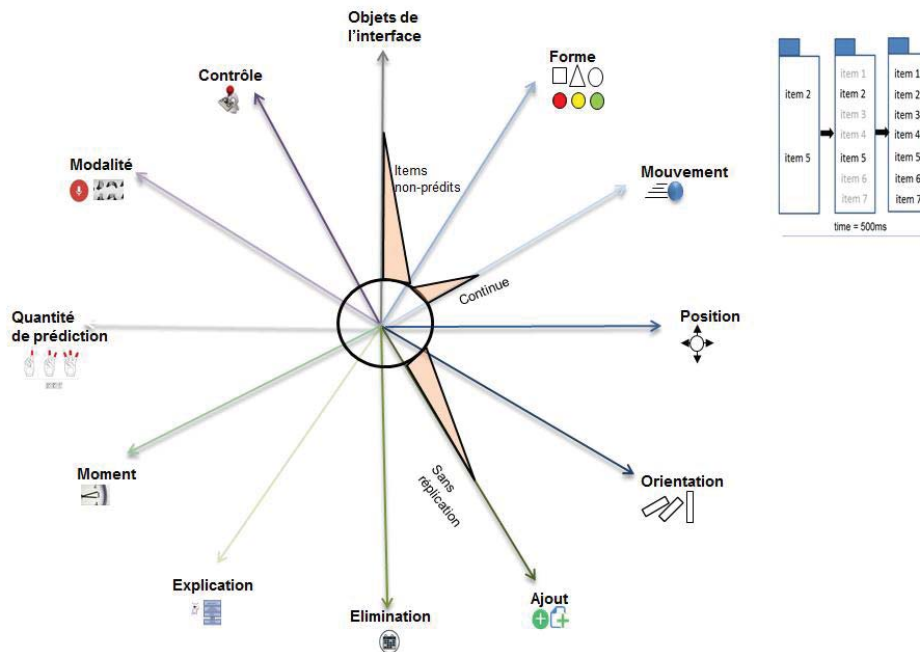


Figure 27. Caractérisation PCA de la technique d'adaptation éphémère.

#### 4. Comparaison et génération de techniques

Les techniques d'adaptation existantes se focalisent principalement sur les variables visuelles de présentation telles que la forme, la position, le mouvement et l'ajout. Nous avons superposé sur la figure 28 certaines des approches actuelles. La majeure partie d'entre elles sont centrées sur les variables visuelles. La variable « haut » de l'axe Position, la variable « continue » de l'axe Mouvement, les variables « couleur » et « taille » de l'axe Forme sont le plus souvent exploitées.

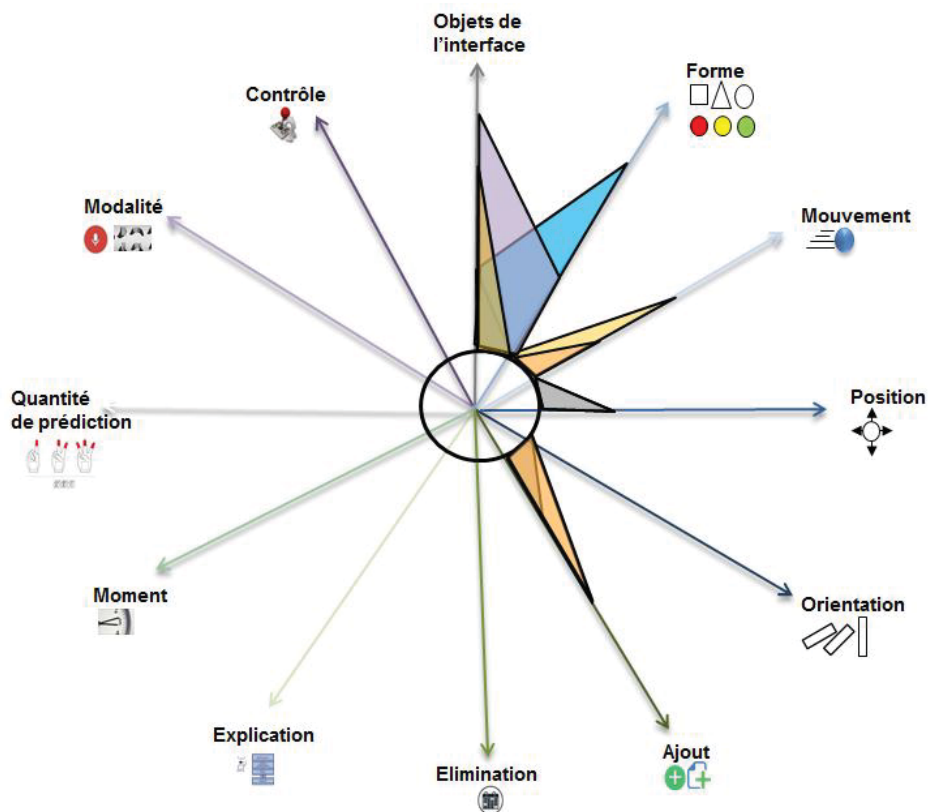


Figure 28. Caractérisation PCA comparative des techniques adaptatives.

Pourtant il n'existe pas de techniques les cumulant toutes. Les variables visuelles étant indépendantes les unes des autres, il est tout à fait possible de les cumuler pour renforcer la présentation de l'adaptation. On peut utiliser, par exemple, plusieurs couleurs pour exprimer la fréquence d'usage des items tout en jouant sur la position pour exprimer la probabilité de prédiction. Plusieurs informations peuvent être ainsi présentées à l'utilisateur via des variables cumulées.

La variable Orientation quant à elle pourrait être davantage exploitée. Par exemple, il est possible de jouer sur l'orientation de l'étiquette pour faciliter l'accès aux items les plus prédits et contraindre l'accès à ceux qui le sont moins (en les rendant moins lisibles).

Néanmoins, tous ces axes visuels sont fortement contraints dès lors qu'on considère un petit écran (*smartphone*). Dans ce cadre, explorer les autres axes Orientation, Elimination, Explication, Quantité de prédiction, Contrôle et Modalité pourrait être une opportunité.

## 5. Support logiciel pour le prototypage rapide

Le modèle PCA explicite un nombre conséquent de dimensions. Il est important d'accélérer l'exploration de cet espace par un outil. Nous avons développé un démonstrateur de concept d'un outil logiciel PCA-S (PCA-Support) pour la modélisation et la simulation rapides de présentations de menus adaptatifs graphiques. PCA-S est une application web en HTML5/JavaScript permettant d'explorer les options de conception de menus graphiques. À tout moment, un concepteur ou un développeur peut spécifier la variable visuelle à utiliser et ainsi créer une simulation visuelle du menu avant qu'il ne soit développé. L'environnement permet de générer différents types de menus en fonction des paramètres sélectionnés. À l'ouverture de l'application, l'utilisateur commence par introduire l'ensemble des items constituant le menu. Ensuite, plusieurs paramètres peuvent être spécifiés :

- *Border* : spécifie le type de bordure de menu à afficher (*solid*, *dashed*, *dotted*, *double*).
- *Layout background* : spécifie la couleur du fond du menu à partir d'une palette de couleurs.
- *Font size* : spécifie la taille de police à utiliser parmi une famille de différentes tailles. Lorsque l'utilisateur sélectionne une case à cocher, par exemple « 10px », un message apparaît pour inciter l'utilisateur à introduire le(s) item(s) à être modifiés.
- *Item background* : spécifie la couleur du fond mais uniquement de l'item sélectionné, contrairement à la propriété « *Layout background* » qui change la couleur du fond du menu entier.
- *Position* : spécifie si les items du menu peuvent être disposés dans une dimension (par exemple : horizontale, verticale), deux dimensions (par exemple : dans un grand menu rectangulaire), trois dimensions (par exemple dans un menu de type *Cloud 3D*).

Lorsque tous les paramètres désirés sont spécifiés, le menu est généré dans une zone d'aperçu (Figure 29). Par exemple, l'utilisateur introduit dans la première étape six items (item 1, item 2, item 3, item 4, item 5, item 6). Il sélectionne ensuite la bordure de type « *solid* ». Il définit le « blanc » comme couleur de fond. Supposons que l'utilisateur souhaite changer la taille de police pour un item, il choisit « 50px » pour l'item 3, ainsi,

qu'une couleur de fond (*Item background*) « bleu » pour ce même item. Le menu généré va être composé des items suivants : item 1, item 2, item 3, item 4, item 5, item 6. Le menu va prendre une couleur de fond blanche. L'item 3 va être affiché avec une taille de police grande et une couleur de fond bleue. Ce menu pourrait être considéré comme une instance du menu *fish-eye* coloré.

The image shows a configuration interface for a menu. It consists of several panels on the left and a preview window on the right.

- Layout:** Includes a checked 'List' checkbox, an 'Add text' button, and a list of items (item 1 to item 5).
- Prediction Layout:** Includes an unchecked 'List' checkbox.
- Border:** Offers four border styles: solid, dashed, dotted, and double, each with a radio button.
- Background:** Offers six color options: Blue, Grey, Red, Brown, Green, and White, each with a radio button.
- Animation:** Offers three animation options: FadeIn, FadeOut, and Transition, each with a radio button.
- Font size:** Offers six font size options: 10px, 20px, 30px, 40px, 50px, and 60px, each with a checkbox. The 60px option is checked. Includes an 'Add text' button.
- Item Background:** Offers six color options: Red, Green, Blue, Pink, Prange, and Yellow, each with a checkbox. The Blue option is checked. Includes an 'Add text' button.

The preview window on the right shows a list of items:

- item 1
- item 2
- item 3 (highlighted with a blue background)
- item 4
- item 5

Figure 29. Conception d'un menu en se basant sur quelques variables visuelles.

L'utilisateur a également la possibilité d'ajouter une zone de prédiction dans laquelle les items prédits peuvent être regroupés et affichés. Pour cela, l'utilisateur sélectionne la case à cocher « *Prediction text layout* ». Il sera interrogé pour introduire les items à afficher dans cette zone. De même, il pourra changer les paramètres d'affichage de cette zone de prédiction. La propriété « *Border* » permet de changer la bordure de la zone de prédiction. La propriété « *Background* » permet d'attribuer ou de changer la couleur de fond de la



zone de prédiction. Aussi, des techniques d'animation pourraient être appliquées à la zone de prédiction, telles que : animation *FadeIn*, *FadeOut*, ou transitions animées (Cockburn *et al.*, 2007). En se basant sur le menu généré dans le Figure 29, l'utilisateur peut ajouter une zone de prédiction en introduisant les items 2 et 4. Il peut aussi définir une bordure de type « *dashed* » pour la zone de prédiction. Pour la couleur de fond de la zone de prédiction, l'utilisateur choisit par exemple le « gris » et *FadeOut* comme animation. Le menu généré est reproduit sur la figure 30 (à droite). La fenêtre de prédiction va être ajoutée par-dessus le menu initial et disparaît progressivement.



Figure 30. Conception d'un nouveau menu avec ajout de la fenêtre de prédiction.

## 6. Conclusion

Dans ce chapitre, nous avons décrit PCA, un modèle de présentation de l'adaptation. Ce modèle est constitué des dimensions perception, cognition, action. Les axes relatifs à la dimension perceptive se basent principalement sur les variables visuelles de Bertin. Notre modèle propose également des axes relatifs aux dimensions cognition et action. Il est descriptif et génératif dans la mesure où il met à plat tous les choix possibles et permet de générer de nouvelles idées.

La caractérisation au sein du modèle PCA des différentes techniques d'adaptation existantes montre que la plupart d'entre elles se concentrent sur les axes Forme, Position, Mouvement et Ajout. Les autres axes qui sont peu ou pas explorés. C'est le cas de l'axe Contrôle dont l'importance a pourtant été soulignée par de nombreux chercheurs (Norman, 1994, Jameson, 2009). De même, l'axe Explication a été peu traité. Enfin, l'axe Quantité de prédiction reste peu exploré. Il est généralement cantonné à deux ou trois items prédits.

Aussi, nous accompagnons le modèle PCA du support logiciel PCA-S, une application web permettant de générer des menus en fonction des variables sélectionnées. Cet outil logiciel facilitera l'exploration et l'évaluation d'alternatives de conception.

## 7. Synthèse

Le modèle de présentation de l'adaptation PCA (Perception-Cognition-Action), permet la caractérisation de rendu adaptatif. Il décrit les propriétés d'une interaction adaptative. Il prend en compte les critères de qualité d'une bonne adaptation (dimension ADP, chapitre 3) et d'une bonne interaction (dimension PDA, chapitre 3).

Le modèle se veut :

- **expressif** pour permettre une caractérisation de toute approche existante ;
- **comparatif** pour discuter des variantes de conception ;
- **génératif** pour permettre d'innover et de créer de nouvelles variantes.

Il est outillé par le logiciel PCA-S, un démonstrateur de concept pour le prototypage rapide de menus.



## **PARTIE III : EXPLORATION DE L'ESPACE DE CONCEPTION SUR PETITS ECRANS**

Dans cette partie, nous illustrons le pouvoir génératif du modèle PCA. Nous favorisons l'exploration des axes qui permettent de présenter l'adaptation dans le contexte de petits écrans, tels que les axes Elimination, Ajout et Mouvement. La technique « adaptation éphémère avec disparition » présentée au chapitre 5 met en œuvre ces trois axes.

L'axe Contrôle est étudié au chapitre 6 via la variante « adaptation éphémère avec disparition contrôlée par l'utilisateur ».

Nous étudions au chapitre 7 dans le menu « *cloud* » les axes Quantité de prédiction et Forme dans l'objectif de présenter un plus grand nombre d'items prédits.

Au chapitre 8, nous avons présentons deux techniques d'adaptations « *step-by-step* » et « *shortcut* » qui jouent sur l'axe Moment. Ces deux techniques concilient accélération et apprentissage.

Enfin, nous explorons l'axe Modalité au chapitre 9. Nous ouvrons les possibilités de rendu et de commande en jouant sur les variables graphique, vocale, gestuelle via le menu «polymodal ».

## **Chapitre 5 : Adaptation éphémère avec disparition, axes Elimination, Ajout et Mouvement**

Ce chapitre a fait l'objet d'une publication en conférence internationale :

**Bouzit, S., Chêne, D., and Calvary, G. (2014).** From Appearing to Disappearing Ephemeral Adaptation for Small Screens. *Proc. of the 26<sup>th</sup> Australian Computer-Human Interaction Conference Oz-CHI'2014* (Sydney, December 2-5, 2014). ACM Press, New York, pp. 41–48.

L'état de l'art souligne que les interfaces adaptatives se répartissent en trois grandes classes de techniques spatiales, graphiques et temporelles (voir chapitre 2). Lorsque la prédiction est correcte, ces techniques accélèrent car elles mettent en avant spatialement, graphiquement ou temporellement les éléments recherchés par l'utilisateur. Cependant, lorsque l'adaptation est incorrecte, ces techniques ralentissent l'utilisateur. En effet, un changement de position des items (techniques spatiales) perturbe l'utilisateur surtout lorsque la prédiction est incorrecte. De même, une surcharge visuelle (techniques graphiques) dégrade inutilement la perception en cas de prédiction incorrecte. Enfin, une interruption de tâche (techniques temporelles) est inacceptable dans le cas d'adaptation inappropriée. Tout cela peut impacter négativement aussi bien les capacités perceptives, cognitives et actionnelles de l'utilisateur.

De plus, l'émergence des petits écrans génère de nouvelles contraintes graphiques et spatiales qui imposent de durcir les critères de présentation de l'adaptation dans ce contexte.

Nous proposons ci-après le principe de l'adaptation éphémère avec disparition. La figure 31 illustre les axes et variables explorés : Mouvement, Ajout et Elimination. Ces choix se justifient par la surface limitée qui réduit d'autant la possibilité d'affichage d'items prédits. En cas d'erreur de prédiction, le défi est de ne pas ralentir l'interaction. Nous présentons deux techniques d'adaptation appelées *In Context Disappearing* et *Out of Context Disappearing* appliquées à des menus sur *smartphone*. Nous présentons l'étude expérimentale ainsi que les résultats obtenus. Enfin, nous concluons sur les préconisations résultantes et proposons de nouvelles pistes de recherche.

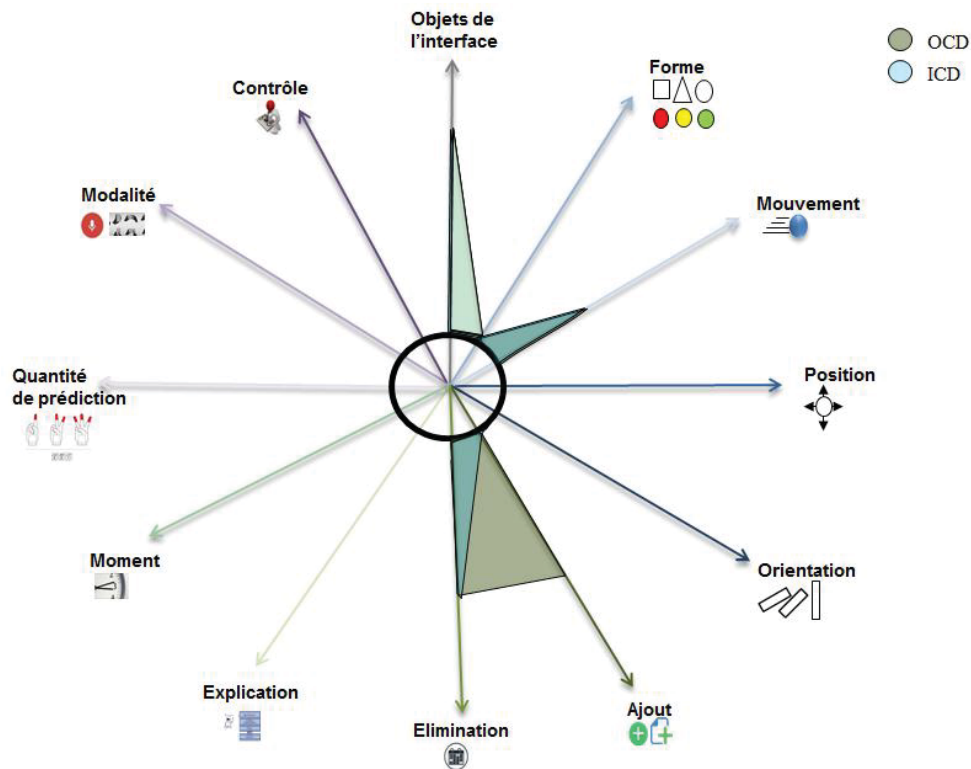


Figure 31. Axes explorés dans l'adaptation éphémère avec disparition.

## 1. Techniques étudiées

Nous présentons les deux techniques ICD et OCD.

### 1.1. In Context Disappearing (ICD)

*In Context Disappearing* (ICD) est une technique « focus + contexte ». Le principe est de garder le contexte (qui est la liste complète des items) visible, accessible et intact à toutes les étapes d'interaction. ICD permet d'afficher les items prédits dans une fenêtre d'incitation présentée simultanément à la liste complète (elle est affichée par-dessus). La fenêtre d'incitation contient trois items prédits. Elle disparaît progressivement dans un délai de 5000 ms. La figure 32 montre le principe de la technique ICD.

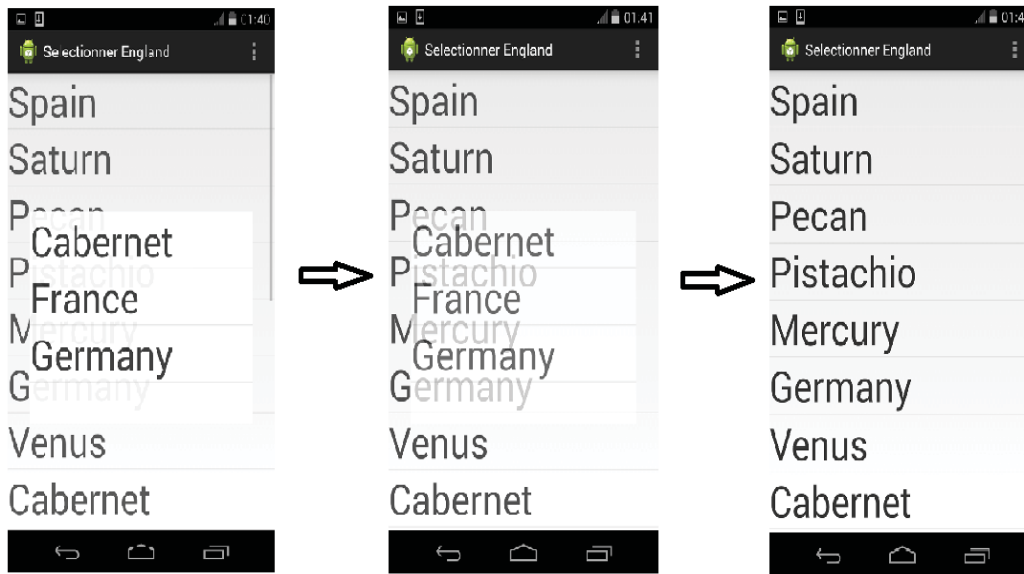


Figure 32. Technique ICD appliquée à des menus sur *smartphone*.

À l'ouverture du menu, l'utilisateur trouve deux listes superposées. Il cherche sa cible sur la fenêtre d'incitation. Dans le cas d'une prédiction correcte, la cible sera présente sur la fenêtre d'incitation. Dans ce cas, l'utilisateur la sélectionne directement. Par contre, dans le cas d'une prédiction incorrecte, la cible ne se trouve pas sur la fenêtre de prédiction. L'utilisateur peut alors naviguer dans la liste complète des items sans attendre la disparition complète de la fenêtre d'incitation. Il peut sélectionner tout item non masqué par la fenêtre d'incitation.

L'objectif de la disparition progressive de la fenêtre de prédiction (axes Elimination et Mouvement) est de ne pas pénaliser l'interaction de l'utilisateur dans le cas où la prédiction est incorrecte.

## 1.2. Out of Context Disappearing (OCD)

La technique d'adaptation OCD est une combinaison de la disparition progressive et de l'apparition progressive. Dans OCD, les items sont présentés en deux étapes avec transition graphique. À l'ouverture du menu, l'utilisateur ne trouve que les trois items prédits. Ces items sont affichés en haut du menu. Dans la seconde étape, les items prédits disparaissent progressivement et la liste complète des items apparaît progressivement. Ce processus dure environ 5000 ms. La figure 33 montre le principe de la technique OCD.



L'utilisateur cherche sa cible parmi les items prédits. Lorsque la prédiction est correcte, l'utilisateur sélectionne directement sa cible. Dans le cas contraire (prédiction incorrecte), l'utilisateur attend jusqu'à l'apparition totale de la liste complète afin de sélectionner sa cible. En résumé, la technique OCD combine deux effets différents : l'effet de disparition de la liste de prédiction (axes Elimination et Mouvement) et l'effet d'apparition de la liste complète (axes Ajout et Mouvement). Comme dans l'adaptation éphémère (Findlater *et al.*, 2009), OCD suit un processus en deux étapes. La première est destinée uniquement à la prédiction sans accès à la liste principale. La seconde est l'accès à la liste principale.

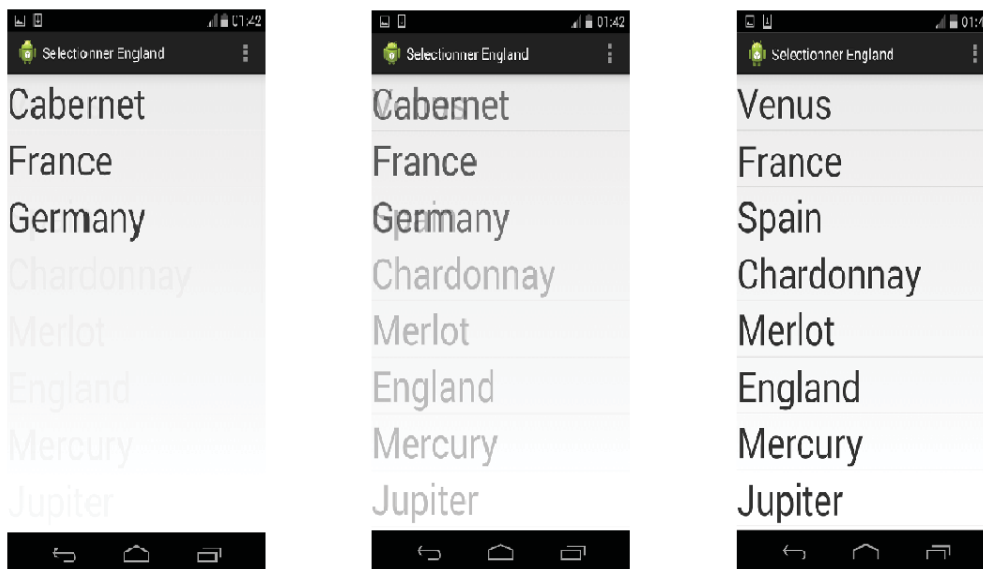


Figure 33. Technique OCD en deux étapes.

## 2. Evaluation expérimentale

L'objectif de l'évaluation expérimentale est triple : (1) comparer la performance de ICD et OCD par rapport à la condition contrôle qui est la condition statique non adaptative ; (2) comparer la performance de ICD par rapport à OCD ; (3) comparer l'effet du facteur topologique vs le facteur d'ordre.

Trois tests sur smartphone ont été implémentés. La liste utilisée dans l'expérience est la même que celle utilisée dans l'étude expérimentale de Findlater *et al.*, 2009. C'est une liste

composée de quatre séries contenant chacune 4 items (England, France, Germany, Spain – Venus, Mercury, Jupiter, Saturn – Cabernet, Chardonnay, Merlot, Shiraz – Almond, Pecan, Pistachio, Walnut).

Dans chaque condition, l'utilisateur doit sélectionner une cible aussi vite que possible. La cible peut être sur la fenêtre d'incitation (fenêtre de prédiction) ou pas. La cible est toujours présente dans la liste principale. Elle se trouve soit sur le premier, soit sur le second écran. Dans le cas du second écran, l'utilisateur doit faire défiler la liste pour atteindre sa cible. La figure 34 présente une vue globale des conditions expérimentales.

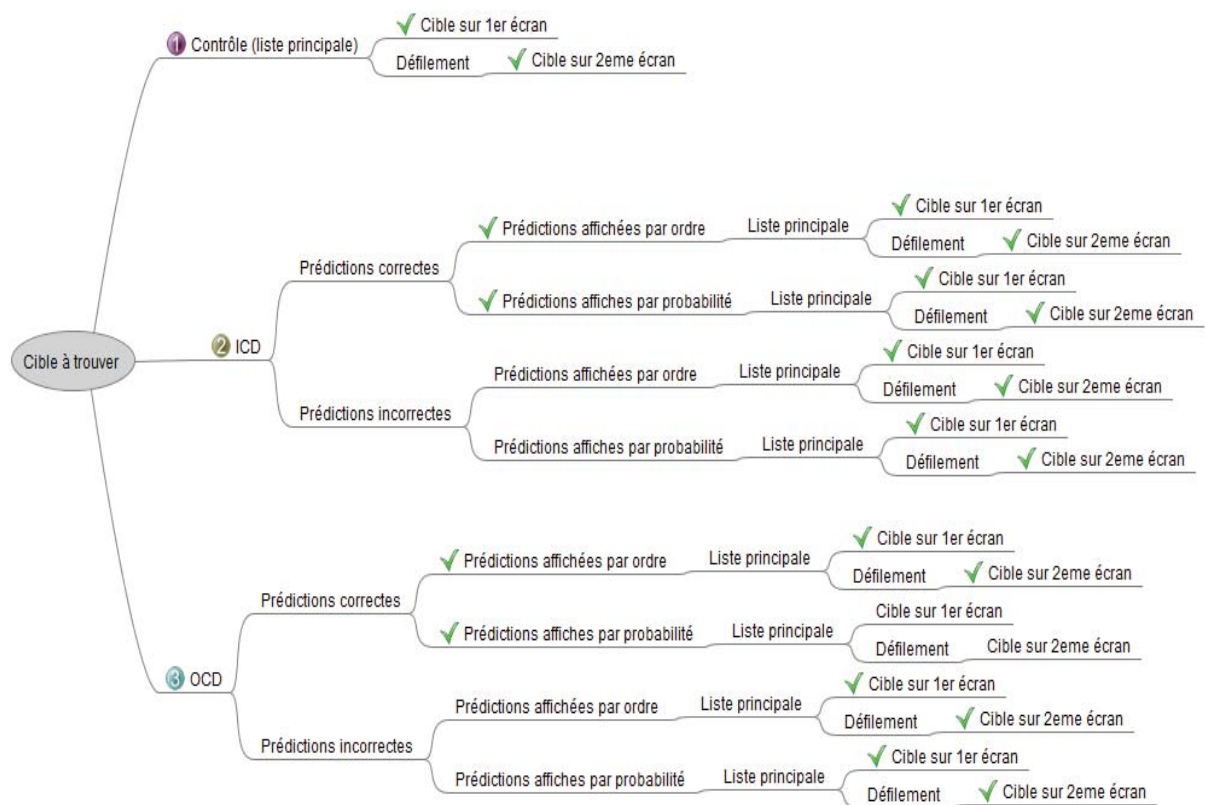


Figure 34. Vue globale des différentes conditions de l'expérimentation.

## 2.1. Méthodologie

Nous manipulons cinq facteurs indépendants :

Le premier facteur est le type du menu (Contrôle vs ICD vs OCD).

Le second facteur est relatif à l’affichage de la fenêtre de prédiction (ICD vs OCD).

Le troisième facteur est l’ordre d’affichage des items prédits sur la fenêtre de prédiction : par ordre prédéfini vs par probabilité. En effet, il est difficile de maintenir la stabilité spatiale dans le cas de *smartphone* où la liste principale des items est souvent répartie sur plusieurs écrans. Au regard des travaux soulignant l’importance de la stabilité spatiale, il convient de respecter quelques propriétés spatiales permettant de préserver la cohérence. Nous avons donc introduit une variable (le troisième facteur) jouant soit sur l’ordre, soit sur la probabilité. Dans le premier cas, l’ordre de la liste initiale (la liste complète) est conservé, alors que dans le second les items sont présentés en fonction de leur prédiction (indépendamment de stabilité la spatiale).

Le quatrième facteur est la localisation de la cible sur le premier ou sur le second écran de la liste principale (liste complète des items). Dans les trois listes principales (Contrôle, ICD, OCD), nous avons 16 items. Chaque liste est divisée en deux écrans. Chaque écran contient 8 items afin d’avoir le même nombre d’items sur chaque écran. La distribution des cibles parmi les deux écrans est contrôlée.

Le cinquième facteur est la précision de prédiction. Dans le cas où la prédiction est correcte, la cible attendue est incluse dans la liste de prédiction (fenêtre d’incitation). Dans le cas où la prédiction est incorrecte, la cible ne se trouve pas dans la liste de prédiction (fenêtre d’incitation). Dans les deux cas, la cible est toujours dans la liste complète, soit sur le premier, soit sur le second écran.

## **2.2. Hypothèses**

Les hypothèses faites pour cette étude expérimentale sont les suivantes :

### **2.2.1. Hypothèse 1 : vitesse d’exécution de la tâche**

#### **- Pour un haut niveau de précision de prédiction**

*ICD et OCD seront plus rapides que la condition contrôle.* Dans les conditions adaptatives (ICD et OCD), la cible est affichée lors de la première phase d’interaction (dans la liste d’incitation). Comme l’utilisateur cherche une cible qui est forcément présente parmi les trois items prédits, alors il sera plus rapide que dans la condition Contrôle contenant 16 items. Cela

sera d'autant plus vrai lorsque l'item ciblé sera disposé sur le second écran de la condition Contrôle.

*L'affichage des items prédits selon le critère de probabilité dans les conditions ICD et OCD sera plus rapide que leur affichage suivant le critère d'ordre.* Cette hypothèse est basée sur le fait que les listes sont couramment lues du haut en bas sur des petits écrans. La cible dans la condition probabiliste va toujours être en tête de liste. Par conséquent, elle va être lue en premier ce qui rend le facteur de probabilité plus rapide que le facteur d'ordre.

- **Pour un faible niveau de précision de prédiction**

*ICD sera plus rapide que OCD.* Lorsque la prédiction est incorrecte et la cible ne fait pas partie de la liste des items prédits, l'utilisateur peut manipuler la liste complète dans ICD avant même la disparition de la fenêtre d'incitation. Ceci n'est pas le cas dans OCD où l'utilisateur est obligé d'attendre jusqu'à l'apparition de la liste complète.

*La condition Contrôle sera plus rapide que la condition OCD.* En se basant sur l'hypothèse précédente, puisque l'utilisateur doit attendre l'apparition de la liste complète dans OCD, l'interaction sera être ralentie par rapport à la condition Contrôle où l'utilisateur a un accès direct à la liste complète des items. Donc OCD devrait être moins rapide.

### **2.2.2. Hypothèse 2 : préférence utilisateur**

- **Pour un haut niveau de précision de prédiction**

*Au moins ICD ou OCD sera préférée à la condition Contrôle.* Dans les conditions adaptatives (ICD et OCD), l'attention de l'utilisateur est attirée par la liste de prédiction. Cela rend la sélection de la cible plus rapide et plus simple que dans la condition Contrôle. Dans cette dernière, l'utilisateur doit trouver la cible qui peut être sur le premier ou sur le second écran.

- **Pour un faible niveau de précision de prédiction**

*La condition Contrôle ne sera pas préférée à la condition ICD ou OCD.* Comme l'objectif visé dans ICD et OCD est de ne pas ralentir l'utilisateur même lorsque la prédiction est incorrecte, la condition Contrôle ne devrait pas être préférée à ICD et OCD.

