



HAL
open science

Contribution à l'évaluation des systèmes d'Information personnalisés

Makram Soui

► **To cite this version:**

Makram Soui. Contribution à l'évaluation des systèmes d'Information personnalisés. Informatique [cs]. Université de Valenciennes et du Hainaut-Cambrésis, 2010. Français. NNT : 2010VALE0006 . tel-03003893

HAL Id: tel-03003893

<https://uphf.hal.science/tel-03003893v1>

Submitted on 13 Nov 2020

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.



Numéro d'ordre : 10/09

THÈSE

Présentée à

l'Université de Valenciennes et du Hainaut-Cambrésis

en vue de l'obtention du grade de

DOCTEUR

Spécialité Automatique et Informatique des Systèmes Industriels et Humains

Mention Informatique

Par

Makram Soui

(Maitre ès Sciences)

*Contribution à l'évaluation des
systèmes d'information personnalisés*

Application au transport collectif de personnes

Soutenue le 9 février 2010 devant le jury composé de :

Franck Poirier
Faouzi Moussa
Mourad Abed
Christophe Kolski
Khaled Ghedira
Philippe Trigano
Guillaume Uster

Professeur à l'Université de Bretagne Sud
Maître de Conférences à Université de Monastir
Professeur à l'Université de Valenciennes
Professeur à l'Université de Valenciennes
Professeur à l'Université de Tunis
Professeur à l'UTC de Compiègne
Chargé de Recherche, INRETS, Villeneuve d'Ascq

Rapporteur
Rapporteur
Co-directeur
Co-encadrant
Co-directeur
Président du jury
Invité



Avant Propos

Le travail présenté dans ce mémoire a été réalisé au Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines (LAMIH) de l'Université de Valenciennes et du Hainaut-Cambrais (UVHC), au sein de l'équipe Raisonnement Automatique et Interaction Homme-Machine (RAIHM), dirigée par le Professeur Christophe Kolski, en collaboration avec l'unité de recherche de Stratégies d'Optimisation des Informations Et de la connaissance (SOIE) de l'université de Tunis, dirigée par le Professeur khaled Ghedira.

Je tiens, tout d'abord, à remercier chaleureusement le Professeur Christophe Kolski, qui m'a accueilli dans son équipe de recherche, guidé, conseillé et encouragé avec enthousiasme et rigueur durant ma thèse. Je lui suis très reconnaissant pour ses apports toujours originaux à nos discussions. Sans lui cette thèse n'aurait pas pu voir le jour.

Je tiens également à exprimer toute ma reconnaissance à Monsieur Khaled Ghedira, pour l'aide et les conseils qu'il m'a apportés depuis mon travail de mastère. Cette thèse n'aurait pas pu avoir lieu sans ses encouragements et ses commentaires toujours forts pertinents concernant mon travail de recherche.

Je suis très reconnaissant aussi envers Monsieur Mourad Abed, Professeur à l'UVHC, pour soutien, son inventivité, sa grande culture scientifique, et pour les judicieux conseils qu'il m'a donnés lors de nos réunions de travail. Je lui exprime ma profonde gratitude et mon plus grand respect pour m'avoir permis de travailler sur un sujet passionnant et innovant.

Pour m'avoir fait l'honneur d'accepter de participer au jury de cette thèse, je remercie vivement Messieurs Franck Poirier, Professeur à l'Université de Bretagne Sud et Faouzi Moussa, Maître de Conférences à l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM) pour m'avoir fait l'honneur d'être rapporteurs de ce mémoire. Je les remercie infiniment pour le temps consacré à cet effet en dépit de toutes les responsabilités qu'ils ont. Leur expertise reconnue dans le domaine des IHM donne un relief particulier à leur jugement sur mes travaux.

Je remercie également Monsieur Philippe Trigano, Professeur à l'Université de Technologie de Compiègne (UTC) et responsable de la Cellule TICE (Technologies de l'Information et de la Connaissances pour l'Enseignement) d'avoir accepté d'examiner mon travail. Mes remerciements vont également à Guillaume Uster, chargé de recherche à l'Institut National de REcherche sur les Transports et leur Sécurité (INRETS), pour ses avis en termes d'information transport.

Je remercie chaleureusement tous les membres du LAMIH et de mon équipe RAIHM pour leur accueil, les échanges et les bons moments que nous avons passés ensemble. Je voudrais aussi remercier tous les membres et partenaires du projet Mouver.Perso et ANR Viatic.Mobilité. Nos réunions étaient enrichissantes et productives.

Enfin, je remercie infiniment mon père Taher qui nous a quittés pour un monde meilleur pour son soutien, ses encouragements tout au long de mes études. J'exprime aussi ma profonde reconnaissance à ma famille et mes amis pour m'avoir soutenu dans ce nouveau projet professionnel.

*A mes parents,
A ma femme,
A ma famille,
A mes amis.*

Table des matières

Avant propos.....	2
Table des matières	4
Liste des figures	7
Liste des tableaux	10
Introduction générale	11
Chapitre 1 Personnalisation dans les systèmes d'information	14
Introduction	15
1.1. Personnalisation : concepts et méthodes	15
1.1.1. Définitions	15
1.1.2. Acquisition d'information sur l'utilisateur	17
1.1.3. Modélisation de l'utilisateur	18
1.1.4. Les méthodes de personnalisation existantes	19
1.2. Etude des systèmes de personnalisation existants	33
1.2.1. Critère d'étude des systèmes d'information personnalisée	33
1.2.2. Systèmes de personnalisation existants.....	33
1.2.3. Synthèse et discussion.....	35
1.3. Contexte applicatif : la personnalisation de l'information transport	35
1.3.1. Apport de la personnalisation dans les systèmes d'informations au profit des usagers des transports collectifs de personnes	36
1.3.2. Systèmes d'informations personnalisés existants dans le domaine de transport.	36
1.3.3. Synthèse et discussion.....	38
Conclusion.....	39
Chapitre 2 De l'évaluation des systèmes interactifs à celle des systèmes d'information personnalisés	40
Introduction	41
2.1. L'évaluation : définitions et principes	41
2.1.1. Définitions	43
2.1.2. Les dimensions et les critères d'évaluation existants	43
2.1.3. Les modèles de développement enrichis sous l'angle de l'évaluation des systèmes	45
2.1.4. Synthèse	46
2.2. Méthodes, techniques et outils d'évaluation des systèmes interactifs	46
2.2.1. Les critères proposés pour l'évaluation de système d'information personnalisé.....	47
2.2.2. Choix d'une classification des méthodes d'évaluation des systèmes interactif	49
2.2.3. Etude de méthodes, techniques et outils d'évaluation des systèmes interactifs	50
2.2.4. Synthèse et discussion.....	65
Conclusion.....	67

Chapitre 3 Contribution à l'évaluation des systèmes d'information personnalisés	68
Introduction	69
3.1. Utilisation d'IDEF0 pour la représentation des activités constitutives des méthodes proposées	69
3.2. La Méthode MetQus	70
3.2.1. Principe et description	70
3.2.2. Synthèse et discussion	73
3.3. La Méthode MetTra	73
3.3.1. Principe et description	75
3.3.2. Synthèse et discussion	92
3.4. La Méthode MetSim	92
3.4.1. Principe et description	92
3.4.2. Synthèse et discussion	97
Conclusion	98
Chapitre 4 Mise en oeuvre des méthodes d'évaluation proposées dans le domaine du transport	99
Introduction	100
4.1. Le projet MOUVER.PERSO	100
4.2. Contextes d'usage envisagés dans le cadre de l'évaluation des systèmes d'information personnalisés	101
4.2.1. La prise en compte de l'utilisateur	101
4.2.2. La prise en compte de l'environnement	102
4.2.3. La prise en compte de la plate-forme	103
4.3. Premier cas d'étude : application de la méthode MetQus	104
4.3.1. Présentation de l'application	104
4.3.2. Mise en oeuvre de MetQus	105
4.3.3. Synthèse et discussion	112
4.4. Second cas d'étude : application de la méthode MetTra	113
4.4.1. Présentation de l'application	113
4.4.2. Mise en oeuvre de MetTra	114
4.4.3. Synthèse et discussion	128
4.5. Troisième cas d'étude : application de la méthode MetSim	129
4.5.1. Présentation de l'application	129
4.5.2. Mise en oeuvre de MetSim	129
4.5.3. Synthèse et discussion	134
Conclusion	134
Chapitre 5 Perspectives de recherche	136
Introduction	137

Table des matières : _____

5.1. Perspectives relatives à la méthode MetQus	137
5.2. Perspectives relatives à la méthode MetTra	139
5.3. Perspectives relatives à la méthode MetSim	144
Conclusion	145
Conclusion générale	146
Bibliographie générale	149
Annexe A Le projet Mouver.Perso.....	165
Annexe B Outils logiciels utilisés dans le cadre de notre travail	166

Liste des figures

Figure 1.1. Arbre de décision.....	24
Figure 1.2. Exemple lié au processus de développement d'ARTStudio pour la production d'interfaces multiplateformes	27
Figure 1.3. Architecture de MOBI-D [Puerta et Eisenstein 98].....	28
Figure 1.4. Modèle simplifié du framework associé aux interfaces graphiques en UsiXML [Francisco et al., 08]	29
Figure 1.5. Architecture globale de l'approche Amacont [Hinz et Fiala 04]	30
Figure 1.6. Méthode globale de génération des IHM plastiques.....	31
Figure 2.1. Principe de base de l'évaluation inspiré de [Senach 90 ; Grislin 95]	42
Figure 2.2. Classification des critères et sous critères ergonomiques selon [Bastien et Scapin, 93]	44
Figure 2.3. Exemples de Modèles enrichis sous l'angle de l'interaction homme-machine	46
Figure 2.4. Classification des méthodes d'évaluation d'interfaces homme-machine proposée par [Huart, Kolski et Bastien 08]	49
Figure 2.5. Interaction directe par interview entre utilisateur et évaluateur	50
Figure 2.6. Principe du mouchard électronique	52
Figure 2.7. Méthode d'évaluation MultiDevice RemUSINE [Paterno et al., 07]	53
Figure 2.8. Règles les plus couramment citées "Top ten" de [Nielsen 96]	54
Figure 2.9. Extrait d'une grille d'évaluation (inspirée de [Ravden et Johnson 89])	56
Figure 2.10. Les niveaux d'une description CLG	58
Figure 2.11. Méthode d'évaluation USINE en utilisant le modèle de tâches CTT, traduite de [Lecerof et Paterno 98].....	61
Figure 2.12. Méthode d'évaluation WebRemUSINE, traduit de [Paganelli et Paternò, 02 ; 03]	62
Figure 2.13. Le prototypage interactif, vu du côté de l'utilisateur de la simulation. Il consiste en l'élaboration d'un SI à travers une évaluation permanente de solutions fonctionnelles simulées [Le Bodic et al., 04]	63
Figure 3.1. Diagramme d'activité en IDEF0	69
Figure 3.2. Les phases de la méthode MetQus	70
Figure 3.3. Activité de test	71
Figure 3.4. Structure du support d'évaluation	71
Figure 3.5. Evaluation de l'utilisation selon les critères proposés (Cas où le sujet est assisté par l'évaluateur)	72
Figure 3.6. Framework de l'approche MUSETTE	73
Figure 3.7. La méthodologie PerMet [Anli 06]	74
Figure 3.8. Les différentes phases de la méthode MetTra.	76
Figure 3.9. Intégration de mécanisme de trace	77
Figure 3.10. La structure d'une fiche de description textuelle d'un cas d'utilisation	79
Figure 3.11. Représentation graphique d'une tâche	79
Figure 3.12. Relations inter-tâches inspirées de la notation CTT (ConcurTaskTrees) [Paternò et al., 97]	80
Figure 3.13. Les types des tâches inspirées de la notation CTT (ConcurTaskTrees) [Paternò et al., 97].....	80
Figure 3.14. Représentation graphique d'une classe	81
Figure 3.15. Modèle des objets interactifs	82
Figure 3.16. Structure du modèle de dialogue	83

Figure 3.17. Recueil des données relatives au comportement de l'utilisateur et réactions des interfaces du système	84
Figure 3.18. Exemple d'extrait d'un fichier de trace sous format XML	85
Figure 3.19. Capture d'écran de l'interface de l'activité de capture des données - exemple ...	85
Figure 3.20. Algorithme de génération du modèle de dialogue à partir d'un fichier texte	86
Figure 3.21. Une partie de la capture d'écran affichant le résultat d'analyse - exemple	87
Figure 3.22. Evolution de taux d'erreur moyen en fonction de nombres de requêtes.....	90
Figure 3.23. Une note graphique UML pour la modélisation d'un problème	90
Figure 3.24. Une note graphique UML pour la modélisation d'une recommandation	91
Figure 3.25. Modélisation avec UML de la tâche prescrite et la procédure proposée par l'évaluateur	91
Figure 3.26. Les phases de la méthode MetSim.....	92
Figure 3.27. Proposition des hypothèses	93
Figure 3.28. Identification des problèmes rencontrés	94
Figure 3.29. Exemple d'une règle d'évaluation décrit en XML	94
Figure 3.30. Exemple d'un conteneur décrit en UsiXML	95
Figure 3.31. Analyse des résultats.....	96
Figure 4.1. Exemple des interfaces de l'application à évaluer.....	104
Figure 4.2. Dynamique de la tâche d'ajout d'un RDV	106
Figure 4.3. Questionnaire préalable	107
Figure 4.4. Evaluation de l'utilisation selon les critères proposés.....	107
Figure 4.5. Interfaces sur PDA.....	108
Figure 4.6. Test de la maquette.....	108
Figure 4.7. Un exemple de fiche d'évaluation.....	109
Figure 4.8. Avis des sujets vis-à-vis de la personnalisation liée au contenu	110
Figure 4.9. Moyennes, par critères de personnalisation liée au contenu, des niveaux de satisfaction des étudiants	110
Figure 4.10. Avis des étudiants vis-à-vis de la personnalisation liée au contenant	111
Figure 4.11. Test avec PDA.....	111
Figure 4.12. Test avec PC.....	111
Figure 4.13. Moyennes, par critère de personnalisation liée au contenant, des niveaux de satisfaction des sujets.....	111
Figure 4.14. Répartition des problèmes en fonction du critère évalué	112
Figure 4.15. Exemple d'interface associé au service de recherche d'itinéraire	113
Figure 4.16. Diagramme des cas d'utilisation du service de recherche d'itinéraire personnalisé	114
Figure 4.17. Fiche de description textuelle pour le cas d'utilisation "Rechercher un itinéraire"	115
Figure 4.18. Le modèle statique pour la tâche « recherche d'itinéraire »	116
Figure 4.19. Diagramme de classe pour le service de « recherche d'itinéraire »	117
Figure 4.20. Modèle des objets interactifs pour la tâche de « recherche d'itinéraire »	118
Figure 4.21. Conception générique pour le service de « recherche d'itinéraire »	118
Figure 4.22. Modèle de dialogue du service « recherche d'itinéraire »	119
Figure 4.23. Diagramme de séquences pour la recherche d'itinéraire personnalisé.....	120
Figure 4.24. Une méthode de définition du champ de texte destination	120
Figure 4.25. Exemple d'extrait d'un fichier de trace sous format XML.....	122
Figure 4.26. Un exemple d'une interface à évaluer associée au service de recherche d'itinéraire.....	123
Figure 4.27. Modélisation avec UML de la tâche prescrite et de la procédure proposée par l'évaluateur	124

Figure 4.28. Les modèles de dialogue de l'utilisateur lors de l'utilisation d'un PC et PDA..	125
Figure 4.29. Evolution de taux d'erreur moyen en fonction de nombres de requêtes.....	127
Figure 4.30. Evolution du coefficient de corrélation de Pearson en fonction du nombre de requêtes	128
Figure 4.31. Exemple d'interface relative à l'affichage des prochains déplacements	129
Figure 4.32. La spécification concrète de l'interface après l'adaptation	130
Figure 4.33. Les réactions réelles (partie gauche) de l'interface du nouveau cas et les réactions du cas similaire antérieur (partie droite)	131
Figure 4.34. Exemple d'adaptation d'une application à la vitesse de déplacement d'un utilisateur : déplacement normal, luminosité normale (a) et déplacement rapide, luminosité faible (b)	132
Figure 4.35. Les réactions réelles (partie à gauche) de l'interface et les réactions prévues par l'évaluateur (partie à droite)	133
Figure 5.1. Architecture de la méthode MetQus étendue	138
Figure 5.2. Fonctionnement du futur module d'analyse automatique de modèle de dialogue	140
Figure 5.3. Comportement de l'agent Critère	142
Figure 5.4. Comportement de l'agent Evalueur	143
Figure A. Exemple d'interfaces homme-machine de l'application.....	165
Figure B. Interface homme-machine de NetBeans	166
Figure C. Exemple d'extrait d'un fichier de trace décrit en XML.....	172
Figure D. Les différents modèles d'IHM dans UsiXML.....	173
Figure E. Exemples d'interface homme-machine de l'outil GrafiXML	170

Liste des tableaux

Tableau 1.1. Classification des méthodes de personnalisation selon le contexte utilisé et le type de filtrage adaptée de [Cinquin et al., 02]	26
Tableau 1.2. Approches d'adaptation d'IHM multiplateforme et/ou orientée contexte	32
Tableau 1.3. Etude comparative des systèmes de personnalisation	35
Tableau 1.4. Etude comparative de systèmes de personnalisation représentatifs dans le domaine du transport	38
Tableau 2.1. Synthèse de l'adaptabilité des méthodes à l'évaluation de système personnalisé	66
Tableau 3.1. Rangs des critères par rapport aux solutions d'une requête	89
Tableau 3.2. Formalisme utilisé pour décrire un problème	97
Tableau 3.3. Formalisme utilisé pour la description de recommandations	97
Tableau 3.4. Etude comparative des méthodes d'évaluation proposées.....	98
Tableau 4.1. Facettes du contexte relatif à l'utilisateur–exemples représentatifs.....	102
Tableau 4.2. Facettes du contexte relatif à l'environnement–exemples représentatifs	103
Tableau 4.3. Facettes du contexte relatif à la plate-forme–exemples représentatifs.....	103
Tableau 4.4. Détail des sujets participant à l'évaluation.....	106
Tableau 4.5. Structure du support d'évaluation.....	107
Tableau 4.6. Définition du contexte d'évaluation.....	121
Tableau 4.7. Les attributs et les règles d'évaluation exploités dans le cadre de l'évaluation statique	123
Tableau 4.8. Exemples d'adaptations représentationnelles lors de l'utilisation d'un PC et d'un PDA.	123
Tableau 4.9. Erreur de prédiction moyenne en fonction du nombre requêtes.....	126
Tableau 4.10. Coefficient de corrélation de Pearson en fonction du nombre requêtes.....	127
Tableau 4.11. Description des problèmes détectés	131
Tableau 4.12. Exemple de description d'une recommandation	132
Tableau 4.13. Description des problèmes détectés	133
Tableau 4.14. Exemple de description des recommandations	134

Introduction générale

La percée des nouvelles technologies de l'information et de la communication permet d'envisager de nouveaux Systèmes d'Information Personnalisés (SIP). De tels systèmes possèdent de plus en plus différentes capacités d'adaptation à leur contexte d'usage, particulièrement concernant la plate-forme d'interaction utilisée, l'utilisateur concerné et l'environnement dans lequel celui-ci se situe. Ils peuvent être disséminés dans l'environnement de l'utilisateur. Les usagers de ces systèmes sont très divers, de par leur âge, connaissances, compétences, centres d'intérêts, besoins, préférences, etc. Parmi ces usagers, on trouve aussi bien ceux du grand public, que ceux de domaines professionnels très ciblés (santé, formation, commerce...). Les dispositifs d'interactions peuvent être également très variés (PC, PDA, téléphones portables, ...). Les caractéristiques de l'environnement de l'utilisateur peuvent s'avérer d'une grande variété.

Avec l'apparition des SIP, de nombreux travaux sont menés aussi bien dans le monde académique que dans celui des entreprises sur la modélisation de l'utilisateur, les méthodes de conception de SIP, les outils de construction d'interfaces adaptées, les algorithmes de personnalisation, etc. Cependant, l'évaluation de ces systèmes a été pour l'instant relativement peu étudiée. Ainsi, de nouvelles problématiques d'évaluation de ceux-ci apparaissent. Parmi les problèmes les plus traités actuellement nous pouvons citer : le manque de méthodes d'évaluation dédiées aux systèmes d'information personnalisés, l'absence d'outils (automatiques ou non) d'aide à l'évaluation destinés aux SIP. Pour combler ce manque, il s'agit d'étudier et mettre en place une batterie de méthodes permettant d'évaluer et d'améliorer leur qualité de personnalisation.

L'évaluation des systèmes interactifs permet de mettre l'accent sur les défauts de conception ; elle peut aussi aider les concepteurs à améliorer les systèmes interactifs déjà conçus en facilitant la détection de leurs problèmes et défauts. La littérature en Interaction Homme-Machine est extrêmement riche de méthodes et outils d'évaluation susceptibles de détecter les problèmes de l'interface utilisateur, à comprendre les difficultés que les utilisateurs rencontrent lorsqu'ils interagissent avec les systèmes interactifs, et à améliorer les IHM. Cependant ces méthodes se focalisent sur l'utilité et l'utilisabilité du système qui sont largement étudiées dans la littérature. Par contre, la notion de personnalisation est généralement négligée, voire bâclée malgré le besoin croissant des systèmes d'information personnalisés. Nous nous situons dans ce contexte.

Dans ce courant de recherche, le travail présenté dans ce mémoire est une contribution à l'évaluation des SIP, l'évaluation étant basée sur le contexte d'usage. Ainsi, différents contextes d'usage peuvent être envisagés pour le SIP ; ces contextes doivent être considérés afin d'évaluer ces systèmes. Nous proposons, dans ce cadre, trois méthodes complémentaires d'évaluation des SIP visant à améliorer leur qualité de personnalisation.

Ce mémoire, organisé en cinq chapitres, débute par un chapitre intitulé " Personnalisation dans les systèmes d'information". Ce chapitre présente un état de l'art des travaux relatifs à la personnalisation. Il explicite les principaux termes employés dans la littérature autour de la personnalisation et expose les principes, approches et méthodes utilisées pour la personnalisation. Ce chapitre présente aussi les principes de plusieurs systèmes d'information transport existants étudiés sous l'angle de l'évaluation.

Le second chapitre, appelé "De l'évaluation des systèmes interactifs à celle des systèmes d'information personnalisés", s'intéresse à l'évaluation dans les systèmes interactifs tout en ayant en visée celle de systèmes d'information personnalisés. Dans ce but, ce chapitre illustre d'abord le principe de l'évaluation ainsi que les critères d'évaluation existants relativement aux systèmes interactifs. Ensuite, nous présentons une étude des méthodes d'évaluation

existantes en fonction des critères de personnalisation que nous proposons pour l'évaluation de SIP.

Ces deux premiers chapitres, correspondent à un état de l'art sur, d'une part, les principes, approches et méthodes utilisées pour la personnalisation, et d'autre part sur les critères et les méthodes d'évaluation existantes. Cet état de l'art nous offre un cadre d'étude pour la proposition des méthodes d'évaluation de SIP.

En effet, intitulé "Contribution à l'évaluation des systèmes d'information personnalisés", le troisième chapitre présente notre principale contribution, la proposition des trois méthodes complémentaires d'évaluation : une première contribution débouche sur la proposition d'une méthode, intitulée MetQus (METHode d'évaluation basée sur le QUestionnaire de Satisfaction). L'objectif de cette technique est de recueillir un ensemble d'appréciations, d'opinions et d'attitudes, de l'utilisateur après qu'il ait interagi avec un système d'information personnalisé. Elle est basée sur l'intervention explicite de l'utilisateur et l'existence d'un système réel (maquette, système final). Une deuxième contribution prend la forme d'une méthode d'évaluation précoce appelée MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage). Elle consiste à prévoir l'évaluation dès les premières étapes de développement d'une application personnalisée en intégrant un système de trace. Partant du principe que la meilleure façon de détecter les problèmes liés à un système d'information personnalisé est d'observer l'utilisateur réel lorsqu'il réalise sa tâche, l'idée de base de cette approche consiste à générer le modèle réel de l'utilisateur dans un contexte d'usage bien défini en se basant sur le mécanisme de trace qui doit être intégré durant la phase de conception. En effet, ce modèle permet à l'évaluateur de détecter un premier ensemble représentatif de problèmes de personnalisation inhérents à l'exploitation d'un SIP. Une troisième contribution correspond à une méthode d'évaluation par simulation basée sur un système à base de cas, intitulée MetSim (METHode d'évaluation par SIMulation). Elle consiste à évaluer une IHM personnalisée en faisant varier des hypothèses d'usages liés à des contextes différents. L'intérêt des simulations est de pouvoir considérer différents scénarios en changeant le contexte d'usage (Plate-forme, Utilisateur, et environnement). Le principe de l'évaluation par simulation est de tester l'application dans des contextes d'usage différents. Elle doit fournir à l'évaluateur des indications lui permettant de l'aider à critiquer l'IHM et proposer des améliorations au concepteur. Elle aide aussi à représenter les problèmes détectés lors des évaluations et les recommandations qui en résultent.

Le quatrième chapitre est intitulé "Mise en œuvre des méthodes d'évaluation proposées dans le domaine du transport". Dans celui-ci, nous validons les méthodes proposés en les appliquant sur trois études de cas de système d'information personnalisé. Étant donné les travaux menés par nos laboratoires (LAMIH, SOIE) dans le domaine de la personnalisation et de la diffusion des informations destinées aux voyageurs dans le domaine des transports, c'est très naturellement que nous nous avons décidé d'utiliser cette expérience acquise au fil des années pour servir de support à nos travaux de recherche. Des résultats issus des applications des trois méthodes d'évaluation proposées sont présentés. L'illustration des méthodes proposées sur des cas d'étude différents permettra de distinguer les points forts et les points faibles de ces méthodes. Chaque cas d'étude a ses propres caractéristiques, ses contextes d'usage spécifiques, et sa fréquence de changement du contexte.

Le projet MOUVER.PERSO (Mobilité et Multimodalité Voyageurs Etudiants en Région Nord-Pas de Calais – Système d'information multimodale personnalisée) impliquant plusieurs laboratoires et instituts de recherche (INRETS, LAGIS, LAMIH), nous servira de cadre d'application au quatrième chapitre. Ce projet vise à mettre en œuvre et évaluer des services liés à la personnalisation des informations. Notre contribution à ce projet concerne l'évaluation du démonstrateur « MOUVER.PERSO », développé au sein de notre laboratoire,

offrant des services centrés sur la mobilité au quotidien, dans un but d'accompagnement du jeune dans son « univers » et ses activités. Le projet ANR VIATIC.MOBILITE portant également sur l'information voyageur a constitué aussi pour nous une source de réflexion importante. Ce projet coordonné par l'INRETS-ESTAS (G.USTER) a impliqué 5 laboratoires de recherche et 10 entreprises. Il vise à faciliter la mobilité urbaine des personnes en les aidant à bouger dans les meilleures conditions possibles depuis la préparation du déplacement jusqu'à l'arrivée au point de destination. Ces projets nous ont fourni de nombreux éléments de réflexion.

Finalement, le cinquième chapitre appelé "Perspectives de recherches", propose des perspectives ouvertes pour des travaux futurs sous la forme d'évolutions et d'extensions de ces travaux ; celles-ci concernent les méthodes MetQus, MetTra et MetSim. Enfin, des perspectives de recherche pourraient s'appuyer sur des études de cas plus complexes que ceux traités dans le quatrième chapitre (ex. système de diffusion d'information personnalisé dans le domaine de transport, etc.). Ceci nous permettrait de continuer à tester les méthodes proposées dans le cadre de notre travail.

Chapitre 1

Personnalisation dans les systèmes d'information

Sommaire

Introduction	15
1.1. Personnalisation : concepts et méthodes	15
1.1.1. Définitions	15
1.1.2. Acquisition d'information sur l'utilisateur	17
1.1.3. Modélisation de l'utilisateur	18
1.1.4. Les méthodes de personnalisation existantes.....	19
1.2. Etude des systèmes de personnalisation existants	33
1.2.1. Critère d'étude des systèmes d'information personnalisée	33
1.2.2. Systèmes de personnalisation existants.....	33
1.2.3. Synthèse et discussion.....	35
1.3. Contexte applicatif : la personnalisation de l'information transport	35
1.3.1. Apport de la personnalisation dans les systèmes d'informations au profit des usagers des transports collectifs de personnes	36
1.3.2. Systèmes d'informations personnalisés existants dans le domaine de transport.	36
1.3.3. Synthèse et discussion.....	38
Conclusion	39

Introduction

Le terme "personnalisation" connaît un vif succès dans la littérature scientifique. Ce mot figure dans un grand nombre d'articles et de publications récentes traitant pourtant de sujets très variés. Cependant, il n'existe pas en ce jour de consensus sur une définition claire et précise de la personnalisation. En effet, le champ de la personnalisation recouvre un nombre important de concepts et a donné lieu à l'émergence d'un grand nombre de termes dont les définitions sont parfois floues.

L'objectif de ce chapitre est de présenter un état de l'art sur la notion de personnalisation dans les systèmes d'information. Dans ce but, une première partie est consacrée aux définitions des principaux concepts rassemblés sous l'unique appellation "*personnalisation*". A partir des différents termes employés dans la littérature autour de la personnalisation, nous identifions dans une deuxième partie plusieurs méthodes de personnalisation existantes. Elles diffèrent principalement dans la manière de modéliser l'utilisateur et la façon de filtrer l'information.

Dans la troisième partie, nous présentons le cadre applicatif de la thèse, nous mettons en évidence les apports de la personnalisation dans les systèmes d'informations au profit des usagers des transports collectifs de personnes et nous étudions quelques systèmes d'information personnalisés existants dans le domaine du transport.

1.1. Personnalisation : concepts et méthodes

1.1.1. Définitions

Lors de notre recherche bibliographique, nous avons pu prendre connaissance de nombreux ouvrages et articles traitant tous de la personnalisation, mais chacun abordant le sujet sous un angle différent. A ce propos, différents termes portant sur la personnalisation sont utilisés. Afin de pouvoir bien les présenter, il est nécessaire de les classer. Pour cela, nous distinguons trois catégories : la première catégorie nécessite l'intervention explicite de l'utilisateur qui est activement engagé dans une interaction avec le système délivrant l'information. On parle alors de personnalisation contrôlée par l'utilisateur. Dans la deuxième catégorie la personnalisation est guidée par le système lui-même [Carmagnola et al., 08] [Zimmermann et Lorenz 08]. Dans ce cas, il s'agit de générer automatiquement une interface homme-machine suivant le profil de l'utilisateur. Et la troisième catégorie porte sur la capacité du système interactif à s'ajuster par rapport à son contexte d'utilisation. La personnalisation est considérée alors comme une réaction au changement de contexte.

La personnalisation contrôlée par l'utilisateur : ce type d'interaction consiste à laisser le soin à l'utilisateur de juger de ce qui l'intéresse ou non. Le système propose par exemple la liste de tous les documents qu'il a pu recenser en réponse à une demande de recherche, charge à l'utilisateur de valider ensuite ceux qui correspondent à ses besoins. Ainsi, la personnalisation est un processus contrôlé par l'utilisateur qui effectue un choix entre plusieurs options. Ce choix guide complètement la réponse fournie par le système [Rosenberg 01]. Dans la personnalisation dirigée par l'utilisateur les auteurs utilisent généralement les termes : customisation et adaptabilité. A titre d'exemple, Cinquin et ses collègues utilisent le terme customisation pour souligner la nécessité de l'intervention de l'utilisateur [Cinquin et al., 02]. Pour [Stephanidis et al., 01] l'adaptabilité caractérise une interface capable de s'adapter en utilisant les connaissances acquises durant la phase d'initialisation de

l'interaction entre l'utilisateur et le système. Notons aussi que ces termes visent la personnalisation de contenu et de contenant.

La personnalisation guidée par le système : le système gère la connaissance qu'il a de l'utilisateur (ses besoins, ses préférences, etc.) pour décider ce qui doit lui être présenté [Enrique et al., 07]. Cette migration de la tâche de décision vers le système évite à l'utilisateur d'interrompre son activité principale. Dans ce cas, la personnalisation s'appuie sur un processus d'apprentissage, par la machine, de ce que veut l'utilisateur. Cet apprentissage est basé sur une observation de son comportement au cours des sessions de l'utilisation du système. Dans ce cadre, les termes principalement employés par les auteurs sont : recommandation, personnalisation et adaptativité. Pour [Dix et al., 98] la personnalisation est l'action de donner un caractère personnel à l'interface, alors que dans le domaine du Web [Bazsalicza et Naïm 01] définissent la personnalisation comme « la capacité d'un site dynamique à produire des ressources en fonction de l'identité du demandeur ». L'adaptativité traduit une vision plus dynamique du processus d'adaptation. Les connaissances sont ici acquises ou modifiées par le système au cours des interactions, via des techniques de suivi de session. Le système procède à des adaptations pendant que l'utilisateur interagit avec lui. Cette vision est également partagée par [Frasincar et al., 02], qui appellent l'adaptabilité adaptation statique et l'adaptativité adaptation dynamique.

La personnalisation centrée sur le contexte : cette catégorie permet de prendre en compte le contexte de la recherche dans le processus de la personnalisation. La définition de contexte d'utilisation ne cesse d'évoluer. Aujourd'hui, la définition la plus couramment adoptée par les équipes travaillant dans le domaine de la personnalisation des systèmes interactifs identifie le contexte par le triplet « Utilisateur, Plate-forme, Environnement » [Calvary et al., 05], [Samaan et al., 04], [Vanderdonck et al., 05], [Miettinen et Oulasvirta 07], [Hariri 08] où :

- **l'utilisateur** est typiquement décrit par ses capacités (perceptuelles, motrices et cognitives), mais aussi par ses préférences et ses caractéristiques culturelles (e.g. langue naturelle) ;
- **la plate-forme** (*user device*) qui se décline en deux sous catégories :
 - la plate-forme physique (matérielle) ;
 - la plate-forme logique (logicielle) ;
- **l'environnement** se réfère à l'environnement physique accueillant l'interaction. Il est décrit par un ensemble d'informations périphériques à la tâche en cours mais susceptibles de l'influencer. Par exemple, la luminosité, le bruit, la localisation géographique, etc.

Ainsi, sont apparues des notions comme les MUI (*Multiple User Interfaces*) [Seffah et al., 04], le concept de plasticité des interfaces [Thevenin et al., 99], défini, par analogie à la plasticité d'un matériau, comme étant « sa capacité à s'adapter aux contraintes matérielles et environnementales dans le respect de son utilisabilité » et le concept de context awareness qui est utilisé dans le cadre de la personnalisation liée au contenu [Abowd et al., 99].

Pour notre part, nous avons retenu deux définitions complémentaires :

"La personnalisation est la capacité d'adapter la communication client en se basant sur la connaissance des préférences et des comportements au moment de l'interaction (avec le client)" [Dyche 02].

"Pour une application informatique, la personnalisation, c'est la capacité de fournir à un utilisateur, à chaque instant, des contenus et des services adaptés à ses besoins et à ses attentes en utilisant des interactions homme-machine appropriées. Pour ce faire, cette adaptation utilise sa connaissance de l'utilisateur (caractéristiques et préférences), sa connaissance de l'environnement, ainsi que des informations sur le comportement de l'utilisateur au moment de l'interaction" [Brossard 08].

A la lumière de ces définitions et des remarques formulées pour chacune d'entre elles, nous nous proposons maintenant d'établir notre propre définition : « la personnalisation est la capacité d'adaptation d'un système interactif par rapport aux préférences, centres d'intérêts et à l'expérience de l'utilisateur ainsi qu'à la plateforme d'interaction et à l'environnement ». Cette adaptation peut aussi bien porter sur le contenu (données, informations, documents, etc.) que sur le contenant (présentation, plate-forme d'interaction, mode de communication, etc.).

1.1.2. Acquisition d'information sur l'utilisateur

Le but d'un système d'information personnalisé est de fournir à ses utilisateurs l'information la plus pertinente, celle qui correspond le plus aux attentes des utilisateurs. Pour répondre à cette problématique, le système d'information doit connaître ses utilisateurs. Nous distinguons deux moyens pour acquérir l'information à propos de l'utilisateur : une acquisition explicite de l'information et/ou une acquisition automatique de celle-ci.

1.1.2.1. Acquisition explicite de l'information

Cette technique permet à un système d'information d'obtenir des informations sur l'utilisateur avec intervention de celui-ci en répondant à un ensemble des questions, sous la forme d'un formulaire ou en attribuant une note à chacune des ressources qu'il souhaite évaluer. Nous pouvons citer à titre d'exemple, le système Smart Radio [Hayes et Cunningham 01] qui propose à ses utilisateurs de noter un ensemble de programmes musicaux sur une échelle de 1 à 5. Outre le fait qu'elle soit facile à implémenter, que l'acquisition des connaissances sur les utilisateurs soit tout aussi aisée et que la validation des connaissances est facilitée, cette méthode souffre d'un inconvénient majeur. La plupart du temps, les questionnaires et/ou les fiches de notes sont longs et fastidieux, de telle sorte que les utilisateurs se découragent assez vite de voir la fin des questions, ce qui les pousse à renoncer à répondre à l'ensemble des questions posées [Petit-Rozé et al., 04d]. Afin de remédier à cet inconvénient, il s'agit de solliciter l'utilisateur avec peu de questions, à différents instants à choisir judicieusement.

1.1.2.2. Acquisition automatique de l'information

Cette technique consiste à observer le comportement de l'utilisateur sans lui poser de question [Petit-Rozé 03a] [Baraglia et Silvestri 07]. Ainsi, l'utilisateur ne spécifie en aucune manière ses préférences. Le système se contente d'observer ce qui semble intéresser ou non l'utilisateur, par le biais de tâches mesurables, comme par exemple le nombre de liens dans un document ou site, que l'utilisateur a suivis ou non.

L'inconvénient de cette méthode est que le système de recommandation ne peut prétendre connaître réellement ses utilisateurs. Puisque le système se charge d'apprendre des informations sur ses utilisateurs, sans que ces derniers ne spécifient leurs préférences, certaines des données risquent de ne pas être prises en compte et, de ce fait, les profils identifiés ne reflèteraient pas nécessairement les intérêts de chacun. Malgré cet inconvénient, une telle méthode permet aux utilisateurs de ne pas être obligés de préciser leurs intérêts. De même que pour évaluer certaines données, il est préférable que ce soit le système qui s'en charge car les utilisateurs ne sont pas toujours à même de pouvoir évaluer ces données.

Différents moyens sont utilisés pour la collecte implicite d'information dans le web : on pourra citer à titre d'exemple l'analyse des fichiers de logs des serveurs web [Murgue 05], l'utilisation des moniteurs de réseau et l'utilisation des paramètres de configuration [Manber et al., 00].

1.1.3. Modélisation de l'utilisateur

L'importance de l'utilisateur dans une démarche de conception de SIP est largement reconnue comme prépondérante. Le système est avant tout conçu pour des utilisateurs auxquels il est nécessaire de s'intéresser en vue d'anticiper leurs attentes à l'égard du système. Le SIP nécessite alors une description plus élaborée de l'utilisateur et de sa représentation en tant qu'objet à part entière du système. Cette représentation interne de l'utilisateur vise à procurer au système les moyens de procéder aux personnalisations souhaitées. Il est donc primordial de définir comment représenter l'utilisateur. En d'autres termes, il faut choisir les informations à retenir pour en construire un modèle. Ces informations sont récupérées par l'une ou l'autre des solutions présentées précédemment (acquisition explicite ou implicite) et sont ensuite mémorisées par le système pour constituer un modèle de l'utilisateur (nous appellerons par la suite profil utilisateur le modèle généré après l'acquisition des données concernant l'utilisateur). Selon Allen le modèle de l'utilisateur peut être vu comme « l'image que le système informatique a de l'utilisateur » [Allen 79]. [Kay 01] définit le modèle utilisateur comme une combinaison du modèle utilisateur construit par lui-même et celui du programmeur. Ces deux modèles sont construits en prenant en compte le contexte par rapport au monde réel, les connaissances, les préférences et les buts de l'utilisateur et du programmeur. Dans ce mémoire on appelle modélisation de l'utilisateur la façon de représenter un utilisateur et ses comportements, il regroupe l'ensemble des connaissances que possède le système sur l'utilisateur.

Les travaux sur la modélisation des utilisateurs remontent aux années 70. Depuis, de nombreux systèmes et de nouvelles techniques sont apparus ayant tous pour objectif l'acquisition de connaissances sur les utilisateurs afin de les exploiter :

- pour assurer une personnalisation de contenant afin de concevoir des interfaces homme-machine destinées à faciliter les tâches utilisateurs grâce à la compréhension de leur activité cognitive et éventuellement la prédiction de leurs actions, et/ou
- pour effectuer une personnalisation de contenu capable d'adapter les données aux besoins, observés et reconnus de l'utilisateur.

Nous distinguons deux catégories de modélisation : la modélisation individuelle et la modélisation commune.

1.1.3.1. Modélisation individuelle

Les SIP sont destinés à un public varié composé d'individus ne présentant pas les mêmes besoins à l'égard du système, n'ayant pas les mêmes centres d'intérêts, faisant preuve d'expériences ou de connaissances variées, exprimant des préférences diverses, etc. La prise en compte d'un utilisateur en tant qu'individu est également nécessaire pour réduire la complexité sous-jacente à une telle hétérogénéité.

La modélisation individuelle considère donc l'utilisateur comme une entité représentée à l'intérieur du système au moyen d'un modèle individuel spécifique ; ce modèle de l'utilisateur constitue la source d'informations nécessaires à la mise en oeuvre de la personnalisation du système d'information. Le modèle individuel essaie de modéliser l'information spécifique à chaque utilisateur. Par exemple, le système WEBMATE [Keeble et al., 00], apprend le profil de l'utilisateur, lui prépare un journal d'actualités personnel et l'aide à améliorer sa recherche d'information.

1.1.3.2. Modélisation commune des utilisateurs

Cette modélisation consiste à proposer un modèle commun ou stéréotype à une catégorie d'utilisateurs. Cette représentation permet de classer *a priori* tout nouvel utilisateur. Cette notion de profil commun à plusieurs personnes peut servir à généraliser des goûts supposés

partagés. Par exemple, le système FIREFLY [Shardanand et al., 95] se base sur la connaissance de similarités entre le profil d'intérêt de l'utilisateur et ceux d'autres utilisateurs dans le domaine musical. Le système présente tout d'abord à l'utilisateur une liste de 125 artistes. Charge à celui-ci de donner un jugement sur chacun de ces artistes (note de 1 à 7). Le système peut ensuite suggérer d'autres artistes ou albums ou faire une prédiction concernant un artiste ou un album.

Les éléments essentiels d'un stéréotype sont :

- les déclencheurs (triggers) qui activent le stéréotype. Il s'agit des caractéristiques définissant ces utilisateurs relativement au reste du groupe. Par exemple, le fait que l'utilisateur soit expert dans l'utilisation d'Internet.
- les règles qui vont définir les stéréotypes. Par exemple, les connaissances supposées connues pour un utilisateur classé comme expert.
- une rétractation facilitant la désactivation d'un stéréotype quand ses déclencheurs essentiels sont invalidés. Par exemple, si un stéréotype expert a été activé, alors que l'observation des actions montre que l'utilisateur n'est pas expert.

Les stéréotypes permettent d'établir des croyances par défaut sur l'utilisateur. Au fur et à mesure que d'autres informations deviennent disponibles, le modèle est affiné et ajusté par la connaissance d'un utilisateur en particulier. Ainsi, le système IFM (Intelligent File Manipulator) [Virvou et al., 02], qui est un système d'aide intelligent utilisé pour la manipulation graphique de systèmes de fichiers, se base sur les stéréotypes. Il incorpore un mécanisme de reconnaissance des buts de l'utilisateur. Le système intervient automatiquement et propose des conseils lorsqu'un utilisateur fait des démarches inconsistantes ou commet des erreurs de manipulation. Citons également TAGUS (Theory and Application for General User/learner modeling System) (décrit dans [Kobsa 01]). Ce système permet la définition de stéréotype hiérarchique et contient un mécanisme d'inférence, un système de maintien de la cohérence, et un sous-système de diagnostic qui contient une bibliothèque des erreurs. Il supporte aussi la simulation de l'utilisateur à travers des mécanismes d'inférence en chaînage avant sur la base du modèle utilisateur, et le diagnostic des comportements imprévus de l'utilisateur.

Les systèmes de recherche de données par Internet utilisent également la notion de stéréotype. Dans CONCALL (CONference and workshop CALL reminder Service) [Waern et al., 98], qui est un service d'information adaptatif, les utilisateurs sont classés selon leur profil (chercheur, éditeur, etc.). Le modèle utilisateur est basé sur un ensemble de mots clés et est utilisé à la fois comme filtre lors de la recherche d'informations et comme moyen d'ordonner les résultats de la recherche (évaluation de la pertinence du résultat). Le système sert à rassembler, filtrer et diffuser des appels pour des conférences, des workshops, etc., sur Internet.

1.1.4. Les méthodes de personnalisation existantes

Les systèmes personnalisés sont généralement conçus avec une ou plusieurs méthodes de personnalisation dédiées pour répondre à un type de personnalisation bien défini. Ces méthodes permettent de fournir à un utilisateur un contenu et/ou un contenant personnalisé (interface adaptée) en se basant sur les modèles utilisateur.

1.1.4.1. Les méthodes de personnalisation de contenu

Il existe deux familles de méthodes pour la personnalisation de contenu : les méthodes cognitives et les méthodes sociales. Notons que ces deux types de méthodes peuvent être combinés.

1.1.4.1.1. Méthodes cognitives

La notion de méthode cognitive a été proposée en 1983 par [Salton 83] ; elle tire son origine de la recherche d'information traditionnelle. Ces méthodes s'appuient sur l'analyse des documents ou sur le modèle individuel de l'utilisateur courant. Ce type de méthode utilise diverses techniques pour représenter et gérer le profil de l'utilisateur.

- **Représentation vectorielle** : ce type de représentation s'appuie généralement sur le modèle vectoriel. Les données manipulées par l'utilisateur et son profil sont représentées par un ou plusieurs vecteurs de mots-clés ou de poids associés à ces mots-clés. La méthode consiste à calculer la distance entre le vecteur représentatif de l'utilisateur et ceux des documents. La distance la plus faible correspond au document le plus pertinent pour la requête de l'utilisateur [Aas 97], [Korfhage 97].

Cette représentation est notamment appliquée dans les systèmes InfoFinder [Krulwich et al., 97] et News Dude [Billsus et al., 99]. Par exemple, dans InfoFinder, des profils d'utilisateurs sont gérés selon une technique simple : l'utilisateur lit des documents ; quand il trouve un document intéressant, il l'indique au système ; InfoFinder permet de créer un arbre de décision à partir des phrases significatives de chaque document jugé intéressant par l'utilisateur, puis il transforme cet arbre en une requête booléenne qui est soumise par la suite à un moteur de recherche afin de trouver de nouveaux documents.

Cette technique offre l'avantage indéniable de la simplicité de mise en œuvre. Cependant, elle permet d'évaluer que des documents sous forme textuelle. De ce fait, les nouveaux types de documents (vidéo, photographie, son, etc.) ne peuvent pas être gérés.

- **Représentation sémantique** : cette représentation met d'avantage en relief des relations de sens entre unités d'informations représentant le profil en apportant des solutions aux problèmes de dissémination et synonymie [Vallet et al., 05]. Le principe de cette technique est la construction d'une hiérarchie de concepts plutôt qu'une liste de structures indépendantes. La hiérarchie peut contenir plusieurs niveaux de préférences de l'utilisateur et des associations latentes entre concepts. Cette représentation utilise généralement des ontologies [Scine et al., 02]. [Ambrosini et al., 97] utilise les réseaux sémantiques pondérés pour structurer des documents afin d'effectuer un filtrage lors d'une recherche d'information. [Pretschner 99] a proposé aussi dans le système OBIWAN (Ontology Based Informing Web Agent Navigation) un modèle innovant pour la construction du profil utilisateur en s'appuyant sur l'ontologie publique de Magellan qui est composée d'approximativement 4.400 noeuds concepts. Semblable à ce travail, on peut citer le système SmartPush [Kurki et al., 99].

- **Réseaux bayésiens** : un réseau bayésien est un graphe orienté acyclique qui représente une distribution probabiliste. Les noeuds sont des variables aléatoires et les arcs décrivent les corrélations probabilistes entre ces variables. Pour chaque noeud, on a une table qui spécifie la probabilité de chaque état du noeud en fonction de toutes les combinaisons possibles des états de ces prédécesseurs [Ragg et al., 02].

[Schiaffino 00] donne un exemple d'utilisation des réseaux bayésiens en combinaison avec la technique de raisonnement par cas. Chaque requête de l'utilisateur est prise comme un cas. Chaque noeud du réseau représente un attribut ou une caractéristique utilisé dans une requête. Les arcs décrivent les relations entre les attributs comme étant les fréquences d'apparitions des attributs dans les requêtes de l'utilisateur. Ces fréquences représentent également l'importance des attributs pour le client. Lorsqu'une nouvelle requête arrive, elle est stockée sous forme de cas et pour chaque attribut un noeud est ajouté dans le réseau. Ensuite, les liens entre ces caractéristiques sont rajoutés et les fréquences d'apparition sont mises à jour.

- **Le raisonnement à partir de cas** : celui-ci permet de résoudre de nouveaux problèmes sur la base des situations survenues dans le passé. Le processus mémorise des instances de cas concrets et ensuite prend une décision en faisant une comparaison entre le nouveau cas et les cas connus [Lajmi et al., 09]. C'est un processus incrémental qui nécessite au début des exemples servant comme base des prochaines classifications. Un exemple de système utilisant cette technique est CASPER (Case-Based Profiling for Electronic Recruitment) [Bradley 00]. C'est un système de recherche d'emploi qui propose à un utilisateur les annonces qui peuvent le concerner selon son profil. Nous pouvons citer aussi le système de [Coyle et Cunningham 02] qui se base sur un raisonnement à partir de cas pour proposer des informations classées par ordre de pertinence suivant les préférences d'utilisateurs des transports aériens.

Les inconvénients de cette approche se trouvent dans le fait que le système doit connaître l'utilisateur pour pouvoir comparer les situations précédentes avec les nouveaux cas et dans le fait que c'est une approche incrémentale qui demande des ressources de calcul considérables parce que chaque nouveau cas est comparé avec tous les cas du profil.

- **Les réseaux de neurones** : un réseau de neurones est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones (humains ou non). Il est basé sur l'interconnexion de nœuds et un principe d'activation par propagation de signaux à travers les connexions depuis les entrées jusqu'aux sorties [Norsham et al., 09]. D'une manière générale les réseaux de neurones sont destinés à résoudre des problèmes de décision non linéaires. Dans le contexte de modélisation de l'utilisateur, l'entrée représente une situation ou des faits observables à partir de l'utilisateur, les sorties représentent des objets cibles du système. Le système proposé dans [Moghrabi et Eid 98] représente les pages web visitées par l'utilisateur sous forme de réseaux de neurones. Un autre système est proposé dans [Jennings et Higuchi 93] qui utilise les réseaux de neurones pour représenter les préférences d'utilisateur pour la recherche des nouveaux articles susceptibles d'intéresser celui-ci. Pour chaque utilisateur, l'approche proposée consiste à concevoir un réseau où les nœuds représentent les mots qui apparaissent dans les articles de l'utilisateur et les relations représentent les associations entre termes apparaissant dans ces articles.

1.1.4.1.2. Méthodes sociales

La notion de méthode sociale a été proposée en 1992 par Goldberg [Goldberg et al., 92]. Les systèmes basés sur les méthodes sociales sont appelés aussi des systèmes de filtrage collaboratif. Celles-ci ne sont pas basées sur l'analyse des documents ou des intérêts individuels des utilisateurs courants pour la personnalisation. Deux stratégies peuvent être adoptées par une méthode collaborative : la première consiste à regrouper les utilisateurs similaires en analysant leurs préférences par rapport aux ressources (documents) et la deuxième permet de classer les ressources en se basant sur les appréciations des utilisateurs. Trois classes d'algorithmes de filtrage collaboratif ont été distinguées [Breese 98] : les algorithmes basés sur la mémoire, les algorithmes basés sur un modèle et les algorithmes de filtrage collaboratif hybride.

Les algorithmes basés sur la mémoire

Ces algorithmes permettent de rechercher les utilisateurs ayant des préférences communes avec l'utilisateur courant pour prédire, à partir des similarités trouvées entre les utilisateurs, l'information qui correspond le plus aux attentes de l'utilisateur courant.

- **Calcul de la corrélation entre les utilisateurs** : le principe de base de cette technique consiste à étudier la corrélation qui peut exister entre l'utilisateur courant et l'ensemble des utilisateurs présentant des préférences semblables, en se basant sur les différents profils

[Candillier 01]. Considérons l'ensemble des items évalués (noté I_i) par l'utilisateur i et notons $v_{i,j}$ les évaluations (généralement des notes) affectées aux items. Nous pouvons, dans un premier temps, définir l'évaluation moyenne pour cet utilisateur par :

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

Supposons qu'il y ait n utilisateurs dans la base de données, chacun ayant évalué un certain nombre de ressources (nous parlerons dans ce cas d'utilisateurs ayant un poids non nul). Soient κ un facteur de normalisation de telle sorte que la somme des valeurs absolues des poids fasse 1 ($\kappa = \sum_{i=1}^n \frac{1}{|w(a,i)|}$), et $w(a,i)$ un poids de similarité entre les utilisateurs déterminé

de façon variable, selon l'algorithme utilisé la prédiction sur l'item j pour l'utilisateur courant a sera donc définie par : $p_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^n w(a,i)(v_{i,j} - \bar{v}_i)$

Pour déterminer la corrélation entre l'utilisateur courant et l'ensemble des utilisateurs du système de recommandation, il suffit de calculer le poids $w(a,i)$. Pour cela, nous pouvons utiliser soit le coefficient de corrélation de Pearson soit la similarité des vecteurs. Nous allons présenter ces deux méthodes.

Coefficient de corrélation de Pearson : il se calcule comme le cosinus de l'écart à la moyenne des évaluations des deux utilisateurs considérés. La formule est la suivante [Resnick et al., 94] :

$$w(a,i) = \cos(\vec{v}_a - \bar{v}_a, \vec{v}_i - \bar{v}_i) = \frac{(\vec{v}_a - \bar{v}_a) \cdot (\vec{v}_i - \bar{v}_i)}{\|\vec{v}_a - \bar{v}_a\| \cdot \|\vec{v}_i - \bar{v}_i\|}$$

$$\text{soit } w(a,i) = \frac{\sum_{j \in I_a \cap I_i} (v_{a,j} - \bar{v}_a) \cdot (v_{i,j} - \bar{v}_i)}{\sqrt{\sum_{j \in I_a \cap I_i} (v_{a,j} - \bar{v}_a)^2 \sum_{j \in I_a \cap I_i} (v_{i,j} - \bar{v}_i)^2}}$$

Notons que plus cette valeur est grande et plus les utilisateurs a et i sont corrélés entre eux.

Similarité des vecteurs : dans ce cas, la formule permettant de calculer le poids $w(a,i)$ est quasiment la même que précédemment, à l'exception du fait qu'il ne s'agit plus d'un cosinus de l'écart à la moyenne des évaluations des deux utilisateurs considérés, mais plutôt d'un cosinus entre les vecteurs formés par les évaluations de l'utilisateur a et de l'utilisateur i [Breese et al., 98]. Nous obtenons la formule suivante :

$$w(a,i) = \cos(\vec{v}_a, \vec{v}_i) = \frac{\vec{v}_a \cdot \vec{v}_i}{\|\vec{v}_a\| \cdot \|\vec{v}_i\|}$$

$$\text{soit } w(a,i) = \frac{\sum_j v_{a,j} \cdot v_{i,j}}{\sqrt{\sum_j v_{a,j}^2 \sum_j v_{i,j}^2}}$$

Les termes du dénominateur servent à normaliser les évaluations afin d'éviter toute inégalité entre utilisateurs qui serait basée sur le nombre de ressources évaluées.

Si $w(a,i)$ est proche de la valeur 1, nous pouvons considérer les utilisateurs comme ayant des profils similaires. Par contre, si $w(a,i)$ avoisine 0, alors les utilisateurs ont des préférences différentes.

- **L'approche de Bayes** : l'approche bayésienne permet de calculer la probabilité d'attribuer une note k à un item j pour un utilisateur i , en se basant sur le profil de cet utilisateur (que nous notons \vec{v}_i : vecteurs des évaluations $v_{i,l}$ de l'ensemble des ressources évaluées), ainsi que l'ensemble des profils des autres utilisateurs [Srisuwan et Srivihok 07]. Cette probabilité est traduite par la formule suivante :

$$P(v_{i,j} = k | \vec{v}_i) = \frac{P(v_{i,j} = k) \cdot P(\vec{v}_i | v_{i,j} = k)}{P(\vec{v}_i)}$$

Il s'agit de trouver la note k qui maximise cette formule.

Il suffit donc de calculer les deux probabilités $P(v_{i,j} = k)$ et $P(\vec{v}_i | v_{i,j} = k)$, la première se calcule facilement en utilisant la base de données utilisateur des évaluations des articles déjà effectuées, et la seconde en calculant $\prod_{p \in I_i} P(v_p = v_{ip} | v_j = k)$, où $P(v_p = v_{ip} | v_j = k)$ est la probabilité d'attribuer la note v_i à l'item p sachant que l'on a attribué la note k sur l'item j .

- **Le vote moyen pondéré de Bayes** : le vote moyen pondéré de Bayes est basé sur l'approche bayésienne, à ceci près qu'il est calculé sur la ressource j , en prenant en compte le profil de l'utilisateur i (noté \vec{v}_i). Notons $MaxEchelle$ le maximum de l'intervalle de notes utilisé. La formule est la suivante :

$$note = \sum_{k=1}^{MaxEchelle} P(v_j = k | \vec{v}_i) * k$$

Les algorithmes basés sur un modèle

L'idée de base de ces algorithmes consiste à estimer ou apprendre un modèle en utilisant la base de données utilisateur, qui sera ensuite utilisé pour les prédictions. Le filtrage collaboratif basé sur un modèle regarde chaque item contenu dans une liste utilisateur/item où chaque utilisateur a choisi les items et trouve l'item ayant les mêmes évaluations. Un modèle est donc créé et le système recommande l'item correspondant à ce modèle.

Pour obtenir un modèle, plusieurs méthodes existent. Nous pouvons citer les approches centrées probabilité et les approches de clustering. Nous allons les présenter dans la suite.

- Les approches centrées probabilité :

Approche probabiliste : de point de vue probabiliste la prédiction est déterminée comme une espérance mathématique, connaissant le profil de l'utilisateur, et le modèle qui a été construit à partir de la base de données utilisateur [Albrecht et Zukerman 07]. Notons I_i l'ensemble des items évalués par un utilisateur u_i , avec $eval(u_i, r_1)$ l'évaluation faite par l'utilisateur u_i sur l'item r_1 et supposons que $[0,v]$ est l'ensemble des valeurs que peuvent prendre les évaluations. Mathématiquement, cette approche s'écrit de la manière suivante :

$$\begin{aligned} \text{predic}(u_i, r_l) &= E(\text{eval}(u_i, r_l)) \\ &= \sum_{k=0}^v k.P(\text{eval}(u_i, r_l) = k \mid \text{eval}(u_i, r), r \in I_i) \end{aligned}$$

Cette formule peut se traduire comme étant la probabilité que l'utilisateur u_i évalue l'item r_l avec une valeur k , connaissant les évaluations qu'il a déjà attribuées sur les autres ressources.

Classifieur « naïf » de Bayes : l'approche par classifieur exige que l'ensemble des valeurs des évaluations doit être partitionné en classes. Le classifieur bayésien permet de déterminer la classe C la plus probable à laquelle l'utilisateur courant va appartenir, sous l'hypothèse d'indépendance des évaluations [Berrut et Denos 03] [Lumineau 03]. Notons v_i les évaluations de l'utilisateur i . Alors nous obtenons la formule suivante :

$$P(C = c, v_1, \dots, v_n) = P(C = c) \prod_{i=1}^n P(v_i \mid C = c)$$

Arbre de décision et réseau bayésien : l'idée de cette approche est de construire un réseau où chaque nœud identifie un item et l'état du nœud correspond à une valeur possible d'évaluation [Hawarah 06]. L'apprentissage consiste ici en la recherche de dépendances entre les ressources. A chaque nœud peut alors être associé un arbre de décision en fonction des nœuds précédents (permettant de prédire l'état du nœud courant). La figure 1.1 montre un exemple d'arbre de décision.

Nous pouvons donc dire que ces arbres correspondent à une représentation des probabilités conditionnelles disponibles au niveau du nœud, exprimée par :

$$P_B(C \mid A_1, \dots, A_n) = \alpha P_B(C) \prod_{i=1}^n P_B(A_i \mid C)$$

Où C représente une variable de classe, A_i sont des attributs et α est une constante de normalisation.

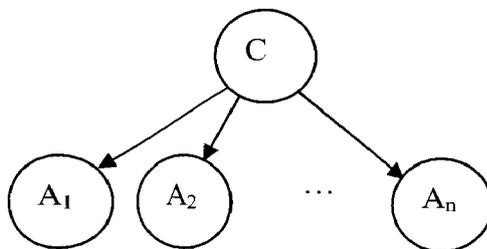


Figure 1.1. Arbre de décision

- Le modèle cluster

L'idée de modèle cluster est de regrouper (en clusters) les utilisateurs ayant les mêmes préférences et de regrouper les items portant sur les mêmes sujets, ou qui tendent à plaire aux mêmes utilisateurs. Ainsi, la prédiction sera calculée en fonction des utilisateurs et/ou items appartenant au même cluster. Parmi les algorithmes de clustering, nous pouvons citer les clustering répété, hiérarchique et l'algorithme des k plus proches voisins [Enrique et al., 07].

Le clustering répété : le principe de cet algorithme est de regrouper les utilisateurs et les items de façon séparée. En fait, Il consiste à affiner les groupes obtenus par une première classification en regroupant (en clusters) les utilisateurs en se basant sur les items, et en regroupant les items en se basant sur les utilisateurs. Ce processus de classification est réitéré

autant de fois que nécessaire mais cette fois en fonction des clusters d'items et les items en fonction des clusters d'utilisateurs [Ungar et Foster 98].

La limite de cette approche tient au fait qu'elle peut aboutir à une surgénéralisation dont la conséquence serait de regrouper des utilisateurs aux profils différents. La prédiction serait, dans ce cas, faussée puisque des utilisateurs très différents dans leurs préférences interviendraient dans le calcul de la prédiction. Or, le but du filtrage collaboratif est que seuls les utilisateurs fortement corrélés entre eux peuvent intervenir dans le calcul de la prédiction, de sorte que celle-ci corresponde effectivement à ce qu'attend l'utilisateur courant.

Le clustering hiérarchique : cette technique se base sur l'utilisation d'un arbre de clusters. On trouve à la racine de cet arbre tous les utilisateurs ainsi que leurs appréciations sur les items de la base. Plus on descend dans l'arbre, plus les clusters sont spécifiques à un certain groupe d'utilisateurs similaires. Le clustering hiérarchique permet de fractionner l'ensemble des utilisateurs en groupes hiérarchisés de telle sorte que les similarités entre les membres d'une même clique sont maximales. Comme le clustering répété, les groupes issus d'une première clusterisation sont reclusterisés. Le système vient alors chercher le groupe qui correspond le plus à l'utilisateur en calculant la matrice de similarité entre les utilisateurs contenus dans la clique, en utilisant par exemple le coefficient de corrélation de Pearson [Resnick et al., 94]. Le calcul de la prédiction se fait donc en ne considérant que les utilisateurs appartenant à la même clique [Fisher 96].

L'algorithme des k plus proches voisins : la méthode des k Plus Proches Voisins (PPV) est un algorithme supervisé classique (Duda et al., 2001), dont le principe est de classer tout nouveau point dans la classe majoritairement présente dans son entourage. L'entourage est constitué de points les plus proches au sens d'une distance prédéfinie en fonction de la nature des points à classer. La distance la plus couramment utilisée est la distance euclidienne. Cette méthode est d'autant plus performante que l'espace de représentation choisi sépare au mieux les différentes classes [Duda et al., 01], [Herlocker et al., 02], [Shakhnarovich 06].

Le filtrage collaboratif hybride

Depuis quelques années, un nouveau courant de filtrage collaboratif est apparu : le filtrage collaboratif hybride. Le principe consiste à combiner un filtrage basé sur la mémoire et un filtrage basé sur un modèle. Cette combinaison a pour objectif de limiter les inconvénients des deux méthodes et de bénéficier de leurs avantages. Dans l'article [Basu et al., 98], les auteurs proposent un système de recommandation de films à partir des deux types de filtrage. La méthode est simple. D'une part, le système se sert du filtrage collaboratif pour savoir si un film a peut plaire à un utilisateur b. Pour cela, le système gère différents ensembles de données (ensemble des utilisateurs aimant a, ensemble des films aimés par b. . .). D'autre part, le système gère les films en fonction de leur contenu. Le filtrage basé sur le contenu est utile pour savoir si un utilisateur peut aimer le nouveau film x. Pour cela, le système compare les caractéristiques de ce film (titre, réalisateur, acteurs, pays. . .) avec ceux d'autres films.

Parmi les approches que nous pouvons aussi trouver dans la littérature, nous pouvons citer l'approche de Horting [Aggarwal et al., 99], l'approche basée sur le diagnostic de personnalité [Pennock et al., 00] et l'approche basée sur la clusterisation et la réduction de dimension [Goldberg et al., 01]. Nous renvoyons le lecteur à [Candillier 01] et [Lekakos et Giaglis 07] ou aux publications se rapportant à ces algorithmes pour de plus amples détails sur ces méthodes.

1.1.4.1.3. Conclusion sur les méthodes de personnalisation de contenu

Les méthodes de personnalisation de contenu permettent de fournir à un utilisateur un contenu qui peut être soit des informations soit des services. Ces méthodes se déclinent en deux familles : les méthodes cognitives et celles qualifiées de sociales. Les méthodes cognitives se basent sur les préférences de l'utilisateur ou sur les données manipulées par ce dernier pour la personnalisation et les méthodes sociales se basent sur les expériences des utilisateurs ayant des préférences communes ou construisent des communautés d'utilisateurs pour la recommandation (cf. Tableau 1.1).

Tableau 1.1. Classification des méthodes de personnalisation selon le contexte utilisé et le type de filtrage adaptée de [Cinquin et al., 02]

		Types de filtrage	
		Méthodes sociales	Méthodes cognitives
Information de contexte	Utilisateur	Filtrage collaboratif	Filtrage basé sur le profil
	Données	Recommandation de communauté	Recommandation contextuelle

1.1.4.2. Méthodes d'adaptation d'IHM multiplateforme et/ou orienté contexte

Nous allons étudier dans cette partie des méthodes de spécification et génération d'IHM multiplateforme et/ou orientée contexte. Ces méthodes permettent de fournir à un utilisateur un contenant personnalisé (interface adaptée) en se basant sur les modèles utilisateur. Sans souci d'exhaustivité mais plutôt avec celui de représentativité, nous allons passer en revue un certain nombre de méthodes d'adaptation d'IHM multiplateforme et/ou orientée contexte.

1.1.4.2.1. Approche de Thevenin

Thevenin a proposé un environnement de développement d'interface homme-machine plastique à base de modèle appelé ARTStudio [Thevenin 01] (Adaptation par Réification et Traduction Studio). Cet environnement s'appuie sur cinq modèles de description : le modèle des tâches, celui des concepts, celui des instances des concepts manipulés par l'interface, celui des interacteurs qui sont présents sur la boîte à outils graphique de la plate-forme visée et celui de la plate-forme cible pour laquelle l'interface sera produite (cf. Figure 1.2).

- Le modèle des tâches décrit, sous la forme d'un graphe, les tâches qui seront réalisées au travers de l'IHM.
- Le modèle des concepts décrit la structure des données qui seront manipulées par l'IHM.
- Le modèle des instances décrit les instances de concepts manipulés par l'interface et le graphe de dépendance entre ces instances.
- Le modèle des interacteurs décrit les interacteurs qui sont présents sur la boîte à outils graphique de la plate-forme visée.
- Le dernier modèle décrit la plate-forme pour laquelle l'interface sera produite.

Cette contribution s'appuie sur cinq principes :

- les principes de description, de factorisation et de décoration qui favorisent la capitalisation et la réutilisation des connaissances, de même que la minimisation des efforts du maintien de la cohérence inter cible;
- les principes de réification et de traduction qui correspondent aux fondements de la génération et de l'adaptation d'IHM multicible.

Cette réflexion contribue à la génération d'interface plastique. Cependant, elle présente quelques limites. En particulier :

- ARTStudio, dans sa version actuelle, ne couvre pas tous les principes du cadre conceptuel. Il ne gère pas le principe de décoration et n'inclut pas d'outils et mécanismes de traduction ;

- ARTStudio n'inclut pas le modèle d'environnement. Il ne génère donc pas d'IHM capable d'adaptation à l'environnement.

Dans sa thèse, Thevenin a proposé un environnement de génération d'IHM multicible mais n'est plastique qu'au regard de la plate-forme. En outre, l'utilisateur ne peut changer de plate-forme à la volée (l'IHM doit, à chaque fois, être renvoyée à l'étape de conception pour passer à une nouvelle plateforme). ARTStudio, dans sa version actuelle, n'a pas incorporé de modèle d'évolution. Ce modèle spécifie les réactions à mettre en oeuvre en cas de changement de contexte.

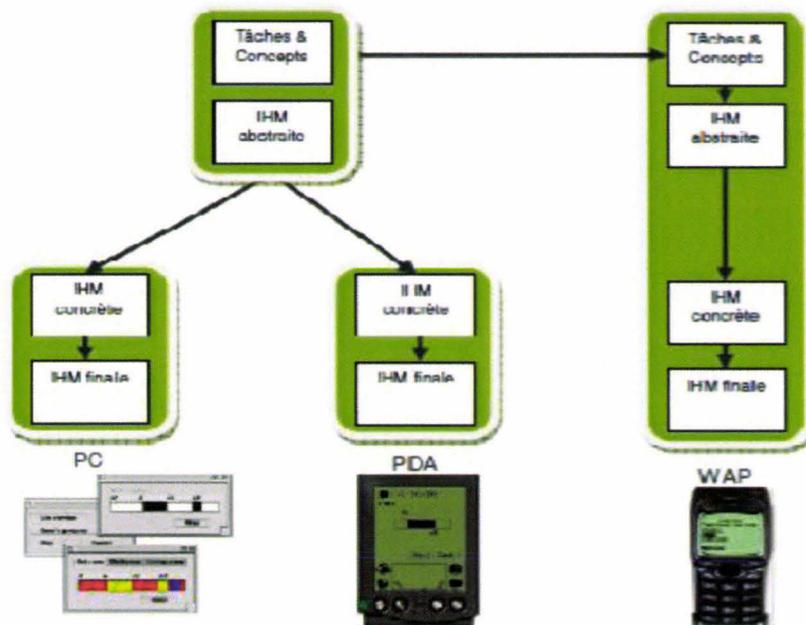


Figure 1.2. Exemple lié au processus de développement d'ARTStudio pour la production d'interfaces multiplateformes

1.1.4.2.2. MOBI-D

MOBI-D (MOdel-Based Interface Designer) [Eisenstein et al., 01] est un environnement de conception qui sert essentiellement à effectuer une conception d'interface homme-machine structurée et automatisée à partir de modèles abstraits (modèle de domaine, modèle de tâche) pour obtenir des modèles concrets (modèle de présentation, modèle de dialogue). Pour une génération de l'interface multicible à base de modèle, nous distinguons deux grandes familles de composants au sein des divers modèles :

- Les composants abstraits qui donnent une vue abstraite des tâches et du domaine.
- Les composants concrets qui se trouvent dans les modèles de dialogue et de présentation.

MOBI-D propose une base de connaissance (guide de style d'IHM, patron de conception, etc.) pour aider le concepteur dans la conception d'interface. MOBI-D intègre un certain nombre d'outils de conception et d'implémentation. Parmi ces outils, on peut citer U-TEL (User-Task Elicitation Tool) [Tam et al., 98], outil de génération du modèle de tâche de l'utilisateur directement à partir des experts du domaine (cf. Figure 1.3).

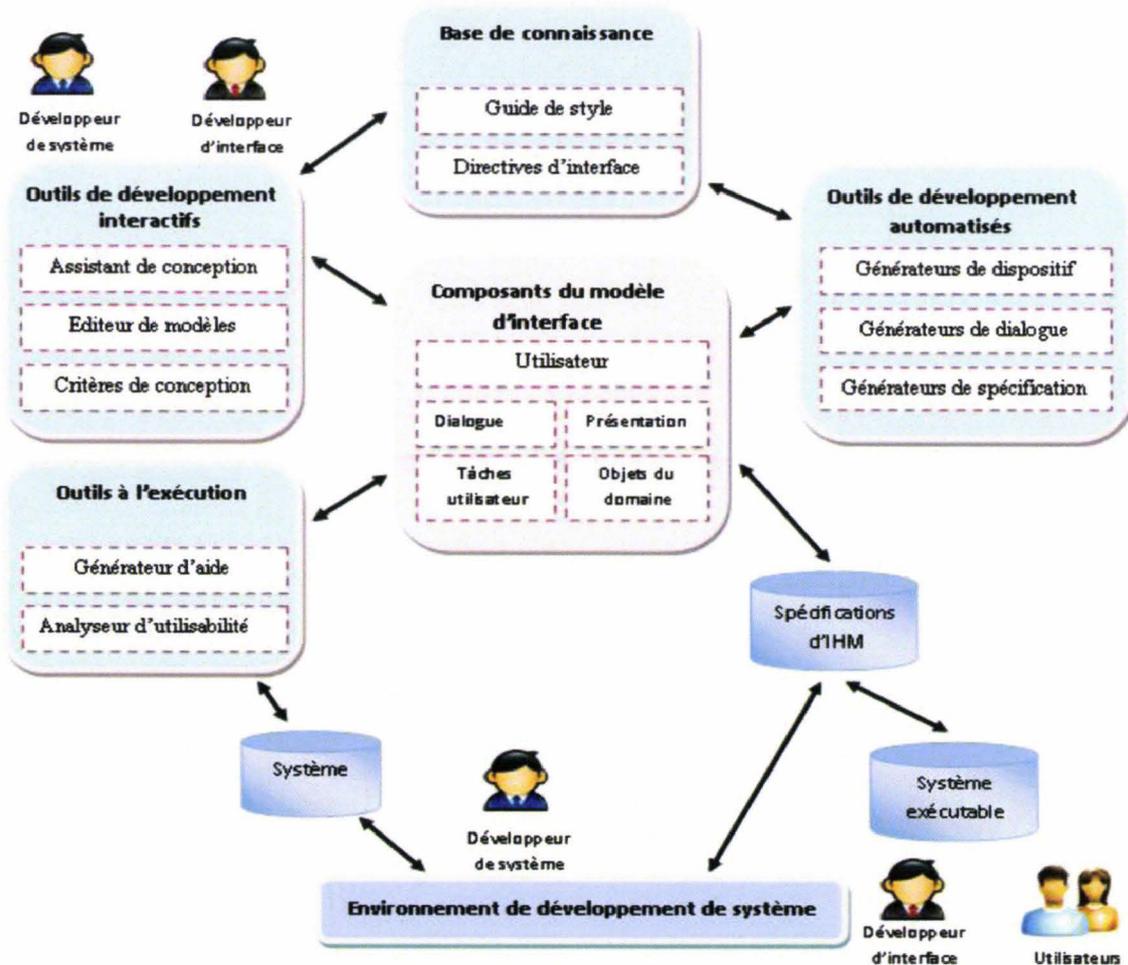


Figure 1. 3. Architecture de MOBI-D [Puerta et Eisenstein 98]

La génération de l'interface graphique par MOBI-D se fait en sélectionnant des interacteurs et une structure de présentation parmi un ensemble de structures de présentation prédéfinies. Donc l'adaptation est statique et prédéfinie par le concepteur du système. Pendant l'exécution du système en utilisant le support d'interaction cible, aucune adaptation ne sera réalisée.

MOBI-D contient une base de connaissances (guides de style, patrons de conception...) à propos de directives et de conseils opérationnels pour la conception d'interface afin d'automatiser la conception des interfaces et de venir en aide au concepteur.

1.1.4.2.3. UsiXML

UsiXML (USer Interface eXtensible Markup Language) [Limbourg 04] est un langage basé sur XML de description d'interfaces utilisateur destinées à différents contextes d'utilisation. Différents types d'interfaces sont visées : graphiques, tactiles, sonores ou multimodales¹. Ce langage s'appuie sur le framework Cameleon [Calvary et al., 03] qui, en définissant un niveau « Tâches et concepts », permet de prendre en charge les tâches de l'utilisateur mais aussi tout

¹ Une interface homme-machine multimodale combine plusieurs moyens de communication entre l'utilisateur et la machine (par exemple l'utilisation combinée du geste et de la parole).

un ensemble de concepts permettant de modéliser les utilisateurs ou encore l'environnement (cf. Figure 1.4).

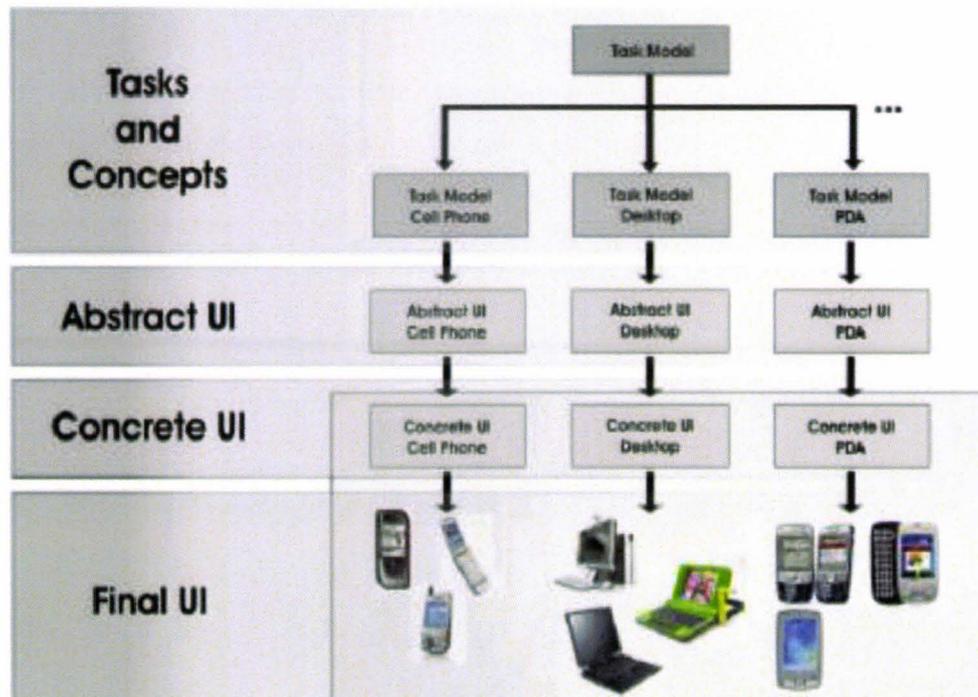


Figure 1.4. Modèle simplifié du framework associé aux interfaces graphiques en UsiXML [Francisco et al., 08]

UsiXML supporte différents modèles pendant le processus de génération d'IHM plastique (modèles de tâche, d'interface abstraite, d'interface concrète, d'interface finale, de contexte, de transformation, et enfin de correspondance). Aujourd'hui, un des grands intérêts d'UsiXML est de proposer un certain nombre d'extensions et surtout de nombreux outils permettant de modéliser et de générer des interfaces homme-machine. Ainsi, on pourra citer l'outil IdealXML [Montero 05] qui permet la modélisation de tâches, de données et du modèle d'interface utilisateur abstrait. Pour une description plus précise d'UsiXML, on pourra se référer à la thèse de [Florins 06] et au site Internet <http://www.usixml.org>.