## **2.3. Mesures qualitatives et quantitatives**

Nous définissons deux variables dépendantes :

- 1- La première variable dépendante est la vitesse de sélection. Elle est mesurée par le temps pris depuis l'ouverture du menu jusqu'à la sélection de la cible correcte. La vitesse est calculée en secondes.
- 2- La seconde variable dépendante est l'accomplissement de la tâche. Le taux d'erreurs est enregistré.

Finalement, nous recueillons aussi des données subjectives sur la difficulté perçue, la satisfaction et l'esthétique selon une échelle de Likert.

## **2.4. Tâche**

La tâche expérimentale consiste à demander aux utilisateurs d'effectuer une séquence de sélections de cibles. Pour chaque condition, un message est affiché en haut de l'écran spécifiant le nom de l'item à sélectionner. Ensuite les listes apparaissent. Le nom de la cible reste affiché afin que l'utilisateur n'oublie pas la cible à sélectionner. L'utilisateur sélectionne la cible à partir de la fenêtre d'incitation ou à partir de la liste complète. Lorsque l'utilisateur réussit à sélectionner la cible demandée, un nouveau message apparaît lui spécifiant le nom de la nouvelle cible à sélectionner. Dans le cas où l'utilisateur fait une mauvaise sélection (la cible sélectionnée ne correspond pas à celle demandée), un message d'erreur est affiché invitant l'utilisateur à rechercher la bonne cible. A la fin du test, un message de remerciement est affiché informant l'utilisateur que le test est terminé.

## **2.5. Matériels**

Des *smartphones* de type Android ont été utilisés avec des prototypes codés en Java pour Android. Une base de données a été créée dans laquelle le temps de sélection (en secondes), le temps de défilement (en millisecondes) et le taux d'erreurs ont été enregistrés.

## **2.6. Participants**

Vingt-quatre sujets ont participé à cette expérimentation dont 5 femmes et 19 hommes. Ils sont âgés entre 23 et 57 ans. Tous les participants ont été recrutés en interne d'Orange Labs. Ils sont tous des utilisateurs réguliers de *smartphones* tactiles.

## 2.7. Procédure

Avant de commencer le test, chaque participant a effectué un pré-test afin de lui permettre de s'entraîner avec des listes contenant des items différents que ceux utilisés dans les vrais tests. Une fois que les 10 cibles du pré-test sont sélectionnées avec succès durant le pré-test, le participant peut commencer le test.

Le test se compose de 100 cibles dont :

- 40 cibles pour chaque condition prédictive (ICD et OCD) réparties en 20 cibles avec prédiction correcte (haut niveau de précision de prédiction) et 20 cibles avec prédiction incorrecte (faible niveau de précision de prédiction).
- 20 cibles pour la condition Contrôle.

## 2.8. Résultats

Les données ont été analysées par une ANOVA *Randomized Blocks* après vérification de l'homogénéité de variance en utilisant le test *Brown & Forsythe's*  $W(9,230) = 2.64, p = .006$ . L'analyse statistique de l'accomplissement des tâches a été effectuée par les tests  $\chi^2$  et *paired student T* pour une analyse détaillée.

### *Effet de disparition*

Dans les conditions prédictives (ICD et OCD), lorsque la prédiction est correcte, les utilisateurs sont significativement plus rapides (respectivement  $M = 1.20, SD = 0.34$  et  $M = 1.32, SD = 0.43$ ) que dans la condition Contrôle ( $M = 2.47, SD = 0.90$ ). Pour ICD vs Contrôle, on a  $t(47) = 9.9, p < .001$ . Pour OCD vs Contrôle, on a  $t(47) = 9.85, p < .001$ . Cela justifie l'usage de la prédiction.

En outre, en ICD, les sujets sont significativement plus rapides  $t(47) = 2.12, p < .02$  que dans OCD, et cela pour un taux d'erreur équivalent.

### *Effet d'ordre topologique*

Lorsque les cibles sont présentées sur la fenêtre d'incitation par ordre ( $M = 1.20, SD = 0.33$ ), elles sont significativement plus rapides ( $t(47) = 2.79, p < .003$ ) que lorsqu'elles sont présentées par probabilité ( $M = 1.33, SD = 0.44$ ), pour un taux d'erreur équivalent.

### *Position de la cible sur le premier ou sur le second écran*

Lorsque la prédiction est incorrecte dans les deux conditions ICD et OCD (respectivement  $M = 3.93$ ,  $SD = 0.84$  et  $M = 4.23$ ,  $SD = 0.67$ ), le temps de sélection est beaucoup plus court dans ICD ( $t(47) = 2.08$ ,  $p < .02$ ).

Les erreurs dans OCD sont plus fréquentes lorsque les cibles sont sur le premier écran (31) que lorsqu'elles sont sur le deuxième écran (18),  $\chi^2(1, N = 480) = 3.841$ ,  $p < .05$ . Ce taux d'erreur est probablement dû à l'enchaînement des deux étapes présentant un chevauchement.

En revanche, dans ICD, les étapes d'interaction sont certes simultanées mais avec des fenêtres clairement différenciées. En effet, dans cette condition, lorsque la cible n'est pas dans la première étape d'interaction (sur la fenêtre d'incitation), l'utilisateur peut cliquer directement sur la partie visible de la liste. Les vitesses d'exécution confirment cela : les utilisateurs sont significativement plus rapides ( $t(23) = 6.41$ ,  $p < .001$ ) lorsque la cible est située sur l'écran 1 ( $M = 3.43$ ,  $SD = 0.64$ ) que lorsqu'elle est située sur l'écran 2 ( $M = 4.42$ ,  $SD = 0.72$ ).

Cela est également vrai dans la condition Contrôle où les sujets sont significativement plus rapides ( $t(23) = 10.3$ ,  $p < .01$ ) lorsque la cible est sur l'écran 1 ( $M = 1.77$ ,  $SD = 0.47$ ) que lorsqu'elle est située sur le l'écran 2 ( $M = 3.17$ ,  $SD = 0.65$ ). De plus, en l'absence de présence de fenêtre incitative, les sujets faisaient significativement moins d'erreurs ( $\chi^2(1, N = 2400) = 17.4$ ,  $p < .01$ ) dans la condition Contrôle que dans les conditions OCD et ICD.

Les temps de sélection sont illustrés sur la figure 35.

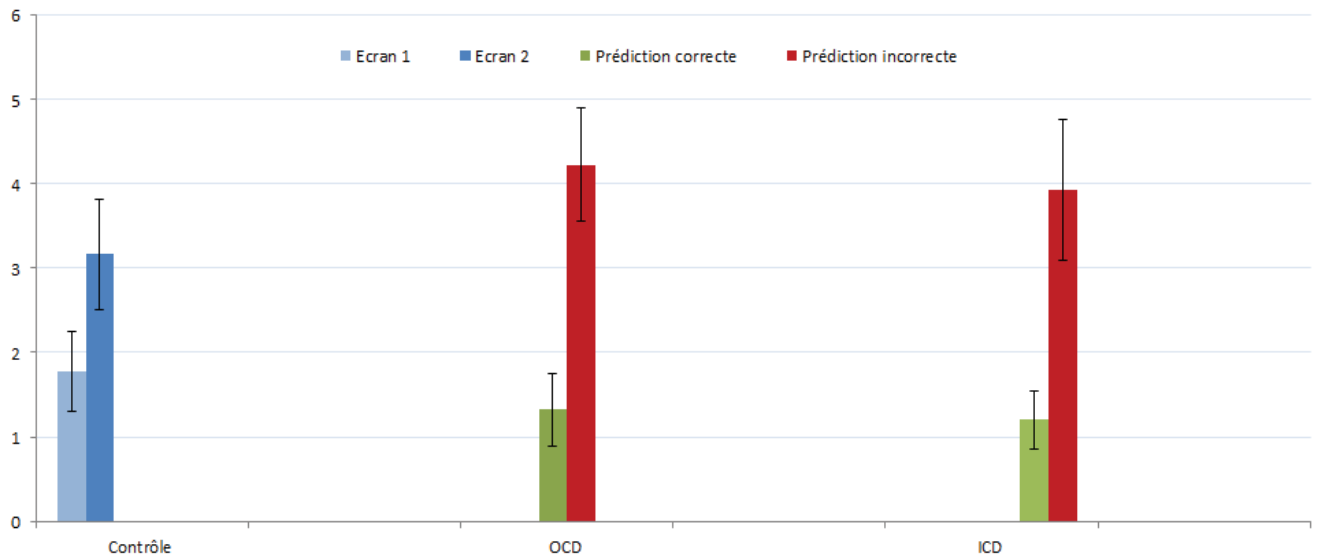


Figure 35. Temps de sélection pour toutes les conditions de l'expérimentation.

En ce qui concerne les préférences exprimées par les utilisateurs, nous n'avons pu mettre en évidence aucune différence significative.

## 2.9. Discussion

Faute de différence significative sur l'hypothèse 2 relative aux préférences utilisateur, la discussion porte uniquement sur l'hypothèse 1 : la vitesse d'exécution de la tâche.

### - Pour un haut niveau de précision de prédiction

*ICD et OCD sont plus rapides que la condition contrôle. Confirmée*

Lorsque la prédiction est correcte et correspond aux besoins de l'utilisateur, ce dernier atteint la cible plus rapidement. Même dans le cas d'une tâche répétitive où l'utilisateur peut apprendre la position des items, la prédiction s'avère efficace, notamment dans le cadre de l'usage du *smartphone* où les applications se multiplient et les zones d'affichage restent contraintes.

*L'affichage des items prédits selon le critère de probabilité dans les conditions ICD et OCD est plus rapide que leur affichage suivant le critère d'ordre. Infirmée*

Les résultats ont montré que l'affichage des items prédits selon le critère de probabilité dans ICD et OCD n'est pas plus rapide que lorsqu'ils sont affichés par critère d'ordre. Cela pourrait



traduire l'importance de la stabilité spatiale qui facilite l'usage de la mémoire visuelle lors de la recherche de cible. Néanmoins, nos listes de prédictions ne contenaient que trois items et il est probable que la stratégie de lecture non ordonnée d'une liste aussi courte soit la cause du phénomène. Le critère de probabilité aurait peut-être plus de bénéfices dans le cas d'une liste de prédiction plus fournie, incitant davantage à une stratégie de lecture de haut en bas.

- **Pour un faible niveau de précision de prédiction**

*ICD est plus rapide que OCD. Confirmée*

Les résultats montrent que lorsque la prédiction ne correspond pas à ce que l'utilisateur cherche, ICD est plus rapide qu'OCD. Cela est justifié par le fait que l'accès à la liste complète des items est plus simple en ICD que dans OCD. Dans ICD, la liste principale est disponible dès le début de l'interaction et peut directement agir sur les items de cette liste (quitte à générer plus d'erreurs). À l'inverse, dans OCD, l'utilisateur doit attendre jusqu'à l'apparition totale de la liste complète des items.

En résumé, lorsque la prédiction est incorrecte, le chemin optimal est de se déplacer aussi rapidement que possible sur la liste complète. Cela est facilité par ICD qui offre déjà un accès visuel à la liste complète, et contraint par OCD qui oblige à attendre l'affichage de cette dernière.

En conclusion, l'utilisation de la disparition progressive ne ralentit pas l'interaction de l'utilisateur dans le cas d'une mauvaise prédiction, contrairement à l'apparition progressive.

*La condition Contrôle est plus rapide que la condition OCD. Confirmée*

La navigation est plus facile dans la condition Contrôle car l'utilisateur accède directement à la liste complète des items.

*Quelle que soit la liste (contrôle, ICD et OCD), lorsque la cible est sur le premier écran, l'interaction est plus rapide que sur le deuxième écran. Confirmée*

Les résultats confirment cette hypothèse. Il est logique que l'accès à la cible sur le premier écran soit plus rapide que sur le second écran où l'utilisateur doit faire un défilement. Ces résultats confortent simplement la cohérence de l'expérimentation. Le plus important, est que l'accès au premier écran dans ICD soit plus rapide que dans OCD. En effet, garder la liste complète directement disponible, sans délai d'attente, est nettement plus efficace. Cela

confirme à nouveau la plus grande efficacité de la technique par disparition progressive sur que la technique d'apparition progressive.

### **3. Conclusion**

Dans ce chapitre nous avons introduit le concept d'adaptation éphémère avec disparition appliquée à des menus sur *smartphone*. Nous avons présenté deux variantes appelées *In Context Disappearing* (ICD) et *Out of Context Disappearing* (OCD) basées sur les variations d'axes PCA Élimination, Ajout et Mouvement. L'objectif est de surmonter les limites contextuelles, en particulier celles liées à la taille d'écran réduite.

Nous avons montré expérimentalement que lorsque la prédiction est correcte, ICD et OCD sont plus rapides que la condition Contrôle, ICD étant plus rapide qu'OCD. De plus, lorsque la prédiction est incorrecte, ICD est comparable à la condition Contrôle.

Néanmoins, ICD provoque des erreurs de manipulation dûes au chevauchement des deux listes lors de la disparition progressive. Le manque de contrôle de l'utilisateur sur cet effet graphique est probablement la source de ces erreurs. Pour augmenter la qualité de l'interaction, nous proposons d'introduire le critère de contrôlabilité en ajoutant l'axe Contrôle dans cette technique. Cela permettra à l'utilisateur de contrôler la disparition de la prédiction. Ce concept fait l'objet du chapitre suivant.

### **4. Synthèse**

L'adaptation des IHM sur petits écrans impose des contraintes supplémentaires et les techniques existantes ne peuvent pas être transposées facilement. Jouer sur les axes visuels tels que le highlighting, la taille de police, le gras, etc. est peu opérant car l'ensemble des éléments ne peut pas être affiché sur une seule fenêtre. De plus, lorsque les prédictions sont incorrectes, ces techniques ralentissent l'interaction. Notre objectif était de proposer une technique d'adaptation applicable à une interface de petite taille et qui d'absorbe les méfaits d'une prédiction incorrecte.

Le principe de l'adaptation éphémère avec disparition est de donner à l'utilisateur l'opportunité de sélectionner sa cible au sein d'une liste de trois items. Progressivement, ces

prédictions disparaissent et laissent la place à la liste complète des items. Cependant, le chevauchement des deux listes génère des erreurs et cette technique doit être améliorée.



## **Chapitre 6 : Adaptation éphémère avec disparition contrôlée, axe Contrôle**

Ce chapitre a fait l'objet d'une publication en conférence internationale :

**Bouzit, S., Chêne, D., and Calvary, G. (2015).** Evanescent Adaptation on Small Screens. *Proc. of the 27<sup>th</sup> Australian Computer-Human Interaction Conference OzCHI'2015* (Parkville, December 7-10, 2015). ACM Press, New York, pp. 62–68.

Dans ce chapitre nous présentons l'adaptation éphémère avec disparition contrôlée par l'utilisateur (DU), une variante de l'adaptation éphémère avec disparition (ICD). Le principe est similaire et consiste à présenter deux couches superposées : une fenêtre incitative contient les items prédits tandis qu'au-dessous se trouve la liste complète des items. La différence avec ICD est que la fenêtre de prédiction qui disparaît progressivement peut être également contrôlée par l'utilisateur : il peut lui-même fermer la fenêtre. Nous explorons ainsi l'axe «Contrôle » du modèle PCA. Les axes explorés dans ce chapitre sont illustrés dans la figure 36.

Le chapitre est structuré en trois sections principales : la première partie présente la généralisation d'ICD et le contrôle utilisateur dans deux prototypes. La deuxième partie décrit l'évaluation du concept. Enfin, les résultats ouvrent de nouvelles perspectives.

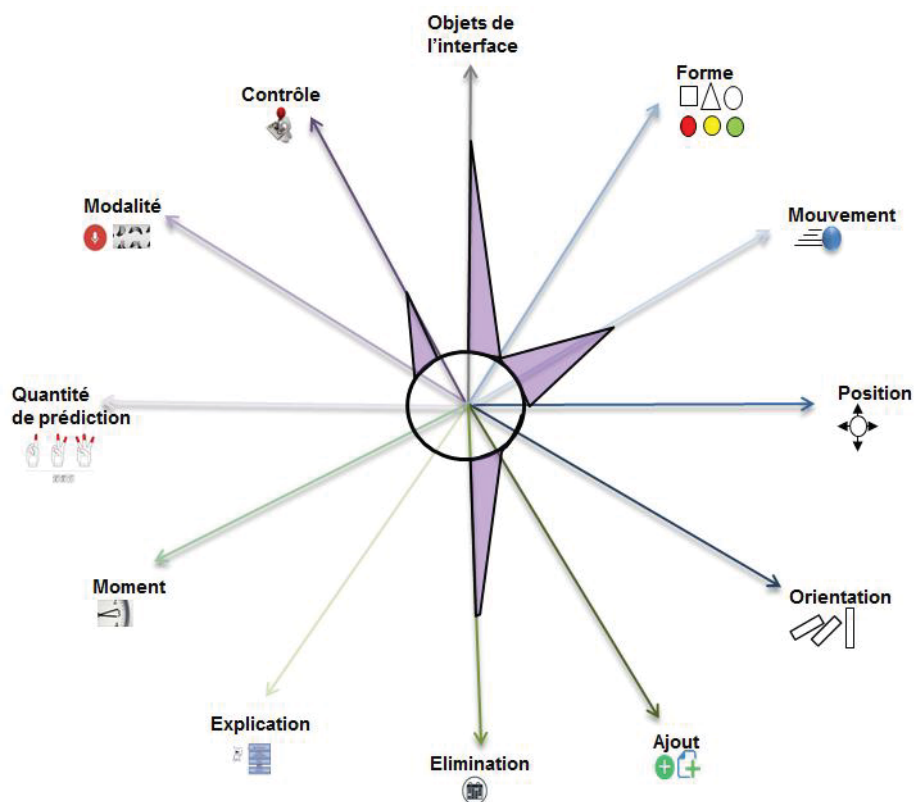


Figure 36. Caractérisation PCA de l'adaptation éphémère avec disparition contrôlée par l'utilisateur.

# 1. Techniques étudiées

Cette section présente les trois prototypes à l'étude.

## 1.1. ICD en deux dimensions

Nous étendons la technique ICD (chapitre 5) limitée à des listes linéaires (1 dimension) en l'appliquant à une grille d'icônes (2 dimensions). L'objectif est de vérifier si ICD conserve ses avantages dans ce cadre. Nous avons produit un prototype nommé D (pour *Disappearing*), (figure 37).

À l'ouverture de la page d'accueil, l'utilisateur trouve deux couches de grille d'icônes. Sur le dessus, la fenêtre d'incitation propose la grille des icônes prédites. En dessous, est visible la grille complète des icônes.

Trois icônes d'applications prédites sont présentes dans la fenêtre d'incitation (étape 1). Celle-ci disparaît progressivement dans un délai de 5000 ms (étape 2) pour laisser place à la grille d'icônes complète (étape 3).

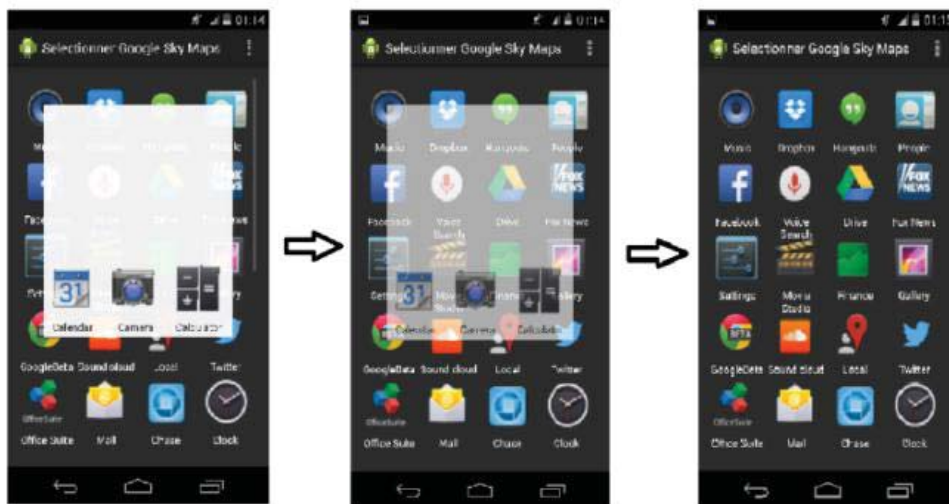


Figure 37. Prototype Disappearing (D).

À la première étape, l'utilisateur cherche sa cible dans la fenêtre d'incitation. Si la prédiction est correcte, la cible s'y trouve, sinon il peut soit attendre la disparition complète de cette fenêtre d'incitation soit directement manipuler la grille sous-jacente. Comme observé sur ICD,

il est fort probable que cela permette d'accélérer les performances, mais génère également des erreurs de pointage, notamment en cas de prédiction incorrecte.

En résumé, dans le prototype D, le contexte (qui est la grille d'icônes complète) est accessible partiellement et sujette à erreurs.

## 1.2. ICD avec contrôle utilisateur

Le second prototype a pour objectif de surmonter ces limitations et réduire le nombre d'erreurs en incorporant un contrôle utilisateur sur la fenêtre d'incitation. Cela donne lieu au prototype nommé DU pour *Disappearing + User control* (figure 38).

Le contrôle utilisateur incorporé est sous forme d'un bouton « Exit » permettant de fermer la fenêtre d'incitation. C'est un processus en trois étapes contrôlées par l'utilisateur où dans la première étape la fenêtre d'incitation et la grille complète sont affichées simultanément. Dans la seconde étape, la disparition de la fenêtre d'incitation est automatique et contrôlable par l'utilisateur. Dans la troisième étape, uniquement la grille complète est affichée.



Figure 38. Prototype Disappearing + User control (DU).

Lorsque la prédiction est correcte, l'utilisateur peut sélectionner sa cible directement sur la fenêtre d'incitation. Lorsque la prédiction est incorrecte, l'utilisateur peut soit attendre la disparition complète de la fenêtre d'incitation qui disparaît progressivement, soit actionner le bouton « Exit » qui accélère l'accès à la grille complète.



### 1.3. Contrôle utilisateur sans disparition

Le troisième prototype consiste à remettre en cause la fonction de disparition progressive automatique. Seul le contrôle utilisateur permet de faire disparaître la fenêtre d'incitation. Cela donne lieu au prototype U pour *User control* (figure 39).

Dans ce prototype, il n'y a pas de disparition progressive de la fenêtre d'incitation. Nous aboutissons à un processus contrôlé par l'utilisateur et à deux étapes où dans la première étape la fenêtre d'incitation et la grille complète sont affichées simultanément et dans la seconde uniquement la grille complète est affichée.



Figure 39. Prototype User control (U).

## 2. Evaluation expérimentale

Une expérimentation a été faite afin de tester les trois prototypes présentés dans la section précédente. Pour l'évaluation, un quatrième prototype est développé il s'agit de la condition Contrôle. Dans ce prototype, la liste complète des icônes est affichée sans aucun effet, ni fenêtre d'incitation, ni effet graphique comme montré sur la figure 40.



Figure 40. Condition Contrôle.

L'objectif de l'expérimentation est double : premièrement, généraliser la technique ICD à une grille d'icônes. Nous vérifierons ainsi si nous obtenons des résultats similaires avec le prototype D à ceux obtenus avec la technique ICD. Deuxièmement, identifier si le contrôle utilisateur améliore l'interaction. Nous vérifierons si les résultats des conditions intégrant le contrôle sont meilleurs à celles qui ne l'intègrent pas.

## 2.1. Méthodologie

Nous avons manipulé quatre facteurs indépendants.

Le premier facteur est le type de condition (D, DU, U et Contrôle).

Le second facteur est l'affichage de la prédiction par ordre ou par probabilité.

Le troisième facteur est la localisation de la cible au sein de la grille d'icônes complète. Nous avons utilisé des grilles de 40 icônes divisées en deux écrans. La cible sur la grille complète peut être sur l'écran 1 ou sur l'écran 2 (accessible par un défilement).

Le quatrième facteur est la précision de prédiction. Lorsque la prédiction est correcte, la cible à sélectionner est dans la fenêtre d'incitation. Lorsque la prédiction est incorrecte, la cible ne se trouve pas dans la fenêtre d'incitation.

Chaque participant a testé la condition Contrôle et les conditions adaptatives (D, DU et U) incluant chacune deux variations (présentation par ordre et présentation par probabilité, voir figure 41).



Figure 41. (a) Condition Ordre. (b) Condition Probabiliste.

## 2.2. Hypothèses

### 2.2.1. Hypothèse 1 : vitesse d'exécution

#### - Pour un haut niveau de précision de prédiction

*D, DU et U seront plus rapides que la condition Contrôle.* Lorsque la prédiction est correcte (la cible est sur la fenêtre d'incitation) dans D, DU et U, l'accès à la cible va être plus rapide que dans la condition Contrôle, qu'elle soit sur le premier ou sur le second écran.

*U sera plus rapide que D et DU.* Dans U, la fenêtre d'incitation ne disparaît pas. Si l'utilisateur ne trouve pas tout de suite sa cible, il devrait avoir plus de temps pour la voir et la sélectionner en U qu'en D et DU.

*Dans les conditions prédictives (D, DU, U), la présentation des items prédits de manière probabiliste sera plus rapide que la présentation par ordre.* Avec une distribution probabiliste, la cible sera toujours en haut de la fenêtre d'incitation et donc a priori d'un accès plus rapide.

- **Pour un faible niveau de précision de prédiction**

*D, DU et U ne demanderont pas plus de temps que la condition Contrôle.* En effet, contrôle utilisateur et disparition ne devraient pas ralentir l'interaction de l'utilisateur dans le cas d'une mauvaise prédiction.

*DU et U seront plus rapides que D.* Lorsque la prédiction est incorrecte, l'utilisateur doit passer le plus vite possible à la grille complète des icônes. Dans les conditions DU et U, l'utilisateur peut contrôler la disparition de la fenêtre d'incitation et ainsi accélérer la transition sans avoir à attendre sa disparition, contrairement à D où l'utilisateur n'a aucun contrôle.

*DU sera plus rapide que U.* En DU, la fenêtre d'incitation est à la fois automatique et contrôlée, la disparition devrait donc être plus rapide.

**2.2.2. Hypothèse 2 : erreurs**

- **Pour un haut niveau de précision de prédiction**

*Les erreurs seront moins fréquentes dans les conditions U et Contrôle que dans les conditions D et DU.* En D et DU, l'utilisateur peut faire des erreurs de pointage en sélectionnant la cible en fin du processus de disparition.

- **Pour un faible niveau de précision de la prédiction**

*Les erreurs seront moins fréquentes dans les conditions DU et U que dans la condition D.* En DU et U, la présence du contrôle utilisateur devrait permettre de passer à la grille complète sans subir la phase de disparition qui est source d'erreurs.

**2.2.3. Hypothèse 3 : préférence utilisateur**

- **Pour un haut niveau de précision de prédiction**

*Les conditions adaptatives D, DU ou U devraient être préférées à la condition Contrôle.* L'adaptation étant *a priori* bénéfique, les conditions prédictives devraient être préférées à la condition Contrôle.

*U devrait être préférée à D et DU.* Cette hypothèse est basée sur le fait que le temps d'affichage de la fenêtre d'incitation est plus long en U qu'en D et DU. Dans la condition U, l'utilisateur contrôle la disparition de la fenêtre d'incitation sans être pressé.

### - Pour un faible niveau de précision de la prédiction

*La condition Contrôle ne sera pas préférée à D, DU ou U.* Les conditions prédictives ne devraient pas ralentir l'interaction lorsque la prédiction est incorrecte.

## 2.3. Mesures quantitatives et qualitatives

Nous mesurons deux variables dépendantes.

La première variable dépendante est la vitesse de sélection en secondes. Elle est mesurée par le temps pris depuis l'ouverture de la grille d'icônes jusqu'à la sélection de la cible demandée.

La seconde variable dépendante est le taux d'erreurs.

Des données subjectives liées à la difficulté, la satisfaction et l'esthétique sont également recueillies selon une échelle de Likert.

## 2.4. Tâche

La tâche expérimentale consiste à demander aux utilisateurs d'effectuer une séquence de sélections de cibles. Pour chaque condition, un message est affiché en haut de l'écran spécifiant le nom de l'icône à sélectionner. Ensuite la grille d'icônes apparaît. Le nom de la cible reste affiché afin que l'utilisateur n'oublie pas la cible qu'il doit sélectionner. L'utilisateur sélectionne la cible à partir de la fenêtre d'incitation et/ou à partir de la grille d'icônes complète. Lorsque l'utilisateur réussit à sélectionner la cible demandée, un nouveau message apparaît lui spécifiant le nom de la nouvelle cible à sélectionner. Dans le cas où l'utilisateur fait une mauvaise sélection (la cible sélectionnée ne correspond pas à celle demandée), un message d'erreur est affiché invitant l'utilisateur à chercher la bonne cible. A la fin du test, un message de remerciement est affiché informant l'utilisateur que le test est terminé.

La figure 42 décrit les différentes conditions du test.

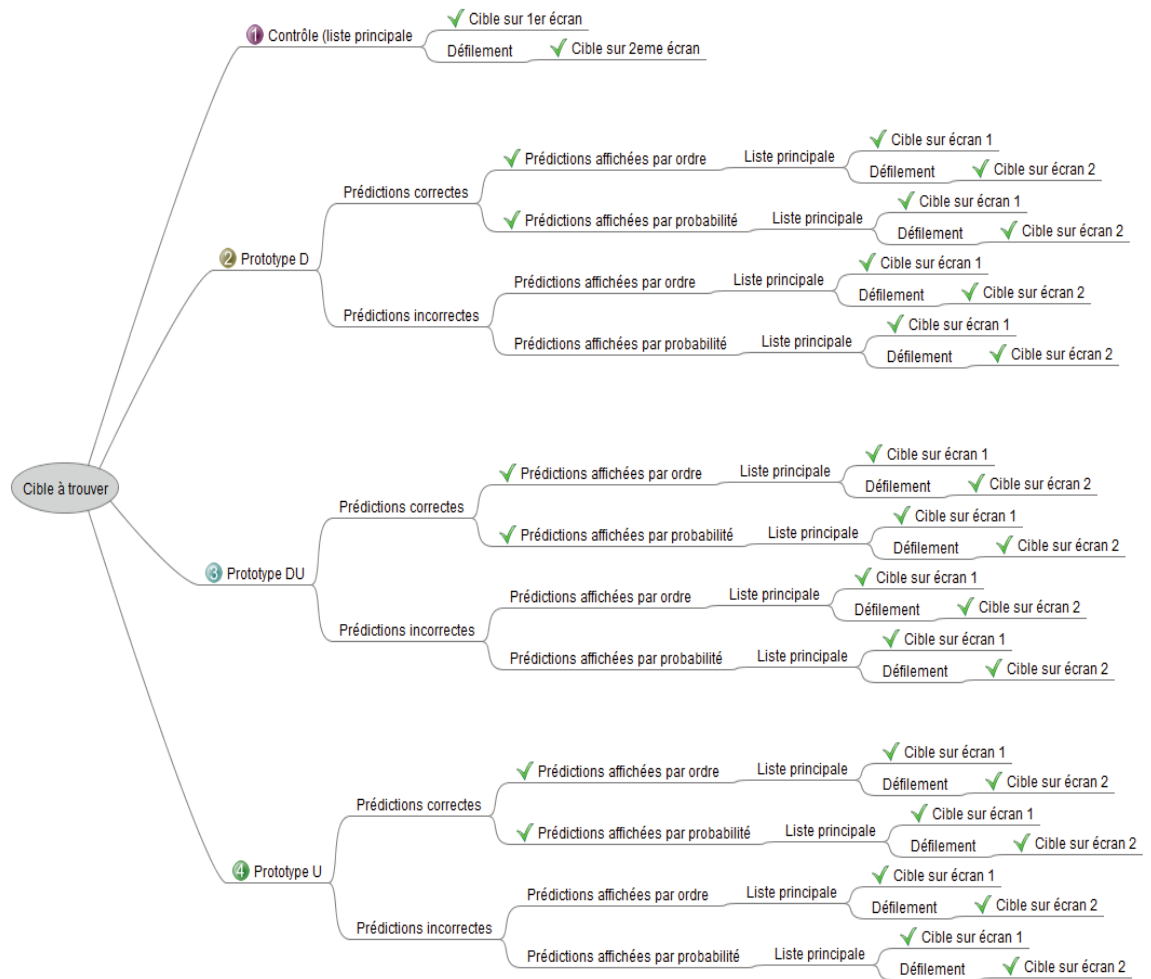


Figure 42. Conditions de l'étude expérimentale.

## 2.5. Matériels

Les appareils utilisés dans cette expérience sont des *smartphones* de type Android. L'expérimentation est codée en Java pour Android. Nous avons créé une base de données dans laquelle le temps de sélection (en secondes), le temps de défilement (en millisecondes) et le taux d'erreurs sont enregistrés.

## 2.6. Participants

Trente-trois volontaires (hommes et femmes) ont participé à cette expérimentation. Ils sont âgés entre 23 et 59 ans. Tous les sujets sont recrutés en interne d'Orange Labs et ils sont tous des utilisateurs réguliers de *smartphones*.

## 2.7. Procédures

Avant de commencer le test, les participants avaient pour consigne de sélectionner rapidement la cible demandée. Chaque participant a effectué un pré-test court (sélection de 10 cibles).

Le test est composé de 140 cibles :

- 120 cibles sont distribuées sur les conditions adaptatives ( $I_3$  pour Interface) (D, DU, U). Chaque sujet effectue 40 sélections pour chaque condition adaptative (D, DU, U =  $40 * 3$ ). 20 sélections correspondent aux prédictions correctes dont 10 sont présentées par ordre et 10 par probabilité ( $O_2$  pour ordre). Les 20 autres sélections correspondent aux prédictions incorrectes dont 10 sont placées sur le premier écran et 10 sur le second ( $E_2$  pour écran). La variable ( $P_2$  pour précision) exprime le faible vs haut niveau de précision de la prédiction.
- 20 cibles dans la condition Contrôle.