1.1.4.2.4. L'approche par Comet

Cette approche a introduit la notion de comet (*Context Mouldable widETs*), c'est-à-dire d'interacteur plastique, qui est définie comme une entité logicielle et/ou matérielle rendant possible à l'utilisateur la réalisation d'une tâche [Calvary et al., 03] [Sottet et al., 06].

Une comet est composée d'abstractions, d'interfaces abstraites (IU abstraites), d'interfaces concrètes (IU concrètes) et d'interfaces finales (IU finales).

L'originalité de cette approche est qu'elle essaie de tirer profit simultanément des approches à base de modèles et des architectures logicielles. Elle propose de regrouper l'ensemble des spécifications dans le même composant (une Comet) en le dotant éventuellement de capacités d'adaptation. En revanche, à notre connaissance, cette approche n'est pas supportée par des outils de conception et d'évaluation des interfaces plastiques.

1.1.4.2.5. L'approche Amacont

L'approche Amacont (System Architecture for Multimedia Adaptive Web Content) de [Hinz et Fiala, 04] vise à permettre à des applications web de s'adapter automatiquement à la plateforme utilisée par l'utilisateur. Le principe général de l'architecture Amacont est présenté à la Figure 1.5.

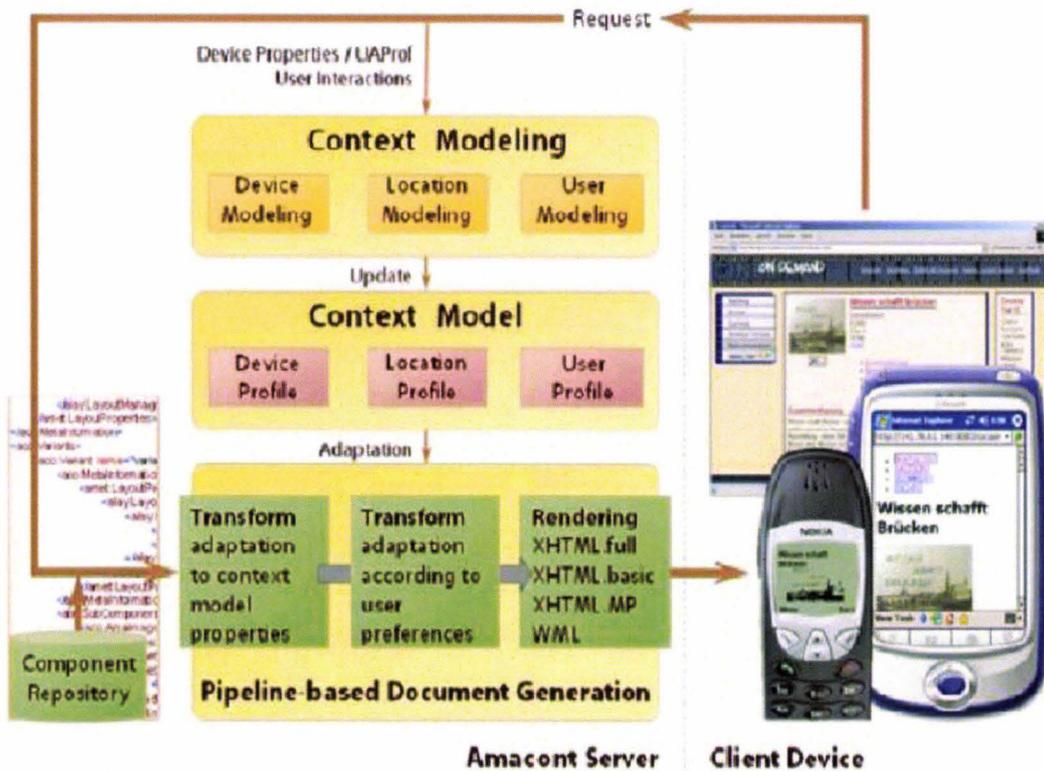


Figure1. 5. Architecture globale de l'approche Amacont [Hinz et Fiala, 04]

De manière globale, Amacont propose une approche et une architecture permettant à des applications web de s'adapter à la plateforme utilisée, au lieu où se trouve l'utilisateur et enfin à l'utilisateur lui-même. Tout ceci passe par la création de trois types de modèles qui sont appliquées sur l'application elle-même à travers des transformations de type XML. Ainsi, toute page web affichée à un utilisateur provient d'une page web générique qui est d'abord adaptée au contexte global, puis à l'utilisateur avant d'être générée dans le langage informatique le mieux adapté à la plateforme de consultation de l'utilisateur.

1.1.4.2.6. Approche de Hariri

La méthode proposée par A.Hariri s'intéresse aux interfaces homme-machine ayant la capacité de s'adapter de manière dynamique à leur contexte d'usage en prenant en compte les changements contextuels sans nécessité de retour à la conception [Hariri et al., 08] (cf. Figure 1.6).

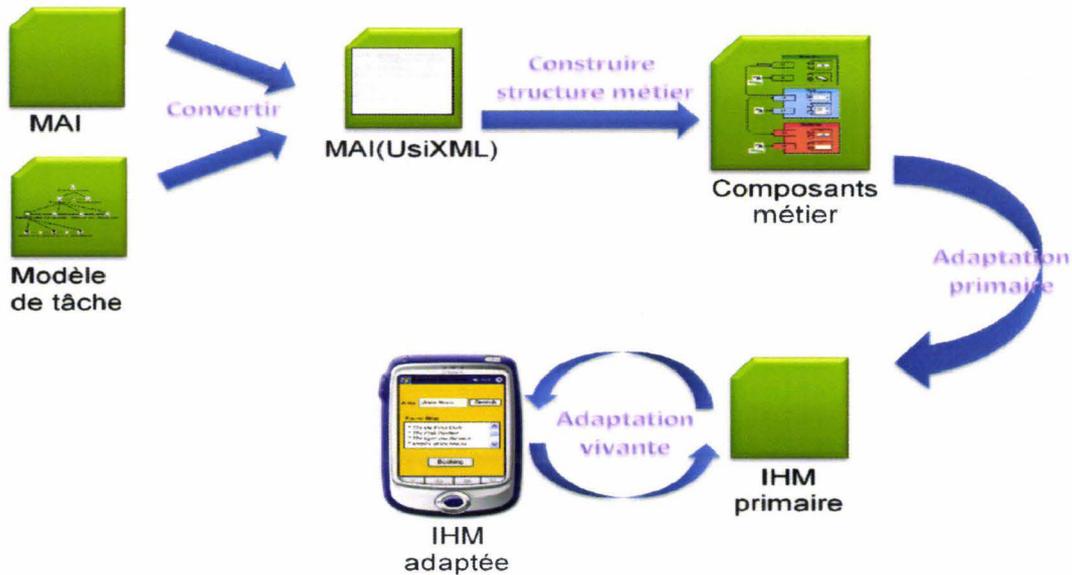


Figure 1.6. Méthode globale de génération des IHM plastiques

Cette contribution consiste à générer une telle IHM à partir d'un modèle abstrait d'IHM spécifié déjà dans une méthode de spécification et conception de systèmes interactifs et/ou un modèle de tâche. Cette méthode s'appuie sur la notion de patrons de conception. Ceux-ci sont utilisés au niveau du passage à l'interface concrète et pendant l'adaptation. L'architecture du système s'appuie sur une composition basée sur les composants métier. Ceux-ci ont la capacité de changer dynamiquement leur facette de présentation. Ce principe est adopté en tant que solution pour l'adaptation dynamique au contexte d'usage. L'IHM est considérée comme un ensemble de composants de présentation. Les composants métier peuvent dynamiquement choisir et changer les composants de présentation de l'IHM, afin de s'adapter à un nouveau contexte, sans nécessité, donc, de retourner à l'étape de conception. Cette méthode s'appuie également sur la notion d'apprentissage. L'intégration d'une technique d'apprentissage permet de continuer à développer la base de connaissance du système à l'exécution afin de préserver l'utilité de l'adaptation.

1.1.4.2.7. Conclusion sur les méthodes d'adaptation d'IHM multiplateforme et/ou orientée contexte

Nous avons passé en revue un certain nombre d'approches d'adaptation d'IHM multiplateforme et/ou orientée contexte. Le Tableau 1.2 présente ces approches en spécifiant les modèles pris en compte, le type de contexte cible, les langages de spécification, et les outils associés.

Tableau 1.2. Approches d'adaptation d'IHM multiplateforme et/ou orientée contexte

	Les modèles pris en compte	Le type de contexte	Les langages de spécification	Les outils associés
Approche de Thevenin	Domaine, Tâches, IHM abstraite, IHM concrète, IHM finale	Plateforme	UML, XML	ARTStudio
MOBI-D	Modèle de domaine, modèle de tâche, modèle de présentation, modèle de dialogue, IHM finale	Plate-forme	MIMIC	U-TEL, model editors, TIMM, interface builder
UsiXML	Domaine, Tâches, IHM abstraite, IHM concrète, IHM finale, Contexte, Transformation, Correspondance	Plateforme, Utilisateur (limité), Environnement (limité)	UsiXML	GrafiXML, VisiXML, SketchiXML, IdealXML, PlastiXML, ComposiXML, KnowiXML, RenderXML, ReversiXML, FlashiXML, QtkiXML, TransformiXML, Vaquita.
L'approche par Comet	Domaine, tâches, IUA, IUC, IUF, Transformation, Evolution	Plate-forme, Utilisateur, Environnement	Non	Non
L'approche Amacont	Modèle utilisateur, modèle plateforme	Plate-forme, Utilisateur, Environnement	XML	Oui, dans le cadre des travaux de recherche, mais les outils ne sont pas librement disponibles.
Approche de Hariri	Modèle abstrait, modèle de tâche, interface primaire, interface adaptée	Plate-forme, Utilisateur, Environnement	UsiXML	PaDev

De nombreuses approches d'adaptation existent. La plupart d'entre elles sont à base de modèles et se concentrent sur le modèle de tâches comme modèle central pour l'adaptation. Chaque approche possède un certain nombre de modèles qui couvrent le cycle de développement d'IHM adaptée. Ces approches visent l'adaptation des systèmes interactifs au contexte d'usage. Elles se différencient par leurs capacités d'adaptation de l'IHM, par l'évolution de l'adaptation, et aussi par la gamme du contexte d'usage prise en compte. Aujourd'hui, et dans un objectif de normalisation, des travaux considérables sont menés (autour de l'OMG), pour reproduire les approches à base de modèles sur l'architecture MDA (Model Driven Architecture) [Sottet et al., 05].

Cependant la possibilité d'évaluer et d'améliorer la qualité des réactions d'adaptation est absente dans les approches présentées ;

1.2. Etude des systèmes de personnalisation existants

Plusieurs systèmes contribuent à des avancées dans le domaine de la personnalisation en interaction homme-machine. La plupart de ces systèmes sont destinés à l'aide à la navigation sur le web. Ils assurent les fonctions d'observation du comportement utilisateur, de recherche et de présentation de ces informations.

1.2.1. Critères d'étude des systèmes d'information personnalisée

Afin d'étudier les systèmes d'informations personnalisés existants, nous proposons quatre critères : la méthode de collecte d'information, le type de personnalisation visé, la méthode de personnalisation, la méthode d'évaluation. Ces critères ont été définis par rapport à notre objectif de personnalisation et d'évaluation des systèmes d'information.

La méthode de collecte d'information : pour pouvoir effectuer de la personnalisation, il est nécessaire de connaître les préférences et les besoins de l'utilisateur. Deux moyens peuvent être utilisés pour collecter les informations à propos de l'utilisateur : la collecte explicite d'information et la collecte implicite d'information (cf. §1.1.2).

Méthode de personnalisation : les méthodes de personnalisation sont utilisées pour l'exploitation des modèles utilisateur. Elles permettent de fournir à un utilisateur un contenu et/ou un contenant personnalisé. Les systèmes sont généralement conçus avec une ou plusieurs méthodes de personnalisation dédiées et figées pour répondre à un type de personnalisation bien défini. (cf. §1.1.4) pour plus de détails sur les méthodes utilisées pour la personnalisation. Nous avons vu que cette évaluation peut porter sur la personnalisation de contenu et/ou de contenant.

Le type de personnalisation visé : la personnalisation peut porter sur le contenu (données, informations, document, etc.) que sur le contenant (présentation, plate-forme d'interaction, mode de communication, canal de communication, etc.). On notera qu'il y a des systèmes d'information qui assurent les deux types de personnalisation.

Méthode d'évaluation intégrée : les méthodes d'évaluation sont utilisées pour l'évaluation de la qualité de personnalisation de système. Ces méthodes permettent de vérifier si les résultats fournis par les systèmes correspondent bien aux attentes des utilisateurs.

1.2.2. Systèmes de personnalisation existants

De nombreux systèmes permettent la personnalisation de l'interaction. Cette section est destinée à présenter une vue globale de systèmes représentatifs existants.

Syskill & webert [Pazzani et al., 96] : c'est un système de recommandation qui permet d'assister l'utilisateur de sites web en lui sélectionnant des liens pertinents d'une page. Ce système représente le profil de l'utilisateur sous forme de classes. Chaque classe représentant un centre d'intérêts, est modélisée à l'aide d'un vecteur booléen des mots clés. Les sélections des pages à recommander est basée sur un calcul de probabilités de prédiction issues d'un classifieur bayésien.

GroupLens [Resnick et al., 94] : il s'agit d'un système qui permet à ses utilisateurs de trouver des articles les intéressant parmi un nombre important d'articles disponibles. Il repose sur une idée simple : « les personnes d'accord avec leurs évaluations des articles passées le seront dans le futur ». Pour connaître les évaluations des utilisateurs, GroupLens leur demande de

noter les articles qu'ils ont lus sur une échelle de notes de 1 à 5, où 5 est la note la plus élevée. Ces notes sont ensuite traitées de deux manières. La première consiste à calculer la corrélation entre les articles afin de déterminer les plus similaires entre eux. Ensuite, il détermine dans quelle mesure les utilisateurs vont être intéressés par les nouveaux articles, connaissant les évaluations d'utilisateurs similaires, pour ainsi prédire les articles considérés comme pertinents aux utilisateurs.

Web personae [Gowan 03] : Web personae est un système qui interagit en mode off line avec des applications Web dans le but de fournir un accès personnalisé. Il est composé de deux parties : le constructeur et l'identificateur de profil. Le constructeur construit progressivement, le modèle de l'utilisateur comme une liste de profils correspondant à diverses préférences inférées à partir de ses interactions avec des applications web. L'identificateur permet de découvrir le profil courant de l'utilisateur lié à une session d'utilisation courante. Web personae combine ainsi les deux types de personnalisation.

Smart Radio [Hayes et Cunningham 01], [Hayes et Cunningham 04] : ce système de recommandation de musique sur le Web, via la technologie de *Streaming Audio* développée par RealNetworks (les morceaux musicaux sont joués à partir d'une playlist sans possibilité de télécharger les titres que l'on écoute, ceci évite le piratage) permet aux utilisateurs d'écouter leur musique préférée ou un programme musical recommandé. Afin de connaître leurs goûts musicaux, Smart Radio demande à ses utilisateurs de noter de 1 à 5 (5 étant la note la plus élevée) un ensemble de morceaux musicaux, ou des programmes musicaux, afin de constituer des profils utilisateurs. L'utilisateur a ainsi la possibilité de se constituer une playlist personnelle pour une écoute ultérieure de ces morceaux préférés. Mais, grâce à ce profil, le système peut trouver un ou plusieurs utilisateurs ayant des goûts communs en matière de musique et ainsi recommander à chacun la playlist des autres.

Letizia [Lieberman 95] : est un assistant intelligent pour le parcours du web qui permet de recommander des liens à explorer en fonction de la page en cours de visite par l'utilisateur. Il construit le profil de l'utilisateur d'une façon implicite à partir du contenu des pages explorées par celui-ci. Puis, il recherche d'autres pages susceptibles d'intéresser l'utilisateur et présente ses résultats sur une fenêtre indépendante. La sélection des pages à recommander est basée sur le calcul d'un score d'appariement vectoriel entre le contenu de la page et le profil ainsi construit.

InfoSleuth [Bayardo et al., 97] [Nodine et al., 03] : ce système est dédié à la recherche d'information dans les environnements dynamiques (en constante évolution), hétérogènes et distribués. L'architecture d'InfoSleuth comprend différents modèles d'agents (User, Ontology, Broker, Ressource, Data Analysis, Task Execution, Query et Monitor) qui communiquent et coopèrent entre eux pour la personnalisation.

PRIAM [Jacquet et al., 06] : cet outil propose de nouveaux types d'interactions entre les systèmes mobiles et les panneaux d'affichage public en vue de mieux guider les usagers en déplacement dans un aéroport. Ce type de personnalisation permet à l'utilisateur de plus rapidement s'orienter dans un aéroport et surtout facilite l'accès à l'information ; celle-ci n'étant pas affichée suite à une demande explicite de l'utilisateur mais parce que le système sait que l'utilisateur en a probablement besoin.

1.2.3. Synthèse et discussion

De nombreux travaux sont menés pour le développement de systèmes de personnalisation (cf. Tableau 1.3). Ces systèmes se basent sur des méthodes de collecte d'information (explicite ou implicite) et des méthodes de personnalisation pour fournir à un utilisateur un contenu et/ou un contenant personnalisé. Nous distinguons deux familles d'approches pour personnaliser un système interactif.

La première famille regroupe les approches qui combinent le système d'information (SI) (application) et le système de personnalisation (SP). Dans ce cas, les fonctions de collecte d'information, de filtrage et de présentation de services personnalisés sont assurés par le SIP comme avec SmartRadio, Letizia et InfoSleuth.

La deuxième famille rassemble les approches qui séparent le SI du SP. Dans ce cas le SP s'occupe de collecter et gérer les préférences des utilisateurs et les applications tierces ou SI qui interagissent avec le SP qui se charge de présenter à l'utilisateur les services personnalisés comme dans GroupLens, Web personae et Broadway.

En revanche, l'étude de ces systèmes montre que la fonction d'évaluation de la qualité de personnalisation est négligée. Ceci diminue la fiabilité des systèmes qui sont proposés et ne permet pas d'envisager des campagnes d'évaluation effectives pour la comparaison entre systèmes.

Tableau 1.3. Etude comparative des systèmes de personnalisation

Systèmes	Méthode de collecte d'information	Type de personnalisation visé	Méthode de personnalisation	Méthode d'évaluation intégrée
GroupLens	Explicite	Contenu	Filtrage collaboratif	?
Web personae	Implicite	Contenu	Filtrage collaboratif	?
Smart Radio	Explicite	Contenu/ contenant	Pearson	?
Syskill & webert	Explicite	Contenu	Classifier bayésien	?
Letizia	Implicite	Contenu	TF-IDF	?
InfoSleuth	Implicite/ Explicite	Contenu	Ontologie	?

“ ? ” signifie que la fonction d'évaluation de la qualité de personnalisation n'est pas ou peu souligné par le système étudié et qu'il y a une interrogation à laquelle la littérature ne nous a pas fourni de réponse.

1.3. Contexte applicatif : la personnalisation de l'information transport

Notre travail s'inscrit dans le cadre applicatif de la personnalisation de l'information dans les transports collectifs de personnes, dans le but, d'une part, de faciliter l'usage et l'accès aux modes de transport collectifs et, d'autre part, d'améliorer le confort des usagers en les accompagnant dans leurs déplacements [Uster 01]. L'information à délivrer doit permettre la préparation d'un déplacement et son suivi. La préparation de déplacement répond aux questions de choix des modes de transports à utiliser et de leur chaînage: "quel(s) mode(s) de transport utiliser pour se rendre d'un point A à un point B ? A quels horaires ? Avec quel(s) changement(s) ? A quel coût ?" etc. Le suivi de déplacement doit permettre d'informer l'utilisateur des modifications éventuelles à apporter en temps réel à son déplacement en fonction des événements extérieurs tels qu'une perturbation du trafic, un incident, etc.

Différents champs d'étude sont donc ouverts dans ce contexte applicatif : recueil de données en réseau; intégration de données hétérogènes; planification; personnalisation de l'information; temps réel.

Notre intervention se situe au niveau de la personnalisation et de l'évaluation de système d'information multimodale² personnalisée à mettre à la disposition des usagers. Ce travail s'inscrit dans des recherches réalisées en collaboration avec l'Institut national de recherche sur les transports terrestres (INRETS), depuis la fin des années 90 [Petit-Rozé 03], [Anli 06d], [Brossard 08].

1.3.1. Apport de la personnalisation dans les systèmes d'informations au profit des usagers des transports collectifs de personnes

Un système d'information est destiné à fournir de l'information à un utilisateur. En fait, il devrait, idéalement, permettre à l'utilisateur de récupérer de l'information à partir des données auxquelles a accès le système. Or, cette transformation des données en information, à savoir cette plus-value apportée aux données qui sont triées, classées, validées, est bien souvent négligeable : le système laisse à l'utilisateur la charge de retrouver l'information dans la masse de données qui lui est fournie. La personnalisation des informations tente de répondre à cette problématique [Bouzeghoub 03]. Elle permet d'aider l'utilisateur dans sa démarche de recherche d'informations en lui fournissant un résultat personnalisé.

En outre, des nouvelles avancées technologiques sont apparues aujourd'hui offrant à l'utilisateur la possibilité d'accéder à l'information en utilisant des supports de plus en plus variés et supportant des modes d'interaction de plus en plus différents. Pour permettre à l'utilisateur de retrouver l'information en tenant compte de ces possibilités technologiques (interaction multimodale, réalité virtuelle, objets nomades, images en 3D, réseaux à large bande), la personnalisation se présente comme une solution appropriée. En effet, la surface de l'interface d'un assistant personnel numérique (PDA) n'est pas la même que celle d'un ordinateur, c'est pourquoi il est nécessaire d'adapter la quantité et la forme des informations, la navigation dans l'interface, le placement des objets graphiques, etc., selon la cible visée. Ainsi, l'utilisateur peut recevoir de l'information sur son ordinateur de bureau, son PDA, son téléphone portable, etc. La plupart de ces plateformes d'accueil permettent une interaction multi-modale combinant le son, l'image et le texte.

D'autre part, les utilisateurs de nouvelles applications sont très divers, de par leur âge, connaissances, compétences, centres d'intérêts, besoins, préférences, etc. Parmi ces utilisateurs, on en trouve un ensemble présentant des handicaps physiques ou intellectuels aussi bien ceux qui présentant des déficiences. Pour répondre à leurs besoins, le système doit être capable de fournir une réponse personnalisée. Prenons l'exemple d'un utilisateur non-voyant : la déficience sera compensée par un système de synthèse vocale en utilisant le téléphone portable plutôt que de lire l'information délivrée sur un PDA.

Pour les industriels (transporteurs, exploitants, développeur de services, etc.) ainsi que pour les agglomérations, la personnalisation représente un atout en matière de qualité de service qui peut fortement améliorer l'attractivité des transports en commun [Perreau 02] et en augmenter ainsi la fréquentation [Uster 00], [Uster 01].

² La multimodalité est l'offre de plusieurs moyens de transport pour un déplacement entre une origine et une destination.

1.3.2. Systèmes d'informations personnalisés existants dans le domaine de transport

Gulliver's Genie [O'Hare et O'Grady 03], [O'Grady et al., 05] est un système développé à l'UCD (University College Dublin) qui permet d'assister les utilisateurs dans la découverte des endroits touristiques ou culturels. Gulliver's Genie est construit à partir de différents modèles d'agents et adopte un mode de traitement client/serveur. Le client est un PDA étendu d'un GPS et d'un GPRS. Le serveur est une station de travail (PC) accessible sur internet. Ce système se base sur le modèle BDI (Beliefs, Desires and Intentions) [Rao et Georgeff 91] pour déduire l'état mental de l'utilisateur.

RATP (Régie Autonome des Transports Parisiens) : le système RATP est un système d'information sur les horaires, les tarifs, l'état de trafic et de calcul d'itinéraire en île de France. Il permet notamment une recherche d'itinéraire, de porte à porte, sans que l'utilisateur ait à consulter plusieurs sources d'information. Ce système sauvegarde les préférences de l'utilisateur pour les utiliser dans une autre connexion. En outre, il permet de diffuser l'information sur un PDA ou téléphone portable. Par exemple, lorsqu'une perturbation supérieure à quinze minutes se produit sur l'une des lignes prédéterminées, l'utilisateur abonné reçoit un message écrit (SMS) indiquant la nature du problème et la durée estimée du retard affectant le trajet.

AGENPERSO (AGENTS logiciels PERSONnels) [Petit-Rozé et al., 03a, Petit-Rozé et al., 03b] : développé dans le cadre d'une collaboration entre notre laboratoire et l'INRETS, ce système se base sur une organisation multi-agents pour la personnalisation de l'information. Il fournit une aide personnalisée dans la préparation d'itinéraires combinant un ensemble de modes de transport (bus, tramway, métro, marche, etc.). Le système se base sur les préférences de l'utilisateur pour lui fournir l'itinéraire correspondant à son profil. Le profil de l'utilisateur correspond aux poids associés à quatre critères (le plus rapide, le plus court chemin, le moins cher et le moins de correspondance) et aux poids associés aux modes de transport (bus, taxi, TER, tramway, marche et métro).

TISONI (Traveller Information System ON Internet) [Lam et Xie 02] : Ce système permet de fournir une information personnalisée en combinant plusieurs modes de transport dans l'Ouest de Singapour. Le système supporte une architecture trois-tiers : la base de données, le moteur de calcul et l'interface utilisateur. L'utilisateur s'inscrit en fournissant ses informations personnelles comme l'âge, son revenu, sexe, vitesse de marche, etc. Le système propose le chemin préféré par l'utilisateur et affiche sur une carte les différents itinéraires en indiquant les arrêts, les temps d'arrêt, ou encore les temps de parcours.

PIEPSER (Personalised information on disruptions to public transport exclusive to users of public transport) [Hoyer et Czogolla 02]: ce système permet de prévenir les usagers des retards et des interruptions éventuelles dans les transports en Allemagne. Le système est personnalisé par rapport aux perturbations des transports. Le passager (régulier) s'abonne au service en donnant l'heure et le lieu de départ, et choisit les lignes de correspondance. Il donne aussi son numéro de portable. Des alternatives combinant plusieurs modes de transport sont proposées (par SMS) à l'utilisateur dans le cas d'une interruption d'un moyen de transport.

PERSYST (PERsonalization SYSTem) a été proposé au sein du LAMIH (en collaboration avec Archimed et l'INRETS) [anli 06d]. PerSyst est un système de personnalisation qui est bâti sur Agenperso (cf. ci-dessous). Il évolutif et distribué pour le développement de système d'information personnalisé. Cette personnalisation couvre deux aspects : la personnalisation

de contenu et celle de contenant. PerSyst est construit à partir d'une architecture multi-agents ce qui le prédispose à des caractéristiques comme l'adaptabilité, l'autonomie et l'assistance lui permettant de prendre en compte différents types de personnalisation. Les services fournis par PerSyst sont dédiés à la personnalisation des itinéraires dans les transports terrestres de personnes [Anli et Abed 05].

Tableau 1.4. Etude comparative de systèmes de personnalisation représentatifs dans le domaine du transport

Systeme	Observations	
Gulliver's Genie	Services fournis	Guide touristique
	Information multi-modale	Non
	Type de personnalisation	Contenant (diffusion d'information sur PDA)
	Méthode d'évaluation intégrée	Non
RATP	Services fournis	Informations sur les horaires, les tarifs, l'état de trafic et de calcul d'itinéraire en île de France
	Information multi-modale	Oui
	Type de personnalisation	Contenu/ Contenant (diffusion multi-cibles : mail, wap, SMS)
	Méthode d'évaluation intégrée	Non
AGENPERSO	Services fournis	Information sur la recherche d'itinéraire dans la région Nord Pas-de-Calais.
	Information multi-modale	Oui
	Type de personnalisation	Contenu
	Méthode d'évaluation intégrée	Non
TISONI	Services fournis	Information sur la recherche d'itinéraire dans l'Ouest de Singapour
	Information multi-modale	Oui
	Type de personnalisation	Contenu
	Méthode d'évaluation intégrée	Non
PIEPSER	Services fournis	Prévenir les usagers des retards et des interruptions éventuelles dans les transports en Allemagne.
	Information multi-modale	Oui
	Type de personnalisation	Contenu/ Contenant
	Méthode d'évaluation intégrée	Non
PerSyst	Services fournis	Information sur la recherche d'itinéraire dans la région Nord Pas-de-Calais
	Information multi-modale	Oui
	Type de personnalisation	Contenu/ Contenant
	Méthode d'évaluation intégrée	Non

1.3.3. Synthèse et discussion

De nombreux systèmes d'information transport personnalisée existent. Ces systèmes visent à personnaliser l'information « avant », « pendant » et parfois même « après » le déplacement. L'objectif de ces systèmes est de mettre à la disposition des usagers des transports, des informations pertinentes leur permettant de préparer leur déplacement en leur présentant toute l'offre disponible sur l'itinéraire demandé en termes de choix de mode de transport, de temps de parcours, de prix, etc., de l'accompagner tout au long de leur déplacement ou voyage, en offrant l'ensemble des éléments susceptibles de réduire leur incertitude tout en prenant en compte la diversité des plates-formes d'interaction dont ils disposent. Il s'agit donc de fournir

à l'utilisateur « *toute l'information nécessaire et uniquement l'information nécessaire* » [Uster 98].

La plupart de ces systèmes présente aussi l'avantage de fournir une information multimodale qui combine plusieurs modes de transports (métro, bus, TER, tramway, etc.). Cependant, les systèmes existants sont généralement destinés à un type particulier de personnalisation. Il est très rare de retrouver un système qui assure aussi bien la personnalisation du contenant que du contenu.

L'étude de ces systèmes a montré qu'il n'y a pas une méthode d'évaluation intégrée permettant d'examiner l'efficacité de ces systèmes. Pour combler ce manque, il serait intéressant d'incorporer une méthode d'évaluation conjointement avec la méthode de personnalisation ou bien de disposer d'une méthode générique qui supporte les deux fonctions. La méthode à intégrer doit être capable d'assurer l'évaluation de contenu et/ou contenant personnalisés fournis à l'utilisateur.

Conclusion

L'objectif de nos travaux est de fournir une batterie de méthodes dédiées pour l'évaluation des systèmes d'information personnalisés. Ces méthodes devraient garantir l'évaluation de ces systèmes en comparant les services fournis à l'utilisateur par rapport à ceux qu'ils attendent. Ce chapitre a présenté un état de l'art sur la personnalisation dans les systèmes d'information. Nous avons fait le constat d'une terminologie riche associée à personnalisation. Nous avons mis en relations les termes employés (customisation, adaptation, adaptabilité, adaptativité, personnalisation, etc.). Ceux-ci correspondent le plus souvent à des façons différentes d'envisager le processus visant à personnaliser le système à l'utilisateur.

Tout processus de personnalisation, quel qu'il soit, repose sur un modèle de l'utilisateur, c'est-à-dire une représentation de ses préférences dont le système tient compte dans la mise en oeuvre de la personnalisation. Les différentes façons d'envisager une telle modélisation des utilisateurs ainsi que les méthodes employées (explicite, implicite) pour construire ce modèle ont été décrites.

Enfin, nous avons présenté plusieurs travaux représentatifs visant à illustrer l'ensemble des propos tenus dans ce chapitre. Nous avons ainsi exposé brièvement quelques propositions faites en matière de personnalisation dans les domaines de la recherche d'information. La plupart de ces travaux visent à fournir des approches, des modèles et des méthodes pour la collecte d'information utilisateur, la gestion des profils utilisateurs et l'adaptation de l'interaction par rapport à ses profils. Mais la fonction d'évaluation de système d'information personnalisé est généralement négligée. Ceci nous incite à poser la question suivante : existe-t-il des méthodes d'évaluations dédiées pour les SIP ? Dans le chapitre suivant, nous allons essayer de répondre à cette question à travers l'étude des méthodes d'évaluations existantes dédiées pour les systèmes interactifs en général.

De l'évaluation des systèmes interactifs à celle des systèmes d'information personnalisés

Sommaire

Introduction	40
2.1. L'évaluation : définitions et principes.	41
2.1.1. Définitions	41
2.1.2. Les dimensions et les critères d'évaluation existants	43
2.1.3. Les modèles de développement enrichis sous l'angle de l'évaluation des systèmes	45
2.1.4. Synthèse	46
2.2. Méthodes, techniques et outils d'évaluation des systèmes interactifs	46
2.2.1. Les critères proposés pour l'évaluation de système d'information personnalisé	47
2.2.2. Choix d'une classification des méthodes d'évaluation des systèmes interactif	49
2.2.3. Etude de méthodes, techniques et outils d'évaluation des systèmes interactifs	50
2.2.4. Synthèse et discussion	67
Conclusion.....	67

Introduction

Il existe énormément de méthodes et techniques, destinées à l'évaluation de système interactif. La littérature est très riche à ce sujet. Toutes présentent des avantages et des inconvénients et aucune d'elle ne peut prétendre à une évaluation exhaustive d'un système. Avec l'apparition de nouvelles architectures et de nouveaux systèmes d'information personnalisés, de nouvelles problématiques d'évaluation de ceux-ci apparaissent. Parmi les problèmes les plus traités actuellement nous pouvons citer : le manque de méthodes d'évaluation dédiées aux systèmes d'information personnalisés (SIP), l'absence d'outils (automatiques ou non) d'aide à l'évaluation destinés aux SIP.

L'objectif de ce chapitre est de présenter un état de l'art sur la notion d'évaluation dans les systèmes interactifs tout en ayant en visée celle de systèmes d'information personnalisés. Dans ce but, une première partie de ce chapitre illustre le principe de l'évaluation ainsi que les critères d'évaluation existants relativement aux systèmes interactifs.

Dans la seconde partie nous présentons une étude des méthodes d'évaluation existantes en fonction des critères de personnalisation que nous proposons pour l'évaluation de SIP. En effet, pour les présenter, nous retiendrons la classification de [Huart, Kolski et Bastien 08] qui s'inspire de [Whitefield 91]. Cette classification classe les méthodes d'évaluation en plusieurs types de méthodes, selon que l'utilisateur et le système sont réels ou représentés. Il est important de préciser que notre objectif dans cette partie n'étant pas de faire un recensement exhaustif de toutes les méthodes d'évaluation existantes dans la mesure où on retrouve plusieurs documents qui traitent ce sujet, mais plutôt de passer en revue un ensemble de méthodes d'évaluation les plus représentatives.

Le SIP se différencie du système interactif par des propriétés spécifiques (sa capacité à s'ajuster par rapport au contexte d'usage, sa capacité d'acquérir l'information à propos de l'utilisateur et sa capacité de lui fournir des contenus et/ou des services adaptés à ses besoins et à ses attentes). Pour cela, les méthodes existantes seront étudiées dans le but de les enrichir afin de supporter l'évaluation des caractéristiques d'un système personnalisé.

2.1. L'évaluation : définitions et principes

Dans cette partie, nous présenterons les définitions et les principes d'évaluation dans les systèmes interactifs, les critères d'évaluation existants selon lesquels un système interactif peut être évalué ainsi que les modèles de développement enrichis sous l'angle de l'évaluation des systèmes.

2.1.1. Définitions

Actuellement, il n'existe pas de définition standard, dans le domaine de la recherche, sur la notion d'évaluation de système interactif. Ainsi, pour [Huart, Kolski et Bastien 08] l'évaluation d'un système homme-machine englobe sa vérification et sa validation. Le système homme-machine est vérifié s'il correspond aux spécifications issues de la définition des besoins. Il est validé s'il correspond aux besoins en respectant les contraintes du domaine d'application. Selon [Bodic 05] l'évaluation peut alors consister en une validation de l'adéquation entre la réalisation fonctionnelle et le scénario d'usage. Des expérimentations réelles, jouant ces scénarios, pourront être réalisées afin de juger du système interactif, selon les critères en question. Pour [Senach 90], « toute évaluation consiste à comparer un modèle de l'objet évalué à un modèle de référence permettant d'établir des conclusions ». D'après le principe illustré en figure 2.1, on applique au système interactif à évaluer des méthodes et techniques d'évaluation en se basant sur des critères d'évaluation ; il s'agit aussi de réaliser

les analyses nécessaires sur les données recueillies (exécution de statistiques, détermination des violations selon les règles d'évaluation, etc.) pour obtenir un modèle appelé « modèle observé ». Ce modèle va être comparé au modèle de référence qui doit être représentatif de l'interface à évaluer par rapport aux besoins spécifiques déterminés par le concepteur. Le résultat de la comparaison doit permettre au concepteur d'estimer le niveau d'adéquation du système interactif par rapport aux besoins déjà déterminés et aider le concepteur à proposer les recommandations pour améliorer celui-ci.

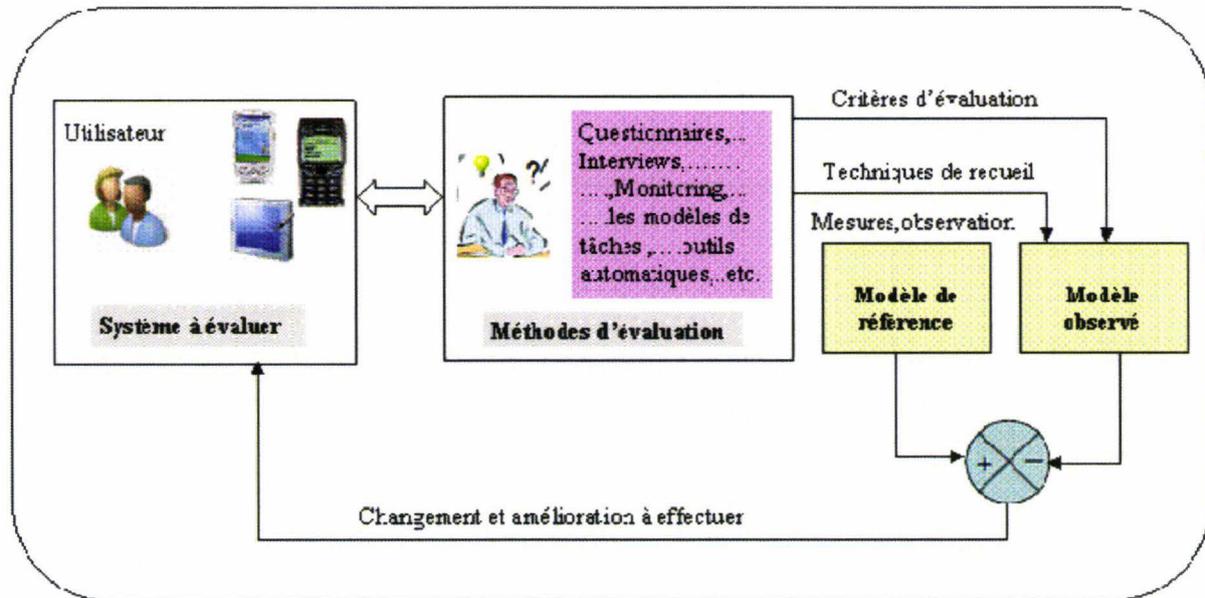


Figure 2.1. Principe de base de l'évaluation inspiré de [Senach 90 ; Grislin 95]

Pour notre part, en nous plaçant dans le cadre de l'évaluation des systèmes d'information personnalisés, l'évaluation consiste à s'assurer que le système fournit à l'utilisateur une information pertinente via une interface adaptée. Il en découle que l'évaluation de la personnalisation peut se faire à deux niveaux :

- Au niveau des contenus et des services ; ceux-ci pouvant être évalués afin de s'assurer que les prédictions proposées par le système correspondant effectivement aux attentes et préférences des utilisateurs.
- Au niveau de l'interface utilisateur ; celle-ci s'adaptant en fonction de préférences de l'utilisateur liée au contenant et du contexte dans lequel est utilisée l'application. L'évaluation de l'interface consiste à vérifier sa capacité d'adaptation dynamique (changement du style d'affichage, changement de la couleur d'arrière plan, etc.) lors du changement contextuel.

Avant d'évaluer, il convient de définir tout d'abord quelles seront les variables à observer et bien sur la manière de les recueillir. L'analyse de l'IHM ou du système interactif doit mettre à jour un ensemble de variables cibles, c'est-à-dire les données à recueillir. Ces sont en effet, le système à évaluer et son utilisation qui définissent le comportement que l'on mesure lors d'une évaluation. Il s'agit de déterminer ce qui permettra de porter un jugement sur l'interface concernée. C'est à ce niveau qu'interviendront les critères d'évaluation sur lesquels se focalise la partie suivante.

2.1.2. Les dimensions et les critères d'évaluation existants

Afin d'améliorer la qualité des interactions entre l'homme et la machine, des études récentes sont orientées vers l'évaluation des systèmes interactifs sous l'angle de deux propriétés principales. Ces propriétés sont l'utilité et l'utilisabilité. Ces deux dimensions sont habituellement explorées dans l'évaluation des interfaces homme-machine [Senach 90], [Nielsen 93], [Baccino et al., 05], [Moha et al., 05]. Les chercheurs du domaine définissent l'utilité et l'utilisabilité comme suit :

- **L'utilité** : ce critère détermine si l'interface répond aux besoins de l'utilisateur ; en d'autres termes, si l'application permet à l'utilisateur d'atteindre ses objectifs de travail. L'utilité englobe la notion de performance du système, la capacité fonctionnelle et la qualité de l'assistance technique proposée [Senach 90], [Nielsen 93].
- **L'utilisabilité** : ce terme provient de la traduction de « usability » en anglais ; c'est un concept qui date du début des années 1980 [Eason 84]. Cette dimension fait généralement référence aux cinq attributs suivants (Cf. [Nielsen 93] [Shneiderman 92]) : la facilité d'apprentissage et d'utilisation ; l'efficacité d'utilisation ; la facilité de mémorisation ; l'utilisation sans erreurs ; et la satisfaction. Ils permettent de déterminer si un système interactif est adapté aux utilisateurs visés et aux tâches pour lesquelles il a été développé.

L'utilisabilité a été étudiée en détail depuis plus d'une vingtaine d'années. Ainsi, en se basant sur plusieurs travaux de recherche [Smith et Mosier 86] [Whitefield et al., 91], une classification de ces propriétés a été proposée par [Bastien et Scapin 91] pour aboutir à un jeu de critères et sous-critères. Dix-huit critères élémentaires sont ainsi énumérés et explicités ci-dessous.

- **“Guidage** : le guidage est l'ensemble des moyens mis en oeuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur (messages, alarmes, labels...), y compris dans ses aspects lexicaux. Quatre sous-critères participent au guidage : incitation, groupement/ distinction entre items, feedback immédiat et lisibilité.
- **Charge de travail** : le critère charge de travail concerne l'ensemble des éléments de l'interface qui ont un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs et dans l'augmentation de l'efficacité du dialogue. Deux sous-critères participent au critère charge de travail : brièveté (qui inclut les critères concision et actions minimales) et densité informationnelle.
- **Contrôle explicite** : le critère contrôle explicite concerne à la fois la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions. Deux sous-critères participent au contrôle explicite : actions explicites et contrôle utilisateur.
- **Adaptabilité** : l'adaptabilité d'un système concerne sa capacité à réagir selon le contexte et selon les besoins et préférences des utilisateurs. Deux sous-critères participent au critère adaptabilité : flexibilité et prise en compte de l'expérience de l'utilisateur.
- **Gestion des erreurs** : le critère gestion des erreurs concerne tous les moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent. Les erreurs sont ici considérées comme des saisies de données incorrectes, des saisies dans des formats inadéquats, des saisies de

commandes avec une syntaxe incorrecte. Trois sous-critères participent à la gestion des erreurs : protection contre les erreurs, qualité des messages d'erreurs et correction des erreurs.

- **Homogénéité** : le critère homogénéité se réfère à la façon avec laquelle les choix de conception de l'interface (codes, dénominations, formats, procédures...) sont conservés pour des contextes identiques et sont différents pour des contextes différents.
- **Signifiante des codes** : le critère signifiante des codes et dénominations concerne l'adéquation entre l'objet ou l'information affichée ou entrée et son référent. Des codes et dénominations « signifiants » disposent d'une relation sémantique forte avec leur référent.
- **Compatibilité** : le critère compatibilité se réfère à l'accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, habitudes, compétence, âge, etc.) et des tâches, d'une part, et de l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part. De plus, la compatibilité concerne également le degré de similitude entre divers environnements ou applications."

1. Guidage	
1.1	Incitation*
1.2	Groupement/Distinction entre items
1.2.1	Groupement/Distinction par la localisation*
1.2.2	Groupement/Distinction par le format*
1.3	Feed-back immédiat*
1.4	Lisibilité*
2. Charge de travail	
2.1	Brièveté
2.1.1	Concision*
2.1.2	Actions minimales*
2.2	Densité informationnelle*
3. Contrôle explicite	
3.1	Actions explicites*
3.2	Contrôle utilisateur*
4. Adaptabilité	
4.1	Flexibilité*
4.2	Prise en compte de l'expérience de l'utilisateur*
5. Gestion des erreurs	
5.1	Protection contre les erreurs*
5.2	Qualité des messages d'erreur*
5.3	Correction des erreurs*
6. Homogénéité/Cohérence*	
7. Signifiante des codes et dénominations*	
8. Compatibilité*	

* Critère élémentaire

Figure 2.2. Classification des critères et sous critères ergonomiques selon [Bastien et Scapin 93]

Dans cette introduction, nous avons présenté les aspects ou dimensions qui doivent être évalués si l'on doit juger de l'adéquation des systèmes interactifs aux utilisateurs et à leurs tâches. Pour ce faire, plusieurs modèles de développement enrichis sous l'angle de l'évaluation des systèmes interactifs, ont été proposés dans la littérature et plusieurs méthodes peuvent être utilisées. Ces modèles font l'objet de la section suivante.

2.1.3. Les modèles de développement enrichis sous l'angle de l'évaluation des systèmes

L'idée d'enrichissement des cycles de vie pour les IHM, repose sur l'intégration, d'un point de vue méthodologique, des aspects fondamentaux de l'interaction homme-machine comme la modélisation des tâches humaines, la réalisation itérative des prototypes ou l'évaluation des systèmes interactifs. Nous avons choisi de présenter ici quatre de ces modèles visant à aller plus loin à ce sujet que les modèles classiques qu'on trouve en génie logiciel (cascade, spirale, en V,) : le modèle de Valentin et al., le modèle en étoile, le modèle ∇ (Nabla), et le modèle en U. Par ailleurs, dans [Kolski 01], on retrouve un recensement des modèles de ce type.

Un premier exemple est le modèle proposé dans [Hartson et al., 89], appelé modèle en étoile. Celui-ci situe l'évaluation au centre même du cycle complet, montrant ainsi des interactions/itérations possibles entre chacune des autres étapes. L'étape d'évaluation est vue comme une étape intermédiaire permettant de protéger l'équipe de développement d'un rejet terminal. Ce modèle se situe assez loin d'un modèle classique mais cette idée le rend intéressant.

Dans le même état d'esprit, un autre modèle intéressant est celui dit en « U », qui positionne des étapes où les facteurs humains devront être considérés par l'équipe de développement [Millot et Roussillon 91], [Abed 90], [Abed 01], [Lepreux et al., 03]. Il comprend une démarche descendante correspondant à la modélisation du système homme-machine et à sa mise en oeuvre ainsi qu'une phase ascendante correspondant à l'évaluation du système global.

Le modèle décrit dans [Valentin et al., 93] se base sur une démarche en Spirale et propose d'un côté les étapes suivies par les informaticiens et de l'autre, les étapes suivies par les ergonomes. Le point central reliant les deux domaines est l'analyse de l'activité.

Le modèle ∇ [Kolski, 97] a pour objectif de situer les différentes étapes du génie logiciel nécessaires pour développer un système interactif, tout en différenciant l'interface proprement dite des modules applicatifs ou d'aide éventuellement accessibles à partir de ceux-ci. Une des caractéristiques marquantes du modèle est de positionner des étapes, inexistantes dans les modèles classiques du génie logiciel, où les facteurs humains devront être considérés par l'équipe de développement. En fait, ce modèle accorde une grande importance à l'évaluation, colonne vertébrale du modèle et aspect central du projet.

Dans ce cadre, [Beuscart 07] a proposé aussi un cycle d'ingénierie de l'utilisabilité. Il s'agit d'un cadre orienté facteurs humains dédié aux projets de développement de systèmes interactifs complexes en milieu hospitalier. Il met en évidence les étapes importantes à réaliser par les ergonomes ainsi que les aspects de coopération à entreprendre avec les informaticiens pour ce type de projet.

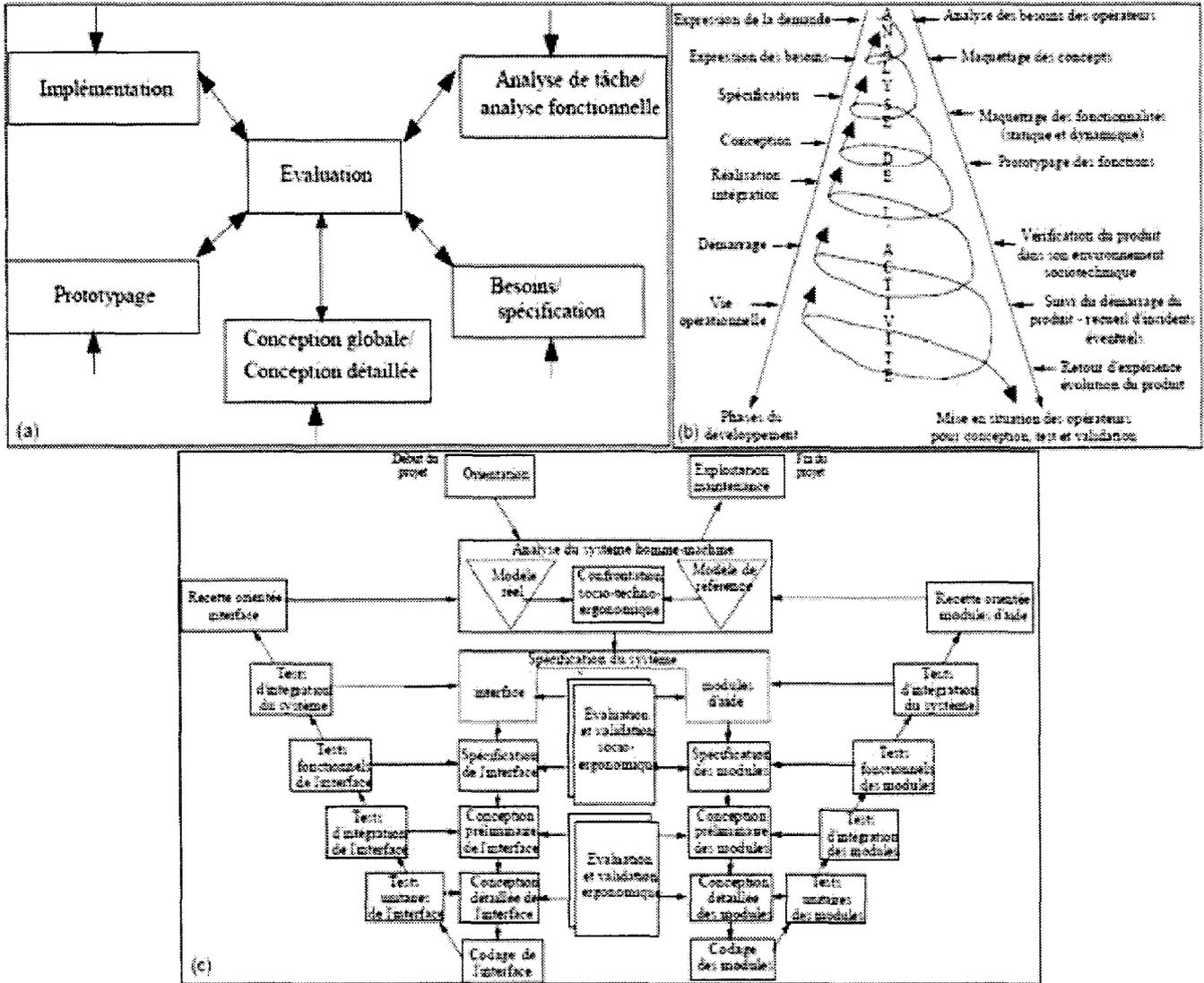


Figure 2.3. Exemples de Modèles enrichis sous l'angle de l'interaction homme-machine
 (a) Modèle en étoile, (b) Modèle proposé par Valentin et al., (c) Modèle Nabla

2.1.4. Synthèse

La plupart des modèles qui ont été présentés, montrent une évolution des processus de développement de plus en plus orientée vers la prise en compte des aspects « facteurs humains ». Nous avons constaté également que ces modèles proposent des principes de développement variés afin de répondre aux exigences des projets (ex : réduction des coûts, évolution des types de systèmes, réduction des erreurs de conception, etc.). Cependant, l'évaluation des aspects liés à la personnalisation est négligée. En effet, pour le système personnalisé, l'évaluateur est régulièrement confronté à des contextes d'évaluation différents. Chacun de ces contextes pose des questions originales et présente des contraintes spécifiques nécessitant la mise en jeu de techniques idoines [Senach 90]. Pour cela, ces modèles doivent être enrichis afin de supporter l'évaluation des caractéristiques d'un système personnalisé.

2.2. Méthodes, techniques et outils d'évaluation des systèmes interactifs

Cette partie présente les critères d'évaluation des SIP. Puis, nous décrivons les méthodes d'évaluation les plus représentatives. Ainsi, nous allons présenter globalement les principes de

chaque type d'approche. Chaque méthode ou outil, tel qu'il est défini et utilisé actuellement, est comparé avec le cadre et les critères d'évaluation de systèmes personnalisés.

2.2.1. Les critères proposés pour l'évaluation de système d'information personnalisé

Dans cette partie, nous présenterons les critères proposés ayant pour but d'aider les évaluateurs à estimer la qualité de personnalisation des SIP. Ces critères sont utilisés pour classer les problèmes détectés et prendre si nécessaire des décisions de modifications et/ou d'améliorations. Nous distinguons sept critères d'évaluation des SIP [Soui et al., 07].

- **l'adaptation du système aux préférences de l'utilisateur.** Les préférences regroupent toutes les informations nécessaires au système concernant les goûts de l'utilisateur. Elles sont classées en deux catégories : les préférences liées au contenu et celles liées au contenant.

Les préférences liées au contenu permettent de cibler les besoins des utilisateurs. Par exemple, un système d'information personnalisé dans le domaine des transports aide les usagers à planifier leurs itinéraires par rapport à leurs préférences (coût, durée, préférence par rapport à un mode de transport, minimisation des correspondances, etc.) [Petit-Rozé et al., 04b] [Anli et al., 06f].

Les préférences liées au contenant portent principalement sur la forme de présentation de l'interface, la disposition des informations ou le style de visualisation.

- **l'adaptation aux centres d'intérêts de l'utilisateur.** Cette dimension du profil est destinée à exprimer les domaines d'intérêts de l'utilisateur. Ceci implique en particulier la profession de l'utilisateur ainsi que sa formation. La majorité des systèmes interactifs utilisent cette dimension comme source de personnalisation du contenu par le filtrage des informations ou le tri des données. Prenons l'exemple d'un homme d'affaire qui est intéressé par le domaine du textile, un système de personnalisation dans le domaine du transport peut avoir pour objectif de chercher des foires qui sont susceptibles d'intéresser l'utilisateur et lui recommander des itinéraires pour y accéder.
- **l'adaptation à l'expérience de l'utilisateur.** Nous entendons par « expérience de l'utilisateur », toute connaissance acquise par une longue pratique jointe à l'observation. Ceci comprend également, la familiarité de l'utilisateur avec le système et l'expérience liée au domaine d'application qui regroupe les concepts que l'utilisateur peut connaître [Tracey 05].

L'expérience est souvent modélisée par un stéréotype (débutant, initié ou expert). Selon le niveau d'expertise de l'utilisateur pour une tâche donnée, le système procédera par étape (guidage de l'utilisateur novice) ou, pour l'expert, présentera un dialogue moins explicite que pour le novice (prévoir le raccourci, des choix d'entrées simples ou multiples selon l'expérience, etc.). Cette dimension est liée à la personnalisation du contenant (adaptation de l'IHM) et à la personnalisation du contenu.

Notons que Bastien et Scapin considèrent l'adaptation à l'expérience comme l'ensemble des moyens mis en œuvre pour permettre au système de respecter le niveau d'expertise de l'utilisateur. Ces derniers ont déjà étudié ce critère dans l'évaluation ergonomique de système interactif mais pas d'une manière explicite [Bastien et Scapin 02].

- **l'adaptation du système à la plate-forme interactive.** L'adaptation des interfaces aux supports avec lesquels l'utilisateur interroge le système se retrouve dans le terme plasticité utilisé dans la communauté IHM [Thévenin et Coutaz 99] [Calvary et al., 03].

Précisons toutefois que la plasticité ne se limite pas à la l'aspect plate-forme, puisque la définition de la plasticité est la suivante : « la capacité des interfaces à s'adapter à leur contexte d'usage dans le respect de leur utilisabilité. Le contexte d'usage se définit comme le triplet utilisateur, plate-forme et environnement » [Thévenin et Coutaz 99]. [Abowd et al., 99]. Toutefois, si on se focalise sur l'aspect plate-forme, il est possible d'en distinguer deux grandes classes. D'une part, la plate-forme physique qui regroupe les informations sur le support employé (PC, PDA, téléphone portable, etc.) pour interagir avec le système interactif [Nigay 95] [Abowd et al., 99] [Dragicevic 04]. Ces informations se déclinent en trois catégories : les dispositifs d'interaction en entrée (souris ou dispositif de pointage, clavier, microphone, etc.), les dispositifs d'interaction en sortie (écran, haut-parleur, etc.) et les capacités physiques du support (processeur, mémoire, capacités de calcul, etc.). D'autre part, la plate-forme logicielle regroupe les informations sur le système d'exploitation et l'ensemble des logiciels, machine virtuelle, et navigateur disponibles sur la plate-forme. Cette dimension est liée à l'adaptation des interfaces ou à la personnalisation du contenant [Florins 04].

- **l'adaptation à la capacité physique de l'utilisateur (accessibilité).** Plusieurs catégories d'utilisateurs peuvent vouloir se servir du système, d'où l'importance d'assurer l'accessibilité de ce dernier [Power et al., 09].

Il y a ici au moins deux catégories de personnes à considérer : les personnes atteintes de déficiences physiques ou sensorielles (malvoyants, non-voyants, daltoniens, etc.) et les personnes âgées ayant des limitations motrices et sensorielles et/ou qui ne savent pas se servir d'un clavier et/ou qui lisent lentement. Cette dimension assure la personnalisation du contenant pour les personnes en situation de handicap. Prenons l'exemple d'un utilisateur non-voyant et d'un téléphone mobile, on peut penser que la déficience sera compensée par un système de synthèse vocale [Plos et Buisine 06] [Vella 05] [Tyler et al., 07].

- **l'adaptation à l'environnement d'utilisation du système.** L'environnement se réfère à l'ensemble des informations périphériques au couple « utilisateur, plate-forme » qui sont susceptibles d'influencer le déroulement de l'activité de l'utilisateur. Par exemple, la luminosité, le bruit, la localisation géographique, etc.

Dans [Schmidt et al., 02] est proposé un système qui exploite plusieurs données de l'environnement : ContextNotePad, service de bloc notes sur Palm Pilot. Ce système est sensible aux mouvements de faible amplitude (le système se met en marche dès que l'utilisateur le prend en main), au déplacement (il change la taille de l'affichage) et à la luminosité (l'écran passe en mode rétro éclairage si la lumière ambiante baisse). Cette dimension est liée à la personnalisation du contenant.

- **l'adaptation au comportement de l'utilisateur.** Ce critère se concentre sur les buts et les plans de l'utilisateur plutôt que sur ses connaissances. Le but se définit comme l'état que l'utilisateur souhaite atteindre, et les plans (activités de navigation) décrivent les étapes parcourues pour y arriver.

Cette adaptation de navigation regroupe deux classes de techniques permettant de modifier la carte de navigation du système ou de guider l'utilisateur : les techniques d'adaptation destructives qui consistent à désactiver ou supprimer des liens inintéressants et les techniques non destructives qui permettent d'ajouter à

l'application des annotations spécifiques de guidage [Henze et Nejd1 99]. L'utilisateur n'a plus qu'à suivre ces informations supplémentaires qui l'aident à effectuer une navigation plus rapide. Cette dimension concerne la personnalisation du contenant.

Dans la partie suivante ces critères sont utilisés pour étudier les méthodes d'évaluation existantes.

2.2.2. Choix d'une classification des méthodes d'évaluation des systèmes interactif

Plusieurs classifications existent dans la littérature, Nielsen et Molich [Nielsen et Molich 90] font la distinction entre les approches expérimentales et les approches prédictives. Sweeney et Maguire [Sweeney et al., 93] proposent trois grandes familles de méthodes où l'évaluation est basée sur l'utilisateur, sur la théorie ou sur l'expertise. [Bastien et Scapin 01] séparent les approches d'évaluation en deux grandes catégories : les méthodes requérant la participation directe des utilisateurs qui se basent essentiellement sur la présence de l'utilisateur et les méthodes s'appliquant aux caractéristiques de l'interface qui se distinguent des précédentes par l'absence de l'interaction directe entre l'utilisateur et le système. Carroll et ses collègues [Carroll et al., 91] classifient les techniques en deux types d'évaluation : les techniques *a priori* (ou « formatives ») qui s'appliquent au produit en conception/réalisation, et les techniques *a posteriori* (ou « sommatives ») appliqués sur le produit fini. Enfin, [Dix et al., 93] proposent deux styles d'évaluation : en laboratoire (où les moyens d'acquisition des données sont meilleurs mais non naturels, car toute tâche est dépendante du contexte), et en situation réelle.

Beaucoup de classifications existent donc. Toutefois comme le soulignent [Grislin 95], [Abed, 01], [Bastien et Scapin 01], il paraîtrait irréaliste de considérer ces regroupements de manière indépendante. En effet, des relations transversales entre méthodes sont souvent très utiles. Pour présenter les principales méthodes d'évaluation existantes, nous retiendrons la classification de [Huart et al., 08] qui s'inspire de [Whitefield 91], [Grislin et Kolski 96], [Senach 90] et [Newman 95]. Cette classification classe les méthodes d'évaluation en quatre types d'approches (cf. figure 2.4), selon que l'utilisateur et le système sont réels ou représentés. Rappelons que notre objectif dans cette partie n'est pas de faire un recensement exhaustif de toutes les méthodes d'évaluation existantes dans la mesure où on retrouve plusieurs documents qui traitent ce sujet, mais plutôt de passer en revue un ensemble de méthodes d'évaluation les plus représentatives, pour ensuite nous orienter vers l'évaluation de systèmes d'information personnalisés.

	Système réel	Système représenté
Utilisateur réel	Méthodes d'observation (approches empiriques)	Rapports d'utilisateurs (approches empiriques)
Utilisateur représenté	Rapports de spécialistes (approches expertes)	Méthodes analytiques

Figure 2.4. Classification des méthodes d'évaluation d'interfaces homme-machine proposée par [Huart et al., 08]

Les différentes méthodes d'évaluation présentées seront donc analysées par rapport à leur pension à l'évaluation de chacune de ces sept dimensions (cf. 2.2.1).

2.2.3. Etude de méthodes, techniques et outils d'évaluation des systèmes interactifs

Cette section présente un panorama des méthodes d'évaluation. Ces méthodes sont regroupées en quatre grandes catégories suivant le principe énoncé au paragraphe 2.2.2 et présenté sur la figure 2.4.

Dans chaque catégorie sont décrites la ou les méthodes d'évaluation les plus représentatives ou celles qui ont servi de base à d'autres méthodes. Ainsi, nous allons présenter globalement les principes de chaque type d'approche. Chaque méthode ou outil, tel qu'il est défini et utilisé actuellement, est comparé avec le cadre et les dimensions de l'évaluation de système personnalisés. L'analyse effectuée porte sur les objectifs et les résultats des méthodes dans leur domaine d'application actuel et ne constitue donc pas un jugement définitif et absolu sur l'adaptabilité des méthodes à l'évaluation de systèmes personnalisés.

2.2.3.1. Approches empiriques

L'évaluation empirique consiste à recueillir des données comportementales sur l'utilisation du système. Ce type d'évaluation nécessite l'existence d'un système réel (maquette, prototype ou système final) et la présence d'utilisateurs. L'utilisation qui est faite du système par ceux-ci est observée et analysée. Une observation individuelle détaillée de l'interaction entre l'utilisateur et le système permet d'identifier les capacités de celui-ci, de détecter les difficultés éventuelles, ou encore de noter les caractéristiques inattendues de la situation [Gagné et al., 88]. Nielsen indique qu'un panel de cinq utilisateurs « représentatifs » permet d'identifier 80% des problèmes [Nielsen 94]. Parmi les nombreuses méthodes empiriques qui existent, on peut citer les plus connues qui sont : les interviews, les questionnaires, les traces d'utilisation, le mouchard électronique et l'oculomètre.

2.2.3.1.1. Interviews

L'interview consiste à s'entretenir avec des sujets représentatifs afin d'avoir un aperçu rapide des problèmes rencontrés par l'utilisateur lors de l'interaction avec le système proposé. L'interview est une conversation entre l'enquêteur et l'utilisateur. Cette discussion permet le recueil de données subjectives relatives aux attitudes, aux opinions des utilisateurs et à leur satisfaction (Cf. à ce sujet [Chin 88], [Kirakowski 88], [Lalomia 90], [Mostert 89], [Shiba 95], [Bisseret et al., 99]). La figure 2.5 illustre cette méthode.

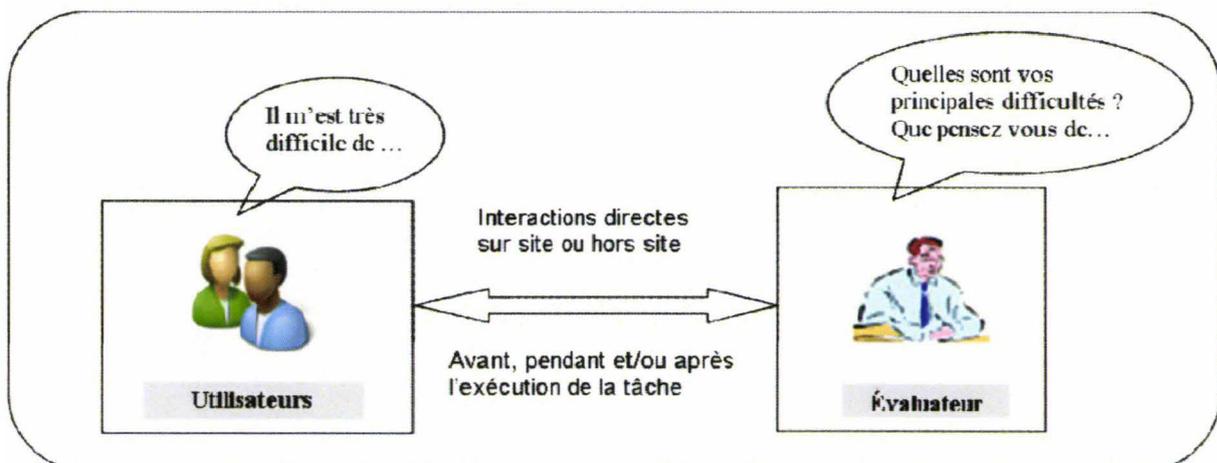


Figure 2.5. Interaction directe par interview entre utilisateur et évaluateur

Intérêts et limites

Des interviews des utilisateurs permettent d'acquérir assez rapidement de nombreuses informations sur la nature de leur activité. L'interview a l'avantage d'être plus flexible que les autres techniques de collecte et d'analyse d'informations par exemple le questionnaire et permet d'orienter les questions directement vers l'information recherchée par l'évaluateur. D'ailleurs [Moha et al., 05] précise, d'après une enquête sur les pratiques de tests d'utilisabilité, que l'interview ou l'entretien est une technique très recommandée par les professionnels de l'évaluation. Néanmoins, L'efficacité de l'interview dépend de sa préparation et des compétences de l'évaluateur qui assure son guidage. En effet, l'évaluateur doit préparer et planifier à l'avance les questions centrales à poser à l'utilisateur. Le fait que les réponses des utilisateurs puissent être influencées par la manière dont l'évaluateur leur pose les questions est un inconvénient de cette méthode.

Méthode générale utilisée dans un certain nombre de disciplines, elle présente l'intérêt d'être potentiellement utilisables pour évaluer chacune des dimensions proposées pour l'évaluation de système personnalisé. L'utilisation de la méthode pour l'évaluation du système interactif plastique EMMA (Embedded Manager for Mobile Adaptation) [Ganneau et al., 08] présente un exemple significatif des perspectives offertes par cette technique.

2.2.3.1.2. Les questionnaires

L'objectif de cette technique est de recueillir, à l'aide de documents en principe structurés, un ensemble d'appréciations, d'opinions et d'attitudes, de l'utilisateur après qu'il ait interagi avec le système. C'est donc une méthode *a posteriori*. Les questionnaires permettent surtout de déterminer la satisfaction ou les problèmes de l'utilisateur qu'il est difficile ou impossible de mesurer autrement qu'avec le questionnaire [Nielsen 93], [Valentin 93], [Kirakowski et al., 98].

Intérêts et limites

Les questionnaires sont des compléments intéressants aux interviews : leurs résultats peuvent être complétés et validés relativement rapidement par la passation d'un questionnaire auprès d'un panel plus grand de la population. Bien que cette technique semble être facile à mettre en place la difficulté réside dans la rédaction de bonnes questions ainsi que dans l'interprétation des réponses des utilisateurs.

Comme les interviews, les questionnaires sont potentiellement aptes à l'évaluation systèmes personnalisés en s'appuyant sur les sept critères proposés. Des questionnaires récents ont été utilisés dans [Soui et al., 07] pour détecter les problèmes liés à ces dimensions pendant l'évaluation du système MoverPerso (MObilité et mUltimodalité Voyageurs Etudiants en Région Nord Pas de Calais). Des détails seront fournis à ce sujet dans les chapitres 3 et 4. [Mersioli et al., 06] a proposé aussi un questionnaire évaluant une maquette de l'interface d'un service de recherche de lignes et d'horaires de bus déclinée en deux versions : une monomodale (graphique) et une multimodale (graphique et vocale).

2.2.3.1.3. Monitoring

Le monitoring, aussi appelé mouchard électronique a pour objectif de recueillir automatiquement des données objectives (les actions des utilisateurs et leurs répercussions sur le système) en situation réelle de travail. Les données recueillies seront analysées ultérieurement (manuellement ou automatiquement) [Bastien et Scapin 02], [Senach 90],

[Drury 90], [Bias et al., 93], [Mariage 05], [Ezzedine et al., 05]. La figure 2.6 illustre l'utilisation de l'outil mouchard électronique comme support pour l'évaluation.

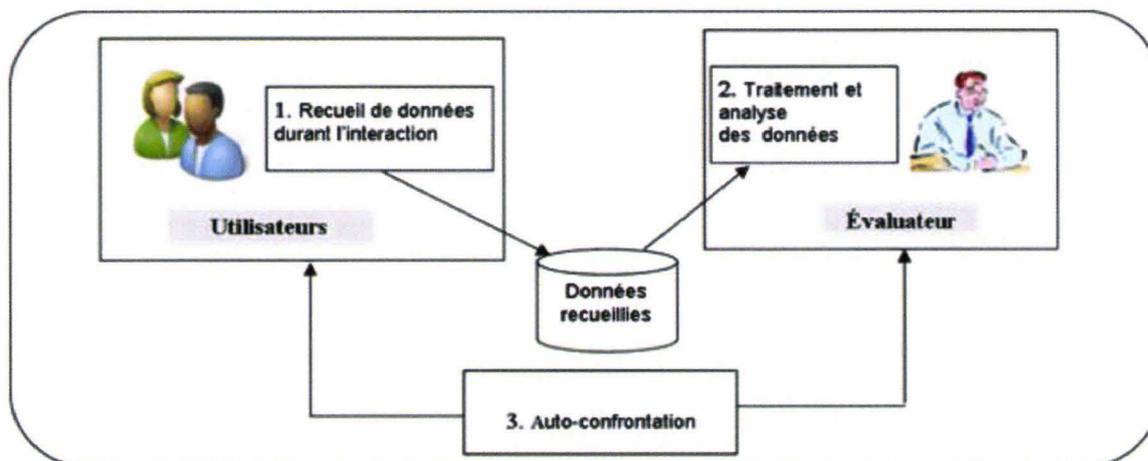


Figure 2.6. Principe du mouchard électronique

Intérêts et limites

Le monitoring présente le sérieux avantage de ne pas perturber l'utilisateur lors de la réalisation de ses tâches, En effet, lors de l'évaluation l'utilisateur ne se sent en principe à aucun moment gêné par la présence du mouchard, et se comporte ainsi d'une façon normale. Pour l'instant, et à notre connaissance, cette technique est orientée globalement vers la détection de défauts d'utilisabilité [Ezzedine et al., 05]. Durant cette évaluation, cette méthode a été utilisée pour examiner la qualité de l'interaction homme-machine en termes de facilité d'apprentissage et d'utilisation en se basant sur un ensemble des paramètres tels que durée réelle du temps de réalisation de chaque tâche ainsi que le taux de succès de sa réalisation.

L'inconvénient des mouchards électroniques réside dans le traitement de la quantité de données qu'il enregistre et dans le niveau de granularité de l'analyse [Bastien et Scapin 02]. En effet, l'évaluateur peut se trouver devant une énorme base de données recueillies qui nécessite des heures (voire des journées) de travail pour le dépouillement. Par ailleurs, les données objectives recueillies au travers des interactions homme-machine peuvent ensuite être analysées et affichées sous une forme synthétique à l'évaluateur, lui facilitant l'analyse des problèmes d'utilisation. Une méthode récente (MultiDevice RemUSINE [Paterno et al., 07] a été utilisée pour évaluer à distance des applications sur dispositifs mobiles. Elle permet de capturer non seulement les interactions entre l'utilisateur et le système, mais aussi vise à capturer les informations contextuelles de l'utilisateur. Ces informations concernent le contexte de travail de l'utilisateur ; elles influencent beaucoup la performance et les activités de celui-ci. Si ces informations contextuelles ne sont pas capturées, il est difficile à l'évaluateur de tirer les conclusions exactes concernant les problèmes que l'utilisateur a rencontrés. Ainsi, les auteurs proposent une autre solution qui utilise l'outil « Mobile Logger » pour détecter les interactions entre l'utilisateur et les applications mobiles en prenant connaissance des conditions environnementales capables d'affecter la performance et les activités de l'utilisateur. Les événements capturés par l'outil « Mobile Logger » sont catégorisés en différentes classes : événements concernant les interactions de l'utilisateur durant la session, événements système générés pour répondre aux interactions de l'utilisateur, événements d'intention concernant les changements de tâches cibles de l'utilisateur. Toutes ces informations sont enregistrées dans des fichiers XML. Cet outil « Mobile Logger » communique avec MultiDevice RemUSINE à travers ces fichiers. La figure 2.7 illustre cette méthode.

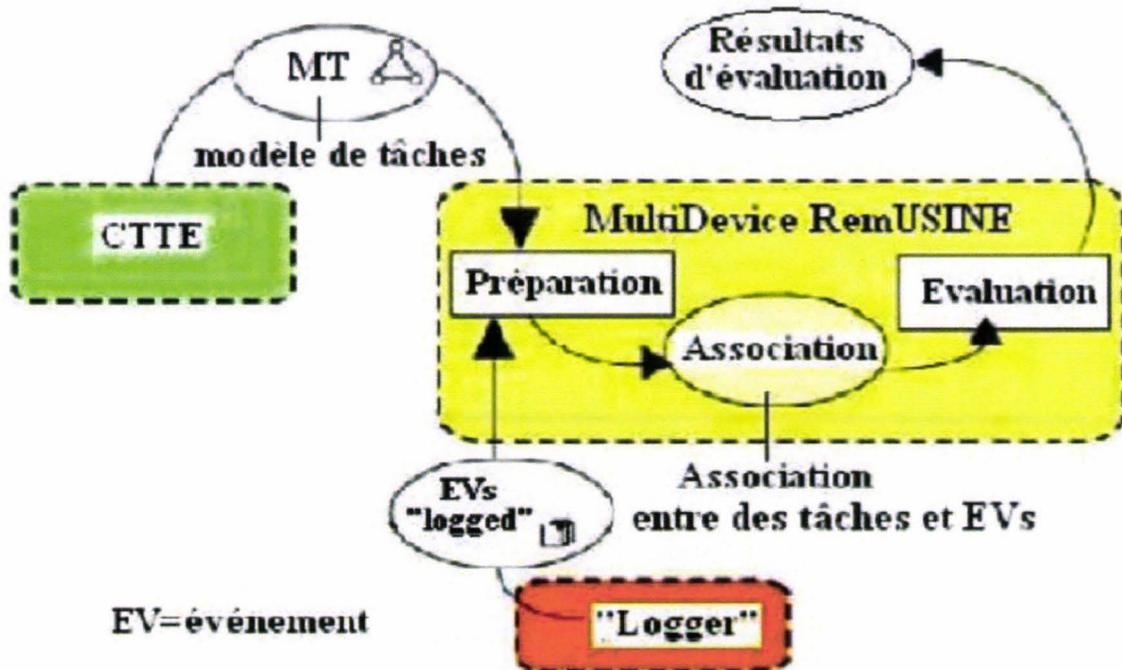


Figure 2.7. Méthode d'évaluation MultiDevice RemUSINE [Paterno et al., 07]

2.2.3.1.4. Conclusion sur les approches empiriques

Les méthodes empiriques reposent sur le recueil de données comportementales de sujets mis en situation. Le recueil s'effectue sur support audiovisuel, et/ou par capture automatique d'actions ("monitoring") ou encore par interviews et questionnaires. Les données observées font ensuite l'objet d'une analyse. L'analyse porte généralement sur les chemins suivis et les procédures adoptées par les utilisateurs, les temps d'exécution, la nature, la cause et la fréquence des incidents [Valentin 93]. Elle conduit à une synthèse qui indique notamment le degré de gravité des incidents et si possible, des recommandations.