Nous avons donc 24 conditions :  $S_{33} * I_3 * E_2 * P_2 * O_2$

## 2.8. Résultats

### *La vitesse de sélection*

En l'absence d'homogénéité des variances (*Brown-Forsythe* et *Levene's*), nous avons appliqué le test non paramétrique *Friedman's ANOVA by Ranks*.

Comme attendu, lorsque la prédiction est correcte, les conditions prédictives (« D »  $M = 1.75$ ,  $SD = 1.31$ , « DU »  $M = 1.70$ ,  $SD = 1.26$ , « U »  $M = 1.40$ ,  $SD = 0.0.64$ ) sont significativement plus rapides ( $F(3) = 102.35$ ,  $p < 0.0001$ ) que la condition Contrôle ( $M = 6.06$ ,  $SD = 3.40$ ).

Cela confirme le résultat de l'étude présentée dans le chapitre précédent : la technique ICD est tout aussi efficace appliquée à une grille d'icônes.

La présentation par probabilité ou ordre ne génère pas de résultats différents entre DU (ordre :  $M = 1.56$ ,  $SD = 0.97$  ; probabilité :  $M = 1.84$ ,  $SD = 1.50$ ) et U (ordre :  $M = 1.40$ ,  $SD = 0.68$  ;

probabilité :  $M = 1.41$ ,  $SD = 0.60$ ). Par contre, dans la condition D, la présentation par probabilité ( $M = 1.46$ ,  $SD = 1.27$ ) est significativement plus rapide ( $F(1) = 6.53$ ,  $p = 0.01$ ) qu'une présentation basée sur ordre ( $M = 2.04$ ,  $SD = 1.30$ ).

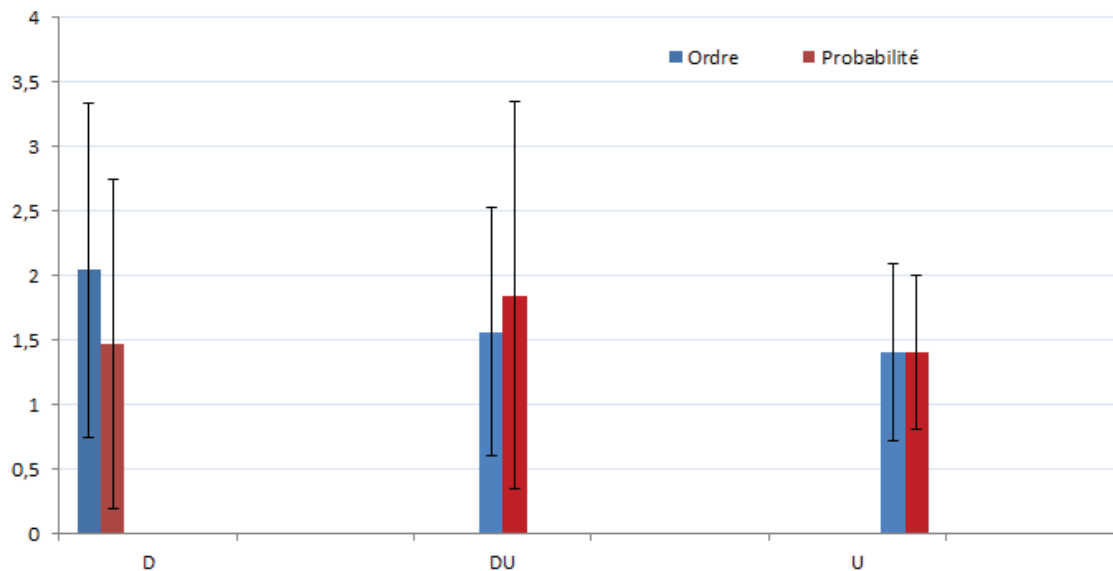


Figure 43. Temps de sélection lorsque la prédiction est correcte.

Lorsque la prédiction est incorrecte, les performances dépendent significativement des conditions  $F(3) = 35.71$ ,  $p < 0.0001$ . Les utilisateurs sont significativement plus rapides dans la condition DU ( $M = 6.97$ ,  $SD = 2.64$ ) et la condition Contrôle ( $M = 6.06$ ,  $SD = 3.40$ ) mais sont plus lents dans les conditions D ( $M = 8.30$ ,  $SD = 3.79$ ) et U ( $M = 8.0$ ,  $SD = 3.37$ ).

Néanmoins, toutes les conditions sont significativement plus rapides ( $F(7) = 103.84$ ,  $p < 0.001$ ) lorsque la cible est sur le premier écran (« D »  $M = 6.27$ ,  $SD = 2.34$ , « DU »  $M = 6.1$ ,  $SD = 2.38$ , « U »  $M = 7.20$ ,  $SD = 3.02$ , « Contrôle »  $M = 3.72$ ,  $SD = 1.75$ ), comparé au second écran (« D »  $M = 10.33$ ,  $SD = 3.90$ , « DU »  $M = 7.85$ ,  $SD = 2.62$ , « U »  $M = 8.81$ ,  $SD = 3.54$ , « Contrôle »  $M = 8.39$ ,  $SD = 3.02$ ).



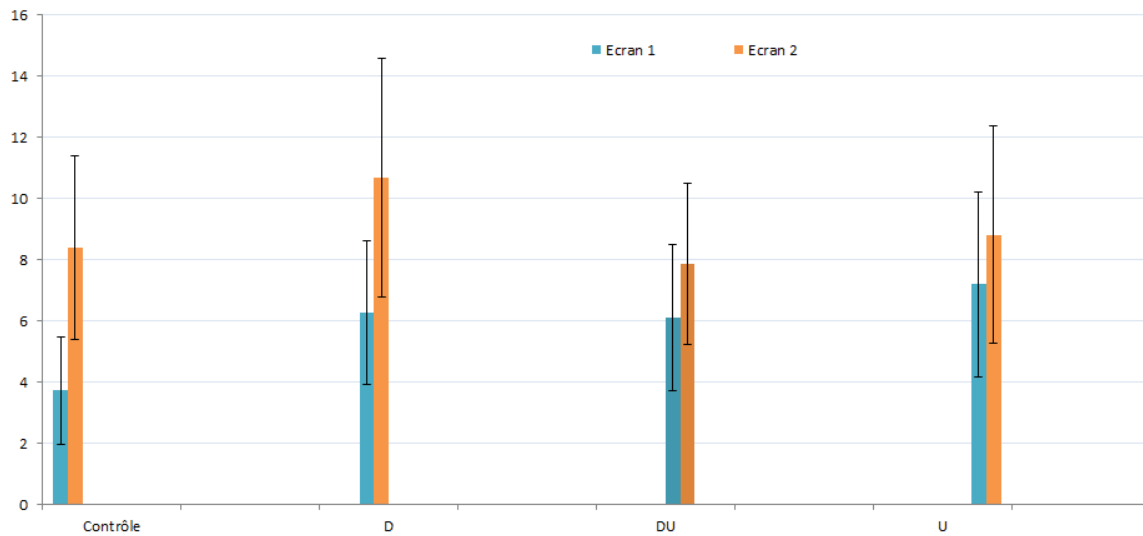


Figure 44. Temps de sélection lorsque la prédiction est incorrecte.

### Les erreurs

Les erreurs sont réparties de façon significativement différente ( $F(3) = 42.21, p < 0.001$ ) selon les conditions Contrôle ( $M = 0.60, SD = 0.83$ ), D ( $M = 4.18, SD = 3.32$ ), DU ( $M = 1.03, SD = 1.10$ ) et U ( $M = 0.63, SD = 0.93$ ). En effet, l'absence de disparition progressive dans la condition U ne génère pas de chevauchement, et induit moins d'erreurs. De plus, la condition DU est significativement plus efficace ( $F(1) = 15.12, p < 0.001$ ) que la condition D, cela étant dû à la présence du contrôle utilisateur.

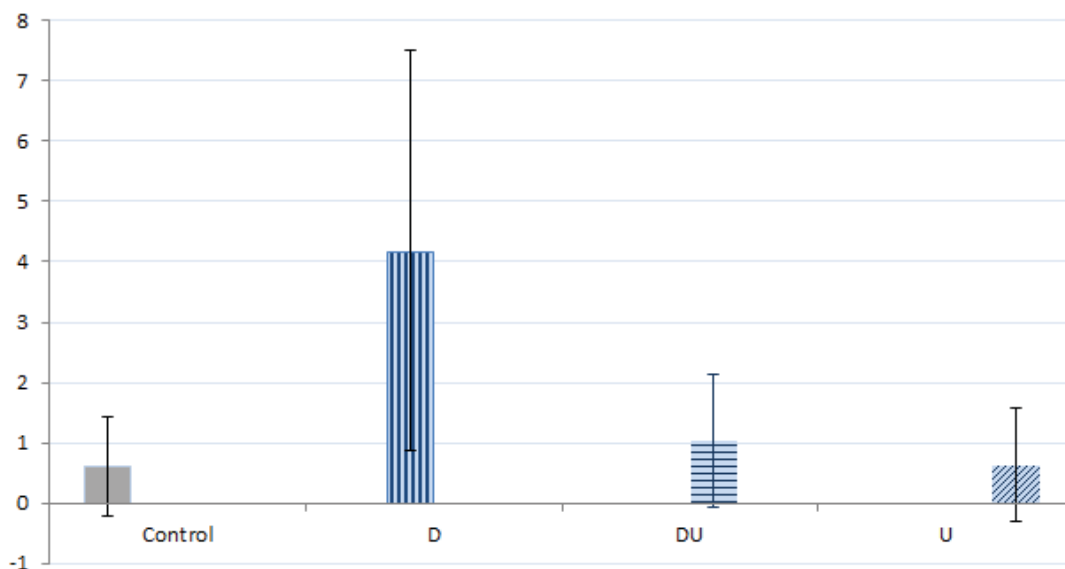


Figure 45. Taux d'erreur pour toutes les conditions.

### *Les préférences utilisateurs*

Les utilisateurs n'ont pas tous la même appréciation des différentes conditions ( $F(3) = 20.82, p < 0.001$ ). Les conditions U ( $M = 3.06, SD = 1.03$ ) et DU ( $M = 2.85, SD = 1$ ) sont appréciées par les utilisateurs, Contrôle (Control  $M = 2.36, SD = 1.08$ ) l'étant moins et D ( $M = 1.73, SD = 0.91$ ) beaucoup moins. Cependant, les résultats montrent une différence significative ( $F(1) = 10.94, p < 0.001$ ) entre D et DU. En présence de disparition progressive, le contrôle est fortement plébiscité.

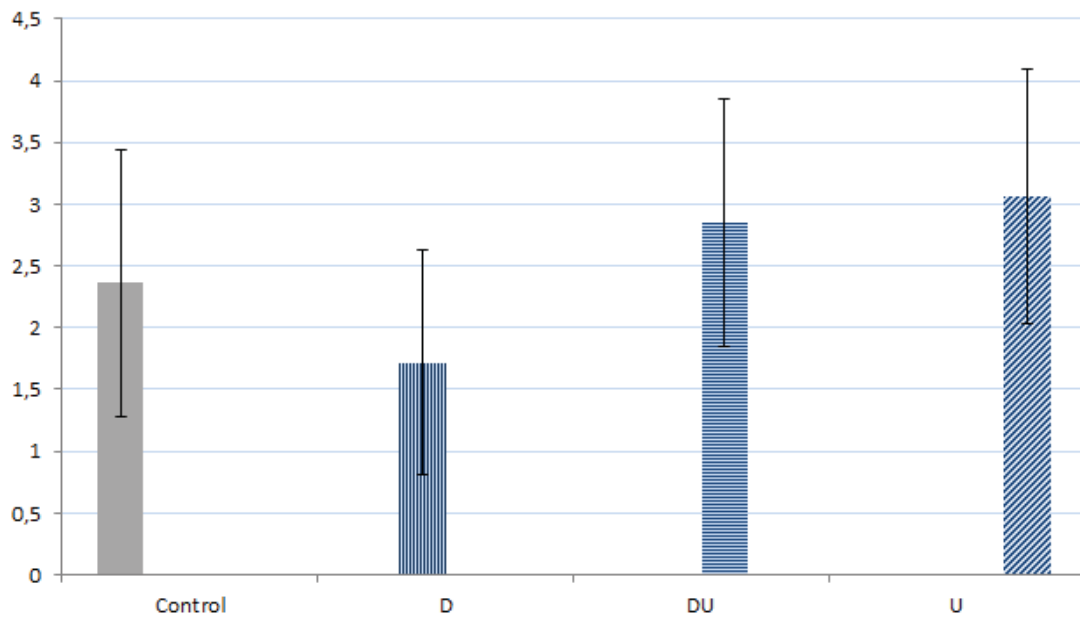


Figure 46. Préférences utilisateur pour toutes les conditions.

## **2.9. Discussions**

### **2.9.1. Discussion sur les vitesses d'exécution**

#### **- Pour un haut niveau de précision de prédiction**

*D, DU, et U seront plus rapides que la condition Contrôle. Confirmée*

Lorsque la prédiction correspond aux besoins de l'utilisateur, celui-ci est plus efficace, contrairement à la condition Contrôle où l'utilisateur doit chercher sa cible sur l'écran 1 ou 2.

*U sera plus rapide que D et DU. Confirmée*

Lors d'une prédiction correcte, l'absence d'effet de disparition est favorable à une performance élevée. L'utilisateur a besoin que la fenêtre d'incitation reste affichée pour sélectionner la cible à son rythme. Or dans D et DU, la fenêtre d'incitation disparaît progressivement.

- **Pour un faible niveau de précision de prédiction**

*DU et U seront plus rapides que D. Partiellement confirmée*

Lorsque la prédiction est incorrecte, l'utilisateur a besoin de faire disparaître la fenêtre d'incitation le plus tôt possible afin de passer à la liste complète. Ainsi, les utilisateurs sont plus rapides dans la condition DU car ils peuvent la faire disparaître rapidement. Dans la condition D, ils doivent attendre la disparition totale de la fenêtre d'incitation. Dans la condition U, c'est à l'utilisateur de fermer la fenêtre. Les bénéfices de la condition U (maîtrise de la fenêtre) sont contrebalancés par les bénéfices de la condition D (aperçu des items sous-jacents).

*DU sera plus rapide que U. Confirmée*

Dans ces deux conditions (DU et U) où le contrôle est présent, la disparition progressive de la fenêtre d'incitation dans DU permet d'obtenir un aperçu de la grille sous-jacente, d'où une plus grande efficacité.

*Dans toutes les conditions (Contrôle, D, DU, U), lorsque la cible est sur le premier écran, l'interaction sera plus rapide que lorsqu'elle est sur le second écran. Confirmée*

Les résultats sont cohérents dans les différentes conditions ce qui logique l'écran 2 étant plus difficilement atteignable que l'écran 1.

## **2.9.2. Discussion sur les préférences utilisateur**

*Les conditions adaptatives D, DU ou U devraient être préférées à la condition Contrôle. Confirmée*

L'étude montre que l'utilisateur est favorable à l'adaptation dans le cas de prédictions correctes.

*La condition U sera préférée aux conditions D et DU. Confirmée*

L'absence de délai lié à la disparition de la fenêtre d'incitation stresse moins l'utilisateur ce qui plébiscite davantage la condition U.

*La condition Contrôle ne sera pas préférée à D, ou DU ou U. Confirmée*

Dans le cas d'une prédiction incorrecte, la présence du contrôle permet de pallier les freins liés à la fenêtre d'incitation qui s'avère inutile. Dans ce cas, l'utilisateur contrôle l'interface : il n'y a pas de délai d'attente ralentissant l'interaction.

### 3. Conclusion

Dans ce chapitre, nous avons généralisé la technique *In Context Disappearing* en passant d'une représentation à une dimension (1D - liste linéaire) à une représentation en deux dimensions (2D - grille d'icônes où la navigation est horizontale et verticale). Nous avons également introduit le contrôle utilisateur. Deux variantes ont été étudiées : DU et U. DU est un processus en trois étapes (prédiction, transition, grille complète), amélioré par un contrôle utilisateur. U est un processus en deux étapes (prédiction, grille complète), avec contrôle utilisateur.

L'expérience montre que DU représente un bon compromis à la fois pour les prédictions correctes et incorrectes. Cette technique cumule les avantages de la disparition progressive et du contrôle utilisateur.

De façon générale, nous notons qu'en cas de prédictions incorrectes, les techniques sont moins performantes. Il est donc important d'améliorer la qualité des prédictions. Une autre voie est d'afficher plus d'items prédits. C'est l'objet du chapitre suivant dans lequel nous explorons l'axe : Quantité de prédiction.

### 4. Synthèse

La technique ICD (chapitre 5) est généralisée à d'autres formats de présentation tels que les mosaïques d'icônes. Elle est améliorée en explorant l'axe Contrôle via un bouton permettant d'accélérer le processus de disparition de la couche prédictive en cas de prédictions incorrectes. L'expérience montre que l'intégration du contrôle **impacte positivement** l'interaction.



## Chapitre 7 : Menu cloud, axes Quantité de prédiction et Forme

De nos jours, la croissance exponentielle de fonctionnalités sur téléphones mobiles entraîne des contraintes d'usage qui peuvent être résolues par de nouvelles techniques adaptatives (Arhipainen *al.*, 2005, Bridle *al.*, 2005, Findlater *et al.*, 2007, Gajos *et al.*, 2006, Shin *et al.*, 2012). Pour que ces techniques soient bénéfiques, elles doivent mettre en œuvre de bonnes prédictions (Bridle et McCreath, 2005, Shin *et al.*, 2012) et éviter de pénaliser l'interaction en cas de prédictions incorrectes.

Dans ce chapitre, contrairement à de nombreuses études qui se centrent sur le rendu des items prédits (Somberg, 1987, Bridle *al.*, 2005, Cockburn *al.*, 2007, Tanvir *al.*, 2008), nous explorons l'axe Quantité de prédiction. En lieu et place des trois items habituellement prédits, nous présentons six items. L'objectif est d'évaluer dans quelle mesure cette augmentation peut impacter l'interaction soit positivement en donnant plus de chances de présenter l'item recherché, soit négativement en saturant l'exploration visuelle.

A cette fin, nous proposons le menu *Cloud*, une nouvelle forme de menu adaptatif. Les six items prédits sont présentés dans une fenêtre de prédiction sous une forme circulaire brisant ainsi l'homogénéité de format avec la liste initiale des items (liste linéaire). Les axes explorés à cette occasion sont illustrés dans la figure 37.

Ce chapitre est structuré comme suit : dans la section 1, nous présentons une sélection des travaux sur les tags cloud. Dans la section 2, nous explicitons nos choix de conception ainsi que l'implémentation du menu *Cloud*. Dans la section 3, nous décrivons le protocole expérimental. Enfin nous concluons ce chapitre par une discussion des résultats obtenus.

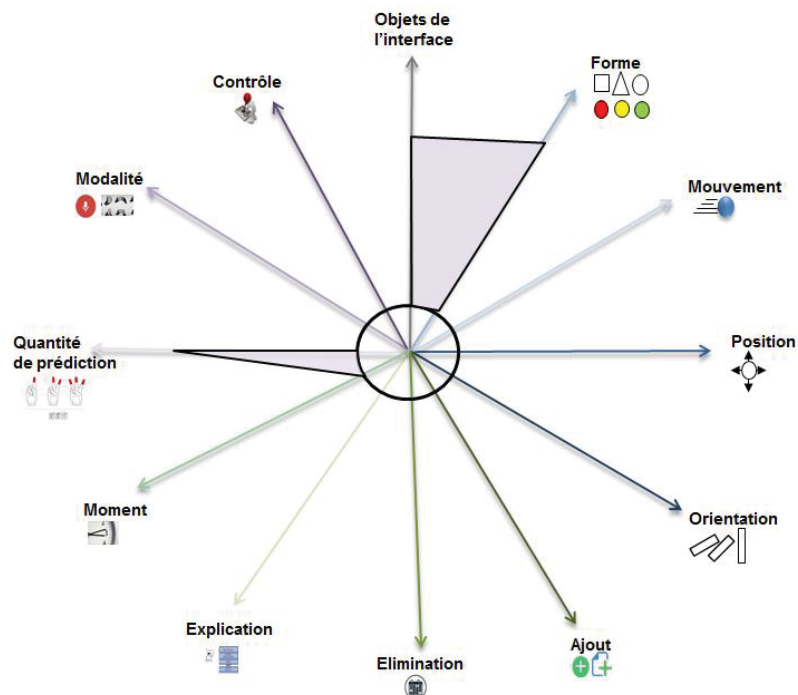


Figure 47. Caractérisation PCA du concept des menus Cloud.

## 1. Techniques étudiées

### 1.1. Travaux connexes sur les Tag Cloud

Le « *tag cloud* » ou *Wordle* (Feinberg, 2010) ou *Word Clouds* (Heimerl *et al.*, 2014) est une représentation visuelle de données textuelles utilisée pour faire ressortir des mots-clés (tags). Cela est utilisé dans le cadre d'un rapport, un site web, une série de documents, une bibliothèque numérique, etc. (Heimerl *et al.*, 2014, Lohmann *et al.*, 2009, Rivadeneira *et al.*, 2007, Seifert *al.*, 2008). Les tags sont généralement des mots simples dont l'importance est représentée à travers plusieurs variables visuelles (Bertin, 1967), telles que la taille de la police ou la couleur.

Les nuages de mots sont à la fois utilisables, simples et visuellement attrayants (Seifert *al.*, 2008). Néanmoins, plusieurs configurations sont possibles (Lohmann *et al.*, 2009, Seifert *al.*, 2008).

Halvey et Keane, 2007, ont comparé le temps de sélection d'un pays à partir d'une liste des pays présentés en 6 formats différents : liste horizontale alphabétique ( $\mu=2.88$  sec), liste verticale alphabétique ( $\mu=2.89$  sec), nuage alphabétique ( $\mu=2.94$  sec), liste horizontale ( $\mu=3.19$  sec), liste verticale ( $\mu=3.24$  sec), et un nuage normal ( $\mu=3.40$  sec). Les résultats ont montré que le nuage est plus lent que les listes horizontale et verticale triées alphabétiquement. Cependant, lorsque la taille de police des items est agrandie dans le nuage, ce dernier atteint ( $\mu=2.74$  sec).

Heimerl *et al.*, 2014 soulignent que les présentations *Cloud* sont intuitives et permettent de bien mettre en évidence certains items tout en laissant l'opportunité de consulter le contexte. Dès lors les menus *Cloud* semblent un candidat intéressant à explorer dans le cadre des techniques visuelles de l'adaptation. Néanmoins, quelques contraintes de conception sont à prendre en compte.

## 1.2. Conception du cloud

Riches des expériences antérieures, nous sommes en mesure de poser des exigences de conception pour l'élaboration de menus *Cloud*. Ces exigences sont :

- (E1) Fournir un menu adaptatif qui accélère l'interaction lorsque la prédiction est correcte et ne la ralentit pas lorsqu'elle est incorrecte ;
- (E2) Augmenter la quantité de prédiction en affichant plus de 3 items prédits (contrairement aux menus adaptatifs habituels) ;
- (E3) Offrir une bonne lisibilité de la fenêtre de prédiction malgré un grand nombre d'items dans un contexte de petit écran (*smartphone*) ;
- (E4) Briser l'homogénéité entre la fenêtre de prédiction et la liste complète des items ;
- (E5) Maintenir la stabilité spatiale de la liste initiale des items ;
- (E6) Faciliter le contrôle utilisateur.

Le menu basé nuage consiste en une liste linéaire statique (E5) superposée d'une fenêtre de prédiction sous forme de nuage de mots circulaire. Afin d'optimiser le rendu visuel du nuage (E3), nous avons catégorisé les items prédits en trois niveaux de précision qui sont rendus soit par une position centrale et une police de grande taille, soit en périphérie avec une police de taille moyenne, soit en périphérie avec une police de petite taille. Le menu *Cloud* présenté sur



la figure 48 est décrit comme suit : l’item « Walnut » possède la plus haute probabilité (centre du cloud avec une grande taille de police) ; les items « Pistachio, Saturn, Spain, Chardonnay et Merlot » ont une probabilité moyenne (périphérie) et sont classés en deux catégories (petite et moyenne taille).

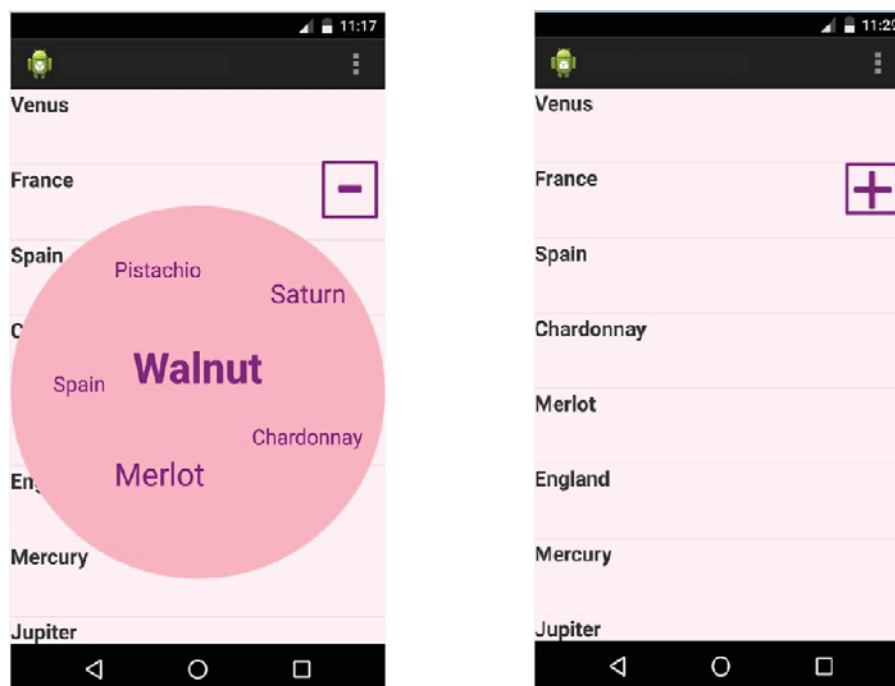





Figure 48. Nuage de mots circulaire avec items prédits (menu Cloud déplié) / liste complète des items (menu Cloud replié).

Tous les nuages présentent six items prédits (E2). Néanmoins, augmenter la quantité d’items au sein de la liste de prédiction risque d’engendrer trop de similarité visuelle avec la liste complète, et par là même des confusions entre la liste complète et la liste de prédiction. Il est donc *a priori* important de briser l’homogénéité graphique (E4). Afin de confirmer ce postulat, nous mettrons en œuvre une condition comparative Linéaire6 dans laquelle toutes les listes présentées verticalement. Le menu linéaire6 est illustré dans la figure 49. Nous introduisons également une autre condition Linéaire3 dans laquelle la fenêtre de prédiction contient uniquement trois items. L’objectif est encore une fois de confirmer si l’exploration d’une grande liste d’items prédits sous forme de *Cloud* est aussi facile qu’une liste de trois items. Le menu Linéaire3 est illustré dans la figure 50.

Deux boutons  et  sont introduits afin de donner le contrôle de la fenêtre de prédiction à l'utilisateur (E6). Ils permettent de faire disparaître et/ou apparaître la fenêtre de prédiction. Lorsque la prédiction est correcte, l'utilisateur sélectionne l'item directement à partir du nuage de mots. Dans le cas contraire (prédiction incorrecte), l'utilisateur peut faire disparaître le nuage de mots (inutile dans ce cas) en appuyant sur le bouton  et ensuite sélectionner sa cible sur le menu statique (liste complète des items).

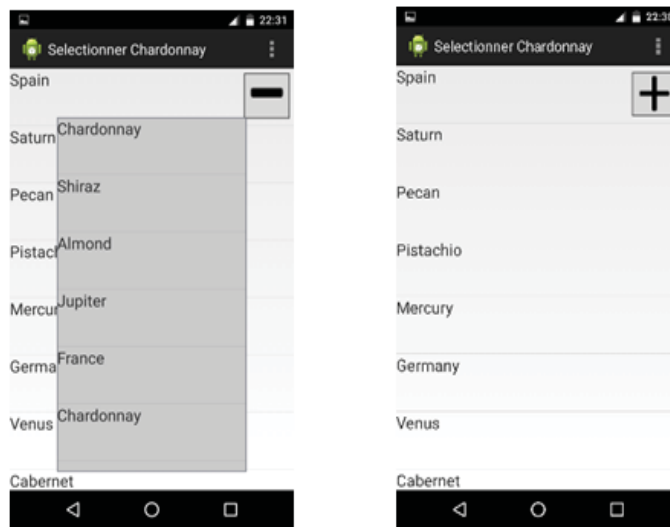


Figure 49. Menu Linéaire6 avec six items prédits (menu Linéaire déplié) / liste complète des items (menu Linéaire replié).



Figure 50. Menu Linéaire3 avec trois items prédits.

L'expérimentation permet de vérifier l'exigence E1, c'est-à-dire la performance de la technique aussi bien dans le cas de prédictions correctes qu'incorrectes.

## **2. Evaluation expérimentale**

L'évaluation vise à vérifier l'efficacité de la technique *Cloud* au regard des techniques Linéaire6 et Linéaire3 aussi bien dans le cas de prédictions correctes qu'incorrectes.

### **2.1. Méthodologie**

Deux variables indépendantes sont manipulées : le type du menu c'est-à-dire la condition Contrôle, le menu *Cloud*, le menu Linéaire6 et le menu Linéaire3 ; le niveau de précision de la prédiction qu'elle soit correcte ou incorrecte.

Chaque menu est composé de 16 items divisés en deux écrans. Dans le menu *Cloud*, les prédictions correctes sont réparties de façon égale entre le centre et la périphérie.

### **2.2. Hypothèses**

Nous avons posé les hypothèses suivantes :

#### **2.2.1. Hypothèse 1 : vitesse d'exécution de la tâche**


##### **- Pour un haut niveau de précision de prédiction**

*Le menu Cloud sera plus rapide que le menu Linéaire6.* Nous supposons que la représentation *Cloud* facilite la recherche visuelle, contrairement à une liste Linéaire incitant à une exploration de haut en bas.

*Il n'y aura pas de différence significative entre le menu Cloud et le menu Linéaire3.* Nous supposons que la représentation *Cloud* facilite la recherche visuelle et donc ne sera pas pire que le menu Linéaire3 avec trois items prédits.

*La cible située au centre du menu Cloud sera sélectionnée plus rapidement que lorsqu'elle est située en périphérie.* En effet, la cible située au centre du *Cloud* est plus facile à localiser et à sélectionner grâce à sa position centrale et à sa grande taille (induisant ainsi une grande zone d'activation comme dans AAMU (Tanvir *et al.*, 2008).

- **Pour un faible niveau de précision de prédiction**

*Le menu Cloud et le menu Linéaire3 ne demanderont pas plus de temps que la condition Contrôle. Le bouton  devrait permettre à l'utilisateur de fermer la fenêtre de prédiction lorsqu'elle est incorrecte.*

*Le menu Cloud sera plus rapide que le menu Linéaire6. Comme dans le cas d'une prédiction correcte, la recherche visuelle devrait être facilitée avec le menu Cloud.*

**2.2.2. Hypothèse 2 : erreurs**

- **Pour un haut niveau de précision de prédiction**

*Aucune différence entre le menu Cloud, le menu Linéaire6 et le menu Linéaire3 et moins que la condition Contrôle. Lorsque la cible est sur la fenêtre de prédiction, la distance de navigation est moindre ce qui réduit le risque d'erreurs.*

*Dans le menu Cloud, une cible située au centre générera moins d'erreurs qu'en périphérie. La cible centrale est de grande taille ce qui réduit les erreurs de sélection.*

- **Pour un faible niveau de précision de prédiction**

*Aucune différence significative entre le menu Cloud, le menu Linéaire6, le menu Linéaire3 et la condition Contrôle. Dans les quatre conditions, l'utilisateur doit trouver la cible dans la liste complète. Nous supposons que ces trois conditions seront similaires en taux d'erreur.*

**2.3. Mesures quantitatives et qualitatives**

Nous avons mesuré deux variables dépendantes : (1) La vitesse de sélection en secondes du temps s'écoulant entre l'ouverture du menu et la sélection de la cible. (2) Le taux d'erreur enregistré lors de l'accomplissement de la tâche.

La figure 51 illustre les conditions de l'expérimentation.

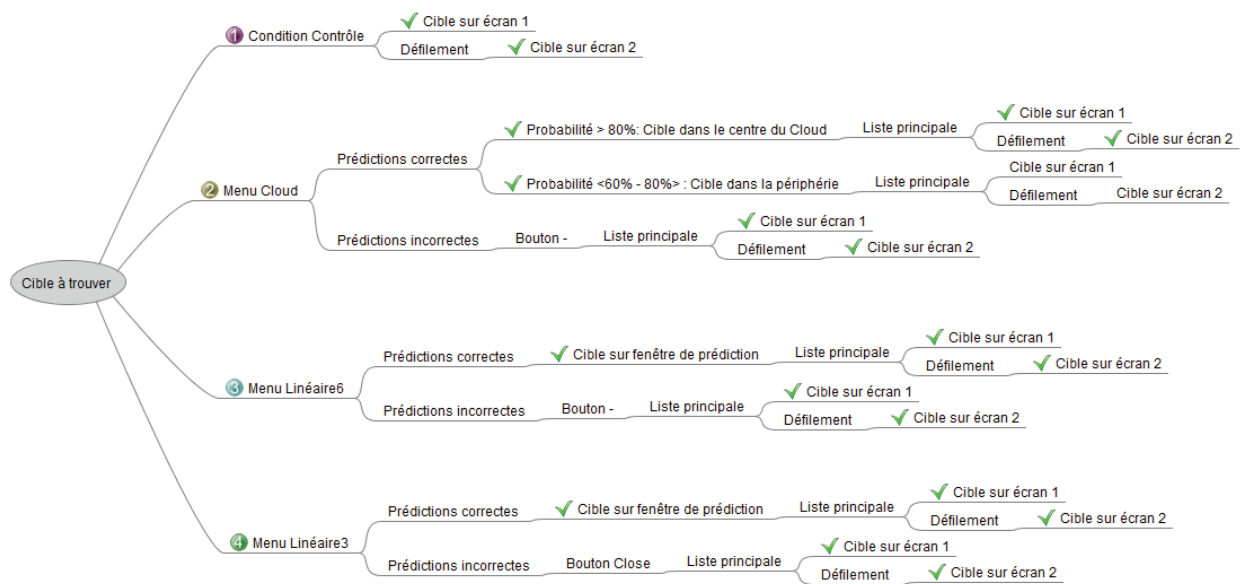


Figure 51. Conditions expérimentales.

## 2.4. Tâche

Deux groupes indépendants de sujets ont participé à l'expérimentation. Dans chaque groupe, nous avons demandé aux utilisateurs d'effectuer une séquence de sélections. Les participants du groupe 1 ont manipulé les conditions *Cloud* et *Contrôle*. Les participants du groupe 2 ont manipulé les conditions *Linéaires* et *Contrôle*. Pour chaque condition, une consigne indiquant la cible apparaît à l'écran puis disparaît. Le nom de la cible reste affiché en haut de l'écran. Dans la condition *Cloud*, les participants sélectionnent la cible à partir de la fenêtre de prédiction sous forme *Cloud* et/ou à partir du menu statique (liste complète des items répartis sur les écrans 1 et 2). Dans les menus *Linéaires*, les participants sélectionnent la cible sur la fenêtre de prédiction sous forme linéaire et/ou à partir de la liste complète des items. Dans la condition *Contrôle*, les participants sélectionnent directement la cible sur l'écran 1 ou sur l'écran 2. Dans toutes les conditions, lorsque l'utilisateur sélectionne la cible demandée, un nouveau message apparaît pour lui indiquer le nom de la nouvelle cible à sélectionner. Lorsque l'utilisateur sélectionne la mauvaise cible, un message d'erreur est affiché. Lorsque l'utilisateur sélectionne la bonne cible, il passe à la cible suivante. A la fin du test, un message de remerciement est affiché indiquant à l'utilisateur que le test est terminé.

## 2.5. Matériels

Les appareils utilisés dans cette expérimentation sont des *smartphones* de type Google Nexus sous un système d'exploitation Android. Une base de donnée a été créée dans laquelle le temps de sélection (en secondes), le temps de défilement (en millisecondes) et le taux d'erreur sont enregistrés.

## 2.6. Participants

Deux groupes indépendants d'un total de trente-huit sujets ont participé à l'expérimentation. Tous les participants sont des utilisateurs réguliers de *smartphones*.

## 2.7. Procédure

Les utilisateurs ont passé un pré-test permettant de s'entraîner préalablement. Chaque participant doit sélectionner correctement 10 cibles dans le pré-test avant de passer au vrai test. Les cibles à sélectionner sont réparties comme suit :

- Pour le groupe 1 : 40 cibles pour le menu *Cloud* dont
    - o 20 sélections lorsque la prédiction est correcte : 10 au centre du *Cloud* et 10 en périphérie.
    - o 20 sélections lorsque la prédiction est incorrecte : 10 sur le premier écran et 10 sur le second écran.
  - Pour le groupe 2 : 40 cibles pour le menu Linéaire6 dont
    - o 20 sélections lorsque la prédiction est correcte
    - o 20 sélections lorsque la prédiction est incorrecte : 10 sur le premier écran et 10 sur le second écran.
- 20 cibles pour le menu Linéaire3 dont
- o 10 sélections lorsque la prédiction est correcte.
  - o 10 sélections lorsque la prédiction est incorrecte : 10 sur le premier écran et 10 sur le second écran.

Les deux groupes ont effectué 20 sélections dans la condition Contrôle : 10 lorsque la cible est sur le premier écran et 10 lorsqu'elle est sur le second.

## 2.8. Résultats

Les données ont été analysées par une One-way ANOVA en inter-sujet et *Wilcoxon Signed Ranks* en intra-sujet.

L'absence de différence significative ( $F(22,14) = 1.43, p < 0.3$ ) en condition Contrôle entre les groupes 1 ( $M = 3.40, SD = 1.07$ ) et 2 ( $M = 3.04, SD = 0.63$ ) confirme l'homogénéité des échantillons.

Lorsque la prédiction est correcte, le menu adaptatif *Cloud* ( $M = 1.76, SD = 0.61$ ) est significativement plus rapide ( $F(22,14) = 3.08, p < 0.005$ ) que le menu Linéaire6 ( $M = 4.12, SD = 3.08$ ).

Il n'y a pas de différence significative ( $F(22,14) = 4.46, p < 0.5$ ) entre le menu *Cloud* et le menu Linéaire3 ( $M = 1.4, SD = 0.35$ ).

Dans la condition *Cloud*, lorsque la cible est située dans le centre ( $M = 1.19, SD = 0.54$ ), les utilisateurs sont significativement ( $W(23), 3.71, p < .0005$ ) plus rapides que lorsqu'elle est en périphérie ( $M = 2.35, SD = 1.04$ ).

A l'inverse, lorsque la prédiction est incorrecte, les utilisateurs sont significativement plus rapides dans le menu Linéaire6 ( $M = 3.40, SD = 2.84$ ) et le menu Linéaire3 ( $M = 4.25, SD = 0.88$ ) que dans le menu *Cloud* ( $M = 5.60, SD = 1.65$ ), ( $F(22,14) = 9.21, p < .005$  respectivement  $F(22,14) = 8.42, p < .005$ ).

Les résultats du temps de sélection dans les différentes conditions expérimentales sont fournis dans la figure 52.

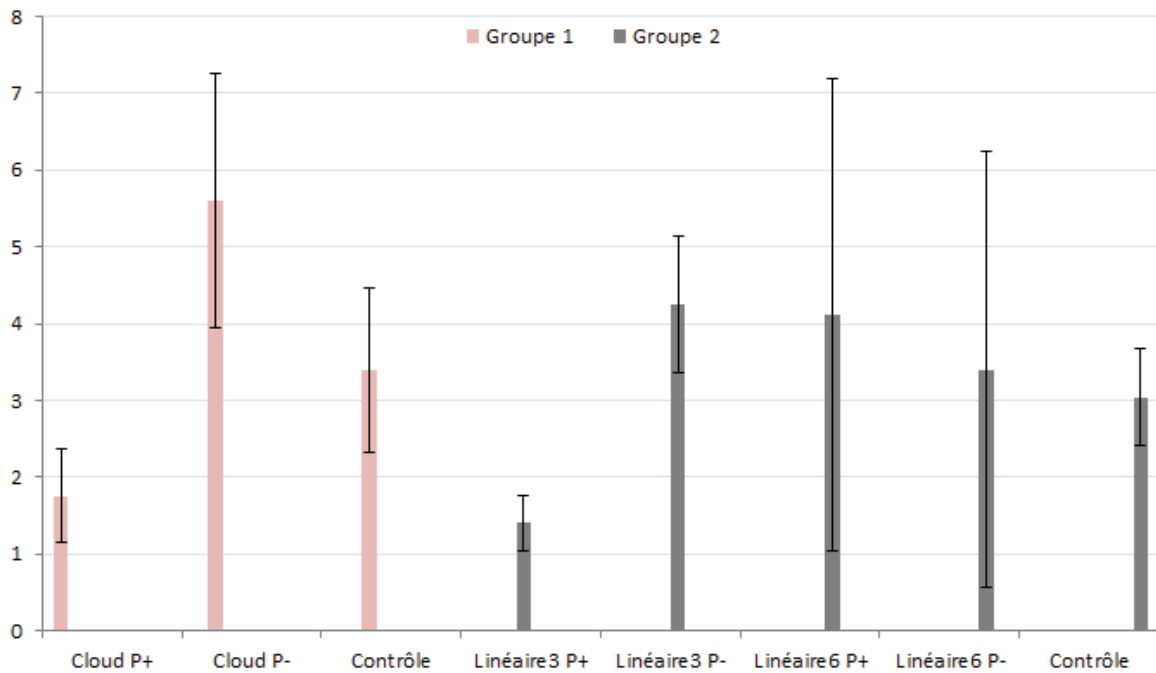


Figure 52. Temps de sélection des conditions Cloud, Linéaire3, Linéaire6 et Contrôle.

Concernant les erreurs, dans les deux échantillons (« groupe1 »  $M = 0.17$ ,  $SD = 0.29$ , « groupe2 »  $M = 0.33$ ,  $SD = 0.62$ ), le nombre d'erreurs est équivalent dans la condition Contrôle ( $F(22, 14) = 1.16$ ,  $p = .29$ ).

Le menu *Cloud* présente davantage d'erreurs ( $W(22) = 3.04$ ,  $p < .005$ ) que la condition Contrôle. Cela est principalement dû à des erreurs de pointage des cibles en périphérie ( $W(22) = 3.44$ ,  $p < .001$  ; « cible centre »  $M = 0.04$ ,  $SD = 0.21$ , « cible périphérie »  $M = 1.87$ ,  $SD = 2.40$ ).

Lorsque la prédiction est incorrecte dans le menu *Cloud* ( $M = 0.74$ ,  $SD = 1.74$ ), aucune différence significative ( $W(22) = 1.22$ ,  $p = .22$ ) n'est mise en évidence par rapport à la condition Contrôle ( $M = 0.17$ ,  $SD = 0.29$ ).

Les erreurs sont significativement moins fréquentes ( $W(14) = 2.56$ ,  $p = .01$ ) dans la condition Contrôle ( $M = 0.33$ ,  $SD = 0.62$ ) que dans les menus Linéaires ( $M = 1.93$ ,  $SD = 2.12$ ).



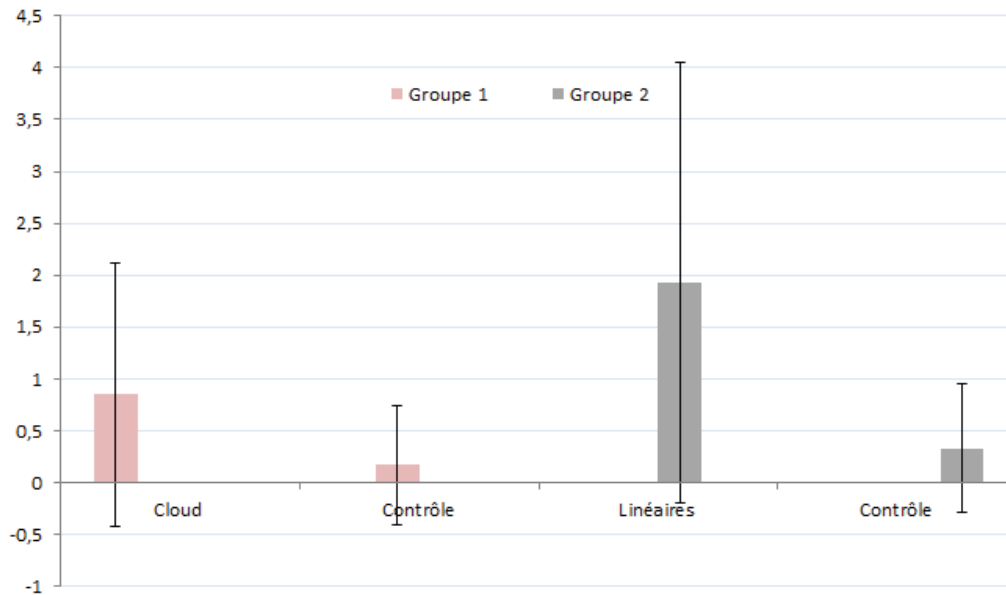


Figure 53. Taux d'erreur pour toutes les conditions.

## 2.9. Discussion

### 2.9.1. Hypothèse 1 : vitesse d'exécution de la tâche

#### - Pour un haut niveau de précision de prédiction

*Le menu Cloud sera plus rapide que le menu Linéaire6. Confirmée*

A priori la représentation du *Cloud* facilite l'exploration d'un grand nombre d'items. L'exploration visuelle semble facilitée par l'agencement en nuage des items. A l'opposé, le menu *Linéaire6* impose un ordre de lecture.

*Il n'y aura pas de différence significative entre le menu Cloud et le menu Linéaire3. Confirmée*

Globalement le menu *Cloud* n'est pas pire que le menu *Linéaire3*. Ce résultat confirme la précédente hypothèse que l'exploration visuelle est facilitée par l'agencement en *Cloud*. Cependant, lorsque la cible est en périphérie du *Cloud*, ce dernier est moins rapide que le menu *Linéaire3*. En conséquence, il est important d'optimiser la qualité des prédictions et d'afficher la cible au centre.

*La cible située au centre du menu Cloud sera sélectionnée plus rapidement que lorsqu'elle est située dans la périphérie. Confirmée*

L'item central est de surcroît présenté dans une taille plus élevée ce qui facilite doublement sa perception.

- **Pour un faible niveau de précision de prédiction**

*Le menu Cloud et le menu Linéaire3 ne demanderont pas plus de temps que la condition Contrôle. Partiellement Confirmée*

En cas de prédiction incorrecte, le menu *Cloud* n'est pas très performant. Les conditions Contrôle et Linéaires restent plus rapides que le *Cloud*. Une explication pourrait être qu'en cas de cible absente, la structure séquentielle imposée par les conditions Linéaires facilite l'identification de l'absence. En revanche, sein d'un nuage, l'utilisateur a tendance à davantage relire les items de peur d'avoir manqué la cible.

### 2.9.2. Hypothèse 2 : erreurs

- **Pour un haut niveau de précision de prédiction**


*Lorsque la cible est située dans le centre du Cloud, les erreurs seront moins fréquentes que lorsque la cible est dans la périphérie. Confirmée*

Les items en périphérie sont plus petits et donc source de davantage d'erreurs.

La présentation des items prédits à l'intérieur du *Cloud* peut être améliorée. Un algorithme pourrait être mis en œuvre afin de calculer la distance entre les items et le centre afin d'assurer une distribution optimale, ou encore, positionner les items en fonction de leur relation sémantique.

- **Pour un faible niveau de précision de prédiction**

*Aucune différence significative entre le menu Cloud et la condition Contrôle. Confirmée*

En cas d'une mauvaise prédiction, les participants ne sélectionnent pas dans le *Cloud* mais le font disparaître par le bouton . Dans ce cas, la liste statique est manipulée comme dans la condition Contrôle. Ceci justifie l'absence de différence entre le menu *Cloud* et la condition Contrôle.

## 3. Conclusion

Dans ce chapitre, nous avons présenté le menu *Cloud*, un menu en nuage de mots : les items prédits sont regroupés dans une couche prédictive sous forme d'un nuage de mots circulaire.

Ce nuage se superpose au menu statique (la liste complète des items) et il est contrôlable par l'utilisateur.

Nous avons exploré les axes Quantité de prédiction et Forme en affichant six items au sein d'un nuage brisant l'homogénéité avec la liste complète. Ce menu *Cloud* s'avère efficace mais uniquement lorsque la prédiction est correcte.

La technique du menu *Cloud* pourrait être étendue à d'autres cas d'usage. Par exemple, pour la navigation dans des menus hiérarchiques où la distance d'un item au centre du *Cloud* peut être calculée en fonction de sa place dans l'arborescence : plus l'item est profond dans l'arborescence, plus il sera loin du centre du *Cloud*. Cette représentation peut augmenter l'intelligibilité et la compréhension de la structure hiérarchique.

Dans le chapitre suivant, nous étudions l'adaptation des menus hiérarchiques sur smartphone.

## 4. Synthèse

Dans ce chapitre, nous avons joué sur l'axe **Quantité de prédiction** en augmentant le nombre d'items prédits (six items). Cette augmentation donne plus de chances à l'utilisateur de trouver sa cible.

Conjointement, nous avons mobilisé l'axe **Forme** en **brisant l'homogénéité graphique** entre la **liste initiale** des items et la **couche de prédiction**. Cette dernière s'affiche sous forme d'un nuage de mots (menu *Cloud*).

L'expérience montre que pour présenter davantage de prédictions, la technique *Cloud* est une piste intéressante.



## Chapitre 8 : Shortcut et step-by-step, axe Moment

Ce chapitre a fait l'objet de deux publications en conférences internationales :

**Bouzit**, S., Calvary, G, Chêne, D., Vanderdonckt, J. (2016). A Comparison of Shortcut and Step-by-Step Adaptive Menus for Smartphone. Proceedings British HCI 2016 – Fusion, Bournemouth, UK.

**Bouzit**, S., Calvary, G, Chêne, D., Vanderdonckt, J. (2016). Automated Evaluation of Menu Guidelines Review. Proceedings of Ro-CHI International Conference on Human Computer Interaction. pp 11-21.

Et en conférence nationale :

Khaddam, I., **Bouzit**, S., Calvary, G., Chêne, D. MenuErgo : Conception assistée de menus par évaluation automatique de règles ergonomiques. Actes de la 28<sup>ième</sup> conférence francophone sur l'Interaction Homme-Machine (IHM), Octobre 2016, Fribourg, Suisse. pp.36-47.

Une technique d'interaction classique en interfaces graphiques est le menu hiérarchique. Par exemple, les menus en cascade de la barre de menu sur ordinateur (sous Windows, Linux, Apple) ou les listes imbriquées sur téléphones mobiles (Ahlström, 2006; Cockburn & Gin, 2006; Bailly *et al.*, 2009) sont des variations du concept de menu hiérarchique.

Les dispositifs à petits écrans réduisent considérablement les possibilités d'affichage. Dès lors, le positionnement d'un menu générique n'est plus envisageable. L'interaction se fait généralement par des menus contextuels liés à la tâche en cours.

Aujourd'hui, les menus sont surtout des listes hiérarchiques. Ces listes sont emboîtées au sein d'une arborescence. L'utilisateur passe progressivement d'un sous-menu à l'autre et progresse ainsi dans l'arborescence. Ce processus de navigation est appelé navigation hiérarchique linéaire (Chêne *et al.*, 2016).

La conception des menus implique une approche multifactorielle (Norman, 1991) équilibrant les facteurs intrinsèques de la navigation hiérarchique (profondeur, largeur, présentation, zone d'activation) et les facteurs extrinsèques (temps de sélection de l'item, temps de réaction au pointage). Le compromis largeur versus profondeur est un facteur déterminant (Jacko et Salvendy, 1996). Lorsqu'une cible est placée profondément au sein d'une arborescence, le temps de navigation augmente de fait. Ce temps peut être réduit par l'utilisation d'un raccourci présenté dans les premiers niveaux, ce qui est rarement le cas dans les menus en cascade (Ahlström, 2006), et jamais dans les listes imbriquées sur *smartphones* (Park *et al.* 2007).

Kiger, 1984 a fortement recommandé de minimiser la profondeur d'un menu puisqu'elle impacte négativement à la fois le temps de sélection d'item et le taux d'erreur. Cependant, avec l'avènement des *smartphones*, la quantité des items affichables sur un seul écran est limitée, les concepteurs ont tendance à accroître la profondeur de l'arborescence pour éviter l'utilisation de l'ascenseur. De plus, selon Pirolli, (2009) les utilisateurs sont prêts à parcourir davantage d'arborescence et à mettre plus de temps pour atteindre leur cible dès lors qu'ils sont guidés. Le temps de sélection de l'item peut être ajusté en fonction du niveau d'aide apportée à l'utilisateur.

Un autre facteur réside dans la complexité de la hiérarchie. La structure de l'arborescence a un impact sur la construction du modèle mental de l'utilisateur. Il doit apprendre la structure, et

constamment la mettre à jour en fonction de ses explorations. Les utilisateurs novices vont passer du temps à découvrir l'arborescence, et plus elle sera profonde, plus l'acquisition sera difficile. Huang *et al.*, 2006 ont montré que les utilisateurs n'apprécient pas la navigation hiérarchique sur *smartphone* et préfèrent les structures de menu peu profondes. Plusieurs études ont traité ce problème soit en accélérant la navigation hiérarchique (John *et al.*, 2004), soit en accélérant la phase de sélection de l'item (Cockbrun et Gin, 2006), soit en optimisant l'organisation de l'arborescence (Bailly *et al.*, 2013 ; Matsui et Yamada, 2008), soit en recommandant à l'utilisateur certains items (Shin *et al.*, 2012).

Nous n'aborderons pas dans ce chapitre la question de l'optimisation de la structure hiérarchique, mais nous nous centrons sur les questions d'apprentissage et d'accélération de navigation au sein de cette structure. Les rapports au temps sont différents que l'on soit en situation d'apprentissage ou d'accélération. Nous explorons donc l'axe Moment du modèle PCA en comparant les avantages et inconvénients de deux menus adaptatifs faisant varier ces critères. Le premier est appelé adaptation *step-by-step*. Il permet d'accompagner l'utilisateur dans l'apprentissage de la structure arborescente. Le second menu est appelé *shortcut*. L'objectif de cette technique est d'accélérer l'interaction en proposant la cible à la racine. Les axes explorés sont présentés dans la figure 54.

Ce chapitre est organisé en 4 sections principales. La section suivante fait la revue critique des menus hiérarchiques adaptatifs. La section 2 introduit les deux menus adaptatifs *Step by Step* et *Shortcut*. La section 3 présente l'expérimentation comparant les deux types de menus adaptatifs. Enfin, la section 4 conclut ce chapitre en soulignant les bénéfices et limitations de ces deux techniques et en suggérant de futures pistes de recherche.

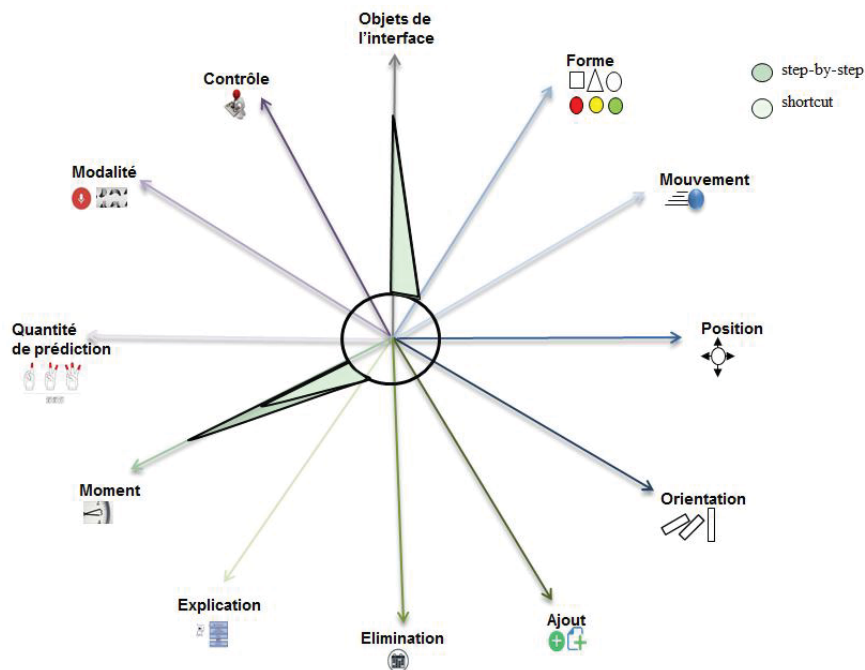


Figure 54. Axes explorés par les techniques step-by-step et shortcut.

## 1. Techniques étudiées

### 1.1. Travaux connexes sur les menus hiérarchiques

De nombreux auteurs se sont attelés à la conception de menus hiérarchiques. Certains ont travaillé davantage sur la représentation graphique d'une structure arborescente (Hesselmann, 2009, Lamping *et al.* 1995, Plaisant *et al.*, 2002, Johnson et Shneiderman, 1991). D'autres ont travaillé sur des outils d'aide à la conception permettant d'optimiser la structure (Vanderdonckt, 1999, Bailly *et al.*, 2013, Khaddam *et al.*, 2016). D'autres se sont concentrés sur des techniques facilitant la manipulation (Ahlström *et al.*, 2010, Cockburn et Gin, 2006, Tsandilas et Schraefel, 2007).

#### 1.1.1. Représentation graphique d'une structure arborescente

Le menu *stacked half-pie* développé par Hesselmann, 2009, consiste à afficher les items au sein de demi-cercles concentriques sur tabletop. Ainsi quelles que soient la largeur et la



profondeur, la représentation graphique est toujours identique. Cependant, cette conception nécessite un grand espace et ne peut être appliquée aux petits écrans.

Le menu hyperbolique (Lamping *et al.* 1995) est une technique dite « focus + contexte » destinée à la présentation et à la manipulation de hiérarchies larges et profondes. Cette technique consiste à afficher plusieurs niveaux de profondeur simultanément. Les problèmes majeurs du menu hyperbolique sont son instabilité spatiale induisant une manipulation difficile et sa nécessité d'être affiché sur un espace important.

*Space Tree* (Plaisant *et al.*, 2002) permet de dynamiquement redimensionner les branches de l'arborescence afin de mieux s'adapter à l'espace disponible. Cette technique a été comparée à l'explorateur de Microsoft et au navigateur hyperbolique (Lamping *et al.* 1995). Les résultats ont montré que le menu *space tree* est significativement plus rapide que le menu hyperbolique et que l'explorateur Microsoft est significativement plus rapide que les deux autres. Néanmoins, les utilisateurs ont préféré *space tree* et le menu hyperbolique à l'explorateur. Dans un contexte de petit écran, les menus Hyperbolique et explorateur ne sont pas applicables. Comme ils affichent simultanément plusieurs niveaux d'arborescence et le contenu de l'arborescence en cours, ils ont besoin d'au moins deux zones d'affichage. A l'opposé, si l'on pousse la logique du *space tree* dans ses retranchements, alors le redimensionnement dynamique de l'arbre dans un petit espace contraint à n'afficher qu'un seul niveau de liste, toutes les branches étant repliées.

Le concept de *Tree-Maps* proposé par Johnson et Shneiderman, 1991, consiste à afficher la hiérarchie sous forme de boîtes imbriquées rectangulaires. Dans un contexte de *smartphone*, la taille réduite de l'écran rend cette technique inapplicable, les zones devenant trop petites.

La visualisation de plusieurs niveaux d'arborescence de façon simultanée au sein d'un petit écran paraît difficile, la présentation d'un seul niveau par contre est tout à fait envisageable. Certains auteurs proposent même de réduire au sein de ce niveau le nombre d'items présentés afin de répondre à des objectifs d'apprentissage. En effet, (Kang *et al.*, 2003, Shneiderman, 2003, Clark et Matthews, 2005, Huang *et al.*, 2006) proposent des interfaces allant dans ce sens. Elles sont évolutives et s'enrichissent au fil du temps telles que les *Training Wheels* (Catrambone et Carrol, 1987).

### 1.1.2. Aide à la conception

Vanderdonckt, 1999, a proposé le MenuDesigner qui assiste le concepteur lors de la génération d'un menu hiérarchique. Le principe est de construire l'arbre en fonction de la tâche utilisateur. Cette approche reste statique car la structure du menu est générée une fois pour toute, sans aucune adaptation.

Bailly *et al.*, 2013, ont proposé le MenuOptimizer permettant d'assister les designers et développeurs dans l'optimisation de la structure de menu tout en tenant compte des critères de cohérence et de performance. La similarité des fonctions permet de créer des catégories, alors que la fréquence d'usage permet d'optimiser la manipulation (items plus grands) et l'ordonnement (soulignement). Ce dernier reste à la main du concepteur ce qui permet d'améliorer la stabilité spatiale contrairement aux adaptations automatiques générées dans le cadre de l'outil proposé par Matsui et Yamada, 2008. Néanmoins, le MenuOptimizer n'offre pas d'évaluation au cours de la conception.

Ce point est assuré au sein du MenuErgo (Khaddam *et al.*, 2016) qui offre aux concepteurs une aide associée à des évaluations ergonomiques automatiques basées sur les règles ergonomiques de Bastien et Scapin (1995). Cependant, toutes ces règles ne peuvent être automatisées.

Soulignons enfin qu'aucune de ces méthodes ne permet d'assister le concepteur dans la mise en place de critères adaptatifs, qu'ils soient liés à la structuration des catégories au sein de l'arbre ou à l'amélioration de la performance.

### 1.1.3. Facilitation de la manipulation

Ahlström *et al.*, 2010 ont proposé les menus *square* permettant d'améliorer les performances de pointage des utilisateurs experts. Les performances de ces menus sont supérieures aux menus linaires traditionnels et aux menus *pie*. En effet, le menu *square* permet de réduire le temps de pointage selon la loi de Fitts (les zones d'activation étant plus grandes, la certitude de pointage est mieux tolérée). Néanmoins, les utilisateurs novices sont contraints par ces menus. Ils ont des performances meilleures avec les menus traditionnels. Les auteurs n'expliquent pas les causes de ces moindres performances.

Le menu Flower proposé par Bailly *et al.*, 2008, permet de tracer les lignes droites ou des courbes pour atteindre une cible. Une étude comparative des menus Flower, linéaire et

polygone montre que les menus polygone et Flower facilitent l'apprentissage du mode expert contrairement au menu linéaire. L'intérêt de ces modes est de relâcher les contraintes de pointage. Néanmoins, de tels menus sont difficilement applicables sur petits écrans.

Cockburn et Gin, 2006, ont rendu la manipulation des menus en cascade plus facile (29% de gain) en élargissant les zones d'activation. L'objectif est d'éliminer le délai de disparition du sous menu.

Le menu *Bubbling* proposé par Tsandilas et Schraefel, 2007, est également une conception destinée aux menus déroulants en cascade. Il vise à accélérer la sélection des items fréquemment utilisés en augmentant la zone d'activation du curseur lorsque ce dernier passe à proximité d'un item fréquemment utilisé. Cela permet *a priori* à l'utilisateur de réduire sa contrainte de pointage. Contrairement aux autres techniques décrites ci-dessus, le menu *Bubbling* intègre l'adaptation en facilitant l'action sur les items les plus fréquemment utilisés. Néanmoins, il génère un grand nombre d'erreurs surtout dans le cas de prédictions incorrectes certainement dûes à la nature instable de la zone d'activation. De plus, comme pour les menus proposés par Cockburn et Gin, 2006, l'amélioration des contraintes de pointage ne saurait être suffisante car sur petits écrans les menus en cascade sont difficilement utilisables du fait du peu d'espace.

De l'état de l'art sur la représentation graphique de la structure arborescente, nous soulignons que dans le cadre d'un petit écran de *smartphone*, un seul niveau d'arborescence peut être représenté avec suffisamment de confort visuel et de pointage. Néanmoins dans ce cas, la vue de l'arborescence est parcellaire. Cela risque de générer des difficultés d'apprentissage de cette arborescence.

De l'état de l'art sur l'aide à la conception, nous soulignons que l'organisation de la tâche est un élément crucial de découpage de l'arborescence mais que peu de critères existent aujourd'hui pour définir une arborescence adaptative.

De l'état de l'art de l'optimisation de la manipulation, nous soulignons qu'agrandir la zone de pointage est certes une idée intéressante mais risque d'occuper trop de place sur un petit écran. Les techniques basées gestes existantes utilisent l'affichage simultané de deux niveaux de l'arborescence et sont donc exclues tout comme les contraintes temporelles et les changements

inopinés des zones d'activation. Ces optimisations de pointage sont inopérantes plus dans le cadre d'un petit écran notamment lors de prédictions négatives.

Dans ce chapitre, nous traitons donc d'arborescences adaptatives facilitant l'apprentissage et/ou la performance sur petits écrans, dans le cas de prédictions correctes et incorrectes.

L'activation de navigation en sein de l'arborescence peut se décomposer en plusieurs sous-tâches. Ainsi, nous pouvons nous interroger sur le Moment durant lequel nous allons présenter l'adaptation : soit tout au long de phase de navigations au sein de l'arborescence, soit en début de tâche. Nous proposons ci-après les menus adaptatifs résultant de cette approche.

## **1.2. Conception des prototypes**

Nous présentons ici les deux techniques *step-by-step* et *shorcut*.

### **1.2.1. Step-by-Step**

La technique *step-by-step* est présentée dans les figures 55 et 56. Ce prototype consiste à afficher une hiérarchie homogène constante : une liste par écran contenant toujours le même nombre d'items. L'item prédit est présenté dans une fenêtre de prédiction qui se superpose au menu initial par un alpha-blending (Baudish et Cutwin, 2004). L'item prédit au sein de cette fenêtre correspond à un des items du niveau courant et permet d'atteindre le niveau suivant (figure 56a) où une autre fenêtre de prédictions sera disponible permettant la navigation vers le niveau directement inférieur (figures 56b, c, d). L'objectif de *step-by-step* est d'accélérer l'interaction tout en facilitant l'apprentissage de l'arborescence.

*Step-by-step* devrait être efficace en cas de prédiction correcte car la manipulation et le fonctionnement sont stables (l'item ciblé est toujours sous le doigt), les actions répétitives devenant alors possibles. De plus, la fenêtre de prédiction est placée sous le contrôle de l'utilisateur qui peut la faire disparaître.