Les approches empiriques ne peuvent être utilisées que lorsque le système est dans un état avancé de développement. Il est nécessaire de disposer d'une interface réalisée ou d'un prototype fonctionnel. L'observation auprès d'utilisateurs, accomplie avec un nombre suffisant d'utilisateurs, donne habituellement de bons résultats en termes de problèmes rencontrés par l'utilisateur. Néanmoins, ces méthodes sont utilisées depuis de nombreuses années, pour l'évaluation de l'utilisabilité des systèmes interactifs en fonction des critères ergonomiques de Bastien et Scapin en ignorant les critères de personnalisation [Bastien et Scapin 02]. Ces approches nécessitent des adaptations afin de pouvoir aider les experts dans l'évaluation des SIP.

2.2.3.2. Approches qualifiées d'expertes

Ce type d'évaluation fait appel au jugement d'un spécialiste (ou expert) en ergonomie des logiciels ou en interaction homme-machine. Ce dernier compare les performances et caractéristiques d'un système, qui sont présentées de spécifications, sous forme de maquette ou de prototype, aux normes ou recommandations existantes dans le but de détecter des erreurs de conception. Parmi les nombreuses méthodes d'évaluation, basées sur des experts, qui existent, on peut citer l'intervention d'un spécialiste, les méthodes d'inspection (cognitive walkthrough [Rieman et al., 91], les directives de recommandations (guidelines), l'évaluation heuristique) et les grilles d'évaluation.

2.2.3.2.1. L'intervention d'un spécialiste

L'évaluation par intervention d'un spécialiste consiste à effectuer une expertise de l'interface et de déterminer l'ensemble des problèmes susceptible d'être rencontrés lors de l'utilisation de l'interface. L'intervention de différents spécialistes pour l'expertise d'une interface montre que chacun se focalise sur des aspects particuliers de l'interface et fonde son évaluation sur une démarche qui lui est propre [Hammond et al., 84], [Nielsen et Molich, 90]. Par conséquent, l'idéal serait de faire intervenir plusieurs spécialistes (psychologues, cogniticiens, graphistes, etc.) l'ensemble des expertises étant ensuite confronté et synthétisé afin de déterminer les améliorations à apporter à l'interface

Intérêts et limites

Cette méthode d'évaluation est avantageuse car elle est relativement peu coûteuse, elle se fait assez rapidement et peut survenir assez tôt dans le processus de conception. Cependant, les études effectuées sur l'activité des « experts » en situation d'évaluation montrent que les performances individuelles sont très variables quant au nombre et au type d'erreurs détectées (par exemple [Hammond 84], [Molich 90], [Nielsen 90]) et quant aux stratégies d'évaluation adoptées [Pollier 92]. L'utilisation de cette méthode est tout à fait envisageable pour l'évaluation de sept dimensions spécifiques aux systèmes personnalisés. Cette méthode a été utilisée dans [Gena et al., 04] pour évaluer un site web personnalisé qui propose des programmes de télévision. En effet, les experts sont invités à vérifier la qualité des programmes proposés en fonction des habitudes, des comportements et des préférences des téléspectateurs.

2.2.3.2.2. L'évaluation heuristique

L'évaluation heuristique est généralement réalisée par un ensemble d'experts qui passent en revue l'ensemble de l'interface au moyen d'une liste de principes appelés "heuristiques". Chaque expert évalue de son côté l'interface avec les guides qui lui sont fournis, et ensuite chacun confronte son point de vue avec celui de son collègue. L'analyse peut être réalisée également à partir de maquettes (papier, logicielle ou prototype) [Ravden et Johnson 89].

Intérêts et limites

L'évaluation heuristique a été exploitée dans énormément de projets et a largement fait ses preuves. A titre d'exemple, [Nielsen et Mack 94], [Garzotto et Matesa 97] s'appuient sur l'évaluation heuristique pour proposer des évaluations de l'utilisabilité à moindre coût. Même si certains évaluateurs ne se basent que sur leur expérience et intuitions, il est recommandé de se baser sur certaines règles, compilées dans des guidelines [Kahn et Prail 93]. Les règles les plus couramment citées sont le "Top ten" de [Nielsen 96] ; (cf. Figure 2.8).

1. L'utilisation des frames ;
2. L'utilisation de nouvelles technologies quand ce n'est pas justifié ;
3. Inclure des animations que l'utilisateur ne peut arrêter ;
4. Des URL trop longues et trop compliquées ;
5. Des pages accessibles par un moteur de recherche et ne présentant pas de lien vers la page d'accueil ;
6. Des pages trop longues qui nécessitent des barres de défilement ;
7. Des fonctions de navigations mal conçues ou inexistantes ;
8. Des liens de couleur non standard ;
9. La présence d'information périmée ;
10. Des pages longues à télécharger ;

Figure 2.8. Règles les plus couramment citées ("Top ten") selon [Nielsen 96]

L'inconvénient de cette méthode réside dans la difficulté de reproduction du contexte d'usage. En effet, les experts ne peuvent pas prévoir les actions de l'utilisateur qui peuvent parfois être inattendues. Ils peuvent se contenter de remarques visant à couvrir la majorité des cas, mais pas les cas particuliers. Pour l'évaluation de systèmes personnalisés, cette méthode nécessite une bonne expérience dans le développement de ces systèmes de la part de l'évaluateur, pour définir des heuristiques spécifiques à ces systèmes et ensuite pour détecter les problèmes liés aux dimensions citées précédemment. Par exemple, [Magoulas et al., 03] a proposé une intégration des heuristiques dans l'évaluation des environnements d'apprentissage adaptés. Ils ont modifié les heuristiques de Nielsen [Nielsen 93] pour tenir compte de caractéristiques de systèmes personnalisés.

2.2.3.2.3. La conformité à des recommandations et à des directives

L'évaluation de la conformité à des recommandations (guideline reviews) consiste à juger la conformité des éléments de l'interface aux recommandations (ergonomiques ou de style) contenues dans divers types de recueils [Vanderdonck 99]. Les problèmes résultent de la non-conformité des caractéristiques de l'interface à ces principes. On trouve ainsi des guides de style, ces recommandations de style permettent généralement de distinguer les interfaces des logiciels développés pour des environnements différents (par exemple Environnement Unix, Windows, Macintosh). Le « Macintosh Human Interface Guidelines » d'Apple [Apple 92], ou encore le « Window interface : An Application Design Guide » de Microsoft [Mic 92] sont des exemples de guides de style.

Intérêts et limites

L'inconvénient majeur des guides de recommandation réside généralement dans leur taille importante (des milliers de règles) [Vanderdonck 94], ainsi leur utilisation devient fastidieuse et complexe. Selon [Bastien et Scapin 01], lorsque on étudie ces guides, on peut constater une absence d'uniformité dans la présentation des recommandations ce qui rend la recherche d'une règle bien précise extrêmement difficile. De plus, le passage en revue de centaines de recommandations pour chaque caractéristique de l'interface n'offre pas une méthode rapide et agréable. Parce que l'émergence de nouvelles recommandations est toujours postérieure à l'apparition d'une nouvelle technologie, l'utilisation de cette méthode nécessite la mise à jour régulière de ces recommandations pour une bonne évaluation des interfaces adaptées qui utilisent ces nouvelles technologies.

2.2.3.2.4. Les grilles d'évaluation

Les grilles d'évaluation (checklists) permettent d'évaluer l'interface selon plusieurs critères. En utilisant des grilles d'évaluation, l'évaluateur note systématiquement l'interface selon une échelle comportant généralement de 3 à 5 points, parfois 10. Une grille d'évaluation regroupe un ensemble de questions auxquelles l'évaluateur, passant en revue l'interface, doit répondre. Généralement la réponse à ces questions se fait via des cases à cocher ou des notes à attribuer à chaque critère à étudier.

Une méthode d'évaluation par « check-list » a par exemple été mise au point par [Ravden et Johnson 89] (Cf. figure 2.9) pour évaluer l'adaptation du système à l'expérience de l'utilisateur ; elle est utilisable par des concepteurs, des spécialistes des facteurs humains et des utilisateurs finaux du système.

GRILLE D'ÉVALUATION	Note				
	1	2	3	4	5
Prise en compte de l'expérience de l'utilisateur, respect du niveau d'expertise					
1 L'information est-elle présentée dans le même état d'esprit que l'utilisateur s'en fait ?			×		
2 L'affichage graphique est-il compatible avec la représentation que se fait l'utilisateur de système ?	×				
3 La structure de système est-elle adaptée à la perception de la tâche de l'utilisateur ?			×		
4 Le déroulement des activités exigées pour effectuer une tâche est-il en accord avec les attentes de l'utilisateur ?		×			
5 Le système travaille-t-il dans la même voie que la pensée de l'utilisateur ?					×
6 Estimation globale du système en terme de prise en compte de l'expérience de l'utilisateur ?					×
Clarté visuelle Caractéristiques lexicales de présentation des informations (cognitif, perceptif) Groupement par localisation, positionnement des items appartenance-distinction.					
1 Chaque écran est-il clairement identifié (titre informatif ou description) ?				×	
2 L'organisation des informations sur l'écran est-elle toujours logique ?			×		
3 La séparation entre les différentes informations		×			

Figure 2.9. Extrait d'une grille d'évaluation (inspirée de [Ravden et Johnson, 89])

Intérêts et limites

La méthode des grilles d'évaluation présente l'intérêt de constituer un guide pour l'évaluateur puisqu'elles l'obligent à n'oublier aucun des aspects de l'interface. En revanche, elle comporte quelques limitations [Bastien 96], [Kolski 97] dues à la nature des grilles : les échelles de mesure et la notation des propriétés sont difficiles à valider, il est aussi difficile de définir des items de la grille d'évaluation adaptés au logiciel évalué. Il existe peu de références à ce sujet dans la littérature et qui sont basés sur questions trop générales [Ravden et Johnson, 89]). Même si la mise au point de check-list est difficile, la généralisation des grilles d'évaluation à des critères plus vastes que l'utilisabilité est tout à fait envisageable. Un travail reste donc à mener pour arriver à une grille d'évaluation de systèmes personnalisés autour des sept dimensions définies dans 2.2.1.

2.2.3.2.5. Conclusion sur les approches qualifiées d'expertes

Les approches expertes autorisent l'évaluation dans des situations de présence ou non du système interactif en question. Comme le précise [Grislin 95], l'intervention d'un spécialiste n'est envisageable qu'une fois que l'interface est réalisée, alors que les grilles d'évaluation et les méthodes d'inspection de l'utilisabilité sont employées en parallèle à la conception. Ces méthodes contribuent à améliorer les caractéristiques ergonomiques de l'interface utilisateur mais négligent relativement les dimensions de personnalisation.

2.2.3.3. Approches qualifiées d'analytiques

Cette catégorie de méthode permet d'évaluer la conception du système et non son utilisation. Le système est évalué au travers d'un ensemble de qualités que doit posséder celui-ci pour être utilisable. Il s'agit donc, dans ce cas là, de comparer le système à un ensemble de référents afin de détecter les déviations. Nous pouvons décomposer les méthodes analytiques et leurs versions automatisées en trois familles : (1) les modèles prédictifs où l'évaluation est menée par rapport à des séquences d'actions en relation avec des tâches à effectuer, (2) les outils d'évaluation automatique qui analysent et vérifient le code de l'IHM et (3) les méthodes par simulation.

2.2.3.3.1. Modèles prédictifs

Les modèles prédictifs sont basés sur l'hypothèse selon laquelle certaines performances de l'utilisateur peuvent être prédites, et donc considérées dès la phase de conception de l'interface. Les modèles prédictifs peuvent être utilisés pour prédire certains aspects liés à l'interaction homme-machine (hiérarchie des tâches, cheminement des actions de l'utilisateur, alternatives, temps requis pour réaliser une tâche) et faciliter la mise en évidence de problèmes potentiels.

2.2.3.3.1.1. Utilisation des modèles des tâches comme support à l'évaluation

Les modèles de tâches consistent en une décomposition hiérarchique de tâches pour représenter l'activité de l'utilisateur. Dans ce type de technique, la tâche représente un support formel pour l'évaluation prédictive de la performance puisqu'elle propose une description mesurable du comportement d'utilisation et de ses performances. Parmi les modèles les plus connus applicables aux systèmes interactifs on peut citer : GOMS [Card et al., 83], [John 95], CPM-GOMS [John et Kieras 96], KML [Card et al., 83], MAD et MAD* [Scapin et al., 89], [Gamboa et al., 98], K-MAD [Lucquiaud 05] et K-MADe [Baron et al., 06], SADT/ Petri [Abed et al., 98]. De nombreuses autres méthodes peuvent être trouvées dans [Diaper et Stanton 03].

- **Le modèle GOMS** : GOMS (Goals, Operators, Methods, and Selection) [Card et al., 83] est un modèle de description du comportement attendu d'un utilisateur. Un modèle GOMS peut se pratiquer à quatre niveaux d'abstraction en fonction des objectifs recherchés par le concepteur : tâche, fonction, argument et actions physiques. Les éléments de GOMS sont les buts (Goal), les opérateurs (Operator), les méthodes (Method) et les sélections (Selection).
 - **un but** définit un état recherché. Il est réalisé par l'exécution d'une suite d'opérateurs et est décomposé de manière hiérarchique ;
 - **un opérateur** est une action atomique dans le niveau de modélisation considéré. Son exécution provoque un changement d'état du modèle mental de l'utilisateur mais aussi de l'environnement et notamment du système. Tous les opérateurs d'un niveau de modélisation GOMS ont une durée d'exécution de même ordre de grandeur. Cette métrique est une façon de définir le niveau d'abstraction d'un modèle GOMS.
 - **une méthode** décrit le procédé qui permet d'atteindre un but. Elle représente le savoir-faire et s'exprime par composition de sous-buts et d'opérateurs.
 - **une règle de sélection** est utilisée lorsque plusieurs méthodes permettent d'atteindre un but donné. Elle offre le moyen de choisir une méthode parmi les candidates possibles.

Notons que le modèle CPM-GOMS [John et Kieras 96] propose une extension de GOMS sous la forme d'un calcul de chemin critique pour réaliser la tâche ; il décrit les tâches de l'utilisateur par une représentation sous forme de réseau.

- Le modèle Keystroke-Level-Model (KLM)** [Card et al., 83] : Etant centré sur la tâche, KLM permet essentiellement de prédire le temps d'exécution des tâches utilisateurs. Une tâche est analysée au niveau de ses tâches-unités. Par exemple : saisir, déplacer, supprimer... Le temps d'exécution de chaque tâche-unité est estimé par addition des temps d'exécution des opérateurs primitifs qui le composent. Ce modèle contient six types d'opérateurs primitifs, et pour chacun de ces opérateurs, un temps d'exécution est estimé ; K : appuyer sur une touche, P : indiquer avec la souris une zone de l'écran, H : diriger la main vers un dispositif, D : dessiner un segment de ligne, M : se préparer mentalement à faire une action, R : temps de réponse du système pendant lequel l'utilisateur est en attente. KLM est un modèle prédictif à utiliser très tôt dans le cycle de développement [Farenc 97]. Le modèle KLM a l'avantage d'être facilement compréhensible par les concepteurs ; par contre sa mise en oeuvre pose rapidement de nombreux problèmes [Coutaz 90]. Par ailleurs, il ne permet pas la prédiction de l'efficacité de l'opérateur quant celui-ci effectue des tâches complexes.

Intérêts et limites

Les modèles des tâches sont des moyens permettant de modéliser les tâches humaines (GOMS), CPM-GOMS, KML, MAD et MAD*, K-MAD et K-MADe en sont des exemples représentatifs. Les modèles des tâches peuvent également servir de support à l'évaluation des interfaces homme-machine personnalisées. En effet, divers outils de développement de systèmes personnalisés ont utilisé un modèle de tâche. Par exemple, RenderXML [Francisco et al. 08] permet de définir un modèle de tâche générique pour une famille de plateformes pour générer une interface multi-plateforme. Ce modèle peut être utilisé aussi pour l'évaluation des interfaces adaptables ou adaptatives. En effet, il peut être employé pour examiner les comportements des utilisateurs en tenant compte du contexte d'interaction.

2.2.3.3.1.2. Utilisation des modèles linguistiques comme support à l'évaluation

Ces modèles tentent de rendre explicite, sous forme d'une grammaire, la structure de l'interface et les actions mises en jeu par l'utilisateur lors de l'exécution d'une tâche. Ces modèles sont utilisables pour décrire les tâches d'interaction entre l'utilisateur et l'interface, dans le but principalement d'évaluation du langage de commande et de sa cohérence [Grislin et Kolski 96]. Parmi ces modèles, nous citons à titre présentatif les modèles suivants :

- Le modèle CLG (Command Language Grammar)** [Moran 81] permet de structurer l'analyse d'une interface par affinements successifs, du niveau tâche jusqu'au niveau des actions physiques. Cette structuration comprend six niveaux regroupés en trois composantes fonctionnelles : la composante conceptuelle, la composante de communication et la composante physique (cf. figure 2.10).

Composante conceptuelle	niveau tâche
	niveau sémantique
Composante de communication	niveau syntaxique
	niveau interaction
Composante physique	niveau de la présentation physique
	niveau des unités d'entrée/sortie

Figure 2.10. Les niveaux d'une description CLG

D'après Moran, le modèle CLG peut contribuer à assurer une évaluation prédictive de la facilité d'utilisation de l'interface et de localiser, selon [Coutaz 90], les décisions d'évaluation qui seront critiques pour l'utilisateur.

- **Le modèle ALG** (Action Language Grammar) [Reisner 84] : il permet de construire un modèle des actions mises en jeu par l'utilisateur, sous la forme de règles de production. Lors de l'évaluation, ces règles permettent :
 - d'identifier des choix de conception susceptibles d'engendrer des erreurs d'utilisation et permettre ainsi de reboucler sur la spécification,
 - de comparer les alternatives de conception selon un critère de facilité d'utilisation.

Intérêts et limites

Ces modèles aident à juger la structure de tâches décrites à l'aide d'un langage, mais celui-ci se limite aux tâches simples, par exemple à la description des fonctions spécifiques à un dispositif d'interaction. Il est souvent difficile dans le système interactive adaptés de prévoir de manière exhaustive les cheminements des actions de l'utilisateur à cause de changements contextuel. Il nous parait donc difficile d'utiliser ces modèles dans l'évaluation prédictive de SIP. Soulignons aussi que ces méthodes n'ont pas fait l'objet d'études récentes à notre connaissance.

2.2.3.3.1.3. Approches prédictives fondées sur une théorie : le cas de PUM (Programmable User Model)

Les approches fondées sur des théories s'appuient sur une représentation de l'activité cognitive de l'utilisateur. Elles permettent d'anticiper les réactions et le raisonnement de l'utilisateur face à un système interactif à l'aide des formalismes de représentation des comportements. Par exemple, PUM (Programmable User Model) consiste à prédire l'utilisabilité d'un système en terme de résolution de plan [Young 90]. L'évaluateur spécifie ce que l'utilisateur doit savoir sur la tâche et sur l'interface. Cette description est ensuite compilée en règles exécutables par PUM. L'exécution des règles fournit une simulation du comportement cognitif de l'utilisateur en train d'exécuter la tâche avec l'interface spécifiée.

Intérêts et limites

PUMs a pour avantage de fournir un feed-back aux développeurs sur le modèle qu'ils sont en train de réaliser, mais également lorsque le modèle est exécuté pour justifier les prédictions qui sont réalisées et améliorer le modèle. Dans beaucoup de cas, des informations utiles en conception peuvent être dérivées de la phase d'analyse des connaissances sans avoir besoin d'exécuter le modèle cognitif du comportement de l'utilisateur [Young 94].

La limite de PUM dans l'évaluation de SIP est de mélanger dans le modèle la représentation de l'utilisateur et la représentation du « device » [Young 94] et donc de ne pas avoir un vrai modèle de l'utilisateur qui constitue la source d'informations nécessaires à la mise en oeuvre de la personnalisation du système d'information. PUMs a également pour inconvénient de ne pas permettre la modélisation de tâches hautement interactives, telles que celles incluant des actions en manipulation directe. Soulignons aussi que PUM n'a pas fait l'objet d'études récentes à notre connaissance.

2.2.3.3.2. Outils d'évaluation automatique

Plusieurs outils logiciels d'aide à l'évaluation ont déjà été proposés. Certains constituent des versions logicielles de documents papier par exemple la version logicielle de la méthode cognitive Walkthrough [Rieman et al., 91], d'autres sont des outils d'accompagnement de l'évaluation, c'est-à-dire qu'ils aident l'évaluateur à structurer et à organiser l'évaluation, comme par exemple l'outil FACE (Fast Audit base on Cognitive Ergonomics) [Hulzebosch et Jameson 96]. FACE permet à l'évaluateur de cerner la situation de l'évaluation, de choisir la technique d'évaluation qui sera appliquée et de présenter les résultats. Finalement d'autres permettent de faire de l'évaluation automatique. C'est de cette dernière catégorie dont il sera question ici. Parmi ces outils automatiques les plus connus, on peut citer sans souci d'exhaustivité les outils suivants : SYNOP, KRI/AG, ERGOVAL, CHIMES. Ils sont décrits brièvement ci-dessous.

- **SYNOP** de Kolski [Kolski 89] [Kolski et Millot 91] est un système expert d'évaluation automatique de la présentation statique des vues pour les systèmes de contrôle temps réel. Les vues sont créées avec le progiciel graphique IMAGIN qui génère automatiquement des fichiers binaires. Ces fichiers sont ensuite traités par l'outil SYNOP, via un module spécifique, pour récupérer la description de l'interface. Le principe d'évaluation et amélioration ergonomique d'une vue graphique par SYNOP peut être décrit par trois phases : (1) interprétation des fichiers graphiques dans le but de traiter et créer un réseau sémantique d'objets structurés LISP ; (2) évaluation ergonomique des objets du réseau à l'aide de règles ; (3) reconstruction des fichiers graphiques correspondant à la vue améliorée.
- **CHIMES** (Computer Human Interaction Models) [Jiang 92] est un système expert développé par le « NASA-Goddard Space Flight Center ». CHIMES est dédié à l'évaluation des interfaces développées avec TAE plus (Transportable Application Environment) sous OSF/MOTIF. Cet outil permet d'évaluer automatiquement la conformité d'une interface au guide standard d'OSF/MOTIF et plus particulièrement le respect des recommandations ergonomiques liées à l'utilisation de la couleur.
- **KRI/AG** (Knowledge-based Review of user Interfaces) de Löwgren [Lowgren et Nordqvist 92] est un système expert connecté directement à l'UIMS (TeleUse, sous environnement X-Window) qui évalue les fichiers générés par ce dernier. Ce système s'appuie sur une base d'environ cent recommandations ergonomiques et de style (Motif) portant sur les aspects syntaxiques et de présentation de l'interface. Notons que ces fichiers contiennent la présentation statique de l'interface. L'évaluation est réalisée durant le développement de l'interface et fournit donc des informations utiles pour améliorer l'interface évaluée.
- **ERGOVAL** [Barthet 94] [Farenc 97] est un système d'évaluation à base de connaissances. Les règles ergonomiques intégrées à la base de connaissance sont relatives aux interfaces graphiques. Il traite de la présentation statique, aux niveaux lexical et syntaxique. ERGOVAL implique trois sous-systèmes : une base de connaissances contenant les règles ergonomiques, la décomposition structurelle de l'interface et une typologie des objets graphiques permettant de factoriser la définition de règles en regroupant les objets candidats à leur application.
- **USINE (USer INterface Evaluator)** [Lecerof et Paterno 98] est une méthode d'évaluation des interfaces de type WIMP (Window, Icons, Menus, Pointing device).

Elle analyse les actions physiques de l'utilisateur sur l'interface à partir d'informations stockées dans des fichiers textuels « log » en utilisant le modèle de tâches CTT (ConcurTaskTrees) [Paterno et al., 97]. La figure 2.11 illustre cette méthode.

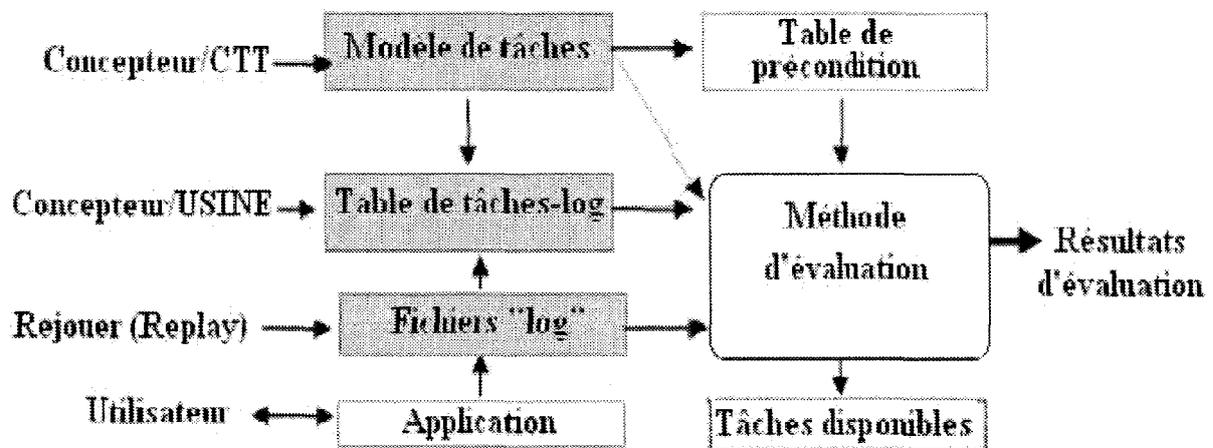


Figure 2.11. Méthode d'évaluation USINE en utilisant le modèle de tâches CTT, traduite de [Lecerof et Paterno 98]

L'outil USINE fournit au concepteur et/ou à l'évaluateur une interface leur permettant d'associer les tâches élémentaires dans le modèle de tâches aux actions physiques correspondantes dans le fichier « log » pour créer la table de tâches-log. Si une action n'est pas associée à une tâche élémentaire, alors cette action est considérée comme une erreur de l'utilisateur. Il y a deux types d'erreur de l'utilisateur :

- les erreurs de précondition (l'utilisateur a réalisé des séquences de tâches violant les contraintes temporelles spécifiées dans le modèle de tâches. Une tâche est réalisée lorsque ses préconditions ne sont pas satisfaites). Ce type d'erreur est une raison d'échec des tâches.
- les erreurs de l'utilisateur concernent les actions de l'utilisateur qui n'existent pas dans les tâches élémentaires associées. Ce sont des actions inutiles.

Si quelques tâches ne sont pas réalisées dans la session considérée pour l'association mais le sont dans une autre session, alors l'outil USINE permet de mettre à jour la table de tâches-Log.

Les résultats d'évaluation concernent les tâches accomplies, les tâches échouées, le temps pris par chaque tâche et les instants où les erreurs sont commises.

Notons que certains outils d'évaluation automatiques ont été utilisés dans l'évaluation ergonomique des sites web. Par exemple, en se basant sur les "balises" ou "tags", les outils WebMetrics Tool Suite [SCH 98b], évaluent l'accessibilité et la lisibilité des pages web en analysant la présence de cadres, de textes de description des images, le nombre de liens par nombre de mots, le nombre d'objets affichés, le nombre de lignes par pages, etc. Ils vérifient également la compatibilité du code avec les différentes versions de navigateurs. On peut citer aussi, la méthode WebRemUSINE [Paganelli et Paternò, 02 ; 03] qui vise à évaluer des interfaces Web. Elle utilise aussi le modèle de tâches CTT et les fichiers « log » (stockant les événements qui se sont produits sur le navigateur Web) pour son évaluation. La figure 2.12 illustre cette méthode.

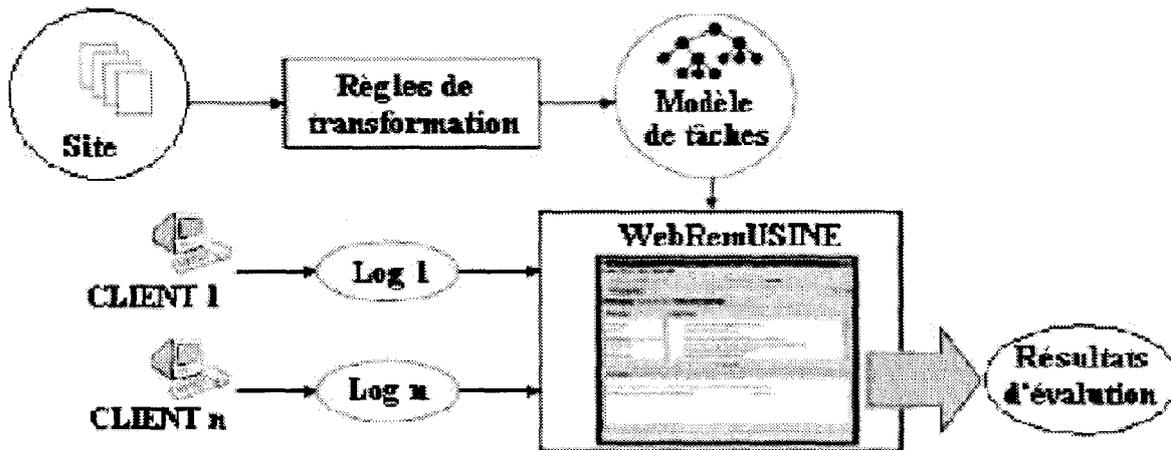


Figure 2.12. Méthode d'évaluation WebRemUSINE, traduit de [Paganelli et Paternò, 02 ; 03]

Cette méthode peut détecter des problèmes similaires à ceux détectés avec USINE, comme les résultats concernant les réalisations des tâches (réussite, échec), les erreurs commises (deux types d'erreurs) ainsi que les informations statistiques les concernant ; mais elle peut aussi détecter d'autres problèmes concernant les pages Web ainsi que proposer les informations statistiques associées :

- les pages visitées et le nombre d'accès, de même que le nombre moyen d'accès à chaque page ;
- les patterns de visite des pages (les séquences de pages visitées qui apparaissent plusieurs fois) durant la navigation et leur fréquence ;
- le temps pris pour visiter ou télécharger chaque page, de même que le temps moyen pris pour visiter ou télécharger une page.

Néanmoins, mise à part la dimension " l'adaptation du système à la plate-forme interactive " évaluable partiellement, ces outils ne permettent pas d'évaluer une majorité des caractéristiques du SP.

Intérêts et limites

Les outils d'évaluation automatiques permettent d'évaluer certains aspects de la qualité ergonomique des systèmes interactifs, et ce de manière automatique [Fortes 97].

KRI et SYNOP sont, à notre connaissance, les seuls systèmes capables de proposer des actions correctives. Mais, leur portée est extrêmement ciblée : KRI n'admet que des interfaces de type remplissage de formulaires, et SYNOP est dédié aux applications de type systèmes de contrôle.

De fait, la modélisation de la tâche, ou plus généralement toute activité cognitive d'ordre supérieur, sont ignorés. Les techniques KRI ou SYNOP évaluent l'image de l'interface mais excluent l'utilisateur et ses préférences ainsi que les remarques constructives qui découlent de l'observation de ce dernier.

CHIMES est un outil d'évaluation dédié spécialement aux interfaces critiques et à hauts risques en aviation. Son inconvénient majeur est qu'il est rattaché à la station de travail SUN et s'exécute seulement sur le système d'exploitation SunOS avec MOTIF.

Grâce à ses principes de structuration d'objets, ERGOVAL a pour avantage d'être automatisable et d'être efficace pour évaluer la présentation statique d'une interface concernant des aspects indépendants de la tâche. L'inconvénient majeur de cette méthode est qu'elle ne tient pas compte la dynamique de l'interface et des aspects liés à la tâche de l'utilisateur.

En utilisant la méthode USINE, l'évaluateur peut déterminer : si les objectifs de l'utilisateur sont atteints, si l'apparence courante de l'interface est efficace, si l'interface est lisible, si l'utilisateur réalise les tâches facilement, les tâches qui causent des problèmes à l'utilisateur, etc. L'évaluateur doit interpréter les résultats d'analyse lui-même pour détecter les problèmes et pour proposer des améliorations. Cette approche nécessite des adaptations afin de pouvoir aider les experts dans l'évaluation des SIP.

2.2.3.3. Vers les méthodes d'évaluation par simulation

En simulant une activité/tâche potentielle, ces méthodes permettent d'avoir une vision de l'interaction et de faire des simulations impliquant différents utilisateurs ou différents contextes. La simulation implique l'usage de modèles décrivant la réalité, non dans son ensemble, car ce travail de modélisation est illusoire, mais d'un point de vue naturaliste, dans un certain nombre de conditions restrictives cohérentes [Le Bodic et al., 03a] [Le Bodic et al., 04b]. Cette simulation permet d'évaluer des hypothèses en termes de modèles et de scénario, d'intervenir sur ceux-ci à tout instant et ainsi de guider, soit une évaluation, soit une conception, (cf. figure 2.13).

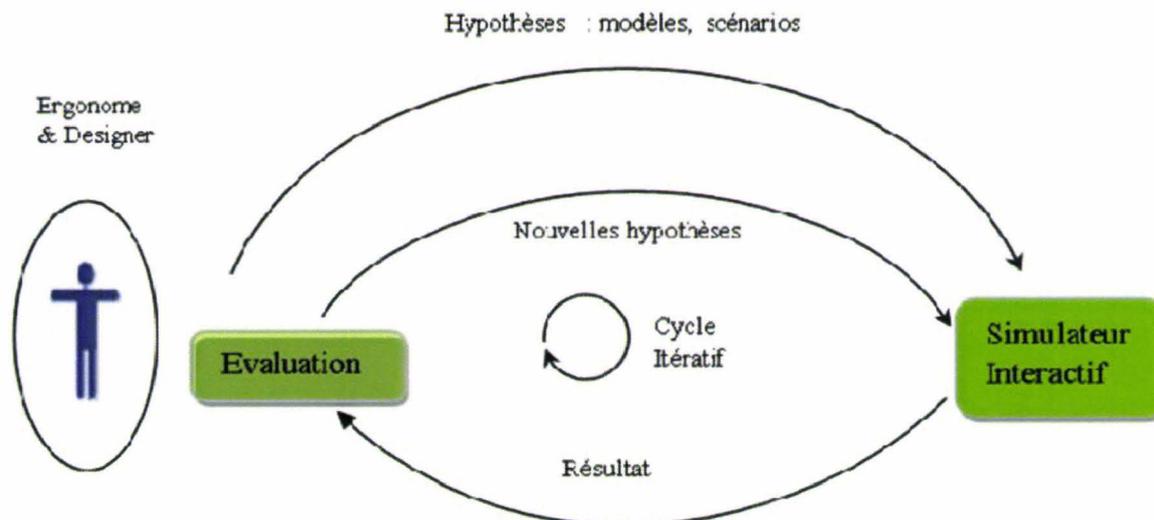


Figure 2.13. Le prototypage interactif, vu du côté de l'utilisateur de la simulation. Il consiste en l'élaboration d'un SI à travers une évaluation permanente de solutions fonctionnelles simulées [Le Bodic et al., 04]

[Le bodic 05] a proposé aussi un simulateur, nommé SIHMM (Simulateur d'interaction Homme Machine Multimodale), permettant de simuler, grâce à la réalité virtuelle, un modèle comportemental d'utilisateur en interaction avec son environnement. SIHMM se distingue des outils précédents par les caractéristiques suivantes :

- La modélisation du contexte environnemental et de sa dynamique qui peut être spécifiée à l'aide d'un éditeur de scénarios.

- La possibilité d'un prototypage interactif, favorisant les tests d'hypothèses à la fois sur le moteur décisionnel de l'utilisateur virtuel, sur les fonctionnalités du système interactif et sur l'environnement lui-même.
- Une modélisation des Systèmes Interactifs Multimodaux orientée tâche facilitant la simulation et l'analyse. Cette modélisation met en évidence les modalités d'interaction ainsi que la multimodalité.

Intérêts et limites

La simulation de scénarios, permet aux évaluateurs-concepteurs d'avoir un aperçu de la réalité possible d'un scénario situé, de faire évoluer celui-ci et d'aboutir plus rapidement à un cahier des charges complet.

L'intérêt des simulations est de pouvoir considérer rapidement différents scénarios en introduisant des événements à des instants donnés et en faisant varier des hypothèses d'usage. Il est par exemple intéressant de simuler l'appropriation de l'interface par l'utilisateur, de tester de nombreuses hypothèses fonctionnelles sur cette interface et de procéder à des analyses statistiques.

Il existe de nombreuses architectures qui tentent de simuler un comportement humain, certaines sont dédiées à l'étude d'IHM. C'est le cas par exemple de CAPS (Cognitive Architecture Production System for modeling reading) [Just et Carpenter 92] de LICAI/CoLides (LInked model of Comprehension-based Action planning and Instruction taking)/ (Comprehension-based Linked model of Deliberate Search) [Kitajima et Polson 96] et de l'Info Scent Simulator [Chi et al., 03]. D'autres sont plus générales comme SOAR (State Operator And Result) [Newell 90] et ACT-R (Adaptative Control of Thought Rational) [Anderson 93], elles sont adaptées et intégrées dans des outils supportant des modules sensori-moteurs comme GLEAN (GOMS Language Evaluation and ANalysis) [Kieras et al., 95] et EPIC (Executive-Process/Interactive Control) [Kieras et al., 97] qui s'intéresse plus précisément à l'évaluation d'interfaces multimodales.

La plupart de ces architectures s'appuient sur des modèles spécifiant le comportement probable de l'utilisateur en fonction des critères de complexité, de la surcharge cognitive en ignorant les dimensions de personnalisation.

2.2.3.3. Conclusion sur les approches qualifiées d'analytiques

Les modèles prédictifs que nous avons présentées ne prennent pas en compte la spécificité d'un SIP et ne préconisent aucun traitement particulier pour ce type de système. Par ailleurs, les techniques de récupération et de capture des comportements des utilisateurs utilisées par ces outils d'évaluation automatiques semblent être prometteuses.

En se basant sur une représentation ou des connaissances des processus cognitifs engagés par l'utilisateur lors de l'interaction avec le système, ces méthodes sont orientées vers des problèmes d'utilisabilité du système. Néanmoins, elles ne permettent pas d'évaluer les caractéristiques d'un SIP. En revanche, les méthodes qui se basent sur la présence des utilisateurs pour la capture de leurs comportements et de leurs interactions avec le système, permettent de mieux apprécier certains problèmes réels liés à ces dimensions.