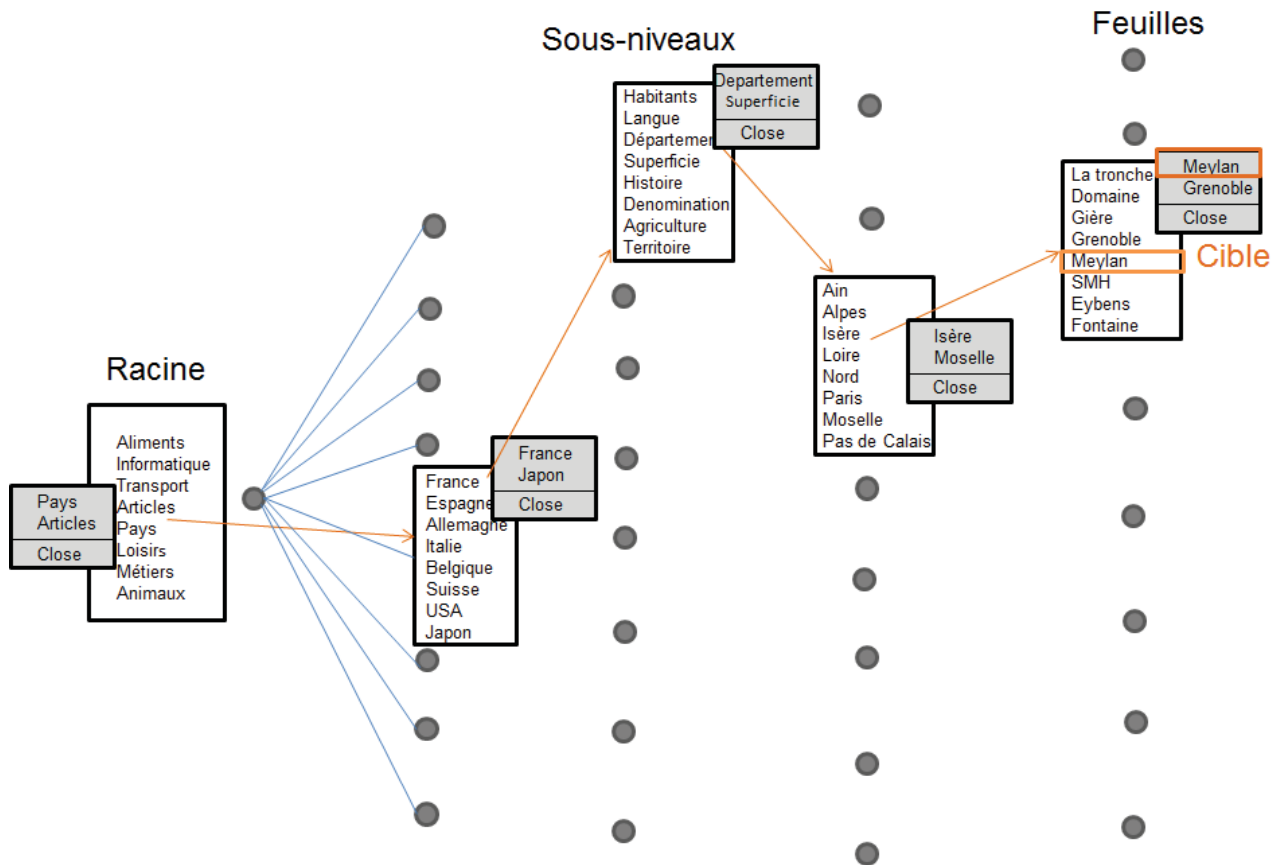


Figure 55. Vue conceptuelle du prototype step-by-step.



Figure 56. Vue utilisateur du prototype step-by-step.

### 1.2.2. Shortcut

L'objectif de la technique *shortcut* est d'accélérer l'interaction en proposant à la racine un raccourci menant directement à la cible finale située aux feuilles de l'arborescence, ce qui rend la navigation simple limitée à une seule opération. A l'inverse de *step-by-step*, *shortcut* devrait par contre freiner l'apprentissage de l'arborescence, l'utilisateur ne parcourant pas cette dernière. Cela devrait surtout être observé dans le cas d'une prédiction incorrecte. La technique *shortcut* est décrite dans les figures 57 et 58.

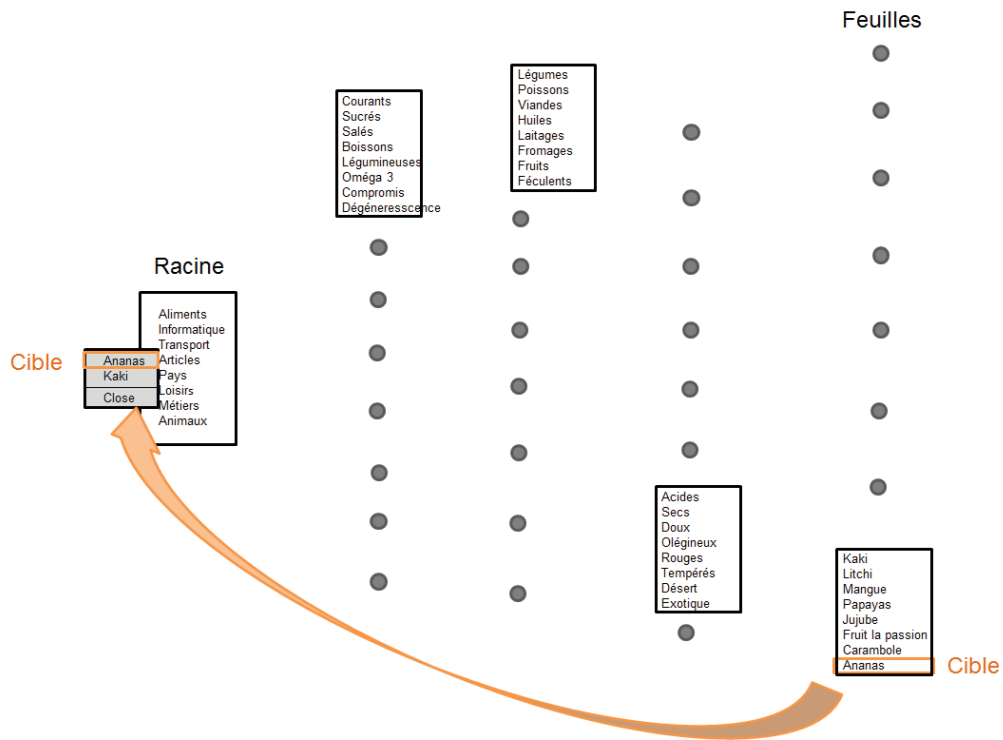


Figure 57. Vue conceptuelle du prototype shortcut.

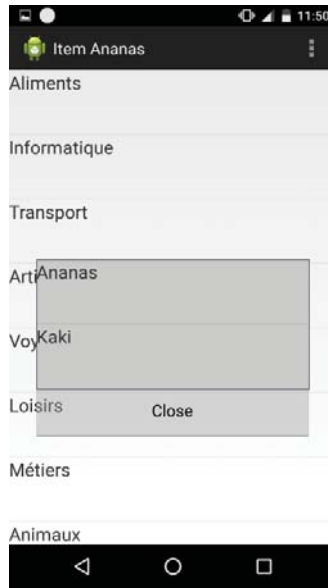


Figure 58. Vue utilisateur du prototype shortcut.

## 2. Evaluation expérimentale

Cette expérimentation vise à évaluer l'efficacité des techniques *step-by-step* et *shortcut* dans les cas de prédictions correctes et incorrectes. Pour l'évaluation, un troisième prototype est développé : il s'agit de la condition Contrôle.

La condition Contrôle permet la navigation au sein de l'arborescence sans proposer d'adaptation (figure 59 et 60).

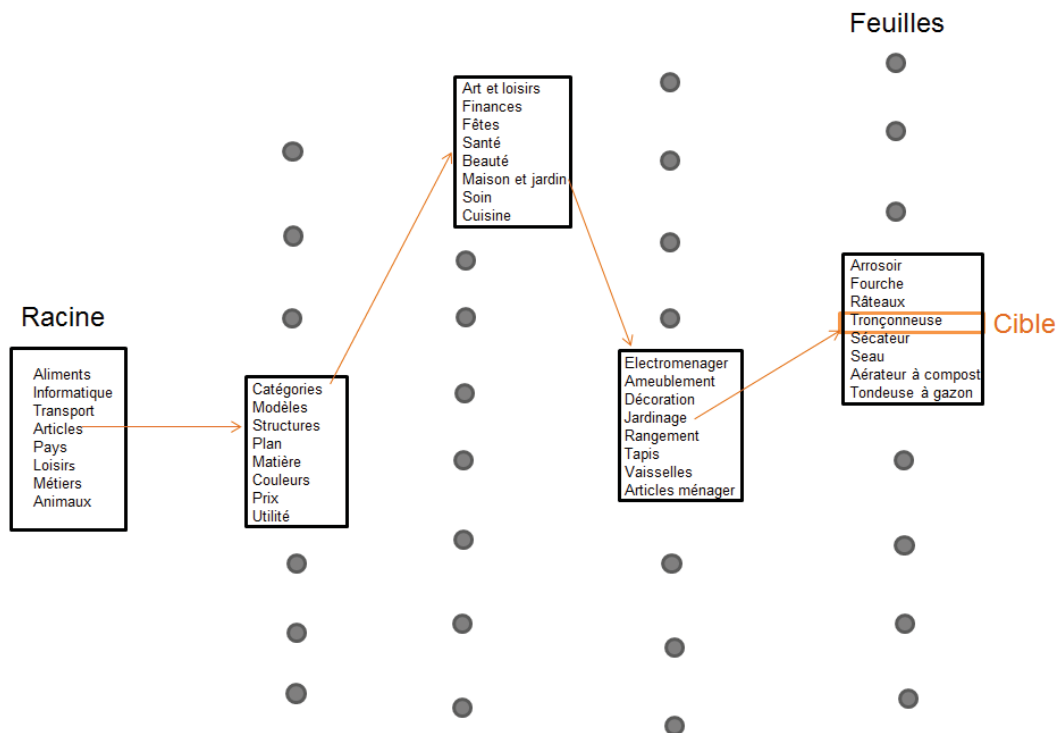


Figure 59. Vue conceptuelle du prototype Contrôle.

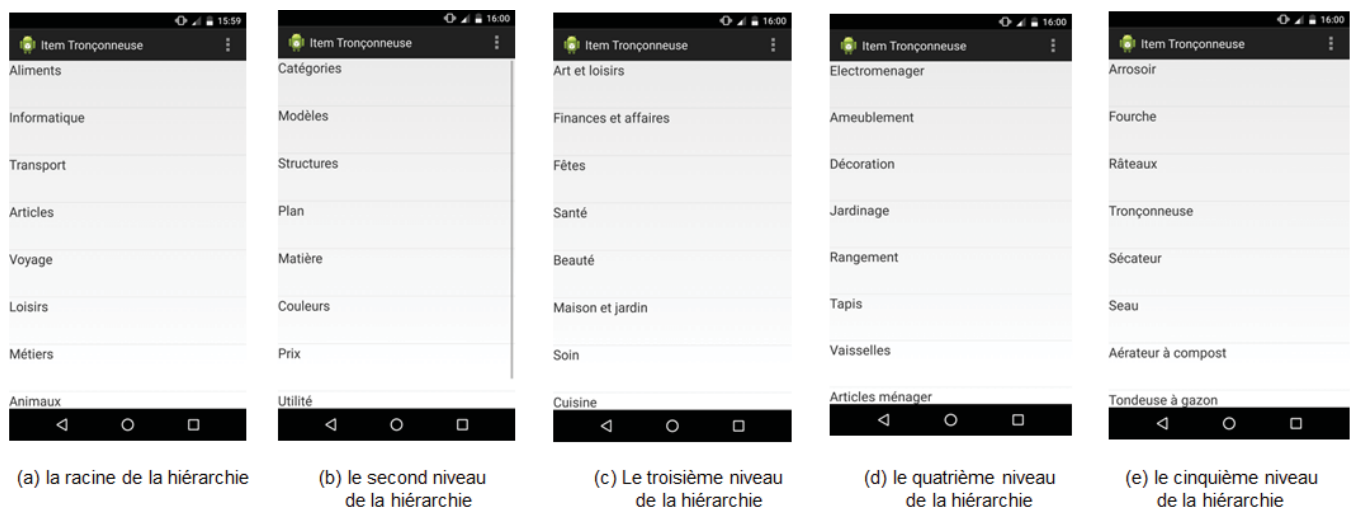


Figure 60. Vue utilisateur du prototype Contrôle.

## 2.1. Méthodologie

Nous avons manipulé deux variables indépendantes :

- La variable type d'interface (I) à trois modalités *step-by-step*, *shortcut* et Contrôle.



- La variable niveau de précision de la prédiction (P) à deux modalités (correcte et incorrecte).

## **2.2. Hypothèses**

Les hypothèses posées pour cette expérimentation sont :

### **2.2.1. Hypothèse 1 : vitesse d'exécution de la tâche**

- **Pour un haut niveau de précision de prédiction**

*La condition shortcut sera légèrement plus rapide que la condition step-by-step. Dans step-by-step, l'utilisateur devra sélectionner progressivement un item dans la fenêtre de prédiction jusqu'à atteindre sa cible qui est une feuille de l'arborescence. Ces actions répétitives ne devraient pas ralentir l'utilisateur outre mesure. De ce fait, les deux conditions devraient être exécutées dans un temps similaire.*

*Les conditions step-by-step et shortcut seront plus rapides que la condition Contrôle. Dans les conditions step-by-step et shortcut, les prédictions permettent d'atteindre la cible plus rapidement.*

- **Pour un faible niveau de précision de prédiction**

*La condition step-by-step sera proche de la condition Contrôle. Les deux conditions facilitant la navigation au sein de l'arborescence, dans le cas d'une prédiction incorrecte, elles devraient permettre d'obtenir des résultats similaires.*

*Les conditions Contrôle et step-by-step seront plus rapides que la condition shortcut. Dans la condition shortcut lorsque la prédiction est incorrecte, l'utilisateur devrait être confus et perdu car ce mécanisme ne lui permet pas d'apprendre la structure de la hiérarchie. Dans la condition step-by-step, l'utilisateur est guidé progressivement par le système et peut apprendre la structure. C'est également le cas dans la condition Contrôle où l'utilisateur est obligé de parcourir cette dernière.*

### **2.2.2. Hypothèse 2 : erreurs**

- **Pour un haut niveau de précision de prédiction**

*Les erreurs seront moins fréquentes dans la condition shortcut que dans la condition step-by-step.* Dans la condition *shortcut* lorsque la prédiction est correcte, la cible est présentée lors de la première étape. Cela devrait réduire la distance de navigation ainsi que les erreurs de sélection.

*Les erreurs seront moins fréquentes dans la condition step-by-step que dans la condition Contrôle.* Dans la condition *step-by-step*, la cible à sélectionner est mise en évidence à chaque niveau de la hiérarchie. Cela devrait réduire les erreurs de pointage.

#### - **Pour un faible niveau de précision de prédiction**

*Les erreurs seront moins fréquentes dans les conditions step-by-step et Contrôle que dans la condition shortcut.* Ces conditions permettant l'apprentissage de la structure. Elles devraient moins générer d'erreurs.

### **2.3. Mesures quantitatives et qualitatives**

Nous mesurons deux variables dépendantes :

La première est la vitesse de sélection (secondes). Elle est mesurée par le temps écoulé depuis l'ouverture du menu jusqu'à la sélection correcte de l'élément cible. Par exemple, si l'utilisateur doit traverser cinq niveaux d'arborescence pour atteindre la cible, le temps de sélection est calculé depuis l'ouverture du premier menu jusqu'à la sélection de la cible finale au niveau N-5.

La seconde variable est le taux d'erreurs. Elle est comptabilisée lorsque l'utilisateur sélectionne à tort une prédiction incorrecte ou parcourt une branche de l'arbre qui ne mène pas à la cible.

### **2.4. Tâche**

Les utilisateurs doivent sélectionner des cibles. Dans chaque condition (*Contrôle*, *step-by-step* et *shortcut*), un message apparaît à l'écran indiquant à l'utilisateur le nom de la cible. Ensuite, la liste des items apparaît.

Dans la condition *step-by-step*, l'item de la liste courante menant à la cible est présenté dans la fenêtre de prédiction à chaque niveau de la hiérarchie. Lorsque la prédiction est incorrecte,

l'utilisateur pourra fermer la fenêtre de prédiction et chercher l'item approprié dans la liste complète. Dans le niveau suivant de la hiérarchie, la fenêtre de prédiction n'est pas affichée et l'utilisateur doit naviguer de la même manière que dans les conditions *shortcut* ou Contrôle (Figure 61). L'objectif est de ne pas perturber l'utilisateur à chaque niveau de la hiérarchie dans le cas des prédictions incorrectes.

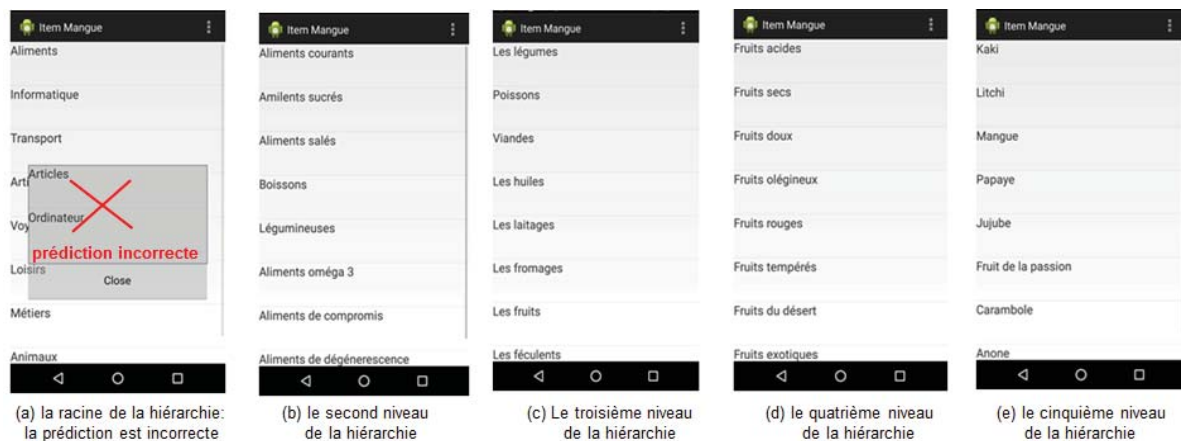


Figure 61. Step-by-step en cas de prédiction incorrecte.

Dans la condition *shortcut*, la cible est présentée dans la fenêtre de prédiction à la racine de la hiérarchie. Lorsqu'elle est correcte, l'utilisateur pourra ainsi la sélectionner directement, mais lorsqu'elle est incorrecte, il pourra fermer la fenêtre de prédiction et chercher cette dernière au sein de la hiérarchie. Elle sera alors toujours placée au dernier niveau (feuilles).

Dans la condition Contrôle, l'utilisateur doit naviguer dans la hiérarchie pour atteindre l'élément cible situé au niveau des feuilles (par exemple au 5<sup>ème</sup> niveau de la hiérarchie).

Dans toutes les conditions, le nom de l'item cible reste affiché en haut de l'écran comme rappel. Si la sélection effectuée ne correspond pas à la consigne, un message d'erreur s'affiche. Si la sélection effectuée correspond à la consigne, alors une nouvelle cible est proposée. A la fin du test, un message de remerciement est affiché.

## 2.5. Matériels

Les appareils utilisés dans cette expérimentation sont des *smartphones* Android. L'expérimentation est codée en Java pour Android. Les données expérimentales (le temps de

sélection en secondes, le temps de défilement en millisecondes et le taux d'erreur) sont enregistrées dans une base de données.

## 2.6. Participants

Treize volontaires masculins âgés de 22 à 58 ans, utilisateurs réguliers de *smartphones* ont participé à l'expérimentation.

## 2.7. Procédure

Chaque participant a passé un pré-test d'entraînement. Les listes hiérarchiques utilisées dans le pré-test sont différentes de celles utilisées dans les conditions du test. Les cibles sont différentes dans chaque condition. Le test entier est composé de 50 cibles réparties comme suit :

- Dans la condition *step-by-step*, 20 cibles sont présentées : 10 cibles avec prédiction correcte et 10 avec prédiction incorrecte. Dans les deux cas, chaque participant effectue cinq sélections pour arriver à la cible finale. Cette dernière est toujours au dernier niveau de la hiérarchie (feuilles).
- Dans la condition *shortcut*, 20 cibles sont présentées : 10 cibles avec prédiction et 10 avec prédiction incorrecte. Respectivement la cible est présentée à la racine de la hiérarchie vs les utilisateurs doivent descendre cinq niveaux hiérarchiques pour arriver à la cible.

Dans les deux conditions adaptatives (*step-by-step* et *shortcut*), chaque cible est demandée deux fois. La première fois dans le cas d'une prédiction correcte et la seconde lorsque la prédiction est incorrecte. Le but est de tester si l'utilisateur crée un modèle mental de la hiérarchie lorsque la prédiction est présentée étape par étape (*step-by-step*), ce qui pourrait faciliter l'achèvement de la tâche dans le cas d'une mauvaise prédiction, par rapport à la condition *shortcut*.

Dans la condition Contrôle, 10 cibles sont présentées. Les participants effectuent cinq sélections pour atteindre une cible qui est toujours au dernier niveau de la hiérarchie (feuille).

Le plan d'expérimentation est le suivant :  $S_{13} * I_3 * P_2$  (S pour Sujet ; I pour Interface ; P pour Précision). 78 conditions expérimentales sont donc créées. Chaque sujet aura à exécuter 50

tâches. En tout 3900 séquences de sélection (78\*50 cibles) sont présentées à l'ensemble des utilisateurs.

## 2.8. Résultats

Nous avons appliqué les tests de *Levene* et *Brown-Frosythe* pour tester l'homogénéité des variances. Cette dernière n'ayant pu être distinguée, l'analyse des données a été effectuée par les tests *Friedman's ANOVA by Ranks* et *Wilcoxon Signed Ranks*.

### *Vitesse*

Les résultats montrent que lorsque la prédiction est correcte, les conditions *step-by-step* (M = 6.42, SD = 2.81) et *shortcut* (M = 1.77, SD = 1.47) sont significativement plus rapides ( $W(24) = -3.17, p < 0.002$ ) que la condition Contrôle (M = 14.73, SD = 4.11).

D'ailleurs la condition *shortcut* est-elle même plus rapide ( $W(24) = -3.18, p < 0.002$ ) que la condition *step-by-step*.

C'est l'inverse, en cas de prédiction incorrecte où les conditions *step-by-step* (M = 14.51, SD = 3.67) et Contrôle sont significativement plus rapide (respectivement  $W(24) = 2.13, p < 0.05$  ;  $W(24) = 2.34, p < 0.05$ ) que la condition *shortcut* (M = 20.94, SD = 8.06). Pour autant les conditions *step-by-step* et Contrôle ne présentent pas de différence significative ( $W(24) = -0.14, p > 0.5$ ).

Comme attendu, dans les deux conditions adaptatives (*step-by-step* et *shortcut*), les participants sont significativement plus rapides (respectivement  $W(24) = 3.18, p < 0.002$  ;  $W(24) = 3.19, p < 0.002$ ) lorsque la prédiction est correcte que lorsqu'elle est incorrecte.

Globalement, la condition *step-by-step* (M = 10.48, SD = 5.24) est plus rapide que la condition Contrôle (M = 14.73, SD = 4.11) qui est elle-même plus rapide que la condition *shortcut* (M = 17.1, SD = 32.39).

Les résultats du temps de sélection sont représentés graphiquement sur la figure 62 et dans le tableau I en annexe.

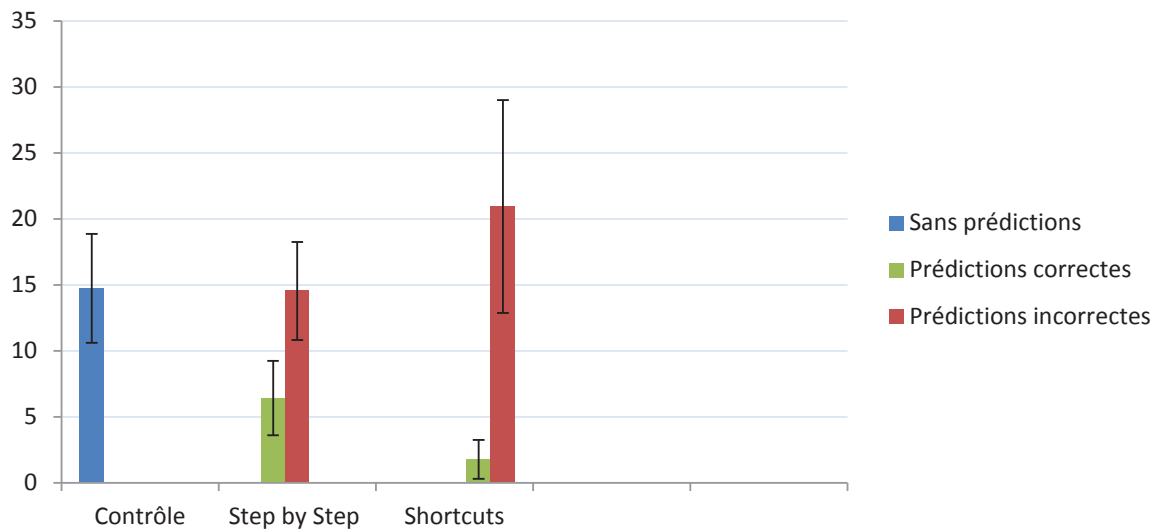


Figure 62. Temps de sélection (secondes) pour toutes les conditions.

### Erreurs

*Step-by-step* et *shortcut* présentent significativement moins d'erreurs ( $W(24) = 3.19, p < 0.005$ ) en cas de prédiction correcte (respectivement  $M = 0.23, SD = 0.6$  ;  $M = 0, SD = 0$ ) qu'en cas de prédiction incorrecte (respectivement  $M = 4.46, SD = 2.93$  ;  $M = 13.46, SD = 15.99$ ).

Lorsque la prédiction est correcte, aucune différence significative ( $W(24) = 1.34, p > 0.1$ ) n'est mise en évidence entre *step-by-step* ( $M = 0.23, SD = 0.6$ ) et *shortcut* ( $M = 0, SD = 0$ ).

Néanmoins, dans ces deux conditions adaptatives, les erreurs sont systématiquement moins fréquentes (respectivement  $W(24) = -2.83, p < 0.005$  ;  $W(24) = -2.94, p < 0.005$ ) que dans la condition Contrôle ( $M = 4.31, SD = 4.37$ ).

Lorsque la prédiction est incorrecte, les erreurs dans les conditions *step-by-step* ( $M = 4.46, SD = 2.93$ ) et Contrôle ( $M = 4.31, SD = 4.37$ ) sont du même ordre ( $W(24) = 0.07, p > 0.5$ ) et sont significativement moins fréquentes (respectivement  $W(24) = -2.34, p < 0.5$  ;  $W(24) = 1.85, p < 0.5$ ) que dans la condition *shortcut* ( $M = 13.46, SD = 15.99$ ).

Les résultats de taux d'erreurs sont représentés graphiquement dans la figure 63 et dans le tableau II en annexe.

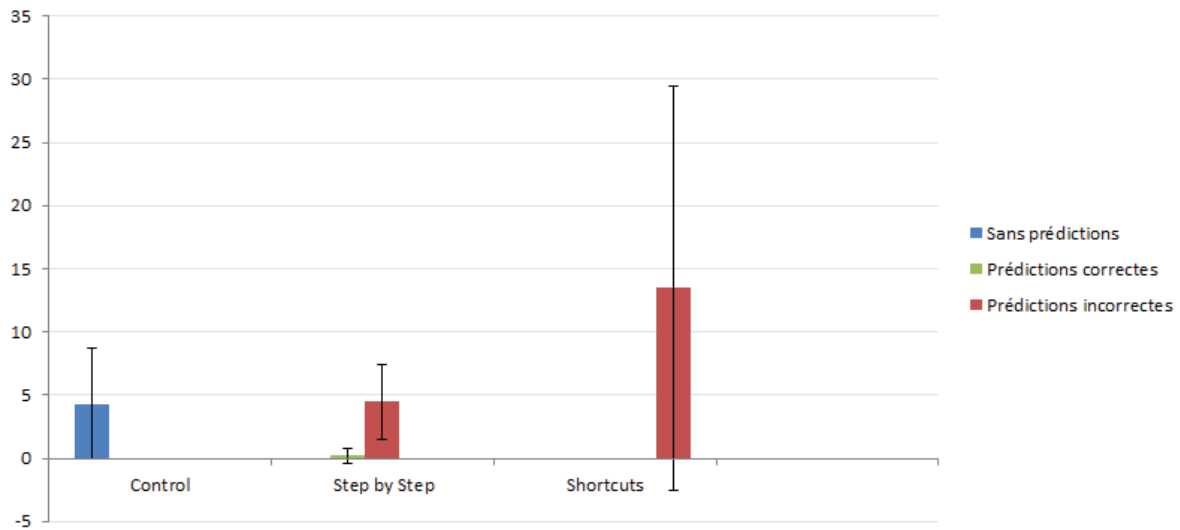


Figure 63. T d'erreur pour toutes les conditions.

## 2.9. Discussion

### 2.9.1. Hypothèse 1 : vitesse d'exécution de la tâche

#### - Pour un haut niveau de précision de prédiction

*Aucune différence significative entre les conditions shortcut et step-by-step. **Infirmée***

Lorsque la prédiction est correcte, les participants sont plus rapides dans la condition *shortcut* car la cible est présentée à la première étape d'interaction. Dans la condition *step-by-step*, les participants sont guidés jusqu'à atteindre la cible. Les résultats montrent que cela prend plus de temps même si, comme nous le prévoyions, de telles actions répétitives auraient dû accélérer l'interaction. Cela suggère qu'aucun effet d'apprentissage n'est observé.

*La condition step-by-step sera plus rapide que la condition Contrôle. **Confirmée***

Le guidage est bénéfique car il permet de réduire le temps de recherche sans pour autant être aussi efficace que la condition *shortcut*.

#### - Pour un faible niveau de précision de la prédiction

*La condition step by Step sera proche de la condition Contrôle et toutes les deux seront plus rapides que shortcut. **Confirmée***

Le guidage des participants dans la condition *step-by-step* leur permet d'apprendre la structure de la hiérarchie. Par conséquent, lorsque la prédiction est incorrecte, les participants sont moins perdus qu'avec *shortcut* qui ne permet pas un tel apprentissage.

En terme d'efficacité, *step-by-step* semble être un bon compromis car elle est plus rapide que Contrôle en cas de prédiction correcte mais n'est pas pire que cette dernière en cas de prédiction incorrecte. A l'inverse, *shortcut* pénalise considérablement l'interaction.

### **2.9.2. Hypothèse 2 : erreurs**

#### **- Pour un haut niveau de précision de prédiction**

*Les erreurs seront moins fréquentes dans la condition shortcut que dans les conditions step by step et Contrôle. Partiellement confirmée*

Il n'y a pas de différence significative entre les conditions *shortcut* et *step-by-step* lorsque la prédiction est correcte, car la cible est identifiée facilement par les utilisateurs. Néanmoins, les erreurs sont enregistrées moins fréquemment que dans la condition Contrôle.

*Les erreurs seront moins fréquentes dans la condition Step by Step que dans la condition Contrôle. Confirmée*

Le guidage réduit le nombre des erreurs.

#### **- Pour un faible niveau de précision de prédiction (P-)**

*Les erreurs seront moins fréquentes dans les conditions step-by-step et Contrôle que dans la condition shortcut. Confirmée*

L'exploration de la structure de l'arbre (*step-by-step* et Contrôle) facilite l'apprentissage et permet de réduire les erreurs. A l'inverse, dans la condition *shortcut*, les participants sont tout de suite perdus lorsqu'ils doivent rechercher eux même une cible inconnue.

## **3. Conclusion**

Dans ce chapitre nous avons présenté deux techniques d'adaptation *step-by-step* et *shortcut* appliquées à des menus hiérarchiques. Elles préservent la présentation du menu initial, et superposent une couche adaptative affichant les éléments les plus fréquents. *step-by-step*



permet aux utilisateurs d'apprendre la structure de la hiérarchie en les guidant à chaque étape. *shortcut* permet de réduire le temps de navigation.

Globalement, la condition *step-by-step* représente un bon compromis en termes de vitesse de sélection et d'erreurs aussi bien en cas de prédiction correcte qu'incorrecte.

Cependant, *shortcut* reste une technique très efficace en cas de prédictions fiables. Par conséquent, elle pourrait être recommandée dans le cas d'un algorithme performant (par exemple, supérieur à 80%) ou quand l'utilisateur est habitué au système et a déjà la connaissance de l'arborescence, voire quand cette dernière est peu profonde. En effet, une arborescence moins fournie pourrait mener à des résultats différents car elle serait plus facile à apprendre. De même, les autres caractéristiques facilitant l'apprentissage pourraient influencer sur le choix de la méthode. Nous savons que la multimodalité est un facteur important. Présenter de façon redondante les informations sur les différents canaux perceptifs facilite l'acquisition. La littérature précise également que l'apprentissage par l'action est le plus efficace qui soit. De fait les modalités de commande utilisées (haptique, commande vocale) peuvent également impacter le choix de la technique de rendu adaptatif. Dans le chapitre suivant, nous étudions les variations de modalité de rendu et de commande utilisées lors d'une prédiction.

## 4. Synthèse

Dans ce chapitre nous avons traité la navigation au sein de listes imbriquées hiérarchiquement sur *smartphone*. Nous avons proposé deux méthodes d'adaptation :

*Step-by-step* permet d'accompagner l'utilisateur dans sa navigation en lui proposant des **prédictions de manière progressive** en tout point d'arborescence, de la racine à la cible.

*Shortcut* permet d'**accélérer** la navigation en faisant remonter à la **racine la cible** visée.

L'expérience montre que l'approche *step-by-step* est un bon compromis en termes de rapidité de navigation et d'apprentissage de la structure de la hiérarchie.