2.2.4. Synthèse et discussion

Les approches d'évaluation empirique consistent à recueillir des données comportementales sur l'utilisation du système. Ces approches empiriques d'évaluation de l'interface homme-machine s'appuient notamment sur des techniques générales (comme les interviews et les questionnaires) qui ont été spécialisées, en particulier pour les besoins d'études en ergonomie du logiciel et en interaction homme-machine en général. Mais, nous pensons que ces méthodes peuvent être utilement appliquées aux SIP. Soulignons aussi que le mouchard électronique a été utilisé pour traiter principalement l'utilisabilité. En effet, il permet la capture et l'analyse essentiellement des interactions entre les utilisateurs et l'interface du système (les actions des utilisateurs : appui sur une touche, mouvement de la souris, sélection d'un objet, etc. ; de même que les réactions du système : apparition d'un message d'avertissement, changement de vue, apparition d'une nouvelle fenêtre, changement du contenu, fermeture d'une fenêtre, etc.). Pour contribuer à l'évaluation de SIP, le mouchard devrait non seulement capturer les interactions entre l'utilisateur et le système, mais aussi capturer les informations contextuelles de l'utilisateur. Ces informations concernent le contexte de travail de l'utilisateur ; elles influencent beaucoup la performance et les activités de celui-ci.

Pour les approches qualifiées d'analytiques, généralement, il n'y pas de prise en compte des contraintes environnementales et sociales dans la modélisation de l'activité humaine. Il s'agit donc d'évaluation décorrélée de la réalité contextuelle. De plus, il faut prendre en compte le contexte [Jones et Mitchell 02], et plus généralement l'environnement de l'utilisateur et de l'interaction.

En outre, il n'existe pas d'extension des méthodes de l'approche analytique d'évaluation aux SI multimodaux. Il n'existe pas non plus de processus automatisé d'évaluation qualitative comme on pourrait facilement le concevoir de prime abord. Il faut alors établir de nouvelles méthodes qui prennent en compte l'environnement, les aspects sociaux du travail de l'utilisateur et, bien évidemment, toutes les possibilités offertes par la multimodalité. Mais aussi le modèle de l'utilisateur qui permettrait de mieux concevoir son comportement et ainsi d'évaluer plus précisément la qualité du SIP. Et pourquoi pas, par la suite créer un processus automatisé de mesure de ces attributs spécifiques à la personnalisation ; [Lucquiaud et al., 02] précise que ce processus n'existe pas encore de manière satisfaisante.

Le tableau 2.1 synthétise l'adaptabilité des méthodes à l'évaluation de système personnalisé.

Tableau 2.1. Synthèse de l'adaptabilité des méthodes à l'évaluation de système personnalisé

Méthodes		C _{Préférence}	C _{Intérêts}	C _{Expérience}	C _{Plate-forme}	C _{Accessibilité}	C _{Environnement}	C _{Comportement}
Approches empiriques	<i>Interviews</i>							
	Les travaux de [Garneau et al, 08]				+		+	
	<i>Questionnaires</i>							
	Les travaux de [Mersiol et al, 05]				++			
Approches expertes	<i>L'intervention d'un spécialiste</i>							
	Les travaux de [Gena et al, 04]	++		+				++
	<i>L'évaluation heuristique</i>							
	Les travaux de [Magouas et al, 03]	+		+				+
	<i>Les grilles d'évaluation</i>							
	Les travaux de [Ravden et al, 89]			+				
Approches qualifiées d'analytiques	<i>Utilisation des modèles des tâches</i>							
	Les travaux de [Francisco et al, 08]				+			
	<i>les méthodes par simulation</i>							
	Les travaux de [Lebodic 05]				++	++	++	

- C_{Préférence} : l'adaptation du système aux préférences de l'utilisateur.
- C_{Intérêts} : l'adaptation aux centres d'intérêts de l'utilisateur.
- C_{Expérience} : l'adaptation à l'expérience de l'utilisateur.
- C_{Environnement} : l'adaptation à l'environnement d'utilisation du système.
- C_{Plate-forme} : l'adaptation du système à la plate-forme interactive.
- C_{Accessibilité} : l'adaptation à la capacité physique de l'utilisateur (accessibilité).
- C_{Comportement} : l'adaptation au comportement de l'utilisateur.
- ++ : Critère traité
- + : Critère partiellement traité

Conclusion

Nous avons présenté dans ce chapitre un panorama des méthodes d'évaluation. Nous avons en premier lieu proposé un ensemble des critères selon lequel un SIP peut être évalué. En second lieu, nous avons présenté quelques classifications existantes des méthodes et techniques d'évaluation dédiées aux systèmes interactifs et particulièrement aux interfaces homme-machine. Nous avons constaté que plusieurs axes de classification des méthodes d'évaluation sont possibles. Notre objectif n'étant pas de critiquer et d'analyser les classifications existantes en vue d'en proposer une nouvelle, dans la mesure où on retrouve explicitement ou implicitement dans la plupart des classifications existantes, les quatre approches proposées par [Huart, Kolski et Bastien 08], nous avons sélectionné cette classification en vue de présenter un panorama des méthodes et techniques d'évaluation existantes.

Ensuite, nous avons présenté les méthodes utilisables pour l'évaluation d'interfaces homme-machine. Celles-ci sont issues plus ou moins directement de domaines variés, tel que le génie logiciel, l'ergonomie, la psychologie, l'acquisition de connaissance ou l'automatique humaine ; elles contribuent de manière générale au domaine vaste et pluridisciplinaire que constitue l'interaction homme-machine. Ces méthodes ont traité principalement l'utilisabilité, mais permettant l'évaluation partielle d'autres dimensions propres aux SIP.

Nous avons constaté, après la présentation d'une grande variété de méthodes contribuant à l'évaluation des interfaces homme-machine, qu'à notre connaissance, il n'existe pas, au stade actuel de la recherche, d'outil ou de méthode spécifique à l'évaluation des SIP. Mais même si elles sont prioritairement tournées vers l'évaluation de l'utilisabilité, il nous semble que certains de ces méthodes peuvent être adaptées et combinées pour l'évaluation des SIP.

Dans la lignée des méthodes, techniques et outils examinés dans ce chapitre, nous avons proposé et développé, dans le cadre de cette thèse, trois démarches d'évaluation des SIP. Ces démarches seront décrites dans le chapitre suivant.

Chapitre 3

Contribution à l'évaluation des systèmes d'information personnalisés

Sommaire

Introduction	69
3.1. Utilisation d'IDEF0 pour la représentation des activités constitutives des méthodes proposées	69
3.2. La Méthode MetQus	70
3.2.1. Principe et description	70
3.2.2. Synthèse et discussion.....	73
3.3. La Méthode MetTra	73
3.3.1. Principe et description	75
3.3.2. Synthèse et discussion.....	92
3.4. La Méthode MetSim	92
3.4.1. Principe et description	97
3.4.2. Synthèse et discussion.....	97
Conclusion.....	98

Introduction

On assiste aujourd'hui à une évolution très rapide des systèmes d'information personnalisés (SIP). En effet, de nombreux travaux sont menés aussi bien dans le monde académique que dans celui des entreprises sur la modélisation de l'utilisateur, les méthodes de conception de SIP, les outils de construction d'interfaces adaptées, les algorithmes de personnalisation, etc. Cependant, l'évaluation de ces systèmes est négligée, voire ignorée par les spécialistes du domaine. Pour combler ce manque, il s'agit d'étudier et mettre en place une batterie de méthodes permettant d'évaluer et d'améliorer leur qualité de personnalisation. C'est dans ce contexte que nous proposons trois démarches complémentaires d'évaluation : la première est, intitulée MetQus (METHode d'évaluation basée sur un QUESTIONNAIRE de Satisfaction) ; la seconde prend la forme d'une évaluation précoce qui consiste à prévoir l'évaluation dès les premières étapes de développement d'une application personnalisée en intégrant un système de trace elle est, intitulée MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage) ; la troisième correspond à une évaluation par simulation d'une maquette de SIP, intitulée MetSim (METHode d'évaluation par SIMulation). Ces méthodes sont successivement présentées, en commençant par la description du moyen de modélisation permettant de les expliquer.

3.1. Utilisation d'IDEF0 pour la représentation des activités constitutives des méthodes proposées

Pour mieux structurer nos propositions, nous utilisons le formalisme IDEF0 (Integrated DEFINition method) [Ross 77] [Ang 99]. Il s'agit d'une représentation graphique. Les primitives utilisées sont très simples : des boîtes représentant des activités, et des flèches représentant les relations entre les activités. Les relations peuvent être de quatre types en regard d'une activité : entrée, sortie, contrôle, moyen ou mécanismes. Les données en entrée sont transformées en données de sortie. Le contrôle régit la manière dont l'activité est réalisée. Le mécanisme représente les moyens (matériaux, logiciels, acteurs, etc.) permettant de réaliser l'activité.

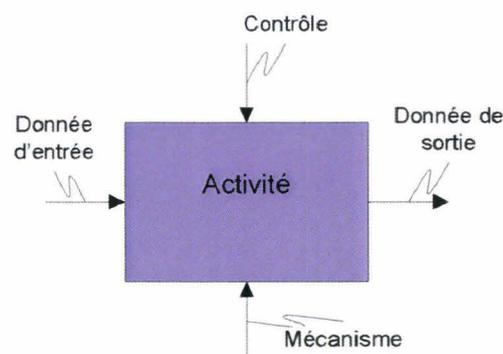


Figure 3.1. Diagramme d'activité en IDEF0

IDEF0 permet une présentation de chaque méthode proposée par une série de diagrammes structurés en arbres (du niveau le plus général aux niveaux les plus détaillés). Chaque activité est représentée graphiquement sur un diagramme par une boîte (Figure 3.1). Celle-ci décrit une action, un processus ou une opération. Les boîtes d'un diagramme communiquent entre elles par échange de données (objets ou informations) ou par envoi de contrôles.

La section suivante présente les trois méthodes proposées dans le cadre de notre travail au travers d'IDEF0.

3.2. La Méthode MetQus

Nous présentons dans cette section le principe et la description de notre méthode MetQus (METHode d'évaluation basée sur un QUESionnaire de Satisfaction).

3.2.1. Principe et description

L'objectif de cette technique est de recueillir, à l'aide de documents structurés, un ensemble d'appréciations, d'opinions et d'attitudes, de l'utilisateur après qu'il ait interagi avec un système d'information personnalisé. Elle nécessite l'existence d'un prototype, ou maquette. Notre méthode repose sur trois phases : (1) Préparation de l'évaluation, (2) Test de modèle, prototype ou maquette, (3) Analyse des données. Chaque phase sera présentée par la suite avec ses activités détaillées. Son déroulement est illustré par la figure 3.2.

Selon le formalisme IDEF0, on retrouve dans la boîte A0 l'objectif global. La figure 3.2 montre l'objectif global de notre démarche qui consiste à évaluer un système d'information personnalisé (SIP). Pour atteindre ce but, nous l'avons décomposé en trois sous-butts correspondant aux boîtes A1, A2 et A3.

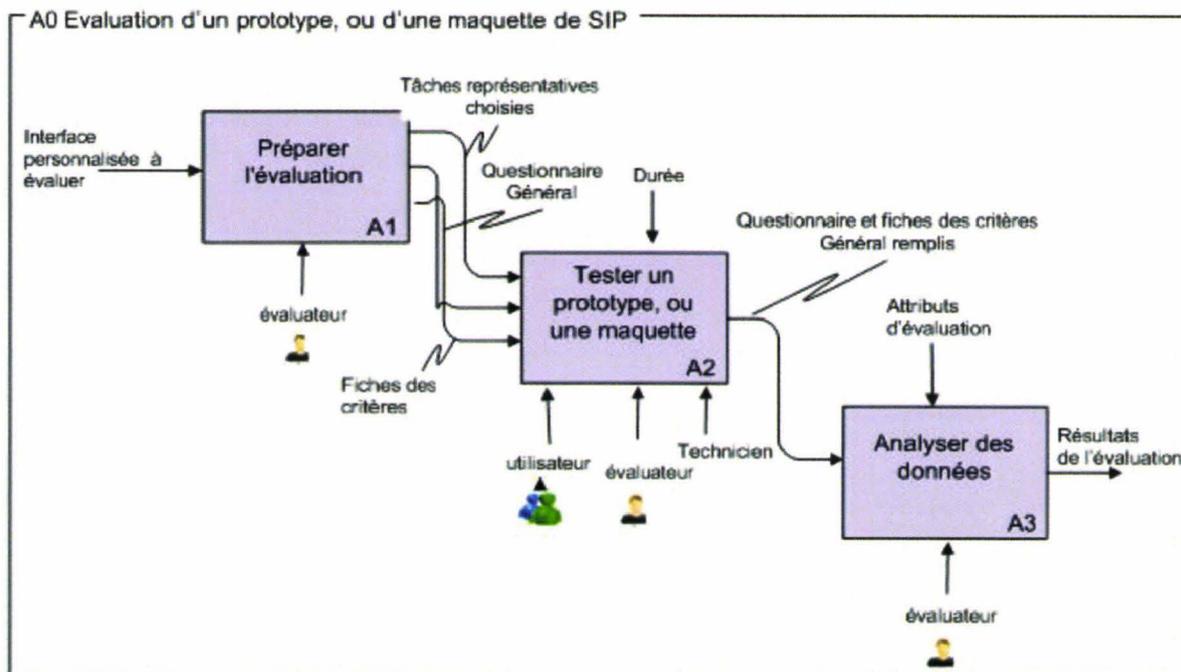


Figure 3.2. Les phases de la méthode MetQus

3.2.1.1. Préparation de l'évaluation

Dans cette phase, l'évaluateur choisit les tâches représentatives du système dans le but d'exploiter ces tâches lors de l'évaluation d'un prototype, ou maquette. Il prépare aussi deux types de documents nécessaires pour l'évaluation. Le premier est un questionnaire général qui contient des informations générales concernant les utilisateurs (sujets de l'expérimentation) et le deuxième est une fiche qui contient la définition de chaque critère ainsi que les parties à remplir par l'utilisateur lors de l'expérimentation.

3.2.1.2. Test d'un prototype ou d'une maquette

Cette phase se compose de deux étapes : La figure 3.3 détaille l'activité A2 «Tester un prototype, ou une maquette». Elle débute par l'étape de pré-expérimentation qui représente l'étape de préparation dans notre démarche (cf. Figure 3.3, A2.1). Ensuite, l'étape d'expérimentation durant laquelle le système sera évalué (cf. Figure 3.3, A2.2).

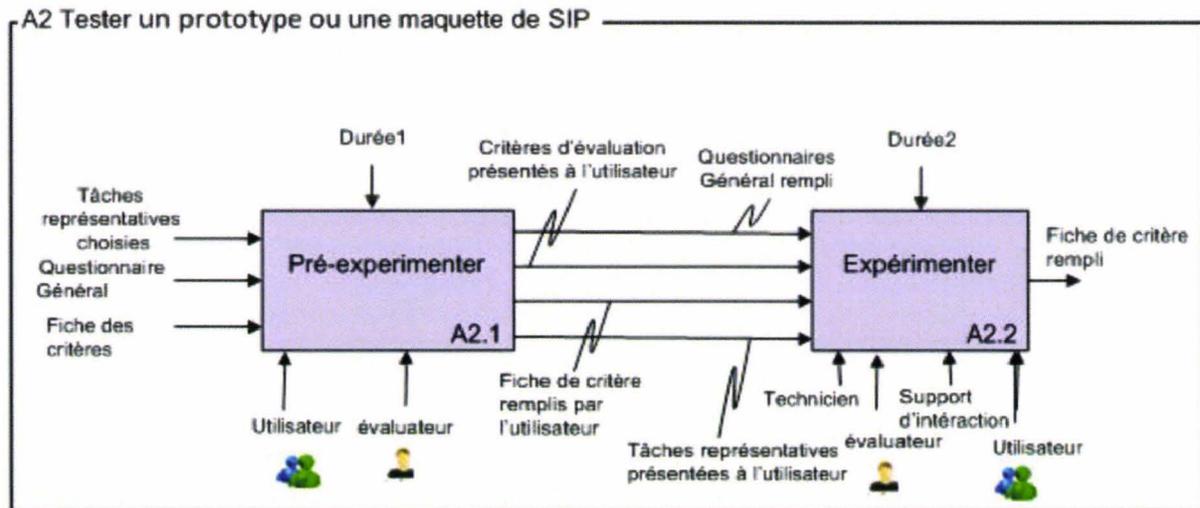


Figure 3. 3. Activité de test

L'activité A2.1 représente l'étape pré-expérimentale. C'est à ce niveau que l'évaluateur présente aux utilisateurs les tâches déjà choisies dans la phase précédente ainsi que les critères d'évaluation et leur définition [Soui et al., 07]. Ensuite il leur demande de sélectionner les critères (cf. 2.2.1) les plus importants selon ses besoins et ses préférences. L'utilisateur doit attribuer pour chaque critère sélectionné un poids représentant l'importance qu'il porte à ce critère, par exemple si l'utilisateur est intéressé par l'itinéraire le moins cher alors son poids sera certainement élevé. Notons que ce poids est borné à la valeur dix (cf. Figure 3.4).

- | | |
|---|---|
| 1 | Définition du critère |
| 2 | Principe de l'intérêt porté ou non au critère |
| 3 | Poids à affecter au critère |
| 4 | Niveau de satisfaction à affecter au critère |
| 5 | Description des problèmes rencontrés |

Figure 3.4. Structure du support d'évaluation

L'activité A2.2 représente l'étape expérimentale. En effet, les utilisateurs évaluent le système au fur à mesure qu'ils exécutent les tâches en fonction des critères déjà définis. Pour cela l'utilisateur affecte à chaque critère, une note selon son niveau de satisfaction et ceci suivant une échelle prédéfinie. En outre, dès qu'ils rencontrent un problème (correspondant au non respect d'un critère) au niveau d'une action d'une tâche, ils le notent et commentent en parallèle sur une fiche de problème de description. La figure 3.5 illustre l'évaluation expérimentale.



Figure 3.5. Evaluation de l'utilisation selon les critères proposés
(Cas où le sujet est assisté par l'évaluateur)

3.2.1.2. Analyse des données

L'étape A3 représente la phase d'analyse. Ainsi des données quantitatives et qualitatives sont recueillies par l'intermédiaire des fiches remplies par les utilisateurs. C'est à ce niveau que l'évaluateur calcule pour chaque utilisateur, le niveau de satisfaction N_s qui traduit le niveau de personnalisation N_p du système en appliquant la formule suivante :

$$N_s = N_p = \frac{\sum_{k=1}^n W_K \times I_a \times N_K}{\sum_{k=1}^n W_K \times I_a}$$

Avec :

n : nombre de critères concernés par l'utilisateur.

W_K : le poids intérêt représentant l'importance que l'utilisateur porte à ce critère

I_a : l'indice d'activation de critère

1 si le critère K est concerné par l'utilisateur U
0 sinon

N_k : Note attribuée par l'utilisateur au critère K

0 si l'utilisateur n'est pas du tout satisfait

0,25 si l'utilisateur est peu satisfait

0,5 si l'utilisateur est moyennement satisfait

0,75 si l'utilisateur est satisfait

1 si l'utilisateur est très satisfait

Le résultat est une valeur bornée entre 0 et 1, donc plus la valeur de niveau de satisfaction est proche de 1, plus le système est jugé adapté à l'utilisateur. Pour juger globalement le système selon son degré de personnalisation, on calcule le niveau de satisfaction moyen N_{sm} de tous les sujets qui participent à l'évaluation. En effet, les évaluations de la personnalisation consistent à estimer la qualité de personnalisation réalisée. Il s'agit de vérifier si la personnalisation effectuée au niveau du service répond bien aux préférences de l'utilisateur. Par exemple, pour un service de recherche d'information personnalisée, une évaluation de la personnalisation consisterait à vérifier si les informations présentées à l'utilisateur correspondent bien à celles que l'utilisateur voulait recevoir. Il s'agira donc de mesurer le degré de satisfaction de l'utilisateur par rapport à la personnalisation fournie.

3.2.2. Synthèse et discussion

Cette évaluation est basée sur l'intervention explicite de l'utilisateur et l'existence d'un système réel (maquette, système final). Cette méthode permet de prendre en compte tous les critères d'évaluation liés au contenu et celles liés au contenant. Bien que cette technique semble être facile à mettre en place, et permet de récupérer des informations précises réellement utiles pour l'évaluation, elle nécessite un traitement assez lourd et un coût relativement important. En effet, pour trouver un nombre satisfaisant de problèmes, de nombreux tests doivent être conduits. De plus, effectuer un seul test prend déjà un temps assez considérable : par exemple, comme on le détaillera dans le chapitre suivant, il a fallu environ 3h par sujet pour l'évaluation du système personnalisé MouverPerso, hors préparation et analyse des données).

3.3. La Méthode MetTra

Selon le même état d'esprit que les recherches sur les traces [Tarby 06], [Tarby et al., 07], [Cram et al., 07], la méthode MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage) consiste à prévoir l'évaluation dès les premières étapes de développement d'un SIP en intégrant un système de trace. Cette approche est basée sur la méthode PerMET [Anli 06] (une méthode adaptée à la conception des SIP) développée au LAMIH durant la thèse d'Anli (cf. 3.3.1.2). La partie suivante définit d'abord la notion de trace informatique, elle expose une critique de la méthode PerMET dont l'objectif est de dégager les contributions apportées dans le cadre de notre travail, puis elle présente les différents phases de notre méthode.

3.3.1. Les traces dans le cadre de l'évaluation de système d'information

Il n'existe pas de définition standard portant sur les concepts de trace et de système de trace. Dans le contexte des environnements informatiques pour l'apprentissage humain (EIAH), les traces sont des données issues d'observations directes ou indirectes de l'interaction de l'apprenant permettant la régulation, le contrôle, l'analyse et la compréhension de l'activité d'apprentissage [Jermann et al., 01]. Dans l'approche MUNETTE [Champin et al., 04], la trace d'utilisation d'un système informatique est structurée en état-transition et est exploitée afin d'être réutilisée selon le principe du Raisonnement à Partir de Cas. La figure 3.6 illustre cette méthode.

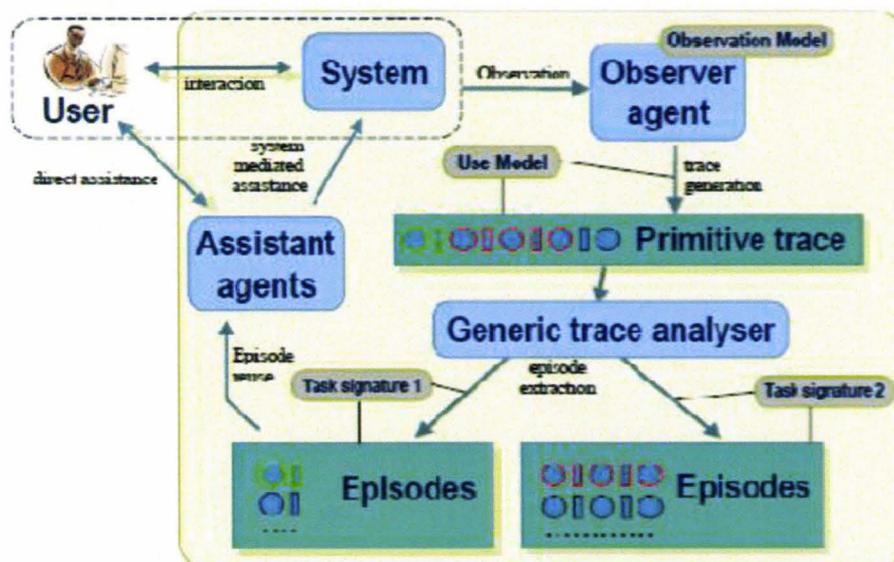


Figure 3.6. Framework de l'approche MUNETTE

[Settoui et al., 06] définissent la trace comme une séquence temporelle d'observées. Cette définition considère que la trace informatique est un document numérique qui révèle des données temporellement situées résultant d'une activité d'observation. Ces traces sont utilisées pour atteindre des objectifs différents, à titre d'exemples la régulation de l'activité d'apprentissage [Heraud et al., 05], la qualité du scénario pédagogique et la réingénierie des EIAH [Barré et al., 05].

3.3.2. Principe de base de PerMet

La méthode PerMet dont la première version avait été proposée dans le cadre de la thèse d'Anli en 2006, est une méthode adaptée à l'analyse, la conception et la modélisation des systèmes d'information personnalisés (SIP) [Anli et al., 05b]. Cette méthode permet aussi bien la mise en place d'un nouveau système d'information personnalisé que la personnalisation d'un système d'information déjà existant. PerMet propose un modèle de développement itératif, incrémental et permet une réalisation parallèle des phases spécifiques liées au développement des services (analyse, conception et implémentation du service) et des phases spécifiques liées à la personnalisation (analyse des agents, conception des comportements des agents et implémentation des comportements des agents). PerMet utilise des formalismes basés sur UML et ses extensions. La figure 3.7 illustre cette méthode.

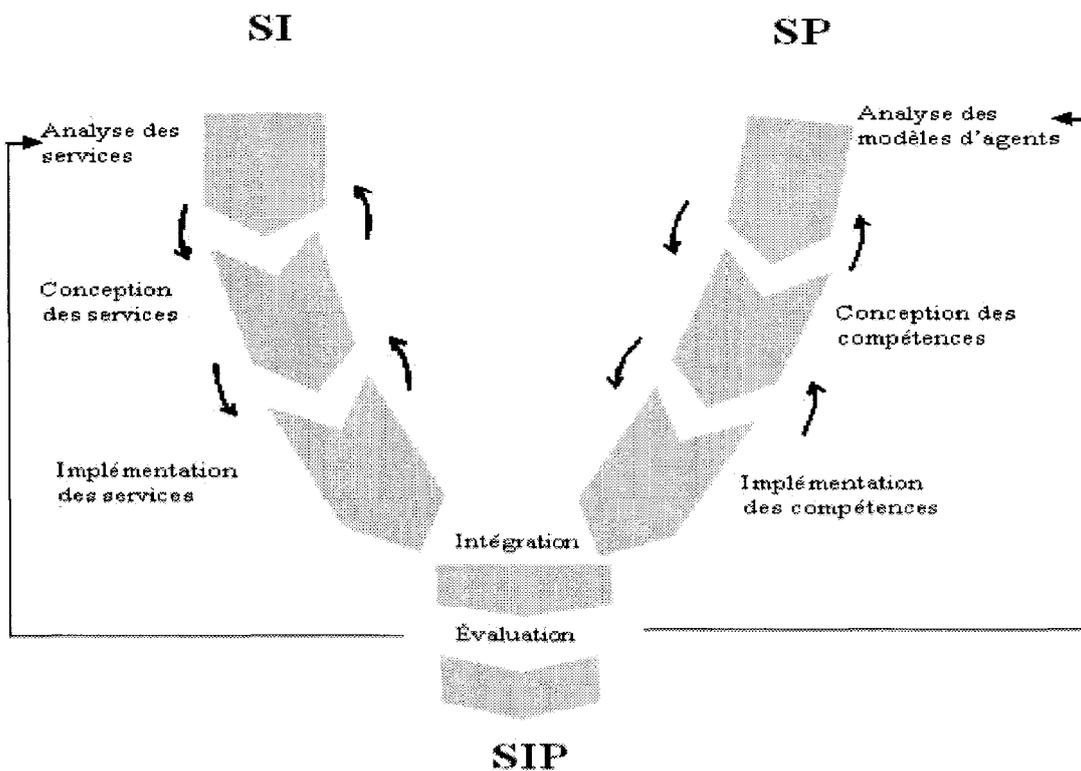


Figure 3.7. La méthodologie PerMet [Anli 06d]

Comme l'indique la figure 3.8, à l'issue de la phase d'intégration, il s'agit de tester et d'évaluer le service personnalisé obtenu. Différents types d'évaluations doivent être effectués ; on peut citer par exemple, des évaluations fonctionnelles, des évaluations de la personnalisation et des performances ainsi que des évaluations techniques et ergonomiques. La phase d'évaluation, dans la méthode PerMet, ne peut être effectuée que lorsque le système est dans un état avancé du développement. Il est nécessaire de disposer d'une interface réalisée ou d'un prototype fonctionnel.

PerMet contribue à l'enrichissement des méthodes de conception des systèmes d'information personnalisés. En effet, PerMet propose un modèle générique de développement itératif, incrémental et permet une réalisation parallèle des phases liées au développement des services d'une part, de celles liées à la personnalisation d'autre part. Les formalismes utiles à la modélisation structurelle et comportementale pour les phases d'analyse et de conception sont basés sur ceux d'UML et de ses extensions ce qui permet en principe une prise en main relativement facile (puisque une grande majorité de développeurs ont adopté UML), dans la mesure aussi où il existe de nombreux ateliers de génie logiciel dédiés; d'où son intérêt.

La méthode PerMet présente plusieurs avantages. Cependant, elle n'intègre pas du tout la notion de tâches telle que nous la connaissons en IHM. Par exemple, PerMet ne contient pas les notions de buts ou d'intentions de l'utilisateur ; elle ne permet pas de modéliser les utilisateurs (connaissances, habitudes, etc.) ; la différence entre tâche interactive, tâche système et tâche manuelle n'existe pas dans PerMet, bien que cette dimension soit primordiale pour l'évaluation de la personnalisation liée au contenant ou à l'adaptation des interfaces des systèmes d'information personnalisés.

La partie suivante présente l'idée de base de l'approche proposée ainsi que leurs différentes phases.

3.3.3. Principe et description de MetTra

Partant du principe que la meilleure façon de détecter les problèmes liés à un système d'information personnalisé est d'observer l'utilisateur réel lorsqu'il réalise sa tâche, l'idée de base de notre approche consiste à utiliser des modèles de tâches conjointement avec les diagrammes de PerMet pour intégrer un mécanisme de trace dès le début de la conception (selon le principe initial proposé par [Tarby 06]). Les avantages de ces modèles sont de plusieurs ordres : les modèles de tâches permettent, d'une part, de faciliter le recueil des informations sur la façon dont les utilisateurs accomplissent une activité, et d'autre part, ils contribuent à la spécification de l'IHM, qui définit la vue que l'utilisateur aura du système interactif. Ces notations peuvent constituer la base de départ de la réalisation d'une IHM. En effet, l'intégration des modèles de tâches issus du domaine de l'IHM est utile pour deux raisons principales : ces modèles de tâches peuvent servir d'outil dans la conception des systèmes interactifs. En plus, ils s'adressent à l'utilisateur pour l'aider dans l'utilisation du système via sa documentation, définie dans la phase de conception. Nous ajouterons à ces deux points que le modèle de tâches est un bon candidat pour servir de support à l'évaluation du système d'information personnalisé. De par sa nature, il permet d'examiner le comportement de l'utilisateur du système.

La méthode proposée repose sur un principe de séparation du système d'information (SI) et du système de personnalisation (SP). Les deux parties SI et SP (les deux branches du 'Y') suivent les phases de développement classique (analyse, conception et implémentation) pouvant se dérouler en parallèle et se rejoignent pour constituer le SIP. Dans la première version de la méthode PerMet, la partie SI est vue comme un ensemble de services. Par exemple, un système d'information personnalisé dans le domaine de transport contient des services comme la recherche d'itinéraire, l'information de trafic routier, etc. En revanche, cette initiative ne sépare pas d'une manière explicite le noyau fonctionnel de l'application et l'interface de l'utilisateur, ce qui représente une contrainte pour estimer la qualité de l'interface d'un SIP. Pour répondre à cet objectif, nous proposons d'intégrer des mécanismes de traces au niveau de la partie SI. Cette intégration préconise une spécification de l'interface de l'utilisateur d'une manière explicite en se basant sur la modélisation de tâche. Cette spécification s'appuie sur deux modèles : le modèle statique et le modèle du dialogue de la tâche. Cependant, rares sont les extensions faites sur UML et qui ont abordé la modélisation

de tâche. Pour cela, nous proposons d'étendre UML par d'autres diagrammes pour introduire cette dimension dans la méthode PerMet qui utilise déjà des formalismes basés sur le langage UML. En effet, nous proposons une représentation d'un modèle de tâche en se basant sur les diagrammes d'UML et de ses extensions. Etendre UML consiste à définir de nouveaux stéréotypes, étiquettes et contraintes pour prendre en compte les spécificités d'un domaine d'application [Booch et al., 00].

Notre méthode repose sur quatre phases : (1) l'intégration du mécanisme de traces pendant la phase de conception (cf. Figure 3.8, A1.1), (2) le recueil des données relatives au comportement de l'utilisateur lors de la réalisation de sa tâche ainsi que les réactions des interfaces homme-machine du système (cf. Figure 3.8, A1.2), (3) l'identification et la description des problèmes rencontrés par l'utilisateur (cf. Figure 3.8, A1.3), (4) l'identification et l'explication des recommandations (cf. Figure 3.8, A1.4).

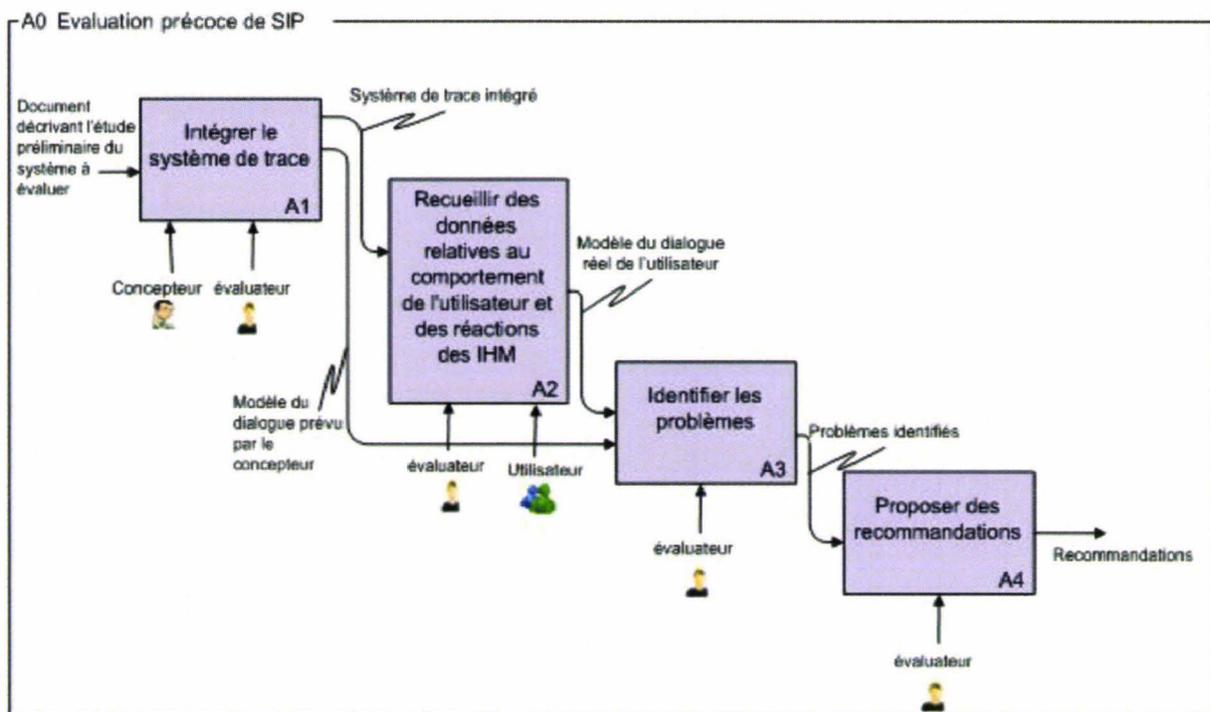


Figure 3.8. Les différentes phases de la méthode MetTra

3.3.1.3. L'intégration du mécanisme de traces pendant la phase de conception

Cette phase se compose de trois étapes (analyse, conception et implémentation du service). La figure 3.9 illustre cette phase. Son déroulement est décrit ci-après.

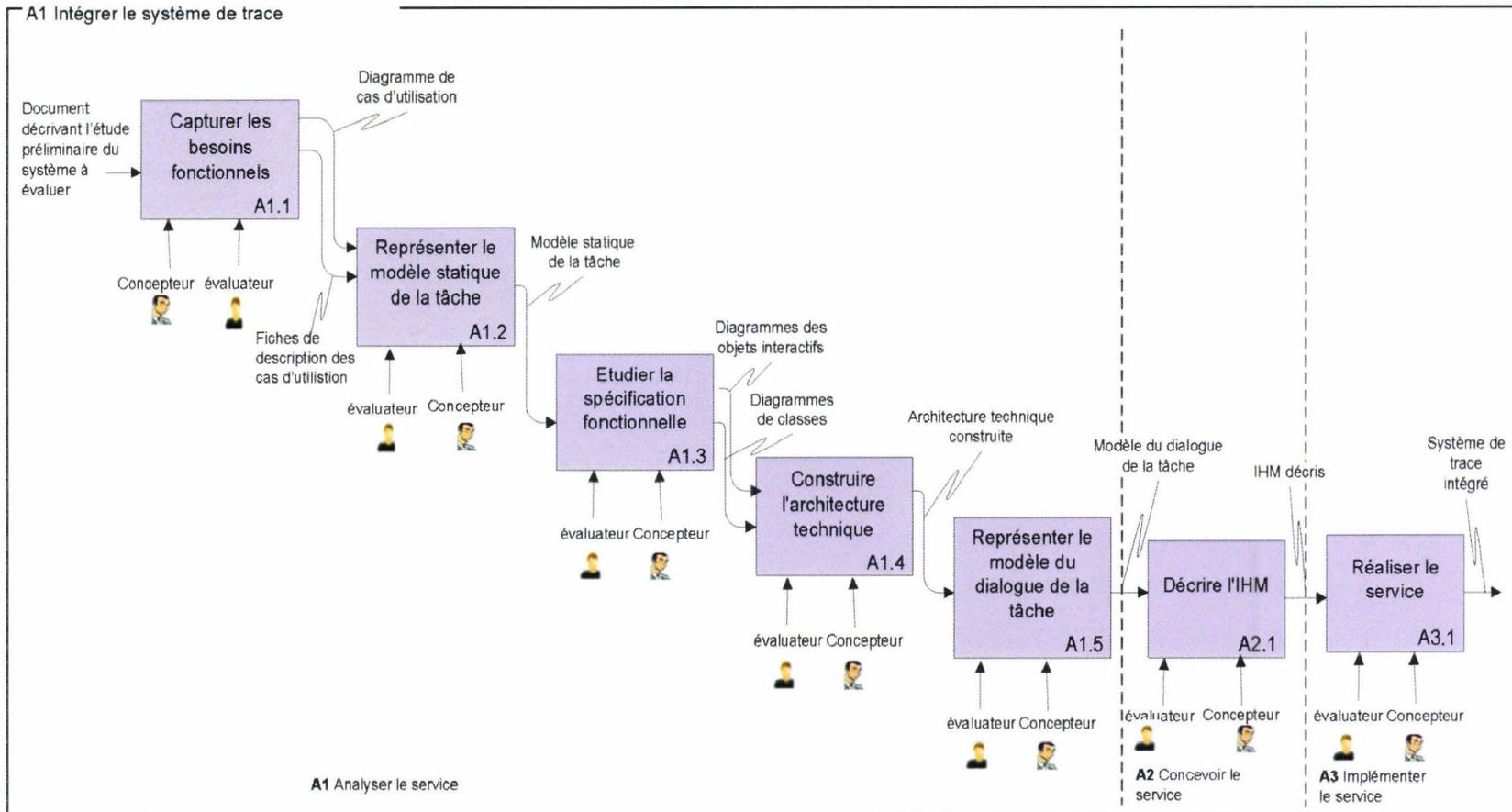


Figure 3.9. Intégration de mécanisme de trace

3.3.1.3.1. Analyse du service

Le point d'entrée fourni par la méthode est l'analyse des besoins des utilisateurs. C'est dans cette étape qu'on spécifie le modèle de données qui seront échangées entre le SI et le SP. C'est à ce niveau que s'effectue aussi la modélisation de la tâche. Ainsi, l'objectif du modèle de la tâche que ce soit MAD (Model for Activity Description) [Scapin et Pierret-Golbeich 90], MAD* [Scapin et Bastien 01], K-MAD (Kernel of Model for Activity Description) [Baron, et al., 06], ADEPT (Advanced Design Environment for Prototyping with Task Model) [Johanson et al., 93], DIANE+ [Tarby 93] ou GLADIS++ (Guide et LAngage de Description des Interfaces homme-machine de Système d'information et de communication) [Buisine 99] est de prendre en compte l'utilisateur et sa tâche le plus tôt possible. Son but est d'abstraire les informations de la tâche utilisateur nécessaires à la conception des interfaces homme-machine. La construction de ce modèle dans la méthode proposée suit le même principe que la méthode TOOD (Task Oriented Object Design) qui propose l'élaboration de deux modèles, l'un statique et l'autre dynamique [Mahfoudi et al., 01]. L'étape d'analyse comprend six étapes successivement décrites.

- **Capture des besoins fonctionnels** : la capture des besoins formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire. Il s'agit d'une description exhaustive des besoins fonctionnels et opérationnels par la modélisation des comportements attendus du service à développer. Les scénarios d'interaction entre l'utilisateur final et le service sont modélisés ainsi que les exigences fonctionnelles du service sans imposer le mode de réalisation de ces fonctionnalités. Arrivés à ce point du cycle, il est trop tôt pour décrire la manière dont l'utilisateur va interagir avec le service. Par exemple, on dira « le système reçoit une demande de recherche d'itinéraire de l'utilisateur » au lieu de dire « l'utilisateur doit cliquer sur le bouton chercher » [Roques 05].

La modélisation des besoins fonctionnels s'effectue au travers des diagrammes de cas d'utilisation. Ces cas d'utilisation vont piloter la réalisation de toutes les autres phases du cycle. Ainsi chaque cas d'utilisation spécifie un comportement attendu du système sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera. Chaque cas d'utilisation doit être suivi d'une fiche de description textuelle. Cette fiche de description n'est pas normalisée par UML. Nous préconisons pour notre part une structuration de cette fiche en quatre parties. La figure 3.10 présente la structure d'une fiche de description textuelle pour un cas d'utilisation. Il peut être utile de détailler certains cas d'utilisation par des scénarios au travers de diagrammes de séquences, d'activités et/ou de collaboration. C'est dans cette étape qu'on choisit aussi les tâches représentatives du système qui seront utilisées dans l'évaluation. Lorsqu'un utilisateur réalise l'une de ces tâches, des traces sont produites et stockées dans un fichier. Ces traces « orientées usage » décrivent comment cet utilisateur met en pratique le système durant la réalisation de cette tâche.

1. Sommaire d'identification : inclut titre, but, résumé, acteur...
2. Description des enchaînements : décrit les enchaînements nominaux en distinguant le type de tâche (utilisateur, interactive, système) (cf. Figure 3.13), les enchaînements exceptionnels, les exceptions, mais aussi les préconditions, et les postconditions. Cette étape représente un support de base pour élaborer le modèle de tâche associé à ce cas d'utilisation.
3. Besoin d'IHM : cette étape ajoute éventuellement les contraintes d'interfaces homme-machine : ce qu'il est nécessaire de montrer, en conjonction avec les opérations que l'utilisateur peut déclencher (tâche interactive).
4. Contraintes non-fonctionnelles : fréquence, temps de réponse, sortie du système.

Figure 3.10. La structure d'une fiche de description textuelle d'un cas d'utilisation

- **Modèle Statique de la Tâche (MST) :** le modèle de tâche statique représente la tâche que l'utilisateur doit pouvoir effectuer au travers de l'interface. Cette tâche déjà définie dans l'étape de capture de besoins fonctionnels sous la forme d'un cas d'utilisation global, sera décomposée en une hiérarchie de tâches et de sous-tâches. La construction de chaque modèle de tâche statique associée à un cas d'utilisation global est guidée par la seconde partie (description des enchaînements) de sa fiche de description déjà décrite dans l'étape précédente. Une extension d'UML, basée sur le diagramme d'activité est proposée pour représenter le modèle de tâche statique. En fait, chaque tâche ou sous tâche est représentée par un état (cf. Figure 3.11). L'ensemble des sous tâches enfants d'une tâche sont représentées par un diagramme d'activités composé d'états (état initial, états intermédiaires, état final) et des transitions (les passages possibles d'un état à un autre). Pour élaborer le modèle de tâche statique on propose de suivre les constructeurs de la figure 3.12 pour identifier les relations inter-tâches.

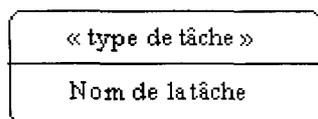


Figure 3.11. Représentation graphique d'une tâche

Notation (Stéréotype)	Signification
Séquence ou « → »	« T1 → T2 » : Quand T1 est terminée T2 devient active.
Séquence et Transfert ou « ≈> »	« T1 ≈> T2 »: Quand T1 se termine elle fournit des informations à T2 et l'active.
Synchronisation « S »	« T1 S T2 » : T1 et T2 doivent se synchroniser sur certaines actions pour échanger des informations.
Imbrication ou « »	« T1 T2 »: Les actions T1 et T2 peuvent être effectuées dans n'importe quel ordre.
Parallèle ou « »	« T1 T2 » : T1 et T2 sont activables en même temps, il s'agit de l'opérateur de parallélisme.
Désactivation ou « # »	« T1 # T2 »: Quand une action de T2 se produit la tâche T1 est désactivée.
Choix ou « © »	« T1 © T2 »: si T1 est activée alors T2 n'est plus activable et réciproquement.
Interruption « »	« T1 T2 » : T2 peut interrompre T1, T1 peut reprendre à la fin de T2.
Décomposition ou « D »	Une tâche T est décomposée en un ensemble de sous-tâches.

Figure 3.12. Relations inter-tâches inspirées de la notation CTT (ConcurTaskTrees) [Paternò et al., 97]

Stéréotypes	Signification
Tâche Interactive ou « I »	Lorsqu'une tâche est associée à une action (ou plusieurs) d'un utilisateur sur le système, alors c'est une tâche « interactive ». La tâche est déclenchée par un utilisateur et réalisée conjointement sur le système.
Tâche Utilisateur ou « U »	La tâche « utilisateur » est une tâche réalisée entièrement par l'utilisateur.
Tâche Système ou « S »	La tâche est réalisée complètement par le système.

Figure 3.13. Les types des tâches inspirées de la notation CTT (ConcurTaskTrees) [Paternò et al., 97]

Le modèle statique représente la structure de décomposition du travail utilisateur sous forme de tâches. Par conséquent, il n'introduit pas la dimension dynamique des tâches. Pour y arriver, ce modèle requiert une transformation en un modèle dynamique de la tâche.

- **Analyse fonctionnelle** : cette phase consiste à étudier la spécification fonctionnelle au travers d'une analyse orientée objet pour trouver les classes fondamentales du système et leurs relations. Ces classes sont organisées suivant un regroupement logique sous forme de catégories (packages). Les classes sont complétées et détaillées itérativement au travers d'une analyse de la dynamique du système modélisée par des diagrammes de séquences, de collaboration et d'états.

En partant du modèle de tâche statique, les premières classes candidates identifiées dans cette étape doivent être des concepts connus des utilisateurs du système, ce qu'on appelle couramment des objets métiers. Le concept de classe dans le cadre de notre travail est étendu faisant apparaître un nouveau compartiment qui s'ajoute au nom de la classe. Ce compartiment est destiné à identifier les classes qui ont une relation directe avec l'intervention de l'utilisateur lors de l'exécution de sa tâche. Les noms de ces entités sont précédés par le stéréotype « tâche » (cf. Figure 3.14). C'est aussi dans cette étape que les modèles des objets interactifs manipulés par l'utilisateur sont définis. Ces modèles sont représentés par des diagrammes d'objets. En effet, en partant de modèle de tâche statique, on définit pour chaque tâche interactive possédant des sous tâches élémentaires, les objets interactifs nécessaires pour la réalisation de ces actions élémentaires. L'identification de ces objets est primordiale afin de générer des traces et vérifier la conformité de l'utilisation du système avec le modèle de tâches qui reste l'objectif de notre travail. Pour l'association entre la classe représentant la tâche interactive et les objets qui lui sont associées, nous proposons le stéréotype « utilise ». Les objets de la même tâche interactive peuvent être regroupés dans une catégorie (package), (cf. Figure 3.15). Cette catégorie sera représentée au niveau de l'interface sous la forme d'un conteneur.

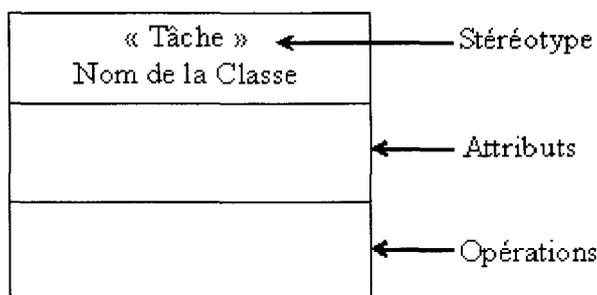


Figure 3.14. Représentation graphique d'une classe

Comme nous l'avons déjà montré plus haut, nous avons besoin d'enregistrer les actions élémentaires de l'utilisateur et les réactions des interfaces dans l'optique d'évaluation. Or ces actions correspondent à l'utilisation des objets interactifs de l'interface. Pour cela, la méthode proposée consiste à enregistrer les noms des objets interactifs utilisés par l'utilisateur et les noms des interfaces qui interagissent dans un fichier de trace au fur et à mesure que celui-ci réalise sa tâche.

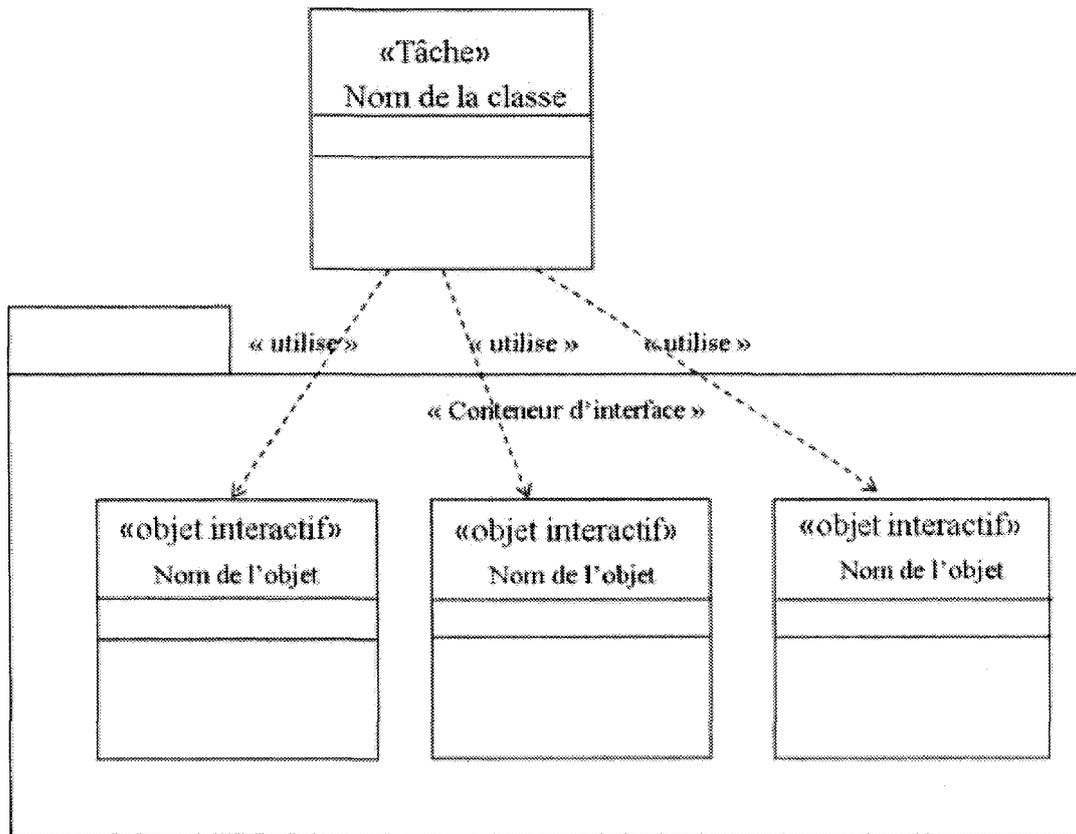


Figure 3.15. Modèle des objets interactifs

- **Capture des besoins techniques** : cette étape couvre, par complémentarité avec celle des besoins fonctionnels, les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description du service. Il s'agit de la spécification des outils (logiciels), de la structure des matériels à exploiter et des contraintes d'intégration avec l'existant. La configuration matérielle est modélisée au travers de diagrammes de déploiement et la spécification logicielle au travers de diagrammes de composants.
- **Conception générique** : la conception générique définit les éléments nécessaires à la construction de l'architecture technique indépendamment des aspects fonctionnels issus de la capture des besoins fonctionnels et de l'analyse fonctionnelle.
- **Modèle du dialogue de la tâche (MDT)** : le modèle de dialogue est destiné au concepteur pour décrire la dynamique de la tâche. Pour représenter ce modèle, plusieurs notations sont utilisables comme les automates à états finis et les réseaux de Petri [Diaz et al., 02], etc. Dans le même esprit, nous proposons un modèle de dialogue à base des diagrammes d'états/transitions pour décrire les conditions d'enclenchement et de terminaison d'une tâche (cf. Figure 3.16.) En effet, chaque transition est conditionnée par une garde ayant pour forme Événement [condition]/Action, qui peut exécuter une action et générer des événements. Chaque garde peut contenir plusieurs événements, pour cela, on propose d'associer à chaque événement un ordre qui représente la priorité de son exécution.

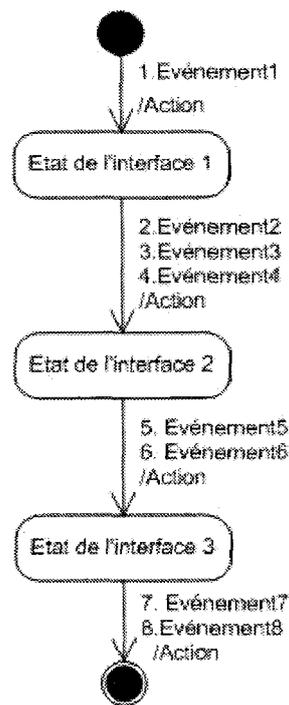


Figure 3.16. Structure du modèle de dialogue

3.3.1.3.2. Conception du service

L'objectif de cette phase est d'une part de décrire l'interface utilisateur à un haut niveau d'abstraction, et d'autre part de préciser le modèle d'analyse de telle sorte qu'il puisse être implémenté avec les éléments de l'architecture. Il s'agit d'exprimer les modèles statiques et dynamiques qui seront traduits directement sous forme de codes exécutables dans la phase d'implémentation du service. En fait, c'est à ce niveau que les diagrammes dynamiques sont utilisés pour décrire les états des classes (diagramme d'états), l'algorithme des méthodes (diagramme d'activités) et les interactions entre les différentes classes (diagramme de séquence et/ou de collaboration).

3.3.1.3.3. Implémentation du service

C'est la phase de la réalisation effective du service. Le service est développé conformément aux modèles conceptuels définis lors de la phase de conception du service. C'est à ce niveau que s'effectue l'intégration du système de traces. En effet, pour implémenter la technique d'enregistrement des événements des utilisateurs et la réaction des interfaces du système pendant la réalisation des tâches à évaluer, nous avons défini une fonction « enregistrer » en ajoutant un attribut qui correspond au nom de l'objet interactif. Lorsqu'un utilisateur utilise un objet interactif, cette fonction permet de renvoyer son nom dans le fichier de trace.

3.3.1.4. Recueil des données relatives au comportement de l'utilisateur et réactions des interfaces du système

Cette phase est composée de trois étapes : (1) la définition du contexte d'évaluation (cf. Figure 3.17, A1.2.1), (2) l'enregistrement des actions élémentaires de l'utilisateur et les réactions des interfaces du système (cf. Figure 3.17, A1.2.2), (3) la représentation du modèle de dialogue (cf. Figure 3.17, A1.2.3).

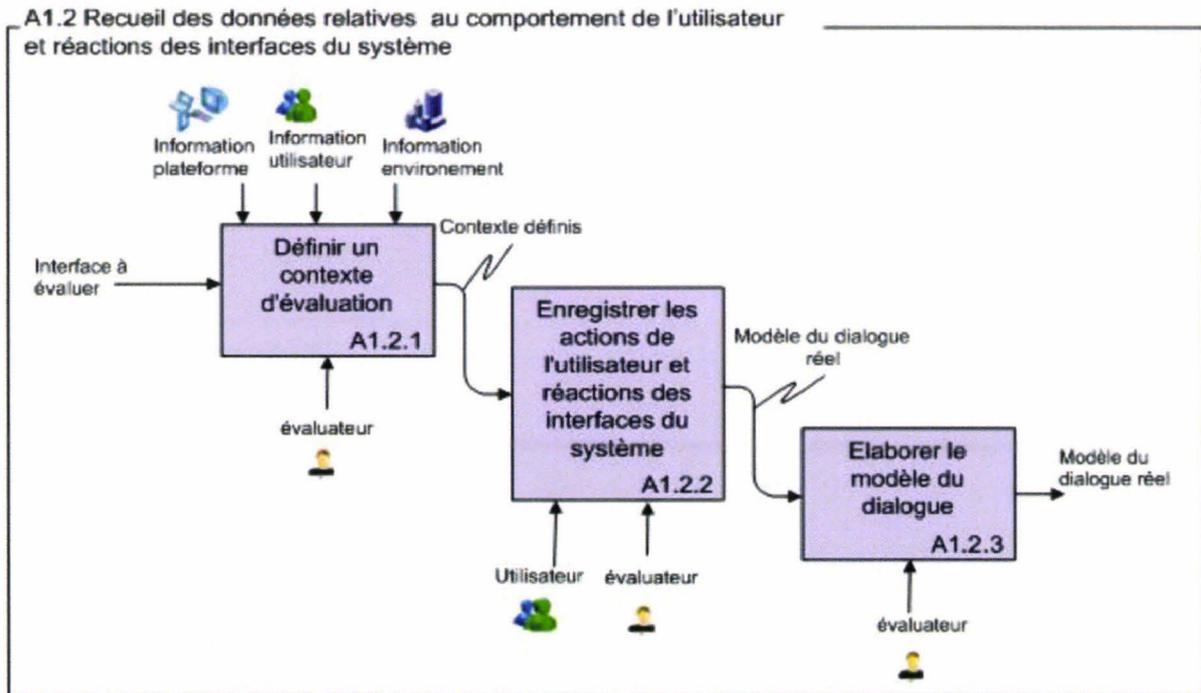


Figure 3.17. Recueil des données relatives au comportement de l'utilisateur et réactions des interfaces du système

3.3.1.4.1. Définition du contexte d'évaluation

Cette étape consiste à définir le contexte d'évaluation. Ainsi, selon le critère à vérifier, l'évaluateur doit choisir la catégorie des sujets qui seront impliqués dans l'évaluation (novices, experts, etc.), le support avec lequel les sujets interrogent le système pendant l'évaluation (PDA, PC, etc.) et l'environnement accueillant l'interaction. Ces choix sont définis en fonction des critères d'évaluation selon lesquels un SIP peut être évalué. Par exemple, pour s'assurer qu'un utilisateur novice réussit facilement à interagir avec une interface de recherche d'itinéraire, une observation de l'utilisation du système auprès d'un ensemble de sujets novices est recommandée. Cette observation permet à l'évaluateur de juger l'adaptation du système au niveau d'expertise des sujets novices. En effet, l'évaluateur peut comparer le modèle de tâche à réaliser et le modèle de tâche observée qui sont déterminés à partir des enregistrements effectués par le système de trace. L'observation des utilisateurs novices permet d'aider l'évaluateur de vérifier si les sujets ont pu mener à terme leur tâche, s'ils sont bloqués ou non pendant la réalisation de leur tâche et si l'interface fournit à l'utilisateur novice une procédure claire et un guidage suffisant pour effectuer sa tâche.

3.3.1.4.2. L'enregistrement des actions de l'utilisateur et réactions des interfaces du système

Cette étape assure la capture des actions élémentaires de l'utilisateur. Ces événements sont enregistrés dans des fichiers de traces sous format texte ou XML. En fait les noms des objets interactifs utilisés par l'utilisateur et les noms des interfaces réagissant sont accumulés dans une liste au fur et à mesure de l'exécution de sa tâche. L'ordre d'insertion dans la liste détermine l'ordre d'ordonnancement des tâches (utilisateur, système). La figure 3.18 illustre le fichier de trace.

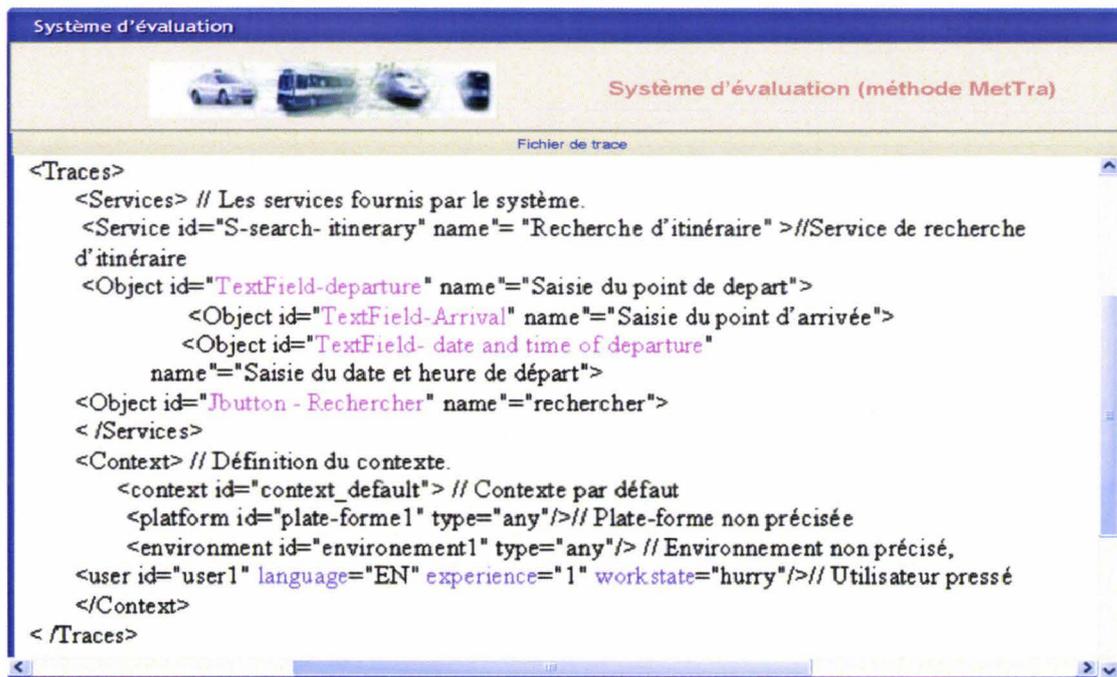


Figure 3.18. Exemple d'extrait d'un fichier de trace sous format XML

Les actions élémentaires de l'utilisateur qui sont capturés par le système de trace (niveau intermédiaire) seront associées aux services correspondants au niveau abstrait (le plus élevé). Les associations créées correspondent aux services que l'utilisateur a réalisés lorsqu'il a utilisé un SIP. Puis, le nombre d'objets interactifs sur l'interface et le nombre des services réalisés par l'utilisateur est comptabilisé et le pourcentage de succès et/ou d'échecs des tâches réalisés est aussi calculé. Les données capturées sont affichées via une interface homme-machine afin d'aider l'évaluateur à les analyser et suggérer les améliorations nécessaires au concepteur. La figure 3.19 est une capture d'écran d'une interface qui montre des résultats d'analyses réalisées sur les données capturées lors d'une expérimentation. Notons que les détails de cette expérimentation seront présentés dans le chapitre suivant.

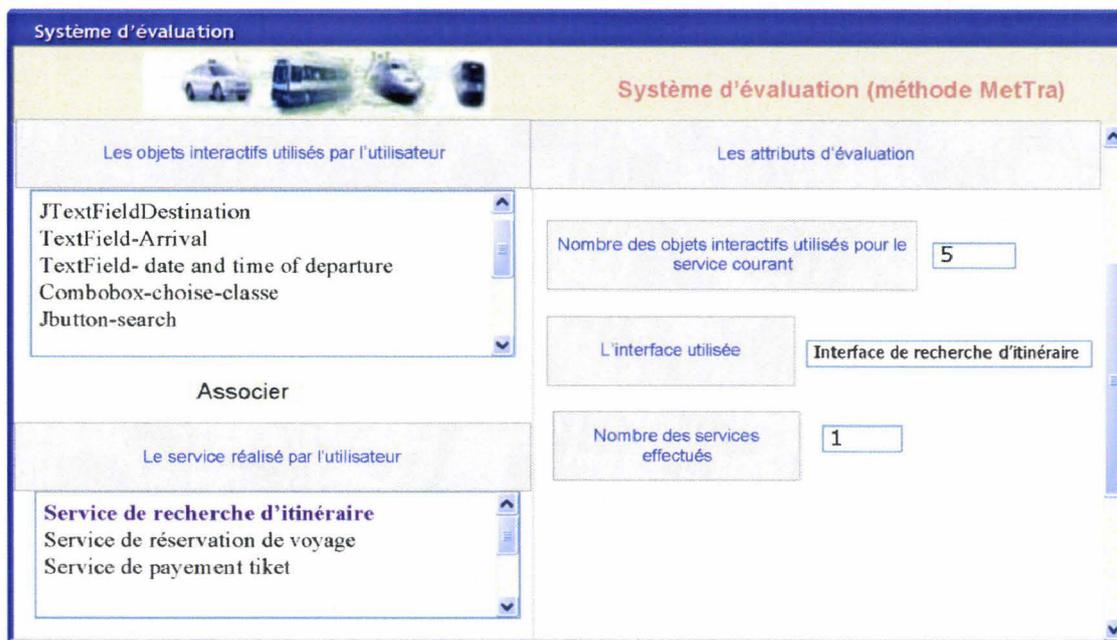


Figure 3.19. Capture d'écran de l'interface de l'activité de capture des données – exemple

Dans l'exemple de la figure 3.19, l'évaluateur peut associer les objets interactifs comme JTextFieldDestination, JTextFieldArrival, JTextField-date-and-time-departure, Combobox-choise-classe, Jutton-search au service correspondant « *recherche d'itinéraire* ». Cette association signifie que ce service correspond à une des tâches réalisées en réalité par l'utilisateur ; l'évaluateur peut l'enregistrer pour une analyse ultérieure.

3.3.1.4.3. Elaboration du modèle de dialogue

Les données enregistrées sous forme de fichier dans l'étape précédente sont exploitées afin de générer un modèle de dialogue représentant l'activité réelle de l'utilisateur ainsi que les réactions de l'interface en se basant sur l'algorithme décrit ci-dessous. Rappelons ici que ce modèle est représenté sous la forme d'un diagramme d'états/transitions étendu. Ce modèle peut servir d'outil d'évaluation qui sera utilisé dans la phase d'identification des problèmes. La figure 3.20 illustre l'algorithme que nous avons proposé pour générer le modèle de dialogue.

```

Algorithme Génération Modèle de dialogue
Début :
Ouvrir (fichier_trace, en lecture)
Tampon : chaîne de caractère ← vide
Ancien_Tampon : chaîne de caractère ← vide
I : entier ← 1
Construire (état_début)
Lire (fichier_trace, Tampon)
Tant que (Tampon ≠ Quitter)
    Si début_chaîne = interface alors
        Ecrire (/Ancien_Tampon)
        Ancien_Tampon ← Tampon
        Construire état_intermediaire
        (Tampon)
        Lire (Fichier, Tampon)
    Sinon
        Ecrire (i)
        Ecrire (Tampon)
        Incréments (i)
        Lire (Fichier, Tampon)
    FinSi
Fin tant que
Ecrire (/Ancien_Tampon)
Construire (état_final)
Fin

```

Figure 3.20. Algorithme de génération du modèle de dialogue à partir d'un fichier texte

3.3.1.5. Identification des problèmes

Cette phase comprend trois types d'évaluation successivement décrites (cf. Figure 3.8, A1.3)

- **L'évaluation statique** : elle consiste à évaluer l'interface dans un contexte bien défini. Elle est guidée par une base de règles définie par l'évaluateur. Pour chaque critère est associé un ensemble d'attributs d'évaluation et un ensemble de règles d'évaluation. En fait, l'évaluateur doit vérifier le respect ou non des ces règles. Par exemple, l'évaluation de l'adaptation du système aux plateformes interactives de type PDA peut se baser sur quelques règles (ex. le nombre maximum des caractères pour un champ de texte est huit, le nombre des valeurs possibles dans une liste des éléments défilants de type List Box est inférieure à six). Soulignons que ces règles sont proposées par l'évaluateur en se basant sur son expérience ou sur des connaissances issues de la littérature. Dans cette évaluation, le modèle de dialogue facilite à l'évaluateur la détection des attributs d'évaluation (les nombres des services traités par interface, les nombre des objets interactifs utilisés par service, etc.). La figure 3.21 présente le principe de l'évaluation statique via un exemple. A ce niveau, les données capturées (partie gauche) sont récupérées pour les analyser à l'aide des règles d'évaluation (partie du milieu). Ensuite, les résultats d'analyse sont affichés sous des formes compréhensibles pour le concepteur. Ces résultats d'analyse peuvent être interprétés par l'évaluateur pour critiquer le système (partie droite) et suggérer les améliorations nécessaires au concepteur. La figure 3.21 montre à l'aide d'une maquette un exemple possible des résultats d'analyses réalisées sur des données capturées pendant une expérimentation.

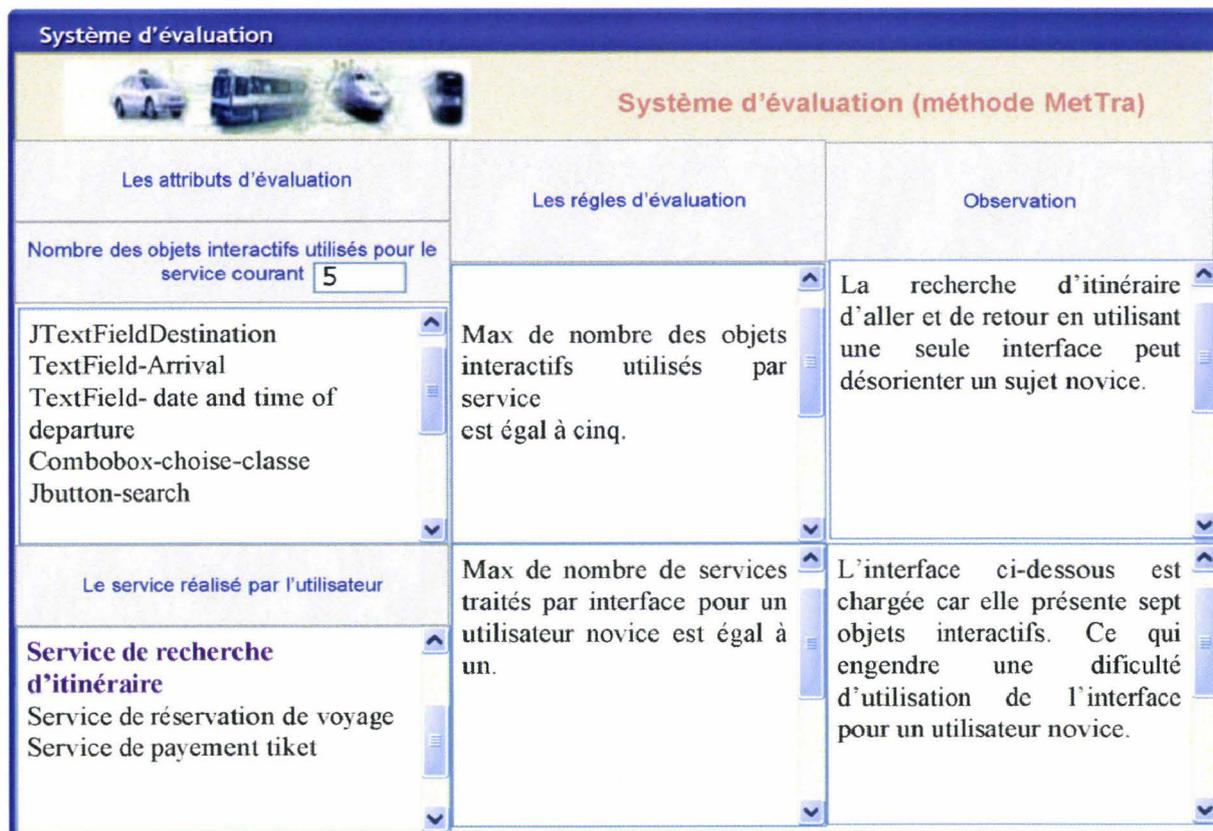


Figure 3.21. Une partie de la capture d'écran affichant le résultat d'analyse – exemple

- **L'évaluation multi-contextuelle** : consiste à vérifier la réaction des interfaces homme-machine suite aux changements de contexte. Rappelons que l'interface peut posséder plusieurs facettes de présentations utilisables dans différents contextes. Cette évaluation évalue la capacité d'adaptation dynamique de l'interface (changement du style d'affichage, changement de la couleur d'arrière plan, etc.) face au changement contextuel. Actuellement, dans notre travail, cette évaluation est basée sur une comparaison entre les adaptations d'une

même interface dans des contextes différents. A titre d'exemple, l'évaluateur peut comparer l'adaptation d'une même interface aux deux plateformes différentes (PC, PDA) en termes de services fournis par interface et objets interactifs opérationnels manipulés par l'utilisateur pour chaque service. Dans cette évaluation, les deux modèles de dialogue relatifs à l'utilisation de l'interface par l'utilisateur dans les deux contextes sont confrontés pour montrer la différence d'adaptation de l'interface dans les deux cas. Les modèles de dialogues générés sont utiles pour l'évaluateur. En effet, ils fournissent à l'évaluateur une vision de toutes les actions effectuées par l'utilisateur et les réactions de l'interface en situation réelle. Donc, l'évaluateur dispose d'une aide pour détecter des problèmes et inconvénients liées aux interactions et à l'interface. L'évaluateur peut effectuer la confrontation entre les modèles de dialogues générés (modèles de dialogue observés ou réels) relatifs aux différents contextes. Cette confrontation peut être effectuée pour chaque service.

Notons que :

- Chaque interface I_j propose un ou plusieurs services $S = \{s_1, s_2, \dots\}$.
- Chaque service s_i est associé à une interface I_j .
- Pour réaliser un service S_i , un ensemble d'objets interactifs O est utilisé, Avec $O = \{O_{ij}\}$; $i = \{1, \dots, n\}$, $j = \{1, \dots, m\}$.
- O_{ij} est associé à un service S_i et appartient à une interface I_j dans un contexte C_k .

- **L'évaluation dynamique (ou de contenu)** : elle consiste à évaluer la capacité d'apprentissage du système, elle concerne principalement l'évaluation de la personnalisation liée au contenu et s'effectue pendant l'exécution (les critères d'évaluation ne sont pas connus dès la conception : préférences des utilisateurs qui peuvent être détectés par un système d'apprentissage). Cette évaluation permet de savoir si la personnalisation proposée par le système est capable de prendre en compte les préférences de l'utilisateur telles que son niveau professionnel, ses domaines d'intérêts, etc. Par exemple, dans le domaine de transport, lorsqu'un utilisateur choisit un itinéraire parmi d'autres, il ne le fait pas au hasard. Son choix est basé sur un ou plusieurs critères de sélection. Nous faisons donc l'hypothèse selon laquelle un utilisateur choisit un itinéraire en fonction de plusieurs critères. Ainsi, l'idée de cette évaluation est de comparer les services fournis à l'utilisateur à ceux choisis réellement par celui-ci en se basant sur le coefficient de corrélation de Pearson et le calcul de taux d'erreur moyen du SIP. Dans le cadre de ce travail, on a besoin de mesurer le degré de la relation entre deux variables, la solution proposée par le système et la solution choisie réellement par l'utilisateur, on parlera alors de corrélation. En statistique, lorsque l'on parle de corrélation, on se réfère avant tout à un coefficient. Il s'agit d'un indice, qui décrit de façon précise la force de la relation entre deux variables. Il existe plusieurs de ces indices. Un de ceux les plus utilisés est le coefficient de Pearson (the Pearson product-moment correlation coefficient) [Paul et al., 94] [Upendra et Pattie 95].

Considérons une requête R , ainsi que l'ensemble des solutions S_i de cette requête ($i = \{1, \dots, n\}$, n étant le nombre de solutions possibles pour la requête R). Nous pouvons, en évaluant chacune des solutions entre elles, les trier par rapport aux différents critères. Ainsi, nous affectons un rang pour chaque solution vis-à-vis de ce critère. Nous pouvons définir une solution (ou un itinéraire) par un quadruplet de valeurs correspondant au rang de cette solution vis-à-vis des critères :

$$S_i = \{v_1, v_2, v_3, v_4\}$$

Où les v_i sont les rangs, par rapport aux solutions, de chacun des critères C_i , $v_i = \text{rang}(C_i)$, $1 < v_i < n$, où n est le nombre de solutions pour la requête considérée.

Considérons deux valeurs v_{ki} et v_{kj} d'un critère C_k de deux solutions respectives S_i et S_j :

$$v_{ki} < v_{kj} \Leftrightarrow S_i \text{ est plus performante que } S_j, \text{ pour le critère } C_k$$

En regardant l'exemple du tableau 3.1, nous pouvons dire, par exemple, que pour le critère C1 (durée du voyage), la solution 1 est plus performante que les autres solutions.

Tableau 3.1. Rangs des critères par rapport aux solutions d'une requête

	C1 : durée du voyage	C2 : Distance	C3 : prix	C4 : durée d'attente
Solution 1	1	2	4	1
Solution 2	4	4	2	4
Solution 3	2	3	1	2
Solution 4	3	1	3	3

Le système affectera des notes aux différents critères C_i par la formule :

$$e_i = \text{note}(C_i) = \frac{1}{\text{rang}(C_i)} * \delta$$

Afin d'évaluer les solutions proposées par le SIP en se basant sur le coefficient de corrélation de Pearson, on doit calculer d'abord pour chaque solution la moyenne des évaluations des solutions suivant tous les critères comme suit :

$$\bar{e}_j = \frac{1}{|I_i|} \sum_{j \in I_i} e_{i,j}$$

Ensuite, on doit calculer la corrélation ou la distance $w(a, b)$ entre la solution proposée par le système et la solution choisie réellement par l'utilisateur en appliquant la formule suivante :

$$w(a, b) = \frac{\sum_{j \in I_a \cap I_b} (e_{a,j} - \bar{e}_a) \cdot (e_{b,j} - \bar{e}_b)}{\sqrt{\sum_{j \in I_a \cap I_b} (e_{a,j} - \bar{e}_a)^2 \sum_{j \in I_a \cap I_b} (e_{b,j} - \bar{e}_b)^2}}$$

Si $w(a,b)$ est proche de la valeur 1, nous pouvons considérer que le système de personnalisation fournit des recommandation de bonne qualité. Cependant, Si l'on obtenait un résultat proche de 0 après plusieurs requêtes d'un utilisateur, alors on peut supposer que le système n'arrive pas à identifier et prendre en compte les préférences de celui-ci.