## Chapitre 9 : Menu polymodal, axe Modalité

Ce chapitre a fait l'objet d'une publication en conférence internationale :

**Bouzit, S., Calvary, G, Chêne, D., Vanderdonckt, J. (2017).** Polymodal Menus: A Model-based Approach for Designing Multimodal Adaptive Menus for Small Screens. Proceedings of the 9<sup>th</sup> ACM SIGCHI Symposium on Engineering Interactive Computing Systems. EICS'17.

Les rendus des items prédits et des items du menu sont souvent effectués simultanément dans la même modalité graphique. Cela peut saturer le canal visuel et réduire l'efficacité de la technique adaptative.

En neurologie, la polymodalité (Belmonte *et al.*, 1994) fait référence à la capacité d'un récepteur unique de répondre à des stimuli provenant de multiples modalités, telles que la lumière, le son, la température, le goût, la pression et l'odeur. Un récepteur polymodal désigne un récepteur qui est sensible à plus d'une modalité sensorielle ou sous-modalité (Merriam-Webster dictionary-<http://www.merriam-webster.com/medical/polymodal>).

Par exemple, nos nocicepteurs humains envoient les signaux de douleur à notre cerveau car ils sont sensibles à la température, pression, air, conditions chimiques. Par analogie à la neurologie, dans ce chapitre nous définissons le menu polymodal. L'item du menu peut être sélectionné par pointage, vocalement, gestuellement ou par toute combinaison des trois modalités. Cela peut être exprimé sur l'axe Modalité du modèle PCA (figure 64).

Dans ce chapitre, nous étudions deux modalités de rendus (graphique et vocale), combinées à trois modalités de commande (pointage, gestuelle et vocale), en cas de prédictions correctes et incorrectes.

Nous présentons tout d'abord les travaux liés aux menus multimodaux menant à la conception et l'implémentation du menu polymodal. Nous présentons une étude expérimentale comparant quatre conditions (gestuelle, pointage, vocale et mixte). Enfin, nous discutons les résultats obtenus et concluons ce chapitre par des propositions de recherche.

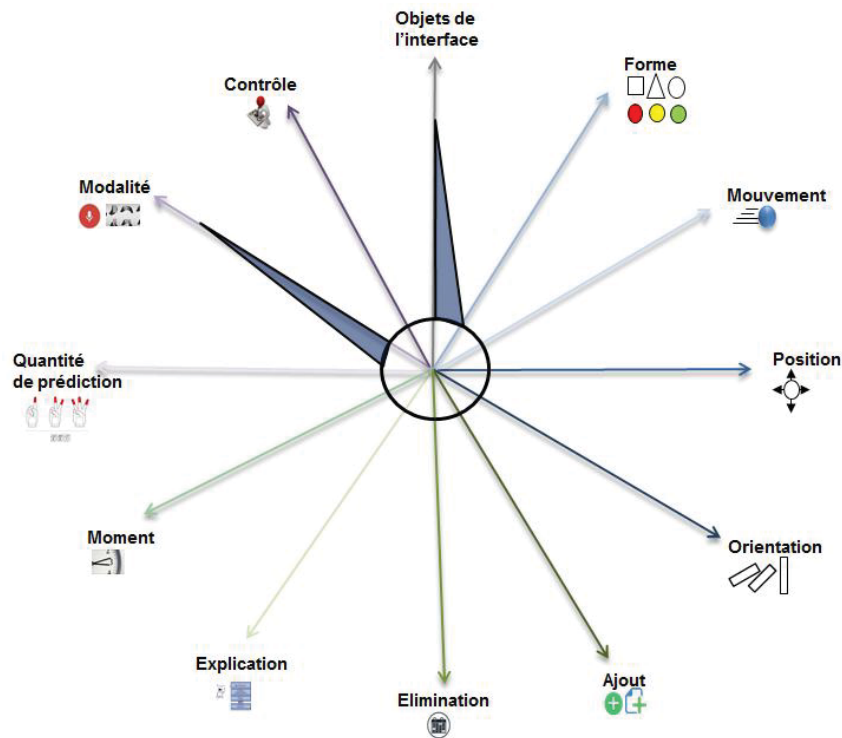


Figure 64. Axes explorés dans le menu Polymodal.

## 1. Techniques étudiées

### 1.1. Travaux connexes sur les menus multimodaux

Pielot *et al.* (2012), ont proposé le PocketMenu qui permet de changer la modalité de sélection. Les items du menu sont disposés le long de la frontière du *smartphone* dans la zone de confort de la main. Un retour audio-tactile facilite l'identification des éléments non-visuels. Cette technique d'interaction est utile pour les utilisateurs visuellement contraints.

Des interfaces graphiques améliorées par un son ont été explorées par Ma *et al.* (2003), Cockburn et Brewster. Plusieurs widgets tels que les menus, boutons, barres de défilement, barres de progression ont été augmentés par un son non verbalisé, nommé *earcons*. Ce dernier est un motif sonore utilisé pour représenter un élément ou un événement spécifique. Les résultats montrent que le menu augmenté d'un son permet une réduction du taux d'erreurs et réduit le temps d'exécution. Cela améliore l'utilisabilité globale du menu.

En 2005, Cockburn et Brewster ont trouvé que l'ajout de feedback audio réduit le temps de sélection à 4.2%. Même si Akamatsu *et al.*, 1995 ne confirment pas cela, ils soulignent néanmoins que le temps passé sur les cibles inappropriées est réduit.

Oakley *et al.* (2001), ont proposé une interface ajoutant un *feedback* tactile. Cela facilite, selon les auteurs, l'interaction.

Zhao *et al.* (2007) ont également montré l'intérêt de la multimodalité sur les petits dispositifs et plus particulièrement dans les conditions où l'utilisateur ne peut compter sur la modalité visuelle pour sélectionner un item.

Les travaux antérieurs sur la multimodalité ont montré que cette dernière représente une piste intéressante pour améliorer les performances. Nous proposons le menu polymodal afin de mettre en évidence les meilleures modalités de rendu et de commande dans le cas des prédictions correctes et incorrectes.

## 1.2. Conception du menu polymodal

La partie statique du menu contenant l'ensemble des items conserve les mêmes modalités de rendu et de commande durant toutes les étapes d'interaction. La partie adaptative contenant les items prédits change de modalités de rendus et de commandes.

Dans ce chapitre, nous ne cherchons pas à évaluer la qualité du rendu graphique, comme cela a pu être fait sur les menus *Split* (Sears et Shneiderman, 1994), *Highlighting* (Tsandilas et Schraefel, 2005) et *Morphing* (Cockburn *et al.*, 2007). Nous évaluons les modalités d'interaction utilisées pour **présenter** et **sélectionner** les items prédits.

Les propriétés CARE pour l'interaction multimodale (Coutaz *et al.*, 1995) sont un bon point de départ pour décortiquer les multiples combinaisons modales possibles.

La propriété d'assignation souligne les situations dans lesquelles le choix de la modalité n'est pas possible : il est fixé par le système.

La propriété d'équivalence souligne les situations où l'utilisateur a le choix entre plusieurs modalités.

La propriété de redondance correspond à l'utilisation conjointe de plusieurs modalités alors qu'une seule suffirait.

Enfin, la propriété de complémentarité dénote la nécessité de combiner plusieurs modalités pour réaliser la tâche.

Les propriétés CARE s'appliquent aux modalités de rendu et de commande. Nous explorons dans ce chapitre les combinaisons les plus fréquemment utilisées en interaction homme-machine en vue d'une transposition à l'adaptation. Nous distinguons quatre agencements de commandes et de rendus multimodaux.

- Le menu **Graphique** où les items prédits sont présentés visuellement dans une fenêtre de prédiction et sont sélectionnés par pointage. La fenêtre de prédiction peut être fermée en appuyant sur le bouton « Close ». La modalité d'entrée (pointage) est en assignation, tout comme la modalité de sortie (visuelle), ces deux modalités étant de natures différentes (visio-haptique déictique / visuelle).
- Le menu **Gestuel** où les items prédits sont présentés visuellement dans une fenêtre de prédiction et peuvent être sélectionnés par un geste tracé sur l'écran tactile en l'occurrence « 1 », « 2 », « 3 » pour sélectionner respectivement le premier, deuxième ou troisième item. La fenêtre de prédiction peut être fermée par un geste «  $\alpha$  ». La modalité d'entrée (geste) est en assignation, tout comme la modalité de sortie (visuelle), ces deux modalités étant de natures différentes (visio-haptique symbolique / visuelle).
- Le menu **Vocal** où les items prédits sont vocalisés par le système et peuvent être sélectionnés par reconnaissance vocale. Un son signale le déclenchement du microphone et le système est en attente de la répétition par l'utilisateur d'un des items prédits. La modalité d'entrée (vocale) est en assignation, tout comme la modalité de sortie (audio), ces deux modalités étant de même nature (audio).
- Le menu **Mixte** où les items prédits sont présentés à la fois visuellement et vocalement et sont sélectionnables par pointage, geste ou reconnaissance vocale. La fenêtre de prédiction peut être fermée par pointage de « Close », geste «  $\alpha$  » ou commande vocale « Escape ». Les modalités d'entrée (pointage, geste, vocale) sont en équivalence, les modalités de sortie (visuelle et vocale) sont en redondance. Toutes ces modalités sont de natures différentes (visio-haptique / visuelle / audio).

Le menu polymodal est décrit dans la figure 65.

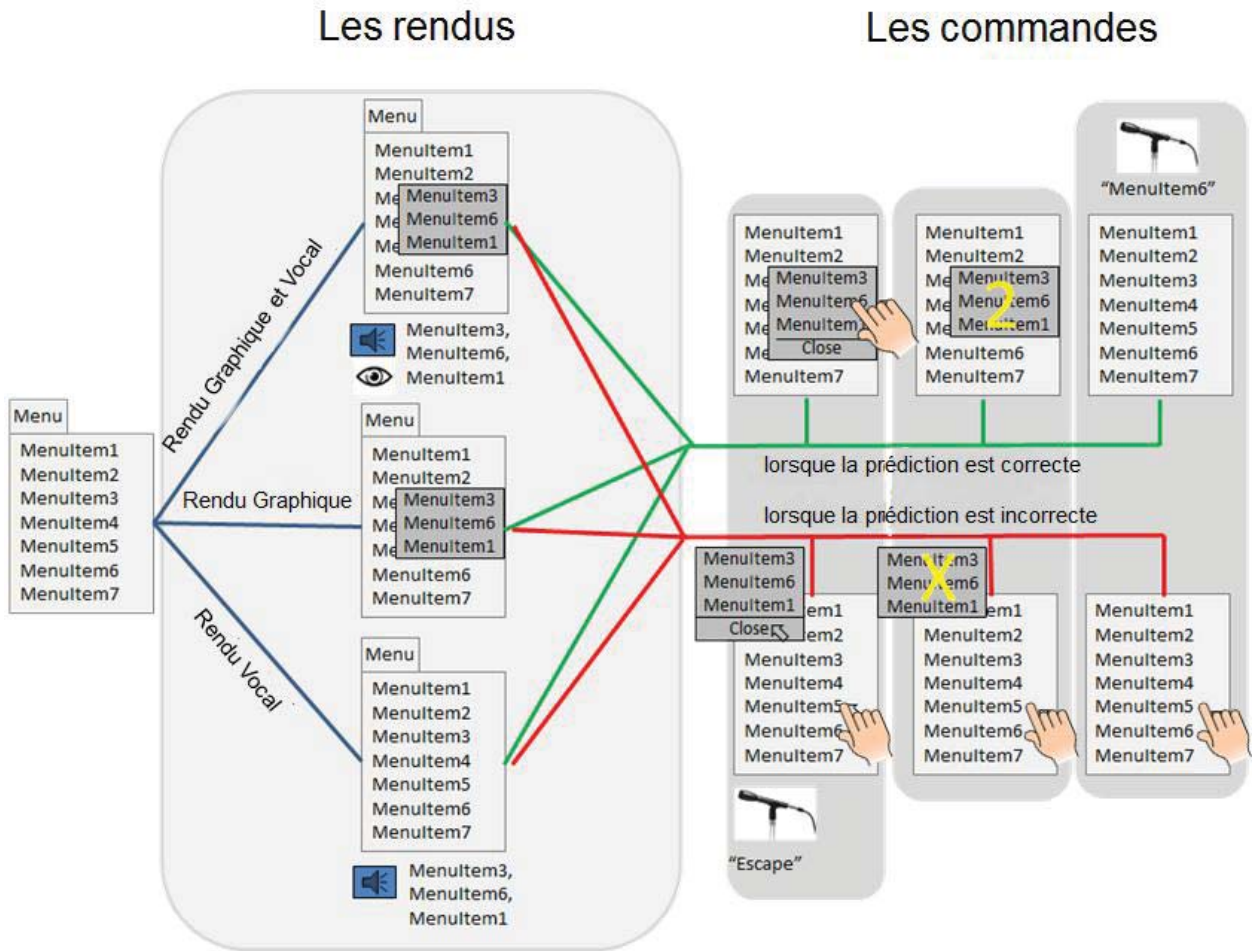


Figure 65. Répartition des modalités de rendus (œil et haut-parleur) et commande (pointeur, geste, micro).

Le menu polymodal est développé en Java pour Android sous Eclipse en utilisant Android Software Development Kit. La reconnaissance vocale est assurée par Android Speech Recognizer. La synthèse vocale est implémentée avec le module Android Text2Speech. La reconnaissance gestuelle est implémentée sur Android Gesture Library. Android Gesture Builder a permis de définir les gestes initiaux qui ont été ajoutés à Android Recognizer.

## 2. Evaluation expérimentale

Nous avons mis en place une expérimentation contrôlée faisant varier deux rendus de prédiction et trois modalités de commande.



## 2.1. Méthodologie

Le premier facteur type de menu (I4) est une variable intra-sujet qui se décline en menu Graphique, Gestuel, Vocal et Mixte.

Le deuxième facteur indépendant est le niveau de précision de prédiction (P2). Lorsque la prédiction est correcte, la cible à sélectionner est présente dans la fenêtre de prédiction des conditions graphique, gestuelle et mixte, et vocalisée par le système dans les conditions vocale et mixte. Dans le cas contraire, la prédiction est incorrecte : la cible n'est donc pas dans la fenêtre prédiction et n'est pas vocalisée par le système.

## 2.2. Hypothèses

Les hypothèses sont les suivantes :

### 2.2.1. Hypothèse 1 : vitesse d'exécution de la tâche

#### - Pour un haut niveau de précision de prédiction

*Aucune différence significative entre les conditions graphique et gestuelle.* Les trois gestes dans la condition Gestuelle sont faciles à produire et à mémoriser mais sont plus longs à exécuter. Ils devraient être aussi efficaces qu'un pointage dans la condition graphique qui nécessite une forte précision, mais plus rapide à exécuter.

*Les conditions Graphique et Gestuelle seront plus rapides que la condition Vocale.* Le temps de présentation vocale étant séquentielle, elle prend plus de temps qu'un affichage global. La prononciation de la commande risque aussi d'être plus longue qu'un simple pointage ou geste.

*La condition Mixte sera plus rapide que les trois autres conditions (Graphique, Gestuelle, Vocale).* Dans la condition Mixte, l'utilisateur est libre d'utiliser n'importe quelle modalité pour valider la prédiction. Cela optimise l'interaction et ses performances.

#### - Pour un faible niveau de précision de prédiction

*La condition Vocale sera plus rapide que les conditions, Graphique, Gestuelle et Mixte.* Lorsque la prédiction est incorrecte dans la condition vocale, l'interface n'est pas surchargée par des informations visuelles inutiles, contrairement aux conditions Graphique, Gestuelle et Mixte où il faut fermer la fenêtre de prédiction.

*Aucune différence significative entre la condition Graphique et la condition Gestuelle.* Dans les deux conditions, lorsque la prédiction est incorrecte, l'utilisateur doit fermer la fenêtre de prédiction et chercher l'item cible dans la liste complète des items. L'utilisateur ferme la fenêtre de prédiction en visant avec le bouton « Close » dans la condition graphique ou en traçant «  $\alpha$  » dans la condition gestuelle. Cette action de traçage est simple à produire mais plus longue qu'un pointage. Elle devrait être équivalente à la sélection du bouton «Close» qui nécessite précision mais qui est plus rapide.

*La condition Mixte sera plus rapide que les trois autres conditions (Graphique, Gestuelle, Vocale).* Dans la condition Mixte, l'utilisateur a la possibilité de choisir entre différentes modalités (pointage, geste, vocal) afin de fermer la fenêtre de prédiction.

### **2.2.2. Hypothèse 2 : erreurs**

Globalement, la condition vocale devrait générer moins d'erreurs.

#### **- Pour un haut niveau de précision de prédiction**

*Les erreurs seront moins fréquentes dans la condition Vocal que dans les conditions Graphique, Gestuelle et Mixte.* Nous supposons que la désignation vocale est plus robuste que les méthodes de commande manuelles.

*Les erreurs seront moins fréquentes dans la condition Gestuelle que dans la condition Graphique.* Dans la condition Gestuelle, la contrainte de précision est moindre, le geste est donc plus facile à produire par rapport au pointage.

#### **- Pour un faible niveau de précision de prédiction**

*Les erreurs seront moins fréquentes dans la condition Vocale que dans les conditions Graphique, Gestuelle et Mixte.* Dans la condition Vocale en absence d'affichage de prédictions, l'utilisateur se trouve directement sur la liste complète des items, ce qui réduit les erreurs.

## **2.3. Mesures quantitatives et qualitatives**

Nous avons mesuré deux variables dépendantes :

- Le temps de sélection mesure en secondes le délai entre l'ouverture du menu et la sélection de l'item cible.

- Le taux d'erreurs cumule les sélections erronées.

## **2.4. Tâche**

Nous avons demandé aux utilisateurs d'effectuer une séquence de sélections d'items. Pour chaque condition (Graphique, Gestuelle, Vocale et Mixte), l'utilisateur doit d'abord sélectionner le bouton « Commencer ». Ensuite, un message apparaît à l'écran indiquant le nom de l'item à sélectionner, puis les listes contenant l'item. Le nom de l'item cible reste affiché en haut de l'écran comme rappel. Dans le cas où l'utilisateur fait une erreur de sélection, un message d'erreur l'invite à sélectionner la bonne cible. Lorsque l'utilisateur sélectionne l'item demandé, un nouveau message indique la nouvelle cible.

## **2.5. Matériels**

Les appareils utilisés dans l'expérimentation sont des *smartphones* de type Samsung Galaxy avec un système d'exploitation Android. Le système crée une base de données dans laquelle le temps de sélection (en secondes), le temps de défilement (en millisecondes) et le nombre d'erreurs sont enregistrés.

## **2.6. Participants**

Treize volontaires ont participé à l'expérimentation. Ils étaient âgés entre 21 et 58 ans ( $\mu = 29.38$ ,  $SD = 11$ ). Les participants sont tous des utilisateurs réguliers de *smartphones*.

## **2.7. Procédure**

Chaque participant a effectué la tâche expérimentale dans un environnement contrôlé. Avant de commencer le test, nous avons expliqué aux participants l'utilisation du menu polymodal. Ensuite, une session de pré-test commence permettant de s'entraîner. La liste des items utilisée dans le pré-test n'est pas la même que celle utilisée dans le test.

Quatre menus de 16 items ont été définis pour le test. Les 16 items sont distribués sur deux écrans, chaque écran contenant 8 items. Les menus utilisés dans les quatre conditions ainsi que les séquences de sélection des cibles sont générés aléatoirement.

Dans chaque condition, nous avons demandé aux utilisateurs de sélectionner 20 items : 10 items lorsque la prédiction est correcte, 10 items lorsque la prédiction est incorrecte.

En résumé, la session du test est conçue comme suit :

13 participants \* 20 cibles \* 4 conditions = 1040 sélections d'items au total.

## 2.8. Résultats

L'homogénéité de variance n'a pas pu être distinguée (*Levene et Brown-Forsythe*). De ce fait, les tests non-paramétriques de *Friedman's ANOVA by Ranks* et *Wilcoxon Signed Ranks* ont été appliqués.

### *Vitesse*

Globalement, lorsque la prédiction est correcte, aucune différence significative  $F(3) = 11.07, p < 0.001$  n'a été mise en évidence entre les quatre conditions.

Cependant, les comparaisons par paires montrent que la condition Graphique (M = 1.461, SD = 0.366) est significativement plus rapide que les trois autres conditions (Gestuelle : M = 2.969, SD = 0.9,  $W(24) = 3.18, p < 0.005$ ), (Vocale : M = 4.438, SD = 0.876,  $W(24) = 3.18, p < 0.005$ ), (Mixte : M = 4.184, SD = 1.154,  $W(24) = 3.18, p < 0.005$ ).

De plus, les utilisateurs sont plus rapides dans la condition Gestuelle que dans les conditions Vocale et Mixte (respectivement  $W(24) = 3.04, p < 0.005$  ;  $W(24) = 2.48, p < 0.05$ ).

Aucune différence significative n'a été trouvée entre les conditions Vocale et Mixte  $W(24) = -0.63, p > 0.5$ ).

Cependant lorsque la prédiction est incorrecte, la condition Vocale (M = 3.115, SD = 0.883) est plus rapide que toutes les autres conditions (Graphique : M = 4.1, SD = 0.886,  $W(24) = -2.76, p < 0.01$ ), (Gestuelle : M = 6.484, SD = 1.643,  $W(24) = -3.18, p < 0.005$ ), (Mixte : M = 6.577, SD = 1.978,  $W(24) = 3.06, p < 0.005$ ).

La condition Graphique est meilleure que les conditions Gestuelle et Mixte (respectivement  $W(24) = 3.18, p < 0.005$  ;  $W(24) = 3.06, p < 0.005$ ).

Au sein d'une même condition, les utilisateurs étaient plus rapides lorsque la prédiction est correcte que lorsqu'elle est incorrecte (sauf pour la condition Vocale)  $F(3) = 11.07, p < 0.001$ , Vocale avec prédiction correcte: M = 4.438, SD = 0.876, Vocale avec prédiction incorrecte: M = 3.115, SD = 0.883,  $W(24) = 2.97, p < 0.005$ .

Les temps de sélection sont illustrés dans la figure 66 et dans le tableau III en annexe.

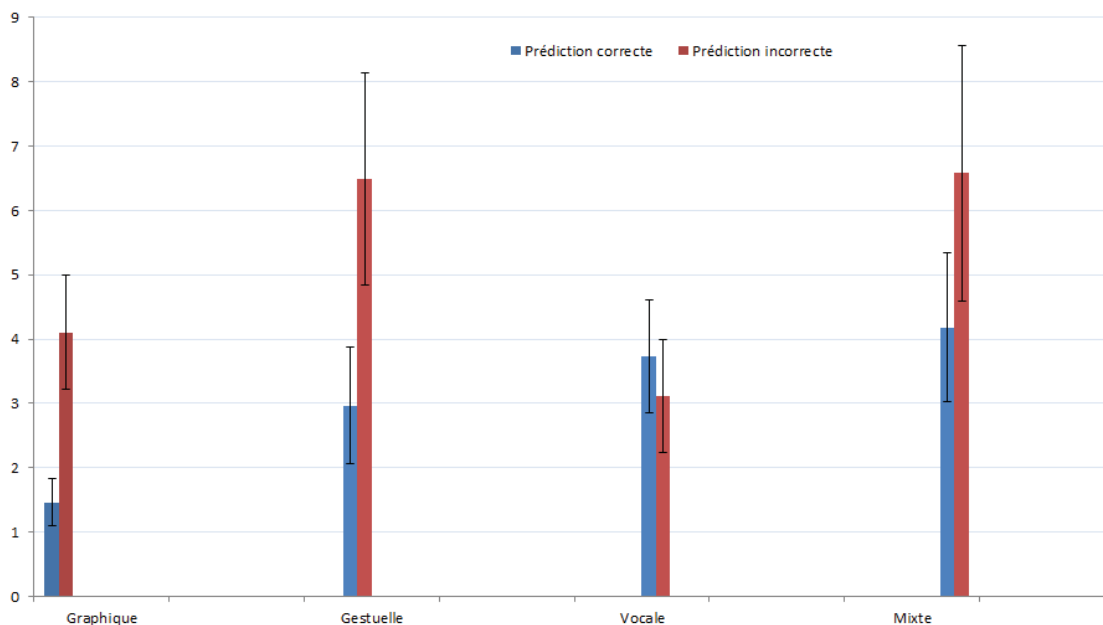


Figure 66. Temps de sélection pour les conditions Graphique, Gestuelle, Vocale et Mixte.

### *Erreurs*

Lorsque la prédiction est correcte, nous ne mettons pas en évidence de différence significative entre les conditions Gestuelle ( $M = 0$ ,  $SD = 0$ ), Vocale ( $M = 0.08$ ,  $SD = 0.29$ ), Graphique ( $M = 0.17$ ,  $SD = 0.39$ ) et Mixte ( $M = 0.33$ ,  $SD = 0.65$ ).

Cependant, il y a significativement plus d'erreurs ( $F(3) = 9.78$ ,  $p < 0.005$ ) lorsque la prédiction est incorrecte que lorsqu'elle est correcte. De plus, la majorité des erreurs ( $F(3) = 15.57$ ,  $p < 0.005$ ) sont produites dans la condition Gestuelle ( $M = 2.08$ ,  $SD = 1.73$ ) alors que la condition Vocale ( $M = 0$ ,  $SD = 0$ ) n'en produit pas.

En résumé, lorsque la prédiction est correcte, les quatre conditions produisent autant d'erreurs, alors que lorsqu'elle est incorrecte ces erreurs sont surtout produites en conditions Gestuelle et Mixte.

Les taux d'erreurs sont illustrés dans la figure 67 et dans le tableau IV en annexe.

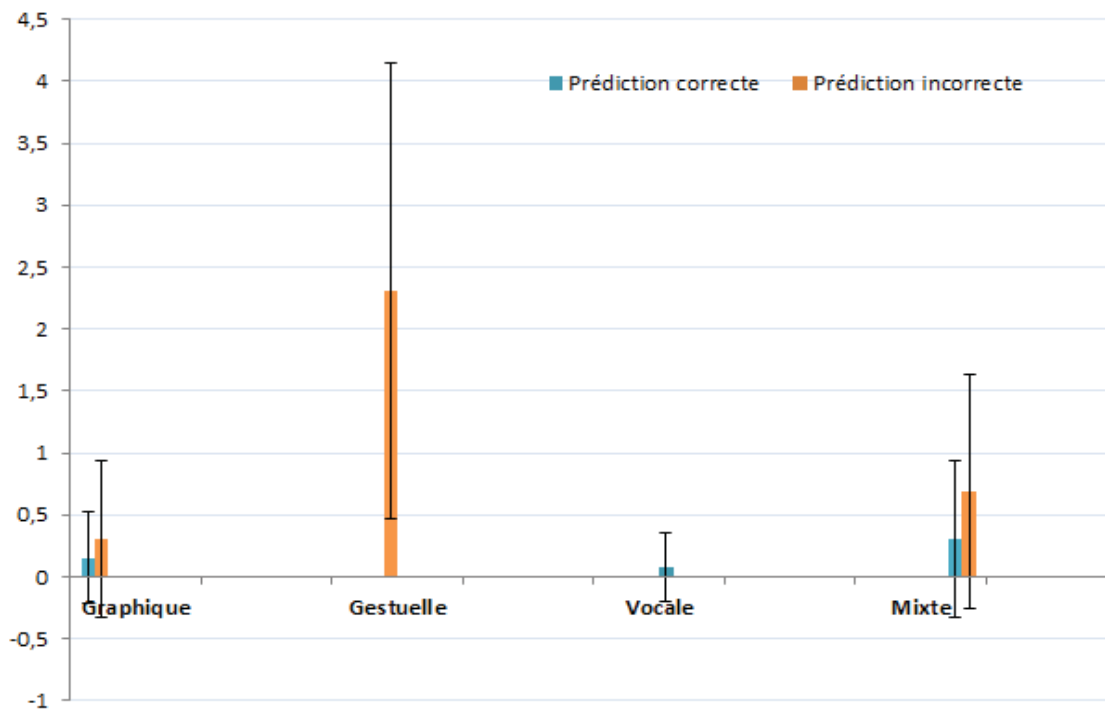


Figure 67. Taux d'erreur pour les conditions Graphique, Gestuelle, Vocale et Mixte.

### *Préférences utilisateurs*

Pour compléter, nous avons interrogé les utilisateurs sur leurs préférences (questionnaire en annexe). L'analyse montre une différence significative entre les quatre conditions  $F(3) = 8.45$ ,  $p < 0.05$ .

Les utilisateurs trouvent que la condition Graphique ( $M = 6.18$ ,  $SD = 0.72$ ) est la plus facile (Gestuelle  $M=5.74$ ,  $SD= 0.76$ , Vocale  $M = 4.82$ ,  $SD = 1.54$  et Mixte  $M = 5.43$ ,  $SD = 1.04$ ).

Cependant, aucune n'est préférée de manière significative ( $F(3) = 3.42$ ,  $p < 0.5$ ).

## **2.9. Discussion**

### **2.9.1. Hypothèse 1 : vitesse d'exécution de la tâche**

#### **- Pour un haut niveau de précision de prédiction**

*Aucune différence significative entre les conditions Graphique et Gestuelle. **Infirmée***

Lorsque la prédiction est correcte, le pointage est plus rapide qu'un geste de commande. Nous prévoyions que les gestes faciles à produire mais plus longs à exécuter qu'un pointage seraient

équivalents aux pointages, certes plus rapides mais nécessitant plus de précision. Il n'en est rien. Il semble que le faible nombre d'items atténue le coup de la précision. Cela peut expliquer la rapidité de la condition Graphique.

*Les conditions Graphique et Gestuelle sont plus rapides que la condition Vocale. **Confirmée***

Dans la condition vocale, le temps exigé pour la reconnaissance et le traitement de la réponse utilisateur est plus important que pour le pointage et le geste.

*La condition Mixte sera plus rapide que les trois autres conditions. **Infirmée***

Dans la condition Mixte, la multimodalité n'est pas présente sur toute l'interface. Elle n'est disponible que lors de la phase de prédiction, la phase suivante s'effectuant par pointage. Ceci explique ses faibles performances. Généraliser la condition Mixte à l'ensemble de l'interface pourrait produire des résultats différents.

#### - **Pour un faible niveau de précision de prédiction**

*La condition Vocale est plus rapide que les conditions Graphique, Gestuelle et Mixte. **Confirmée***

Le résultat est justifié par le fait que l'interface dans la condition Vocale n'est pas surchargée par des prédictions incorrectes et inutiles (en Vocale il n'y a pas de rendu visuel des prédictions incorrectes), alors que dans les conditions Graphique, Gestuelle et Mixte, il faut en plus fermer la fenêtre de prédiction.

*Aucune différence significative entre la condition Graphique et la condition Gestuelle. **Infirmée***

Lors de la fermeture de la fenêtre de prédiction, si le bouton est suffisamment grand comme c'est le cas dans l'expérience, alors le bénéfice du geste ne nécessitant pas de pointage et de précision est trop faible. La condition Graphique reste quand même plus rapide.

*La condition Mixte sera plus rapide que les trois autres conditions. **Partiellement confirmée***

La condition Mixte n'a pas été meilleure que la condition Graphique bien que les utilisateurs puissent utiliser le pointage. Ce résultat peut être expliqué par le fait que plusieurs modalités sont offertes à l'utilisateur en même temps et dans la même condition (Mixte), ce qui peut être source de confusion entre les modalités d'interaction dans la fenêtre de prédiction et la liste complète des items.

### **2.9.2. Hypothèse 2 : erreurs**

En termes d'erreurs, les quatre conditions sont similaires pour les prédictions correctes. Par contre, en cas de prédictions incorrectes, les erreurs sont moins fréquentes dans les conditions Vocale et Graphique. En effet, dans la première les utilisateurs ont une étape de moins à produire ; dans la seconde, ils n'ont pas à changer de modalité de commande.

## **3. Conclusion**

Ce chapitre a présenté le menu Polymodal, un nouveau type de menu adaptatif dans lequel les items prédits sont présentés graphiquement sur une fenêtre de prédiction et/ou vocalement. La sélection d'un item prédit peut être effectuée graphiquement en pointant l'item, gestuellement en traçant un geste, ou vocalement en prononçant le nom de l'item cible.

Le choix d'une modalité d'entrée (graphique ou vocale) et de sortie (graphique, vocale, ou gestuelle) a été analysé dans les deux cas de prédictions correcte et incorrecte.

Le pointage graphique est dans tous les cas plus rapide. Les conditions Gestuelle et Vocale présentent de faibles résultats, cela étant certainement dû, d'une part, au changement de modalité de commande, et d'autre part, au temps de traitement du système. Les utilisateurs en pointage graphique restent sur la même modalité de commande lors de la phase de prédiction et lors de la sélection de la liste complète. Cette continuité monomodale semble être importante lorsqu'elle se situe au niveau des commandes mais pas forcément au niveau de l'articulation rendu-commande. En effet, les performances de la condition Vocale sont bonnes alors que (1) les prédictions sont présentées sur une modalité audio, (2) la liste complète est présentée sur la modalité visuelle et (3) les commandes sont effectuées sur la modalité par pointage en cas de prédiction incorrecte.

Néanmoins, la condition Mixte n'a pas donné de bons résultats dans les deux cas de prédictions (correcte et incorrecte). Aussi, proposer trop de modalités en simultanée semble source de confusion.



## 4. Synthèse

Dans ce chapitre, nous avons exploré l'axe Modalité du modèle PCA et proposons le menu polymodal dans lequel nous faisons varier les modalités de présentation et de commande de la prédiction.

Dans la modalité graphique, les items prédits sont présentés dans une fenêtre de prédiction superposée au menu initial alors que dans la modalité vocale, le système les vocalise. Les trois modalités de commande mises en œuvre sont graphique (par pointage), gestuelle (en traçant un symbole) et vocale (en prononçant le nom de l'item).

Nous recherchions la meilleure combinaison modale dans un cadre adaptatif. Lorsque la prédiction est correcte, la modalité graphique s'avère la plus pertinente. Dans le cas d'une prédiction incorrecte, la modalité vocale est la plus pertinente car le canal visuel n'est pas surchargé par des données inutiles.



# CONCLUSION ET PERSPECTIVES

L'objectif de cette thèse était d'étudier l'adaptation de l'interaction homme-machine, en particulier dans le cas de surface d'affichage restreinte. Ces contraintes complexifient l'interaction et justifient le besoin d'adaptation. Néanmoins, adapter est une prise de risque exposant l'utilisateur à des évolutions inappropriées de l'IHM. La question de recherche était comment au mieux accélérer l'interaction sans la pénaliser lorsque la prédiction est incorrecte. Nous rappelons ici les contributions puis ouvrons des perspectives de recherche.

## 1. Rappel des contributions

Les contributions sont doubles. D'un point de vue théorique, elles comportent un cadre pour la conception d'interactions adaptatives. D'un point de vue pratique, elles incluent de nouvelles techniques d'interaction induites par l'exploration du cadre théorique.

### 1.1. Espace de conception

Le cadre théorique comprend deux modèles : d'une part, le modèle systémique AIMS ; d'autre part, le modèle PCA.

Le modèle **AIMS** permet de décrire l'interaction adaptative selon les points de vue utilisateur et système. La boucle d'interaction est décrite par les dimensions PDA (Perception-Décision-Action), tandis que la boucle d'adaptation est décrite par les dimensions ADP (Apprentissage-Décision-Prédiction). Nous avons considéré les prédictions incorrectes en plaçant au sein d'AIMS un ensemble de critères qualité devant être pris en compte dans la conception d'une interaction adaptative.

Conjointement à ce modèle systémique, nous avons proposé le modèle de présentation de l'adaptation **PCA** (Perception-Cognition-Action). Ce modèle vise principalement à couvrir les critères qualité d'une interaction adaptative. Il explore des variables permettant de présenter l'adaptation dans le but d'améliorer la perception, de réduire la charge cognitive et de faciliter

la manipulation par l'utilisateur. Le volet Perception comprend les axes Forme, Mouvement, Position et Orientation. Le volet Cognition comprend les axes Ajout, Elimination, Explication et Moment. Le volet Action comprend les axes Quantité de prédiction, Modalité et Contrôle. Nous avons utilisé ce modèle pour caractériser l'état de l'art et identifier des voies de recherche inexplorées.

## 1.2. Exploration de l'espace de conception

Le pouvoir descriptif et génératif du cadre théorique nous a permis d'identifier un ensemble de techniques innovantes. Chaque technique a fait l'objet d'un prototypage rapide et d'une expérimentation contrôlée.

Les axes PCA que nous avons explorés sont Forme, Mouvement, Elimination, Ajout, Moment, Quantité de prédiction, Modalité et Contrôle, ils ont donné lieu à sept techniques.

L'exploration des axes Elimination et Mouvement nous a amenés à proposer le concept **d'adaptation éphémère avec disparition** selon deux variantes : *In Context Disappearing* et *Out of Context Disappearing*. Nous avons montré expérimentalement que ce concept permet d'accélérer l'interaction dans le cas d'une bonne prédiction sans la pénaliser lorsque l'adaptation est incorrecte. Néanmoins, l'inconvénient majeur de cette technique est qu'elle est source d'erreurs. L'exploration de l'axe Contrôle nous a permis de pallier cette faiblesse et de proposer **l'adaptation éphémère contrôlée**.

Si cette technique induit moins d'erreurs, il peut néanmoins être pertinent de réduire le risque de prédictions erronées. Pour ce faire, nous avons exploré l'axe Quantité de prédiction et avons proposé le **menu cloud** permettant d'augmenter le nombre d'items prédits pour accroître la probabilité que le bon item soit affiché. Le rendu étant impacté, nous avons conjointement exploré l'axe Forme.

La définition du paramètre « Distance au centre » au sein du menu *cloud* nous a amenés à nous interroger sur l'utilisation de cette distance à d'autres finalités. Jusqu'alors nous avons focalisé sur l'accélération de l'interaction sans considérer l'apprentissage. Or, une interface est une structure dont les items doivent être accédés plus au moins rapidement (raccourcis prédictifs) mais dont la position doit également être apprise. Pour ce faire, nous avons exploré

l'axe Moment et avons proposé deux techniques d'adaptation appelées *step-by-step* et *shortcut*. La première technique consiste à accompagner l'utilisateur en lui proposant des prédictions à chaque niveau de la hiérarchie jusqu'à ce qu'il atteigne son objectif. La seconde consiste à accélérer l'interaction en présentant un raccourci de prédiction uniquement à la racine. L'expérimentation révèle que la technique *step-by-step* est le meilleur compromis permettant à la fois de faciliter l'apprentissage et d'accélérer l'interaction, qu'il s'agisse de prédictions correctes ou incorrectes.

Afin d'augmenter encore les possibilités d'apprentissage du système et d'alléger la charge perceptive, nous avons exploré l'axe Modalité en proposant le menu *polymodal*. Les modalités de rendu graphique et vocal ainsi que les modalités de commande, gestuelle, vocale et par pointage ont été explorées. En cas de prédictions incorrectes, la continuité modale entre les phases de prédiction et de sélection dans la liste complète s'avère appréciée et l'articulation audio et visuelle est pertinente car elle allège la perception visuelle.

Les études expérimentales montrent l'intérêt des techniques proposées et confirment donc le potentiel génératif du modèle PCA. Les perspectives incluent bien entendu l'exploration intensive des axes. Ceci inclut ceux partiellement explorés (Forme, Mouvement, Elimination, Ajout, Moment, Quantité de prédiction, Modalité et Contrôle) et non explorés (Position, Orientation et Explication).

Le choix des axes explorés dans la thèse se justifie par une volonté de couverture de l'espace. Les axes non explorés ont été écartés pour une raison de temps, en cette approche méthodologique pour les raisons suivantes :

L'axe Position permet de changer l'emplacement des éléments dans l'interface afin de les mettre en avant ou en arrière-plan. Cet axe risque d'impacter la stabilité de l'interface et, en conséquence, de compliquer la construction pour l'utilisateur d'un modèle mémoire. Aussi, nous avons écarté l'axe Position de l'étude mais il mériterait d'être reconsidéré conjointement avec d'autres axes pour atténuer le changement (par exemple l'axe Mouvement).

L'axe Orientation permet de changer l'alignement des objets de l'interface. Nous n'avons pas exploré cet axe car nous avons supposé que ce type de présentation sur une interface de taille

réduite n'est pas favorable et dégrade la perception. Grâce à l'outil PCA-S, le prototypage rapide mériterait néanmoins d'être réalisé.

L'axe Explication vise à réduire la charge cognitive et à faciliter la compréhension. Cet axe n'a pas été abordé dans la thèse car nous centrons nos travaux principalement sur les techniques d'accélération de l'interaction plutôt que sur l'apprentissage. Nous avons montré dans *step-by-step* qu'il était possible de concilier les deux. Aussi, l'exploration de cet axe est une perspective pour ce travail.

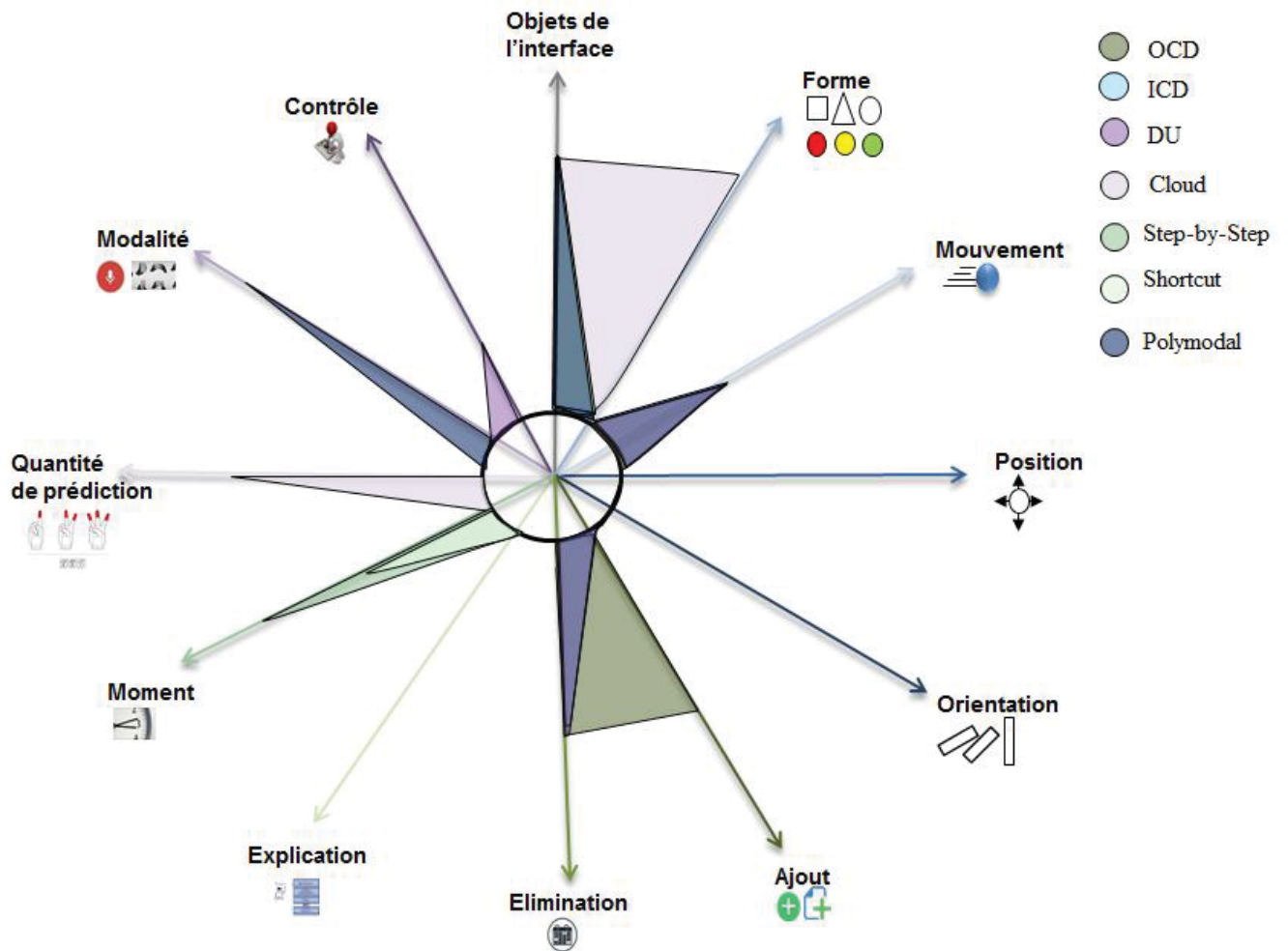


Figure 68. Synthèse des dimensions explorées dans les sept techniques proposées.

Etant donnée la combinatoire des axes, l'exploration intensive de l'espace de conception est chronophage. Aussi, la vision est l'opérationnalisation du cadre théorique en un environnement de prototypage rapide. Nous avons présenté PCA-S, un premier démonstrateur de concept.

## 2. Perspectives

Les perspectives comportent deux aspects : d'une part, la phase de conception, d'autre part, la phase d'exécution.

### 2.1. Environnement de conception pour l'exploration intensive

Pour chaque axe PCA, les idées sont nombreuses. Par exemple :

L'axe Contrôle pourrait être exploré au-delà du contrôle de la manipulation des objets de l'interface. En effet, les données d'adaptation elles-mêmes pourraient être contrôlées afin d'assurer par exemple la confidentialité de cette dernière (*Privacy by Design*).

L'exploration de l'axe Modalité pourrait être poursuivie afin de déterminer l'ensemble des combinaisons modales les plus favorables à l'adaptation.

L'axe Moment pourrait également donner lieu à des études supplémentaires visant à faire varier l'échelle de temps considérée par l'adaptation. Ainsi, cette dernière pourrait être produite suite à une inactivité de l'utilisateur ou suite à une longue période d'activité. Cela pourrait être mis en œuvre aussi bien au niveau des micros tâches qu'au niveau d'une activité mensuelle voire annuelle.

L'axe Quantité de prédiction mériterait d'être également exploré notamment concernant le rapport entre le nombre d'items prédits et la taille de la liste complète.

La plupart des études d'adaptation se basent sur l'ajout d'effets graphiques ou de fenêtre prédictive, or l'axe Ajout pourrait être exploré par l'ajout d'items au sein même de la liste complète. Par exemple, une interface pourrait progressivement proposer des items supplémentaires en fonction du niveau d'apprentissage de l'utilisateur.

L'axe Orientation couplé à l'axe Position devrait permettre de présenter encore différemment des éléments de l'interface par exemple en périphérie de celle-ci.

Il serait intéressant d'explorer l'axe Explication afin de rendre le dialogue homme-machine davantage intelligible.

Clairement, l'exploration intensive se heurte au passage à l'échelle : prototyper chaque configuration et l'évaluer dans les différents contextes d'usage est coûteux. Nous prôtons le développement d'un outil logiciel en ligne permettant le paramétrage de l'IHM, sa génération automatique puis son évaluation par des utilisateurs en ligne. Un premier démonstrateur de concept PCA-S a été réalisé pour le cas des menus avec une couverture réduite à la présentation. Le développement de cet outil est la perspective immédiate à ce travail.

## **2.2. Plasticité pour le meilleur compromis à l'exécution**

Conjointement aux axes PCA (perspective 2.1), l'ultime objectif est la qualité de la collaboration homme-machine. Le défi est la qualité de la prédiction et l'ajustement optimal des dimensions ADP du modèle AIMS via l'ajustement des dimensions PDA. Aujourd'hui, dans les systèmes adaptatifs tels que les *smartphones* ou les ordinateurs, les prédictions sont faites sur les fonctions offertes par le système : les prédictions sont le résultat de statistiques d'usage. La précision est importante. Aussi, au-delà du niveau macroscopique (les fonctions utilisées), est-il important de nuancer par un grain plus fin (micro tâche) et une capture du contexte d'usage.

Les systèmes adaptatifs existants manquent encore de capacité d'adaptation au contexte d'usage. Pour exemple, l'interaction n'est toujours pas accessible à tous. Un novice rencontre encore des difficultés à manipuler des IHM sans compter les déficiences qui sont encore mal gérées. De façon générale, l'utilisateur fait encore bien souvent des efforts pour comprendre et manipuler l'IHM. Cette charge cognitive devrait être réduite par un dialogue homme-machine plastique. En d'autres termes, l'intelligence du système devrait être augmentée pour accélérer l'interaction et faciliter l'apprentissage. Il serait garant du meilleur compromis à l'exécution pour le bien de l'humain.

Clairement, cette perspective milite pour une ouverture de la communauté Interaction Homme-Machine (IHM) à l'Intelligence Artificielle (IA), (Winograd, 2006, Grudin, 2009, Kolski et al., 2014). Ce serait l'opportunité d'explorer conjointement de nouveaux algorithmes



de prédiction et d'adaptation qui prennent en compte les critères qualité de l'interaction adaptative.

## Bibliographie

1. Ahlström, D., Cockburn, A., Gutwin, C., and Irani, P. (2010). Why it's quick to be square: modelling new and existing hierarchical menu designs. Proc. of the ACM SIGCHI Conf. on Human Factors in Computing Systems CHI'2010 (Atlanta, April 10-15, 2010). ACM Press, New York, pp. 1371–1380.
2. Ahlström, D. (2006). Modeling and improving selection in cascading pull-down menus using Fitt's law, the steering law and force fields. Proc. of the ACM SIGCHI Conf. on Human Factors in Computing Systems CHI'2006 (Portland, April 2-7, 2006). ACM Press, New York, pp. 61–70.
3. Arhippainen, L., Rantakokko, T., and Tähti, R. M. (2004). Navigation with an adaptive mobile map-application: user experiences of gesture- and context-sensitiveness. Proc. of 2<sup>nd</sup> Int. Symposium on Ubiquitous Computing Systems UCS'2005 (Tokyo, Nov. 8-9, 2004). Lecture Notes in Computer Science, vol. 3598. Springer, Berlin, pp. 62–73.
4. Bailly, G., Lecolinet, E., Nigay, L., and Lebiere, C. (2008). Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. Proc. of the ACM Working Conf. on Advanced Visual Interfaces AVI'2008 (Napoli, May 28-30, 2008). ACM Press, New York, pp. 15–22.
5. Bailly, G., Lecolinet, E., Nigay, L. (2009). MenUA: A Design Space of Menu Techniques. Accessible at [www.gillesbailly.fr/menua/](http://www.gillesbailly.fr/menua/)
6. Bailly, G., Oulasvirta, A., Kötzing, T., and Hoppe, S. (2013). MenuOptimizer: interactive optimization of menu systems. Proc. of the 26<sup>th</sup> ACM Symposium on User interface software and technology UIST'2013 (St Andrews, October 8-11, 2013). ACM Press, New York, pp. 331–342.
7. Bastien., C and Scapin., D.L. 1995. Evaluating a User Interface with Ergonomic Criteria, International Journal of Human-Computer Interaction 7, 2, 1995, 105–121. <http://dx.doi.org/10.1080/10447319509526114>.

8. Baudish, P. and Cutwin, C. (2004). Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. Proc. of ACM SIGCHI Conf. on Human Factors in Computing Systems CHI'2004 (Vienna, April 24-29, 2004). ACM Press, New York, pp. 367–374.
9. Beaudouin-Lafon, M. (1997). Ingénierie des systèmes interactifs. <http://www-ihm.lri.fr/mbl/ENS/IHM/ecole-in2p3/Cours/index.html>.
10. Beaudouin-Lafon, M. (2000). Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In: Proceedings of ACM Conference on Human Aspects in Computing Systems, CHI'2000, The Hague, April 1-6, 2000. pp. 446–453, ACM Press, New York.
11. Bellotti, V., Sellen, A. Design for privacy in ubiquitous computing environments. In Proceedings of the Third Conference on European Conference on Computer-Supported Cooperative Work, ECSCW'93, pp. 77–92, Norwell, MA, USA, 1993. Kluwer Academic Publishers. ISBN 0-7923-2447-1.
12. Benaboud, R. and Sahnoun, Z. 2006. Implémentation par agents du processus d'adaptation des informations en interface homme-machine. In Proc. of the 18th Int. Conf. of the Association Francophone d'Interaction Homme-Machine (IHM '06). ACM, New York, NY, USA, 273-276.
13. Bederson, B.B. (2000). Fisheye menus. Proc. of 13<sup>th</sup> ACM Symposium on User Interface Software and Technology UIST'2000 (San Diego, November 6-8, 2000). ACM Press, New York, pp. 217–225.
14. Benyon, D. and Murray, D. (1993). Developing adaptive systems to fit individual aptitudes. Proceedings of the 1st international conference on Intelligent User Interfaces, pp. 115-121, Orlando, Florida, United States, ACM Press, 1993.
15. Bérard F., Coutaz J., Crowley J.L. Le tableau Magique, Ergonomie et Informatique Avancée-Interaction Homme-Machine (ErgoIHM'2000), D.L Scapin & E. Vergisson (eds), CRT ILS & ESTIA, Octobre 2000, Biarritz, pp. 33-40.
16. Bertin, J. Sémiologie graphique. Paris, Mouton/Gauthier-Villars, 1967.
17. Bratman, M.E. (1987). Intention, Plans, and Practical Reason. Harvard University

Press, Cambridge.

18. Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, pp. 2119–2128, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7.
19. Bridle, R. and McCreath, E. (2005). Predictive Menu Selection on a Mobile Phone. Proc. of Workshop W7 on mining spatio-temporal data, ECML/ PKDD, Portugal.
20. Browne, D.P., Sharratt, B. and Norman, M.A. (1986). The formal specification of adaptive user interfaces using Command Language Grammar. In: Proceedings of the ACM Conference on Human Aspects in Computing Systems, CHI'1986, Boston, April 13-17, 1986. pp. 256–260. ACM Press, New York.
21. Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*. 6 (2-3), 87–129.
22. Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.). 11 (1/2), 87-110.
23. Bunt, A., Conati, C. and McGrenere, J. (2004). What role can adaptive support play in an adaptable system? In: Proceedings of the 9<sup>th</sup> International Conference on Intelligent User Interfaces, IUI'2004, Funchal, January 13-16, 2004. pp. 117–124, ACM Press, New York.
24. Bunt, A., Conati, C. and McGrenere, J. (2009). A Mixed-Initiative Approach to Interface Personalization. *AI Magazine*, vol. 30, no. 4, pp. 58-64, 2009.
25. Bunt, A., Lount, M., Lauzon, C. (2012). Are explanations always important?: A study of deployed, low-cost intelligent interactive systems. In Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, IUI bibliography 209 '12, pp. 169–178, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1048-2.
26. Calvary, G., Coutaz, J. and Thevenin, D. (2001). Supporting Context Changes for Plastic User Interfaces: a Process and a Mechanism. In: Blandford, A., Vanderdonckt, J., Gray, P. (Eds.), *Interactions sans frontières – Interactions without frontiers*, Proceedings of the Joint AFIHM-BCS Conference on Human-Computer Interaction, IHM-HCI'2001, Lille, September 10-14, 2001. Vol. I, pp. 349–363, Springer, London.

27. Calvary G., Coutaz J., Thevenin D., Limbourg Q., Bouillon L., Vanderdonckt J. (2003). A unifying reference framework for multi-target user interfaces, *Journal of Interacting With Computer*, Elsevier Science B.V, June, Vol 15/3, pp 289-308.
28. Carroll, J. M. and Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM*, 27(8):800-806.
29. Carpendale. M.S.T. (2003). Considering Visual Variables as a Basis for Information Visualisation, Univ. of Calgary, Dept. of Computer Science, 2001-693-16.
30. Catrambone, R., and Carroll, J. M. (1987). Learning a word processing system with training wheels and guided exploration. *Proc. of the ACM SIGCHI/GI Conf. on Human Factors in computing systems and graphics interface CHI/GI'87* (Toronto, April 5-9, 1987). ACM Press, New York, pp. 169–174.
31. Chaari, T., Zouari, M., Laforest, F. Ontology based context-aware adaptation approach. In *Book of Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications*, pp. 26–58, Hershey - New York, 2009. Information science reference.
32. Chaari T., Laforest F., Celentano A. (2004) Design of context-aware applications based on web services, Technical Report RR-2004-033, LIRIS, Lyon.
33. Clark, B., and Matthews, J. (2005). Deciding layers: adaptive composition of layers in a multi-layer user interface. *Proc. of 11<sup>th</sup> Int. Conf. on Human-Computer Interaction HCI International'2005* (Las Vegas, July 22-27, 2005). C. Stephanidis (Ed.), Volume 7. Mira Digital Publishing.
34. Chêne, D., Petit, E., Zijp-Rouzier, S. (2016). How to achieve design for All:. In: Antona M., Stephanidis C. (eds) *Universal Access in Human-Computer Interaction. Interaction Techniques and Environments. UAHCI 2016. Lecture Notes in Computer Science*, vol 9738. Springer, Cham.
35. Cockburn, A. and Gin, A. (2006). A faster cascading menu selection with enlarged activation areas. *Proc. of Graphics Interface GI'2006* (Quebec, June 7–9, 2006). Canadian Information Processing Society, Toronto, pp. 65–71.
36. Cockburn, A., Gutwin, C., Greenberg, S. (2007). A Predictive Model of Menu

- Performance. In Proceedings of the 2007 Conference on Human Factors in Computing Systems (CHI 2007), 627-636.
37. Cockburn, A., and Brewster, S. (2005). Multimodal Feedback for the Acquisition of Small Targets. *Ergonomics*. 48(9): 1129--1150. Taylor and Francis. 2005.
  38. Cockton, G. (1987). Some Critical Remarks on Abstractions for Adaptable Dialogue Managers. *BCS HCI 1987*: 325-343.
  39. Constantine, S. (2001). Adaptive Techniques for Universal Access. *User Modeling and User-Adapted Interaction* 11: 159-179, 2001.
  40. Cote-Munoz, J.A. AIDA: An adaptive system for interactive drafting and CAD applications. In M. Schneider-Hufschmidt, T. Kuhme, and U. Malinowski, editors, *Adaptive User Interfaces*, pp. 225–240. Elsevier Science Publishers, 1993.
  41. Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., Young, R. M. (1995): Four Easy Pieces for Assessing the Usability of Multimodal Interaction: the CARE Propertie. In: Nordby, Knut (eds.) *Proceedings of INTERACT 95 - IFIP TC13 Fifth International Conference on Human-Computer Interaction June 25-29, 1995, Lillehammer, Norway*. pp. 115-120.
  42. Coutaz, J. Meta-user interfaces for ambient spaces. In *Proceedings of the 5<sup>th</sup> International Conference on Task Models and Diagrams for Users Interface Design, TAMODIA'06*, pp. 1–15, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-70815-5.
  43. Coutaz, J., Balme, L., Alvaro, X., Calvary, G., Demeure, A., Sottet, J.S. (2007). An MDE-SOA Approach to Support Plastic User Interfaces in Ambient Spaces. In: Stephanidis, Constantine (eds.) *Universal Access in Human-Computer Interaction. Ambient Interaction, 4th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2007 Held as Part of HCI International 2007 Beijing, China, July 22-27, 2007 Proceedings, Part II July 22-27, 2007, Beijing, China*. pp. 63-72.
  44. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. G. (2001). Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18(1), 32-80.

45. Debevc, M., Meyer, B., Donlagic, D., Svecko, R. Design and evaluation of an adaptive icon toolbar. *User Modeling and User-Adapted Interaction*, 6(1):1–21, 1996.
46. Dessart, C.-E., V. Genaro Motti, and J. Vanderdonckt. (2011). Showing user interface adaptivity by animated transitions. *Proc. of ACM Conf. on Engineering Interactive Computing Systems EICS'2011 (Pisa, June 13-16, 2011)*. ACM Press, New York, pp. 95–104.
47. Dey, A.K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), pp. 4-7.
48. Dieterich, H., Malinowski, U., Kühme, T. and Schneider-Hufschmidt, M. (1993). State of the Art in Adaptive User Interfaces. In: Schneider-Hufschmidt, M., Khüme, T., Malinowski, U. (Eds.), *Adaptive User Interfaces: Principle and Practice*. pp. 13-48, North Holland, Amsterdam.
49. Eisenstein, J. and Puerta, A. (2000). Adaptation in Automated User-Interface Design. In: *Proceedings of ACM Conference on Intelligent User Interfaces, IUI'2000*, New Orleans, January 9-12, 2000. pp. 74–81, ACM Press, New York.
50. Feinberg, J. (2010). Wordle. In *Beautiful Visualization: Looking at Data through the Eyes of Experts*, J. Steele, N. Iliinsky (eds.). O'Reilly, Sebastopol, 37-58.
51. Findlater, L., and Mc Grenere, J. A comparison of static, adaptive, and adaptable menus. In *proceedings of the SIGCHI conference on Human factors in computing systems*, pages 89-96. ACM, 2004.
52. Findlater, L. and McGrenere, J. Evaluating reduced functionality interfaces according to feature findability and awareness. In *Proc. IFIP Interact 2007*, (2007), 592-605.
53. Findlater, L., and McGrenere, J. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp. 1247-1256. ACM, 2008.
54. Findlater, L., K. Moffatt, J. McGrenere, and J. Dawson (2009). Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. *Proc. of ACM SIGCHI Conf. on Human Aspects in Computing Systems CHI'2009 (Boston, April 4-*

- 9, 2009). ACM Press, New York, pp. 1655–1664.
55. Findlater, L., and Gajos., K.Z. Design space and evaluation challenges of adaptive graphical user interfaces. *AI Magazine*, 30(4) : 68, 2009.
  56. Florins, M. and Vanderdonckt J.: 2004, Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems. In: *Proceedings of ACM Conference on Intelligent User Interfaces, IUI'2004*, Funchal, January 13-16, 2004. pp. 140–147, ACM Press, New York.
  57. Fischer, S., Schwan, S. Adaptively shortened pull down menus: Location knowledge and selection efficiency. *Behaviour & Information Technology*, 27(5):439–444, 2008.
  58. Francone, J., Bailly, G., Lecolinet, E., Mandran, N., Nigay, L. (2010). Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-Free Selection for Expert Mode. In *Proc. of Working Conf. on Advanced Visual Interfaces (AVI'2010)*, 173-180.
  59. Gajos, K. Z., and Weld, D. S. SUPPLE: automatically generating user interfaces. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interface*, pp. 93-100, New York, NY, USA, 2004. ACM Press.
  60. Gajos, K. Z., and Weld, D. S. Preference elicitation for interface optimization. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pp. 173-182, New York, NY, USA, 2005. ACM Press.
  61. Gajos, K.Z., Czerwinski, M., Tan, D.S., and Weld, D.S. (2006). Exploring the design space for adaptive graphical user interfaces. In *Proc. AVI'06*, 201-208.
  62. Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S. (2008). Predictability and accuracy in adaptive interfaces. In *Proc. CHI '08*, 1271-1274.
  63. Gardiner, M. and Christie, B. (1987). *Applying Cognitive Psychology to User Interface Design*. John Wiley, New York.
  64. Gómez, J.M. and Tran, T. (2009). A Survey on Approaches to Adaptation on the Web. In: Lytras, M.D., Ordóñez de Pablos, P. (Eds.), *Emerging Topics and Technologies in Information Systems*. pp. 136-152, IGI Global Publishing, Hershey.
  65. Gong, Q., Salvendy, G. An approach to the design of a skill adaptive interface.



- International Journal of Human-Computer Interaction, 7(4):365–383, 1995.
66. Greenberg, S., Witten, I. (1985). Adaptive personalized interfaces- A question of viability. *Behaviour and Information Technology* 4, 1, 31-45.
  67. Grossman, T., Dragicevic, P., Balakrishnan, R. Strategies for accelerating on-line learning of hotkeys. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1591–1600, 2007.
  68. Grudin, J. "AI and HCI: two fields divided by a common focus". *AI Magazine*, 30(4):48-57, Winter 2009. AAAI.
  69. Halvey, M., Keane, M. T. (2007). An Assessment of Tag Presentation Techniques. In *Proc. of WWW'2007*, 1313–1314.
  70. Hartmann, M. Challenges in Developing User-Adaptive Intelligent User Interfaces. In: Hauger, D., Köck, M. & Nauerz, A. (Hrsg.), *17th Workshop on Adaptivity and User Modeling in Interactive Systems*.
  71. Heimerl, F., Lohmann, S., Lange, S., Ertl, T. (2014). Word Cloud Explorer: Text Analytics based on Word Clouds. In *Proc. of the 47<sup>th</sup> Hawaii Int. Conf. on System Science (HICSS'2014)*, 1833-1842.
  72. Hesselmann, T., S. Flöring, and M. Schmitt. (2009). Stacked half-pie menus: navigating nested menus on interactive tabletops. *Proc. of the ACM Int. Conf. on Interactive Tabletops and Surfaces. ITS'2009 (Banff, November 23-25, 2009)*. ACM Press, New York, pp. 173–180.
  73. Holleis, P., Rukzio, E., Otto., F, Schmidt, A. Privacy and curiosity in mobile interactions with public displays. In *Proceedings of the CHI 2007 Workshop on Mobile Spatial Interaction*, 2007.
  74. Hook K (2000) Steps to take before intelligent user interfaces become real. *Interacting with computers* 12:409–426.
  75. Horvitz, E., Herckerman, D., Hovel, D., Rommelse, R. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of UAI'98, Conference on Uncertainty in Artificial Intelligence*, pp. 256–265, 1998.
  76. Horvitz, E. (1999). Principles of Mixed-Initiative User Interfaces. In: *Proceedings of ACM Conference on Human Factors in Computing Systems CHI'99*, Pittsburgh, May

- 15-20, 1999. pp. 159–166, ACM Press, New York.
77. Huang, S.C., Chou, I-F, and Bias; R.G. (2006). Empirical evaluation of a popular cellular phone's menu system: theory meets practice. *Journal of Usability Studies*, vol. 1, no. 2, pp. 91–108.
78. Huhtala, J., Mantyjärvi, J., Venta, L. and Isomursu, M. Animated Transitions for Adaptive Small Size Mobile Menus. In *Proc of the 12<sup>th</sup> IFIP TC 13 Int. Conf on Human-Computer Interaction Interact'2009*.
79. Huhtala, J., Sarjanoja, A.-H., Mäntyjärvi, J., Isomursu, M. and Häkkinen, J. (2010). Animated UI transitions and perception of time: a user study on animated effects on a mobile screen. In: *Proceedings of ACM Conference on Human Aspects in Computing Systems, CHI'2010, Atlanta, April 10-15, 2010*. pp. 1339–1342, ACM Press, New York.
80. Jacko, J.A., and Salvendy, G. (1996). Hierarchical menu design: breadth, depth, and task complexity. *Perceptual and Motor Skills*, vol. 82, pp. 1187–1201.
81. Jameson, A. (2003). Adaptive Interfaces and Agents. In: Jacko, J.A., Sears, A. (Eds.), *Human-Computer Interface Handbook*. Lawrence Erlbaum, Mahwah, pp. 305–330.
82. Jameson, A. (2009). Understanding and Dealing with Usability Side Effects of Intelligent Processing. *AI Magazine* 30(4), 23-40.
83. John, C., Raymond, M. and Carolyn, Watters, R. (2004). A comparison of simple hierarchy and grid metaphors for option layouts on small-size screens. *Int. J. Hum.-Comput. Stud.* 60(5-6): 564-584.
84. Johnson, B. and Shneiderman, B. (1991). Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. *Proc. of the 2<sup>nd</sup> Conf. on Visualization Vis'1991 (San Diego, October 22-25, 1991)*. IEEE Computer Society Press, Piscataway, pp. 284–291.
85. Jones, M., Marsden, G. Mohd-Nasir, N., and Boone, K. (1999). Improving Web Interaction on Small Displays. In *8<sup>th</sup> World Wide Web conference, Toronto 1999*.
86. Kang, H., Plaisant, C., and Shneiderman, B. (2003). New approaches to help users get started with visual interfaces: multi-layered interfaces and integrated initial guidance.

- Proc. of the National Conf. on digital government research (Boston, May 18-21, 2003), pp. 1–6.
87. Kang, E., Jackson, E., and Schulte, W. (2010). An Approach for Effective Design Space Exploration. In *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, Lecture Notes in Computer Science, vol. 6662, Berlin: Springer, pp 33-54.
  88. Karagiannidi, C., koumpis, A., and Stephanidis, C. (1996). Deciding ‘What’, ‘When’, ‘Why’, and ‘How’ to Adapt in Intelligent Multimedia Presentation systems”, *ECAI’96 Workshop on Towards a standard reference model for intelligent multimedia presentation systems*, Budapest, Hungary.
  89. Kellen, V. (2003). *Business Performance Measurement. At the Crossroads of Strategy, Decision-Making, Learning and Information Visualization*. DePaul Univ., Chicago.
  90. Kiger, J.I. (1984). The depth/breadth trade-off in the design of menu-driven user interfaces. *International Journal of Man-Machine Studies*, vol. 20, no. 2, pp. 201–213.
  91. Kobsa, A. User Modeling: Recent Work, Prospects and Hazards. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski, Eds.: *Adaptive User Interfaces: Principles and Practise*. Amsterdam: North Holland Elsevier. 1993.
  92. Kobsa, A., Koenemann J. & Pohl W. (2001). Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships, (pp. 111-155). *The knowledge Engineering Review*, 16(2).
  93. Kolski C., Tendjaoui M., Millot P. (1992). A process method for the design of "intelligent" man-machine interfaces : case study : "the Decisional Module of Imagery". *International Journal of Human Factors in Manufacturing*, 2, pp. 155-175.
  94. Kolski C., Boy G., Mélançon G., Ochs M., Vanderdonckt J. (2014). Fertilisation croisée entre interaction personne-système et IA. In P. Marquis, O. Papini, H. Prade (Ed.), *L'intelligence artificielle : frontières et applications*, Cépaduès Editions, Toulouse, France, pp. 1117-1141.
  95. Kurtev, I, Bézivin, J., Aksit, M. (2002). Technological spaces: An initial appraisal, *CoopIS, DOA'2002 Federated Conferences, Industrial track*.
  96. Lamping, J., Roa, R., and Pirolli, P. (1995). A focus+context technique based on

- hyperbolic geometry for visualizing large hierarchies. Proc. of the ACM SIGCHI Conf. on Human Factors in Computing Systems CHI'1995 (Denver, May 7-11, 1995). ACM Press, New York, pp. 401–408.
97. Langley, P. (1999). User Modeling in Adaptive Interfaces. In: Kay, J. (Ed.), Proceedings of the 7<sup>th</sup> International Conference on User Modeling UM'1999, Banff, July 1999. pp. 357-370, Springer, Berlin.
98. Lavie T. and Meyer, J. (2010). Benefits and costs of adaptive user interfaces. *Int. J. of Human-Comp. Studies*, vol. 68, no. 8, pp. 508–524.
99. Lee, D.S. and Yoon, W.C. (2004). Quantitative results assessing design issues of selection-supportive menus. *International Journal of Industrial Ergonomics* 33 (1), 2004, pp. 41-52.
100. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López-Jaquero, V. (2005). UsiXML: a Language Supporting Multi-Path Development of User Interfaces. In: Proceedings of 9<sup>th</sup> IFIP Working Conf. on Engineering for Human-Computer Interaction jointly with 11<sup>th</sup> Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004, Hamburg, July 11-13, 2005. Vol. 3425 of Lecture Notes in Computer Science. pp. 200–220, Springer, Berlin.
101. Linton, F., Joy, D., Schaefer, H., Charron, A. OWL: A recommender system for organization-wide learning. *Educational Technology & Society*, 3(1), 2000.
102. Lohmann, S., Ziegler, J., Tetzlaff, L. 2009. Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. In Proc. of INTERACT'2009, 392–404.
103. López-Jaquero, V., Vanderdonckt, J., Montero, F., and González, P. (2007). Towards an Extended Model of User Interface Adaptation: the ISATINE framework. In: Proceedings of IFIP WG2.7/13.4 10<sup>th</sup> Conference on Engineering Human Computer Interaction jointly organized with IFIP WG 13.2 1<sup>st</sup> Conference on Human Centred Software Engineering and DSVIS - 14<sup>th</sup> Conference on Design Specification and Verification of Interactive Systems, EIS'2007, Salamanca, March 22-24, 2007, J. Gulliksen, M.B. Harning, Ph. Palanque (Eds.). Vol. 4940 of Lecture Notes in Computer Science. pp. 374–392, Springer, Berlin.
104. López Jaquero, V., Montero, F. and González, P. (2008). Quality of Adaptation: User

- Cognitive Models in Adaptation Quality Assessment. In: Proceedings of 6<sup>th</sup> International Conference on Computer-Aided Design of User Interfaces, CADUI'2008, Albacete, June 11-13, 2008. López Jaquero, V., Montero, F., Molina, J.P., Vanderdonckt, J. (Eds.). pp. 265–276, Springer, Berlin.
105. López-Jaquero, V., Montero, F. and Gonzalez, P. (2009). AB-HCI: an interface multi-agent system to support human-centred computing. *IET Software* 3, 14. pp. 14–25.
106. Mackinlay, J. (1986). Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics* 5, 2, 110–141.
107. MacLean, A., Young, R.M., Bellotti, V. and Moran, T.P. (1991). Questions, options, and criteria: elements of design space analysis. *Hum.-Comput. Interact.* 6, 3 (September 1991), 201-250. DOI=10.1207/s15327051hci0603\&4\_2.
108. Malinowski, U. Adjusting the presentation of forms to users' behaviour. In Proceedings of the ACM Conference on Intelligent User Interfaces (IUI'93), pp. 247–249, 1993.
109. Mark T. Maybury and Wolfgang Wahlster, editors. Readings in intelligent user interfaces. Morgan Kaufmann Publishers Inc., 1998.
110. McBurney, S., Howard Williams, M., Taylor, N., Papadopoulou, E. Managing user preferences for personalization in a pervasive service environment. *Advanced International Conference on Telecommunications*, 0 :32, 2007.
111. McGrenere, J., Baecker, R.M., and Booth, K.S. (2007). A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Transactions on Computer-Human Interaction*. 14(1), Article no 3, 1-43.
112. McKinley, P. K., Sadjadi, S. M., Kasten, E. P., and Cheng, B.H. (2004). Composing Adaptive Software. *IEEE Computer* 37, 7. pp. 56–64.
113. Matejka, J., Grossman, T., Fitzmaurice, G. (2013). Patina: Dynamic Heatmaps for Visualizing Application Usage. In Proc. of ACM Conf. on Human Aspects in Computing Systems (CHI'2013), 3227–3236.
114. Matsui, S. and Yamada, S. (2008). Genetic algorithm can optimize hierarchical menus. Proc. of the ACM SIGCHI Conf. on Human Factors in Computing Systems CHI'2008 (Florence, April 5- 10, 2008). ACM Press, New York, pp. 1385–1388.

115. McGrenere, J., Baecker, R., Booth, K. (2002). An evaluation of a multiple interface design solution for bloated software. *CHI Letters* 4, 1, 163–170.
116. Miah, T., Karageorgou, M., Knott, R.P. Adaptive toolbars: An architectural overview. In *Proceedings of the 3rd ERCIM Workshop on User Interfaces for All*, 1997.
117. Mitchell, J., Shneiderman, B. (1989). Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin* 20, 4, 33–37.
118. Moiscu, B. (2006). Adaptation dans les systems d'information spatio-temporelle interactifs, *INFOSID 2006*, 1er-3 juin, Hammamet, Tunisie.
119. Myers, B., Weitzman, D., Ko, A., Chau, D. Answering why and why not questions in user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '06*, pp. 397–406, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7.
120. Norcio, A. F. and Stanley, J. (1988). *Adaptive Human- Computer Interfaces*. NRL Report 9148, Naval Research Laboratory.
121. Norman, K.L. (1991). *The Psychology of Menu Selection: Designing Cognitive Control at the Human/computer interface*. Intellect Books.
122. Norman, D.A. (1986). Cognitive Engineering. In: Norman, D.A., Draper, S.W. (Eds.): *User Centered System Design*. Lawrence Erlbaum Associates, Hillsdale, pp. 31–61.
123. Norman, D.A. *The design of everyday things*. Doubleday, NY, 1988.
124. Norman, Donald A. (1994). How Might People Interact with Agents. In *Communications of the ACM*, 37 (7) pp. 68-71.
125. Norman, D. A. (1996). Design as practiced. In T. Winograd (Ed.), *Bringing design to Software*. Reading, MA: Addison-Wesley. pp. 233-247.
126. Oppermann, R. (1994). Adaptively supported adaptability. *International Journal of Human-Computer Studies* 40, 3. pp. 455–472.
127. Oppermann, R., Specht, M. (2000). A Context-Sensitive Nomadic Exhibition Guide. In: Thomas, Peter J., Gellersen, Hans-Werner (eds.) *Handheld and Ubiquitous Computing - Second International Symposium - HUC 2000 September 25-27, 2000, Bristol, UK*. pp. 127-142.
128. Papadopoulou, E., McBurney, S., Taylor, N., Williams, M. H. A dynamic approach to

- dealing with user preferences in a pervasive system. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 409–416, Washington, DC, USA, 2008. IEEE Computer Society.
129. Park, J., Han, S.H, Cho, Y., Park, W., Im, H., and Hong., S.W. (2007). Usability of a Roll-Up Style Menu on a Mobile Phone. Proc. of the 8<sup>th</sup> Pan-Pacific Conf. on Occupational Ergonomics, Bangkok 2007.
130. Patel, D., Marsden, G., Jones, M., and Jones, S. An Evaluation of Techniques for Image Searching and Browsing on Mobile Devices. Proceedings SAICSIT 2009, pp. 60-69.
131. Pielot, M., Kazakova, A., Hesselmann, T., Heuten, W., and Boll, S. (2012). PocketMenu: non-visual menus for touch screen devices. Proc. of the 14<sup>th</sup> Int. Conf. on Human-computer interaction with mobile devices and services MobileHCI'2012 (San Francisco, September 21–24, 2012). ACM Press, New York, pp. 327–330.
132. Pignotti, E., Edwards, P., Grimnes, G.A. Contextaware personalised service delivery. In Proceedings of the European Conference on Artificial Intelligence, pp. 1077–1078, 2004.
133. Pirolli, P. (2009). An elementary social information foraging model. Proc. of the 27<sup>th</sup> Int. Conf. on Human Factors in Computing Systems CHI'2009. ACM Press, New York, pp. 605–614.
134. Plaisant, C., Grosjean, J., and Bederson, B.B. (2002). SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Proc. of the IEEE Symposium on Information Visualization InfoVis'2002 (October 28 -29). IEEE Press, Piscataway, pp. 57–64.
135. Ponsard, A., Ardekani, K., Zhang, K., Ren, F., Negulescu, M., McGrenere, J. (2015). Twist and Pulse: Ephemeral Adaptation to Improve Icon Selection on Smartphones. In Proc. of GI'2015, 219–222.
136. Puerta, A., Eisenstein, J. (2001). XIIML: A Universal Language for User Interfaces. Whitepaper.
137. Quan, W., Jianmin, H. Personalized recommendation services based on service-

- oriented architecture. In Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing, pp. 356–361, Washington, DC, USA, 2006. IEEE Computer Society.
138. Raptis, D., Tselio, N., Kjeldskov, J., Skov, M. B. (2013). Does size matter?: investigating the impact of mobile phone screen size on users' perceived usability, effectiveness and efficiency. In Proc. of MobileHCI'2013, 137–136.
139. Rekimoto J., Nagao, K. The World through the Computer: Computer Augmented Interaction with Real World Environments, Actes du symposium UIST'95, 1995.
140. Rivadeneira, A.W., Gruen, D.M., Muller, M.J., Millen, D.R. (2007). Getting Our Head in the Clouds: Toward Evaluation Studies of Tagclouds. In Proc. of ACM Conf. on Human Aspects in Comp. Sys. (CHI'2007), 995–998.
141. Rogers, S. and Iba, W. (2000). Adaptive User Interfaces: Papers from the 2000 AAAI Symposium. AAAI Press, Technical Report SS-00-01.
142. Rosaci, R., Sarnè, G.M.L. (2006). MASHA: A multi-agent system handling user and device adaptivity of Web sites. *User Modeling and User-Adapted Interaction* 16, 5. pp. 435–462.
143. Roudaut, A., Bailly, G., Lecolinet, E., Nigay, L. (2009). Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. In Proc. of IFIP Int. Conf. on Human-Computer Interaction (INTERACT'2009), 616–619.
144. Ryan N., Pascoe J., Morse D., (1997). Enhanced Reality Fieldwork : the Context-Aware Archeological Assistant. In Dingwall, L., S. Exon, V. Gaffney, S. Laflin and M. van Leusen (eds.), *Archaeology in the Age of the Internet. CAA97. Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 25th Anniversary Conference, University of Birmingham, April 1997 (BAR International Series 750)*. Archaeopress, Oxford, pp. 269-274.
145. Samaan K., Tarpin-Bernard F. (2004). Task models and interaction models in a multiple user interfaces generation process. In Proceedings of TAMODIA 2004, Czech Republic, vol. 86, ACM Press, New York, NY, pp. 137-144.
146. Samp, K., Decker, S. Supporting menu design with radial layouts. In Proc. of AVI'2010, 155–162.
147. Schilit B.N., Adams N.I., Want R. (1994). *Context-Aware Computing Applications*.



- In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), IEEE Press, pp 85-90.
148. Schlee, M. and Vanderdonckt, J. (2004). Generative Programming of Graphical User Interfaces. In: Proceedings of 7<sup>th</sup> International Working Conference on Advanced Visual Interfaces, AVI'2004, Gallipoli, May 25-28, 2004. pp. 403-406. ACM Press, New York.
149. Schlienger, C., Conversy, S., Chatty, S., Anquetil, M. and Mertz, Ch. (2007). Improving Users' Comprehension of Changes with Animation and Sound: An Empirical Assessment. In: Proceedings of IFIP TC 13 Conference on Human-Computer Interaction, Interact'2007, Rio de Janeiro, September 10-14, 2007. Vol. 4662 of Lecture Notes in Computer Science. pp. 207–220, Springer, Berlin.
150. Schneider-Hufschmidt, M., Kühme, T. and Malinowski, U Eds.: Adaptive User Interfaces: Principles and Practise. Amsterdam: North Holland Elsevier. 1993.
151. Sears, A., Shneiderman, B. (1994). Split menus: Effectively using selection frequency to organize menus. *ACM Trans. on Comp.-Human Interaction* 1, 1, 27–51.
152. Seifert, C., Kump, B., Kienreich, W., Granitzer, G., Granitzer, M. On the beauty and usability of tag clouds. In *Proc. of Int. Conf. on Information Visualization (IV'2008)*, 17–25.
153. Sendín, M., López-Jaquero, V. and Collazos, C.A. (2008). Collaborative Explicit Plasticity Framework: a Conceptual Scheme for the Generation of Plastic and Group-Aware User Interfaces. *Journal of Universal Computer Science* 14, 9. pp. 1447–1462.
154. Sherman, R., Alpert, J.K., Karat, C., Carolyn, B. and Vergo, J. (2003). User attitudes regarding a user-adaptive e-commerce web site. *User Modeling and User-adapted Interaction* 13, 4. pp. 373–396.
155. Shin, Ch., Hong, J-H., Dey, A.K. (2012). Understanding and Prediction of Mobile Application Usage for Smart Phones. In *Proc. of UbiComp'2012*. 173–182.
156. Shin, Ch., Hong, J.H., and Dey, A.K. (2012). Understanding and prediction of mobile application usage for smart phones. *Proc. of UbiComp'2012 (Pittsburgh, September 5-8, 2012)*. ACM Press, New York, pp. 173–182.
157. Shneiderman, B. (2003). Promoting universal usability with multi-layer interface

- design. Proc. of the 2003 Conf. on Universal Usability CUA'2003 (Vancouver, November 10-11, 2003). ACM Press, New York, pp. 1–8.
158. Shaun K. Kane, Jacob O., Wobbrock Ian, E. Smith. Getting off the treadmill: evaluating walking user interfaces for mobile devices in public spaces. In *MobileHCI '08: Proceedings of the 10<sup>th</sup> International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 109–118, 2008.
159. Sischer, G. (2000). *User Modeling in Human Computer Interaction, User modeling and user-adapted interaction*, pp. 65-68, 11, Kluwer Academic Publishers.
160. Somberg, B.L. (1987). A comparison of rule-based and positionally constant arrangements of computer menu items. In *Proc. of CHI'87*, 255–260.
161. Stephanidis, C., Paramythis, A, Sfyarakis, M, Stergiou, A, Maou, N, Leventis, A, Paparoulis, G, and Karagiandidis, C. (1998). Adaptable and adaptive user interfaces for disabled users in AVANTI Project in Triglia, S., Mullery, A., Campolargo, M., Vanderstraeten, H. and Mampaey, M. (Eds) *Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services*, pp. 153-166, Springer.
162. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E. (2008). *EMF: Eclipse Modeling Framework*, Addison-Wesley Professional.
163. Tanvir, E., Cullen, J., Irani, P., Cockburn, A. (2008). AAMU: Adaptive Activation Area Menus for Improving Selection in Cascading Pull-Down Menus. In *Proc. of CHI'2008*. 1381–1384.
164. Tendjaoui M., Kolski C., Millot P. (1991). An approach towards the design of intelligent man-machine interfaces used in process control. *International Journal of Industrial Ergonomics*, 8, pp. 345-361.
165. Thevenin, D. and Coutaz, J. (1999). *Plasticity of User Interfaces: Framework and Research Agenda*. Proceeding of IFIP Conference on Human Computer Interaction, INTERACT.
166. Totterdell, P. and Rautenbach, P. (1990). Adaptation as a Problem Design. In: Browne, D., Totterdell, P., Norman, M. (Eds.), *Adaptive User Interfaces*. Computer And People Series, Academic Press, pp. 59–84.
167. Trigg, R., Moran, T. & Halasz, F. (1987). Adaptability and Tailorability in NoteCards,

- in Bullinger and Shackel (Eds.) Proc. INTERACT'87, North-Holland.
168. Tsandilas, T., Schraefel, M.C. 2005. An empirical assessment of adaptation techniques. In Proc. of CHI'2005 Extended Abstracts, 2009–2012.
169. Tsandilas, T. and Schraefel, M.C. (2007). Bubbling menus: a selective mechanism for accessing hierarchical drop-down menus. Proc. of the ACM SIGCHI Conf. on Human Factors in Computing Systems CHI'2007 (San Jose, April 28–May 03). ACM Press, New York, pp. 1195–1204.
170. Vanderdonckt, J. (1999). Computer-aided design of menu bar and pull-down menus for business oriented applications. Proc. of 6<sup>th</sup> Int. Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'99 (Braga, June 2-4, 1999), D.J. Duke, A.R. Puerta (Eds.). Springer, Vienna, pp. 73–88.
171. Vanderdonckt J., Grolaux D., Van Roy P., Limbourg Q., Macq B., Michel B. (2005). A Design Space For Context-Sensitive User Interfaces. In Proceedings of IASSE 2005.
172. Vermeulen, J. (2010). Improving Intelligibility and Control in Ubicomp. In Proceedings of the 12<sup>th</sup> ACM Interaction Conference on Ubiquitous Computing-Adjunct Papers, Ubicomp'10 Adjunct, pp. 485-488, New York, NY, USA, 2010. ACM. ISBN 18.978-1-4503-0283-8.
173. Weibelzahl, S. (2002). Evaluation of Adaptive Systems, Graduate Programme Human and Machine Intelligence, Kunzenweg 21, University of Education Freiburg, Germany.
174. Weiser M. The computer for the 21st century, Scientific American, 265(3), 1991, pp. 94-104.
175. Wiebe, M., Geiskkovitch, D.Y., Bunt, A. (2016). Exploring User Attitudes Towards Different Approaches to Command Recommendation in Feature-Rich Software. In Proc. of 21<sup>st</sup> Int. Conf. on Intelligent User Interfaces (IUI'2016). 43-47.
176. Weld, D.S., Anderson, C., Domingos, P., Etzioni, O., Gajos, K., Lau, T., Wolfman, S. Automatically personalizing user interfaces. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2003), August 2003.

177. Winograd, T. "Shifting viewpoints: artificial intelligence and human-computer interaction". *Artificial Intelligence* 170(18):1256–1258, December 2006. Elsevier.
178. Yu, E. (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: *Proceedings of the 3<sup>rd</sup> IEEE International Symposium on Requirements Engineering, RE'97*, Washington, January 6-8, 1997. pp. 226-235, IEEE Computer Society Press, Los Alamitos.
179. Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In *Proc. of ACM Conf. on Human Factors in Computing Systems (CHI'2007)*. ACM Press, New York, NY, 1395–1404. DOI= <http://dx.doi.org/10.1145/1240624.1240836>

## Liste des tableaux

Tableau I.	Comparaisons deux à deux (Step by Step, Shortcut, Contrôle) pour la vitesse de sélection (P+ signifie prédiction correcte et P- signifie prédiction incorrecte). .....	vi
Tableau II.	Comparaisons deux à deux (Step by Step, Shortcut, Contrôle) pour les erreurs (P+ signifie prédiction correcte et P- signifie prédiction incorrecte).....	vii
Tableau III.	Paire de comparaisons (Graphique, Gestuelle, Vocale, Mixte) pour les temps de sélection (P+ signifie prédiction correcte et P- signifie prédiction incorrecte). .....	viii
Tableau IV.	Paire de comparaisons (Graphique, Gestuelle, Vocale, Mixte) pour le taux d'erreurs (P+ signifie prédiction correcte et P- signifie prédiction incorrecte). .....	viii

## Liste des figures

Figure 1.	Une configuration d'adaptation selon Dieterich. ....	26
Figure 2.	Etapas d'exécution et d'évaluation selon Norman.....	27
Figure 3.	<i>Framework</i> ISATINE pour l'adaptation des IHM (extrait de Lopez-Jaquero <i>et al.</i> , 2007).	28
Figure 4.	Exemples d'interfaces adaptatives.....	42
Figure 5.	(a) Dimension Perception-Décision-Action, (b) Boucle d'interaction. ....	48
Figure 6.	Modèle d'adaptation du système pour l'interaction.....	49
Figure 7.	Modèle AIMS côté système.....	50
Figure 8.	Modèle AIMS côté utilisateur.....	51
Figure 9.	Les plans Interaction et Adaptation du modèle AIMS.....	52
Figure 10.	Ajustement des dimensions PDA système et PDA utilisateur.....	53
Figure 11.	Ajustement à la tâche.....	54
Figure 12.	Caractère auto-descriptif.....	54
Figure 13.	Contrôle utilisateur.....	55
Figure 14.	Conformité aux attentes de l'utilisateur.....	55
Figure 15.	Tolérance aux erreurs.....	56
Figure 16.	Aptitude à l'individualisation. ....	56
Figure 17.	Facilité d'apprentissage. ....	57
Figure 18.	Description des conséquences générées par une adaptation inappropriée.....	58
Figure 19.	Ajustement des dimensions ADP système et ADP utilisateur.....	59
Figure 20.	Critères de qualité au sein du modèle de l'interaction adaptative AIMS. ....	63
Figure 21.	PCA modèle de présentation de l'adaptation.....	75
Figure 22.	Caractérisation PCA du menu basé sur la fréquence.....	76
Figure 23.	Caractérisation PCA de la technique Split avec réplication. ....	77
Figure 24.	Caractérisation PCA des techniques basées sur le changement de taille.....	78
Figure 25.	Caractérisation PCA de la technique Highlighting.....	79
Figure 26.	Caractérisation PCA de la technique du menu temporel. ....	80
Figure 27.	Caractérisation PCA de la technique d'adaptation éphémère.....	81

Figure 28.	Caractérisation PCA comparative des techniques adaptatives. ....	82
Figure 29.	Conception d'un menu en se basant sur quelques variables visuelles. ....	84
Figure 30.	Conception d'un nouveau menu avec ajout de la fenêtre de prédiction. ....	85
Figure 31.	Axes explorés dans l'adaptation éphémère avec disparition. ....	91
Figure 32.	Technique ICD appliquée à des menus sur <i>smartphone</i> . ....	92
Figure 33.	Technique OCD en deux étapes. ....	93
Figure 34.	Vue globale des différentes conditions de l'expérimentation. ....	94
Figure 35.	Temps de sélection pour toutes les conditions de l'expérimentation. ....	100
Figure 36.	Caractérisation PCA de l'adaptation éphémère avec disparition contrôlée par l'utilisateur. ....	106
Figure 37.	Prototype Disappearing (D). ....	107
Figure 38.	Prototype Disappearing + User control (DU). ....	108
Figure 39.	Prototype User control (U). ....	109
Figure 40.	Condition Contrôle. ....	110
Figure 41.	(a) Condition Ordre. (b) Condition Probabiliste. ....	111
Figure 42.	Conditions de l'étude expérimentale. ....	114
Figure 43.	Temps de sélection lorsque la prédiction est correcte. ....	116
Figure 44.	Temps de sélection lorsque la prédiction est incorrecte. ....	117
Figure 45.	Taux d'erreur pour toutes les conditions. ....	117
Figure 46.	Préférences utilisateur pour toutes les conditions. ....	118
Figure 47.	Caractérisation PCA du concept des menus Cloud. ....	123
Figure 48.	Nuage de mots circulaire avec items prédits (menu Cloud déplié) / liste complète des items (menu Cloud replié). ....	125
Figure 49.	Menu Linéaire6 avec six items prédits (menu Linéaire déplié) / liste complète des items (menu Linéaire replié). ....	126
Figure 50.	Menu Linéaire3 avec trois items prédits. ....	126
Figure 51.	Conditions expérimentales. ....	129
Figure 52.	Temps de sélection des conditions Cloud, Linéaire3, Linéaire6 et Contrôle. ....	132
Figure 53.	Taux d'erreur pour toutes les conditions. ....	133
Figure 54.	Axes explorés par les techniques step-by-step et shortcut. ....	140
Figure 55.	Vue conceptuelle du prototype step-by-step. ....	145

Figure 56.	Vue utilisateur du prototype step-by-step. ....	145
Figure 57.	Vue conceptuelle du prototype shortcut. ....	146
Figure 58.	Vue utilisateur du prototype shortcut.....	147
Figure 59.	Vue conceptuelle du prototype Contrôle. ....	148
Figure 60.	Vue utilisateur du prototype Contrôle.....	148
Figure 61.	Step-by-step en cas de prédiction incorrecte. ....	151
Figure 62.	Temps de sélection (secondes) pour toutes les conditions.....	154
Figure 63.	T d'erreur pour toutes les conditions. ....	155
Figure 64.	Axes explorés dans le menu Polymodal. ....	161
Figure 65.	Répartition des modalités de rendus (œil et haut-parleur) et commande (pointeur, geste, micro). ....	164
Figure 66.	Temps de sélection pour les conditions Graphique, Gestuelle, Vocale et Mixte. 169	
Figure 67.	Taux d'erreur pour les conditions Graphique, Gestuelle, Vocale et Mixte. ...	170
Figure 68.	Synthèse des dimensions explorées dans les sept techniques proposées. ....	178



## Liste des acronymes

IHM : Interaction Homme-Machine

IU : Interface Utilisateur

IUG : Interface Utilisateur Graphique

AIMS : Adaptive Interaction Model with Symmetry

PDA : Perception – Décision – Action

ADP : Apprentissage – Décision – Prédiction

PCA : Perception – Cognition – Action

ICD : In Context Disappearing

OCD : Out of Context Disappearing

D : Disappearing

DU : Disappearing + User control

U : User control

## Annexe

Menu	Moyenne	Ecart type	Mesure	Valeur
Contrôle	14.73	4.12	-3.18	0.001
Step By Step (P+)	6.42	2.82		
Contrôle	14.73	4.12	-3.18	0.001
Shortcut (P+)	1.77	1.47		
Step By Step (P+)	6.42	2.81	-3.18	0.001
Shortcut (P+)	1.77	1.47		
Contrôle	14.73	4.12	-0.14	0.888
Step By Step (P-)	14.51	3.67		
Contrôle	14.73	4.12	-2.34	0.019
Shortcut (P-)	20.94	8.06		
Step By Step (P-)	14.51	3.67	-3.18	0.001
Shortcut (P-)	20.94	8.06		

Tableau I. Comparaisons deux à deux (Step by Step, Shortcut, Contrôle) pour la vitesse de sélection (P+ signifie prédiction correcte et P- signifie prédiction incorrecte).

Menu	Moyenne	Ecart type	Mesure	Valeur
Contrôle	4.31	4.37	-2.83	0.004
Step By Step (P+)	0.23	0.56		
Contrôle	4.31	4.37	2.94	0.003

Shortcut (P+)	0	0		
Step By Step (P+)	0.23	0.56	1.34	0.179
Shortcut (P+)	0	0		
Contrôle	4.31	4.37	0.07	0.944
Step By Step (P-)	4.46	2.93		
Contrôle	4.31	4.37	1.85	0.064
Shortcut(P-)	13.46	15.99		
Step By Step (P-)	4.46	2.93	-2.34	0.018
Shortcut (P-)	13.46	15.99		

Tableau II. Comparaisons deux à deux (Step by Step, Shortcut, Contrôle) pour les erreurs (P+ signifie prédiction correcte et P- signifie prédiction incorrecte).

Menu	Moyenne	Ecart type	Mesures	Valeur
<b>Graphique(P+)</b>	1.461	0.366	3.18	0.005
<b>Gestuelle (P+)</b>	2.969	0.9		
<b>Graphique(P+)</b>	1.461	0.366	3.18	0.005
<b>Vocale (P+)</b>	4.438	0.876		
<b>Graphique(P+)</b>	1.461	0.366	3.18	0.005
<b>Mixte (P+)</b>	4.418	1.154		
<b>Gestuelle(P+)</b>	2.969	0.9	3.04	0.005
<b>Vocale (P+)</b>	4.438	0.876		
<b>Gestuelle(P+)</b>	2.969	0.9	2.48	0.05
<b>Mixte (P+)</b>	4.418	1.154		
<b>Vocale(P+)</b>	4.438	0.876	-0.63	0.5
<b>Mixte (P+)</b>	4.418	1.154		
<b>Graphique(P-)</b>	4.1	0.886	3.18	0.005
<b>Gestuelle (P-)</b>	6.484	1.643		
<b>Graphique(P-)</b>	4.1	0.886	-2.76	0.01
<b>Vocale (P-)</b>	3.115	0.883		
<b>Graphique(P-)</b>	4.1	0.886	3.06	0.005

<b>Mixte (P-)</b>	6.577	1.978		
<b>Gestuelle(P-)</b>	6.484	6.484	-3.18	0.005
<b>Vocale (P-)</b>	3.115	0.883		
<b>Gestuelle(P-)</b>	6.484	1.643	0.42	0.5
<b>Mixte (P-)</b>	6.577	1.978		
<b>Vocale(P-)</b>	3.115	6.577	3.06	0.005
<b>Mixte (P-)</b>	0.883	1.978		

Tableau III. Paire de comparaisons (Graphique, Gestuelle, Vocale, Mixte) pour les temps de sélection (P+ signifie prédiction correcte et P- signifie prédiction incorrecte).

Menu	Moyenne	Ecart type	Mesures	Valeur
<b>Graphique (P+)</b>	0.17	0.39		
<b>Gestuelle (P+)</b>	0	0		
<b>Vocale (P+)</b>	0.08	0.29		
<b>Mixte (P+)</b>	0.33	0.65		
<b>Graphique (p-)</b>	0.33	1.73	2.50	0.05
<b>Gestuelle (P-)</b>	2.08			
<b>Gestuelle (P-)</b>	2.08	1.73	2.67	0.01
<b>Vocale (P-)</b>	0	0		
<b>Gestuelle (P-)</b>	20.8	1.73	1.92	0.1
<b>Mixte (P-)</b>	0.75	0.96		
<b>Vocale (P-)</b>	0	0	2.26	0.05
<b>Mixte (P-)</b>	0.75	0.96		

Tableau IV. Paire de comparaisons (Graphique, Gestuelle, Vocale, Mixte) pour le taux d'erreurs (P+ signifie prédiction correcte et P- signifie prédiction incorrecte).

Le questionnaire formulé pour l'expérimentation du menu Polymodal

Q1 : En général, je suis satisfait(e) de la facilité d'utilisation de ce prototype

Q2 : Ce prototype est simple à utiliser

Q3 : J'ai accompli les tâches demandées correctement en utilisant ce prototype

Q4 : Je me sens à l'aise avec ce prototype

Q5 : J'ai eu de la facilité à apprendre comment utiliser ce prototype

Q6 : Je crois être devenu(e) rapidement efficace en utilisant ce prototype

Q7 : L'organisation de l'information dans les écrans du système est claire

Q8 : L'interface de ce système est agréable

Q9 : En général, je suis satisfait(e) de ce système

Chaque question a été mesurée en utilisant une échelle de Likert à 5 points.

1 = fortement en désaccord, 2 = en désaccord, 3 = neutre, 4 = d'accord, 5 = fortement d'accord.

Six questions ont été destinées à mesurer la facilité d'utilisation perçue (Q1, Q2, Q4, Q7-9) et trois questions pour mesurer l'utilité perçue (Q3, Q5, Q6). Les questions ont été localisées aléatoirement dans le questionnaire.