Concernant le taux d'erreur moyen, il s'agit de vérifier si les prédictions proposées correspondent effectivement à ce qui était prévu. Pour cela, il faut compter dans un premier temps le nombre de prédictions erronées pour chaque utilisateur. Puis, on calcule pour chaque utilisateur le taux d'erreur en appliquant la formule suivante :

$$TE(i) = \frac{\text{nbPrédictionsErronées}(i)}{\text{nbRequêtes}(i)}$$

Où $\text{nbPrédictionsErronées}(i)$ désigne le nombre de prédictions erronées pour l'utilisateur i et $\text{nbRequêtes}(i)$ correspond au nombre de requêtes passées par l'utilisateur i .

Par la suite, on calcule le taux d'erreur moyen TEM de tous les utilisateurs comme suit :

$$TEM = \frac{\sum TE(i)}{\text{nbUtilisateurs}}$$

Où nbUtilisateurs correspond au nombre d'utilisateurs qui ont participé à l'évaluation. La figure 3.22 illustre un exemple d'évolution de taux d'erreur moyen en fonction de nombres de requêtes d'un utilisateur.



Figure 3.22. Evolution de taux d'erreur moyen en fonction de nombres de requêtes

Notons que si TEM est faible (proche de 0), nous pouvons considérer que le système de personnalisation fournit des recommandations de bonne qualité.

Les problèmes détectés pendant la phase d'identification des problèmes sont ensuite catégorisés selon les critères d'évaluation proposés (cf.2.1.1). Ces problèmes sont ensuite modélisés sous forme de notes graphique UML [Roques et Vallée 00] qui portent le stéréotype « problème ». Ces notes sont destinées à expliquer textuellement les problèmes détectés (cf. Figure 3.23).

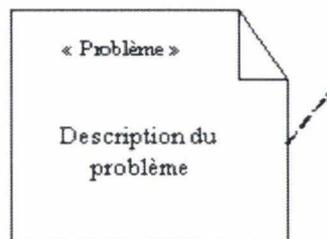


Figure 3.23. Une note graphique UML pour la modélisation d'un problème

3.3.1.6. Proposition des recommandations

A la suite de l'identification des problèmes, les évaluateurs proposent des solutions aux problèmes identifiés dans la phase précédente. Pour cela, les actions du modèle de dialogue de la tâche prescrite sont modifiées et d'autres événements sont rajoutés pour constituer un nouveau modèle de tâche. Ce modèle est représenté sous forme d'un modèle de dialogue. Une description textuelle complémentaire est également ajoutée pour expliquer chaque recommandation associée à un problème bien défini. Celle-ci est modélisée sous la forme d'une note graphique UML qui porte le stéréotype « recommandation ». Dans le cadre de notre travail les évaluateurs se basent sur leur expérience et leur intuition pour proposer ces recommandations (cf. Figure 3.24).

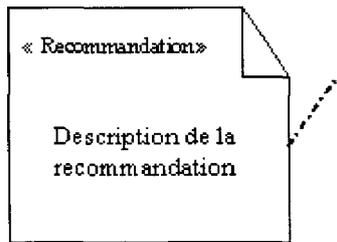


Figure 3.24. Une note graphique UML pour la modélisation d'une recommandation

Pour mieux structurer cette évaluation, l'évaluateur place dans un tableau le modèle de dialogue de la tâche prescrite et le modèle de dialogue intégrant les recommandations. Pour mieux comprendre la correspondance entre la note graphique UML qui décrit le problème identifié au niveau du modèle de dialogue de la tâche prescrite et celle qui explique la recommandation proposée au niveau du modèle de dialogue intégrant les recommandations, nous affectons aux deux notes le même caractère alphabétique. La figure 3.25 illustre un exemple d'une modélisation avec UML de la tâche prescrite et de la procédure proposée par l'évaluateur.

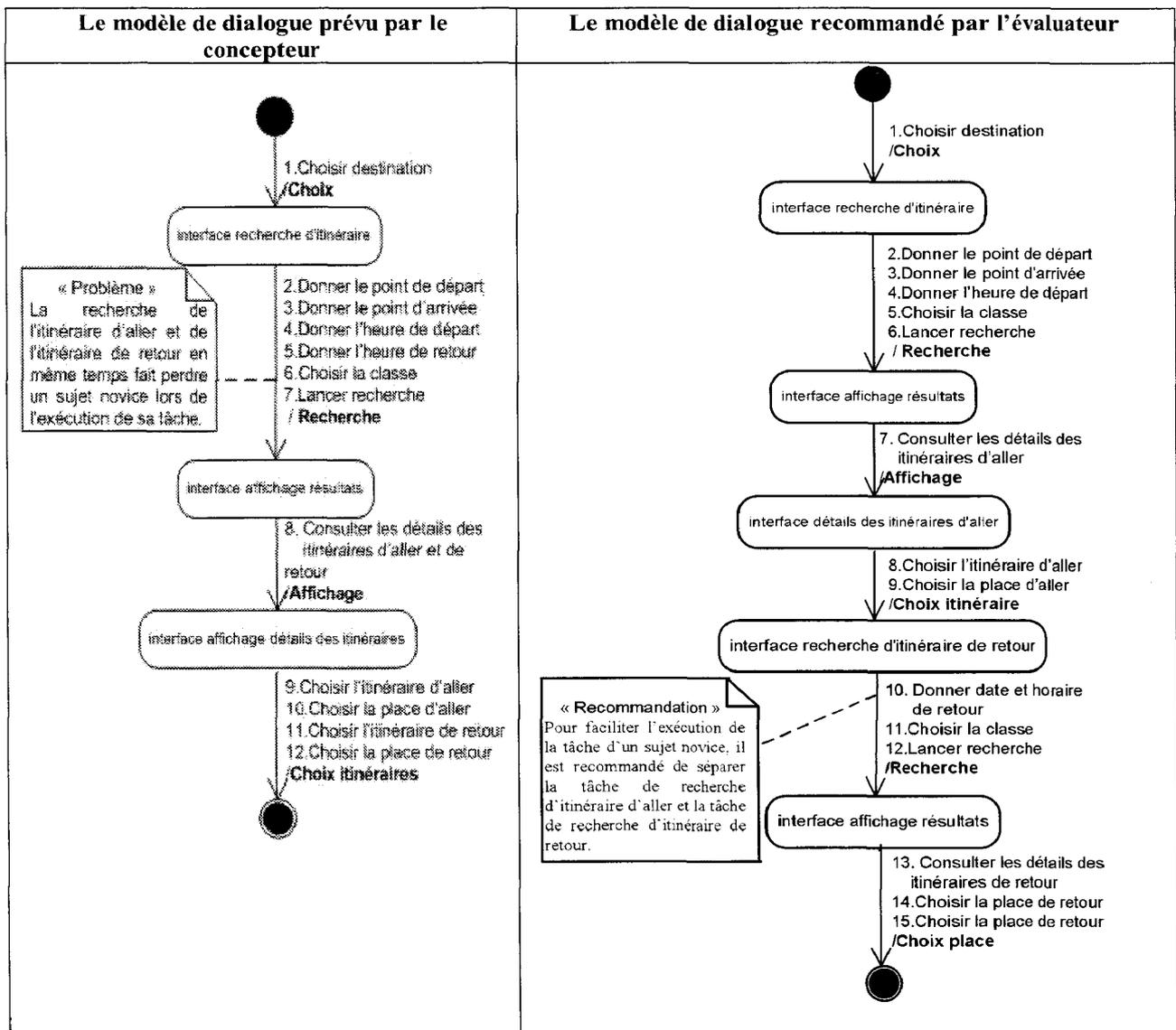


Figure 3.25. Modélisation avec UML de la tâche prescrite et la procédure proposée par l'évaluateur

3.3.2. Synthèse et discussion

MetTra présente dans l'état actuel de nos travaux de recherche de nombreux avantages. En premier lieu, elle possède l'avantage de la prise en compte de la réalité de l'utilisation du système dans le cadre de tâches (réelles ou représentatives), tout en impliquant l'utilisateur (réel ou représentatif). En effet, le système de trace produit des traces « orientées usage » qui fournissent rapidement des informations complètes et facilement exploitables lors des phases d'évaluation. MetTra est basée sur un ensemble de critères selon lesquels un SIP peut être évalué. Ces critères représentent un moyen pour aider l'évaluateur à identifier les problèmes rencontrés par les utilisateurs. Un autre avantage de notre approche est la représentation graphique des modèles de dialogue relatifs à des contextes différents en se basant sur le diagramme d'états de transitions étendu issu du langage UML. Cette modélisation permet une prise en main relativement facile (puisque une grande majorité de développeurs ont adopté UML), dans la mesure aussi où il existe de nombreux ateliers de génie logiciel dédiés.

3.4. La méthode MetSim

3.4.1. Principe et description

La troisième méthode proposée est appelée MetSim (METHode d'évaluation par SIMulation). Elle consiste à évaluer une IHM personnalisée en faisant varier des hypothèses d'usages liés à des contextes différents. L'intérêt des simulations est de pouvoir considérer différents scénarios en changeant le contexte d'usage (Plate-forme, Utilisateur, et environnement). Dans MetSim l'IHM est considérée comme un ensemble de composants de présentation. Nous partons de l'hypothèse que cette IHM est présentée en UsiXML (User Interface eXtensible Markup Language)³ [Limbourg et al., 04]. Grâce au langage à balises utilisé pour décrire l'abstraction de l'IHM, il est possible d'accéder à cette description et de la modifier pendant la conception, mais aussi à l'exécution. Notons que MetSim sera présentée également avec la technique IDEF0. MetSim repose sur trois phases : (1) Proposition des hypothèses (cf. Figure 3.26, A1), (2) Identification des problèmes (cf. Figure 3.26, A2), (3) Analyse des résultats (cf. Figure 3.26, A3). Chaque phase sera présentée par la suite avec ses activités détaillées.

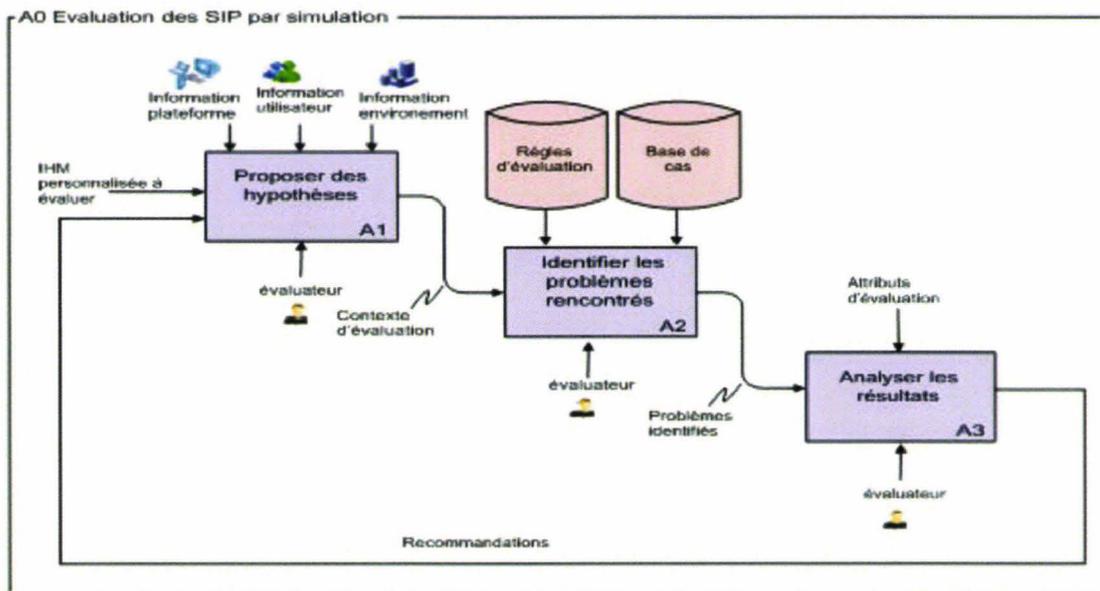


Figure 3.26. Les phases de la méthode MetSim

³ Une description complète de Usixml est disponible à : <http://www.usixml.org>

3.4.1.1. Proposition des hypothèses

La figure 3.27 détaille l'activité A1 « Proposer des hypothèses ». La proposition débute par la définition d'un contexte d'évaluation. Cette définition prend en entrée l'interface utilisateur à évaluer décrite en UsiXML.

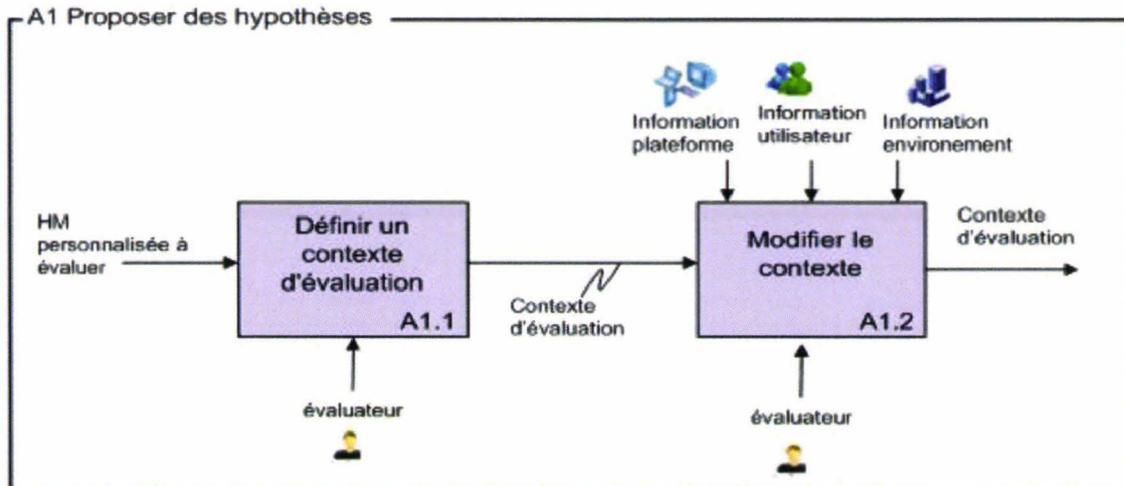


Figure 3.27. Proposition des hypothèses

L'activité A1.1 consiste à définir le contexte d'évaluation. Ainsi, selon le critère à vérifier, l'évaluateur doit choisir la catégorie des sujets qui seront impliqués dans l'évaluation (novices, experts, etc.), le support avec lequel les sujets interagissent avec le système pendant l'évaluation (PDA, PC, etc.) et l'environnement accueillant l'interaction. Ces choix sont définis en fonction des critères d'évaluation selon lesquels un SIP peut être évalué.

Ensuite, l'évaluateur du système peut modifier la balise <contextModel> qui comprend les contextes d'usage prévus selon le triplet <plate-forme, utilisateur et environnement>. Grâce au langage à balises utilisé pour décrire l'IHM, il est possible d'accéder à cette partie et de la modifier pendant la conception et à l'exécution.

3.4.1.2. Identification des problèmes

La phase A2 permet de détecter les problèmes engendrés suite au changement du contexte en se basant sur un système à base de cas. Le principe général du raisonnement à partir de cas (ou Case Based Reasoning : C.B.R.) consiste à traiter un nouveau cas (cas cible) en se remémorant les expériences passées voisines (cas de référence). Ce type de raisonnement repose sur l'hypothèse suivante : si une expérience passée et la nouvelle situation sont suffisamment similaires, alors tout ce qui peut être expliqué ou appliqué à l'expérience passée (base de cas) reste valide si on l'applique à la nouvelle situation qui représente le nouveau problème à résoudre [Bradley et al., 00] [Coyle et Cunningham 02] [Lajmi et al., 09]. Le raisonnement à base de cas se subdivise en plusieurs étapes reliées formant un cycle présenté à la figure 3.27.

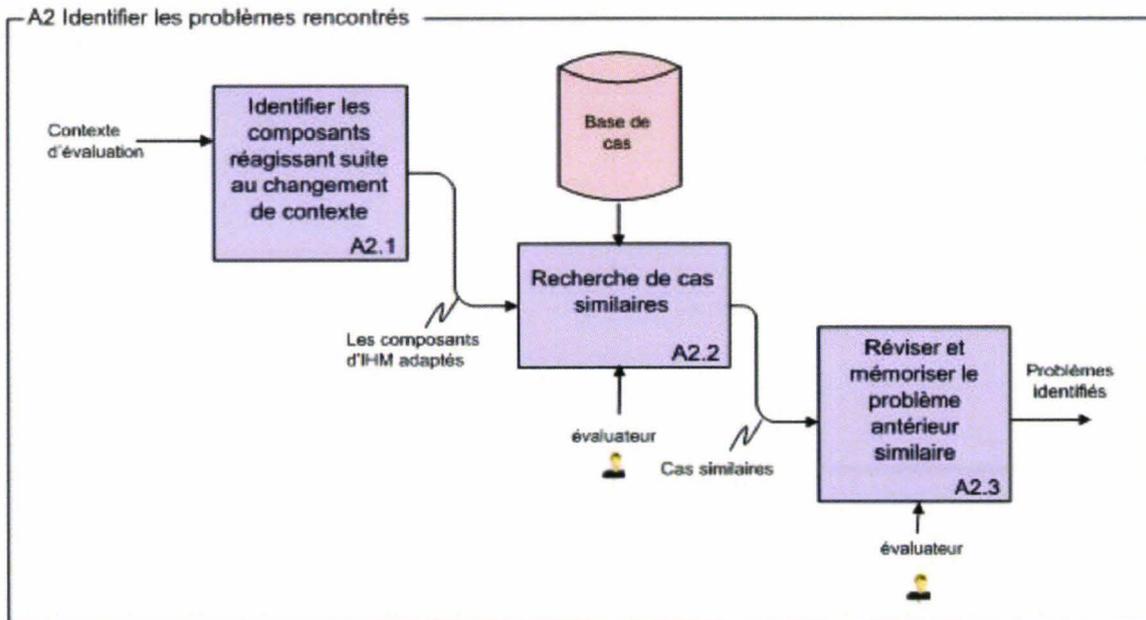


Figure 3.28. Identification des problèmes rencontrés

Les cas sont des expériences réelles qui se présentent sous la forme : triplet <R, P, J> où :

- R est la description de la réaction de l'interface utilisateur
- P est la description du problème
- J est la justification du problème

Pour être utilisable, une base de cas doit contenir un certain nombre de cas. Une base de cas vide ne permet aucun raisonnement. Il est ainsi important d'initialiser la base de cas avec des cas pertinents. Dans notre travail, la base de cas est initialisée en se basant sur un ensemble de règles d'évaluation. Ces règles sont définies initialement par l'évaluateur. En effet, à chaque critère d'évaluation (par exemple lié à l'environnement) est associé un ensemble d'attributs d'évaluation (bruit, luminosité, température). A chaque attribut est associé un ensemble de règles d'évaluation. Le but de ces règles est de proposer et décrire les réactions d'interface attendues par l'utilisateur suite à un changement contextuel. Ainsi, l'évaluateur vérifie le respect ou non de ces règles. Dans notre méthode, la règle d'évaluation prend la forme «Action→Réaction» : une action indique un changement qui apparaît dans le cadre de l'utilisation (un état du contexte d'usage à considérer par l'IHM), une réaction est réalisée en conséquence de cet état. La figure 3.29 présente un exemple d'une règle d'évaluation décrite en XML. Cette règle présente la réaction prévue de l'interface (fond d'écran mis en gris) lorsque la luminosité augmente à 80%.

```

<rule> // règle d'évaluation (Luminosité atteint plus de 80)
<id>R1</id> // Num de règle
<action>
<luminosity> // Luminosité est arrivée à plus de 80
<morethan>80</morethan>
</luminosity>
</action>
<reaction> // Fond de l'IHM graphique : mis en gris
<background>
<color> clGray </color>
</background>
</reaction>
</rule>
    
```

Figure 3.29. Exemple d'une règle d'évaluation décrit en XML

L'activité A2.1 représente l'étape d'identification des composants suite au processus d'adaptation. Rappelons ici que le composant d'IHM peut posséder plusieurs facettes de présentation utilisables dans différents contextes. En cas de changement au niveau du contexte, une adaptation est lancée. Ainsi, les composants de présentation adéquats au contexte d'usage sont arrangés dans des conteneurs et mis en forme sur l'IHM. L'identification des composants de présentation et les conteneurs qui les arrangent s'appuie sur la balise `<cuiModel>` qui est composée de conteneurs concrets `<Concretecontainer>` qui sont attachés aux tâches principales (par exemple, un conteneur concret est défini pour la tâche de *recherche d'itinéraire préféré*). Les conteneurs peuvent inclure des composants concrets associés aux sous-tâches (par exemple, un composant concret est défini pour la sous-tâche *saisie du point de départ*). Dans cette étape, les nouvelles valeurs des caractéristiques des conteneurs et des composants réagissant (telle que la modification de la couleur du fond d'IHM, le changement de la taille d'une fenêtre, etc.) et la réorganisation des facettes de présentation sur l'IHM sont détectées. L'identification des conteneurs et composants réagissant suite au changement contextuel et leurs caractéristiques se fait en comparant la spécification concrète de l'interface avant et après l'adaptation. Chaque conteneur ou composant adapté sera décrit en UsiXML (cf. Figure 3.30).

```

<comment>Generated by GrafXML 1.2.0 build id : 200612141408</comment>
<comment>WARNING : AUI Model save is a work in progress. Use it at your own risk</com
</head>
<auiModel id="projet1-aui_30" name="projet1-aui"/>
<cuiModel id="projet1-cui_30" name="projet1-cui">
  <window id="window_component_0" name="window_component_0"
    width="200" height="150">
    <gridBagBox id="grid_bag_box_1" name="grid_bag_box_1"
      gridHeight="17" gridWidth="20">
      <constraint gridx="10" gridy="2" gridwidth="6"
        gridheight="1" weightx="1.0" weighty="1.0"
        fill="both" insets="0,0,0,0">
        <inputText id="input_text_component_2"
          name="input_text_component_2" isVisible="true"
          isEnabled="true" textColor="#000000"
          maxLength="50" numberOfColumns="15" isEditable="true"/>
      </constraint>
      <constraint gridx="2" gridy="2" gridwidth="4"
        gridheight="1" weightx="1.0" weighty="1.0"
        fill="both" insets="0,0,0,0">
        <outputText id="output_text_component_4"
          name="output_text_component_4"
          content="/uiModel/resourceModel/cioRef[@cioId='output_text_component
          defaultContent="point de départ"
          isVisible="true" isEnabled="true"
          fgColor="#000000" isBold="true" textColor="#000000"/>
      </constraint>
    </gridBagBox>

```

Figure 3.30. Exemple d'un conteneur décrit en UsiXML

Ensuite, le système à base de cas utilise ces caractéristiques extraites afin de comparer les réactions d'IHM du nouveau cas à celles des cas antérieurs. Pour cela, une recherche (retrieve) est effectuée dans une base de cas (Case-Base) afin de trouver les cas précédents les plus susceptibles de satisfaire l'objectif de raisonnement, à savoir résoudre le cas présent (cf. Figure 3.28, A2.2).

Afin de pouvoir comparer les cas entre eux, il doit être possible d'extraire des caractéristiques du contenu des cas. [Kolodner 93] définit une caractéristique ou un descripteur par une paire (attribut, valeur) associée à un cas. Dans cette même optique, pour notre approche, chaque cas est décrit par un ensemble de caractéristiques de l'interface (par exemple, la taille des caractères) et un ensemble d'attributs de contexte d'usage.

L'activité A2.2 représente l'étape de recherche du cas utile (pertinent). C'est à ce niveau que le cas le plus pertinent est déterminé en utilisant les méthodes de calcul de similarité afin de guider la recherche. L'algorithme général de recherche se passe en plusieurs étapes.

1. Un préfiltrage de la base de cas fournit un ensemble B de cas potentiellement similaires au nouveau cas. Ce processus de préfiltrage se base sur les attributs de contexte d'usage communs du nouveau cas et le cas antérieur. L'ensemble résultat doit être relativement petit tout en contenant le cas le plus similaire.

2. Idéalement en parallèle, pour chaque cas $C \in B$,

(a) Déterminer les caractéristiques de l'interface communes de réaction de l'interface du nouveau cas RN et celle du cas antérieur RA. Soit A l'ensemble de ces attributs.

(b) Calculer, pour chaque attribut $a \in A$, une fonction de similarité dimensionnelle $D_a(RN_a, RA_a)$, où RN_a représente la valeur de l'attribut a dans RN et RA_a , la valeur dans RA.

$$D_a(RN_a, RA_a) = \frac{RN_a - RA_a}{DM} \in [0..1]$$

DM est la différence entre la borne haute et la borne basse du domaine de définition de l'attribut

(c) Déterminer le degré de similarité global $S(RN, RA)$

$$S(RN, RA) = \sum_{a=1}^A D_a(RN_a, RA_a)$$

3. Classer les degrés de similarité et retourner le ou les cas les plus similaires à Q.

Le problème est extrait du cas le plus pertinent à des fins de réutilisation (reuse) ; elle devient un problème proposé. Elle subit un processus de révision (revise) et d'adaptation afin de devenir compatible avec le cas actuel (cf. Figure 3.28, A2.3). Le résultat de ce processus engendre un nouveau cas contenant la réaction actuelle de l'interface et le problème confirmé qui peut alors être mémorisé dans la base de cas pour usage futur (retain). Les évaluateurs se basent sur leur expérience et leur intuition pour réviser les problèmes.

3.4.1.3. Analyse des résultats

Cette phase décrit les problèmes détectés concernant la prise en compte des facteurs du contexte et les recommandations qui en résultent (cf. Figure 3.31). Notons qu'à chaque critère d'évaluation est associé un ensemble d'attributs d'évaluation. Par exemple, pour vérifier l'adaptation du système au facteur luminosité, l'évaluateur vérifie les réactions d'interface si la luminosité ambiante baisse ou augmente.

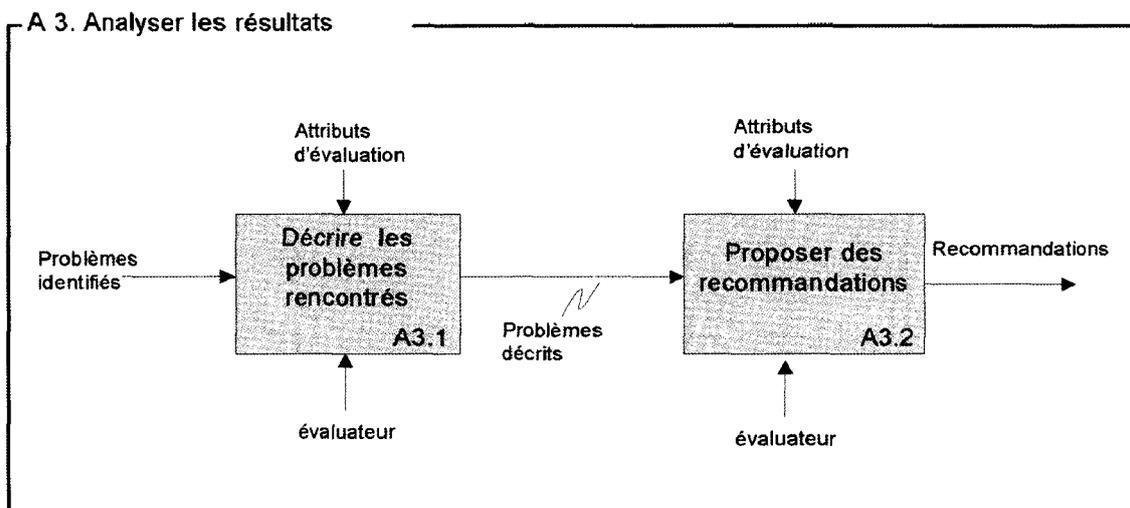


Figure 3.31. Analyse des résultats

Les résultats peuvent être communiqués sous forme de fiche problème : à chaque fois que l'évaluateur détecte un problème, celui-ci est intégré dans une fiche de type tableau et basé sur le texte. Le tableau 3.2 montre le formalisme typique pour la description de problèmes, utilisé dans notre méthode. Le tableau comprend une colonne « contexte d'usage » indiquant un changement qui apparaît au niveau de l'attribut de contexte et correspondant au problème détecté, une colonne « description du problème » contenant une explication textuelle du problème et une colonne « conséquence » indiquant les risques éventuels du problème.

Tableau 3.2. Formalisme utilisé pour décrire un problème

Contexte d'usage	Description du problème	Conséquence
Intitulé de l'action de contexte d'usage.	Description textuelle du problème	Description textuelle de la conséquence

Les résultats obtenus autorisent la réalisation de statistiques concernant par exemple le taux de réussite de présentation d'interface personnalisée (adaptation adéquate), les problèmes les plus fréquemment rencontrés, etc.

Des fiches contenant des descriptions textuelles des recommandations seront préparées par l'évaluateur. Chaque fiche est présentée sous forme de tableau. Le tableau comprend une colonne « action » indiquant un état du contexte d'usage à considérer par l'IHM (par exemple utilisation d'un PDA) et une colonne « description des recommandations » proposées par l'évaluateur (cf. Tableau 3.3).

Tableau 3.3. Formalisme utilisé pour la description de recommandations

Action	Description des recommandations
Intitulé de l'action	Description textuelle des recommandations

Ces descriptions concrètes des recommandations doivent apporter des solutions possibles claires, aidant au mieux les développeurs pour la conception ou la ré-ingénierie des interfaces homme-machine [Hornbaek et Frokjaer 05], plutôt qu'une description textuelle vague qui n'apporte pas vraiment de piste pour les concepteurs.

3.4.2. Synthèse et discussion

L'intérêt des simulations est de pouvoir considérer rapidement différents scénarios en introduisant des événements à des instants donnés et en faisant varier des hypothèses d'usage. Il est par exemple intéressant de simuler l'appropriation de l'interface par l'utilisateur, de tester de nombreuses hypothèses fonctionnelles sur cette interface dans des contextes différents et de procéder à des analyses statistiques. Cette méthode ne s'appuyant pas sur la réalisation d'un prototype fonctionnel de l'IHM, elle peut être utilisée pour effectuer des tests prospectifs, étudier de nouveaux modes d'interaction induits par une future technologie et ainsi être intégrée au sein du cycle de conception d'un SIP.

Conclusion

Ce chapitre a décrit notre contribution pour l'évaluation de la personnalisation de l'interaction homme-machine. Nous avons proposé trois méthodes pour l'évaluation des systèmes d'information personnalisés. Les méthodes proposées sont basées sur les critères d'évaluation des SIP que nous avons présentés dans le chapitre 2. Nous avons en premier lieu proposé une approche d'évaluation *a posteriori* centrée sur l'utilisateur en s'appuyant sur des documents structurés (questionnaire, fiche de problème, etc.). En second lieu, nous avons présenté une

méthode d'évaluation qualifiée de précoce basée sur les traces d'usage. Ensuite, nous avons présenté une approche d'évaluation par simulation. Nous avons présenté les différentes étapes de chaque méthode avec les activités détaillées au travers du formalisme IDEF0. Le tableau 3.4 synthétise les trois méthodes d'évaluation proposées selon le type de personnalisation à évaluer (contenu et/ou contenant) et les intervenants de l'évaluation (utilisateur, concepteur, évaluateur).

Tableau 3.4. Etude comparative des méthodes d'évaluation proposées

Méthodes d'évaluation	Le type de personnalisation à évaluer	Les intervenants de l'évaluation
MetQus	Contenu, Contenant	Utilisateur, Evalueur
MetTra	Contenant	Utilisateur, Concepteur, Evalueur
MetSim	Contenu, Contenant	Concepteur, Evalueur

Dans le chapitre suivant, nous allons présenter les résultats d'applications de trois méthodes d'évaluation proposées : une première étude de cas concerne l'évaluation de démonstrateur MoverPerso en se basant sur MetQus. Nous nous sommes focalisés sur une tâche représentative de l'application : *la recherche d'itinéraire après l'ajout d'un rendez-vous.* ; une seconde étude de cas montre comment, à l'aide de la méthode MetTra, il est possible d'évaluer un SIP de manière semi-automatique. Puis, nous terminerons en présentant quelques exemples d'évaluation permettant d'étudier notre méthode MetSim. Cette méthode a été appliquée pour évaluer un système permettant d'afficher les prochains départs dans une gare ou une station, accessibles à travers une interface web.



Chapitre 4

Mise en œuvre des méthodes d'évaluation proposées dans le domaine du transport

Sommaire

Introduction	100
4.1. Le projet MOUVER.PERSO	100
4.2. Contextes d'usage envisagés dans le cadre de l'évaluation des systèmes d'information personnalisés	101
4.2.1. La prise en compte de l'utilisateur	101
4.2.2. La prise en compte de l'environnement.	102
4.2.3. La prise en compte de la plate-forme.	103
4.3. Premier cas d'étude : application de la méthode MetQus	104
4.3.1. Présentation de l'application	104
4.3.2. Mise en oeuvre de MetQus.....	105
4.3.3. Synthèse et discussion.....	112
4.4. Second cas d'étude : application de la méthode MetTra	113
4.4.1. Présentation de l'application	113
4.4.2. Mise en oeuvre de MetTra.....	114
4.4.3. Synthèse et discussion.....	128
4.5. Troisième cas d'étude : application de la méthode MetSim	129
4.5.1. Présentation de l'application	129
4.5.2. Mise en oeuvre de MetSim.....	129
4.5.3. Synthèse et discussion.....	134
Conclusion	134

Introduction

Nous avons proposé dans le chapitre précédent trois démarches d'évaluation des SIP. Nous allons dans ce chapitre proposer une première validation des démarches proposés en les appliquant sur trois études de cas de système d'information personnalisé. Étant donné les travaux menés par notre laboratoire dans le domaine de la personnalisation et de la diffusion des informations destinées aux voyageurs dans le domaine des transports, c'est très naturellement que nous nous avons décidé d'utiliser cette expérience acquise au fil des années pour servir de support à nos travaux de recherche.

Dans ce chapitre, nous allons présenter les résultats issus des applications des trois méthodes d'évaluation proposées. Mais, avant d'étudier ces différents exemples, nous allons commencer par présenter le projet MOUVER.PERSO que nous avons eu la chance d'intégrer pendant la durée de notre thèse, et auquel nous avons contribué ; ce projet nous a fourni de nombreux éléments de réflexion⁴.

Puis nous présenterons un premier cas d'étude en suivant les étapes de la première méthode proposée dans le cadre de notre travail. L'objectif de cette expérience était d'évaluer le démonstrateur MOUVER.PERSO en se basant sur la démarche proposée ainsi que sur les critères que nous avons définis pour l'évaluation d'un SIP. Nous nous sommes focalisés sur une tâche représentative de l'application : la recherche d'itinéraire après l'ajout d'un rendez-vous.

Ensuite nous étudierons à travers un deuxième cas d'étude, comment, à l'aide d'une méthode d'évaluation, qualifiée de précoce, (c'est-à-dire utilisée lors des premiers étapes d'un cycle de développement logiciel), basée sur les traces d'usage, il est possible d'évaluer un SIP de manière semi-automatique.

Puis, nous terminerons en présentant quelques exemples d'évaluation permettant d'étudier la méthode MetSim proposée dans le chapitre précédent. Cette méthode a été appliquée pour évaluer un système permettant d'afficher les prochains départs dans une gare ou une station, accessibles à travers une interface web.

L'illustration des méthodes proposées sur des cas d'étude différents permettra de distinguer les points forts et les points faibles de ces méthodes. Chacun des trois exemples a ses propres caractéristiques, ses contextes d'usage spécifiques, et sa fréquence de changement de contexte.

Les parties suivantes décrivent tour à tour la mise en pratique des trois méthodes proposées. Mais commençons par présenter le projet qui a servi de support à nos travaux.

4.1. Le projet MOUVER.PERSO

Le projet intitulé MOUVER.PERSO correspond à la deuxième phase du projet MOUVER (Mobilité et Multimodalité Voyageurs Etudiants en Région Nord-Pas de Calais). La première

⁴ Le projet ANR VIATIC.MOBILITE portant également sur l'information voyageur a constitué aussi pour nous une source de réflexion importante. Ce projet coordonné par l'INRETS-ESTAS (G.USTER) a impliqué 5 laboratoires de recherche et 10 entreprises.

phase, financée par la Prédim – Direction des Transports Terrestres en 2003, a permis de mettre en avant des pistes de travail pour l'amélioration de l'information de mobilité à destination d'une population de jeunes et d'étudiants.

Cette deuxième phase est consacrée à la mise en oeuvre concrète des services et à leur évaluation. La personnalisation de l'information en est l'axe central.

Parmi les pistes de travail proposées dans le domaine de l'information multimodale, nous nous intéressons ici particulièrement à « l'articulation entre l'individuel et le collectif », à savoir l'adaptation de l'information fournie aux besoins spécifiques de l'utilisateur. Lors de cette seconde phase du projet MOUVER, nous souhaitons mettre en oeuvre et évaluer des services liés à la personnalisation des informations. Ces services seraient centrés sur la mobilité au quotidien, dans un but d'accompagnement du jeune voyageur (de type étudiant) dans son « univers » et ses activités [Uster, 01].

L'objectif est de faciliter l'accès à l'information multi-modes de transport d'un usager désireux effectuer un déplacement. En effet, l'utilisateur se trouve confronté à un ensemble de sources d'informations disparates (horaires et tarifs délivrés par les différents exploitants, cartes, etc, sur papier, borne interactive ou par Internet) qui s'avèrent parfois difficiles à intégrer en un plan de déplacement précis et unique, adapté à ses besoins et préférences. Comme souligné dans le chapitre 1, la personnalisation s'avère une approche pertinente pour relever ce défi. Dans le contexte de l'information aux usagers intégrant plusieurs modes de transports et leurs connexions, notre objectif est, d'une part, d'aider l'utilisateur dans sa démarche de recherche d'informations et, d'autre part, de lui fournir un résultat personnalisé, c'est-à-dire toute l'information nécessaire et uniquement l'information nécessaire en fonction de son destinataire [Petit-Rozé et al., 04].

Plus précisément, notre objectif est d'évaluer le démonstrateur « MOUVER.PERSO », développé au sein de notre laboratoire et permettant de [Grislin et al., 07] :

- assurer une passerelle entre le monde de la mobilité et la société de l'information ;
- extraire des gisements de données transports de l'information pertinente pour les usagers ;
- organiser la mobilité quotidienne en fonction de l'agenda et des offres de transport afin de gagner du temps et mieux organiser le déplacement de l'étudiant ;
- présenter l'information de manière immédiate et naturelle. Seront distinguées l'information statique et l'information temps réel ;
- diffuser l'information sur plusieurs supports, tout en assurant une présentation ergonomique et un contenu homogène.

4.2. Contextes d'usage envisagés dans le cadre de l'évaluation des systèmes d'information personnalisés

Rappelons que les SIP doivent avoir la capacité de s'adapter à leur contexte d'usage en découvrant et utilisant des informations contextuelles, telles que la localisation de l'utilisateur, la date et l'heure, la proximité d'autres utilisateurs, les ressources disponibles, les caractéristiques de la plate-forme cible, les conditions ambiantes, etc. Ainsi, différents contextes d'usage (P : plate-forme, U : utilisateur, et E : environnement) peuvent être envisagés pour le SIP ; ces contextes doivent être considérés afin d'évaluer ces systèmes.

4.2.1. La prise en compte de l'utilisateur

Nous présentons dans cette partie, les différentes facettes du contexte relatif à l'utilisateur. Pour pouvoir, dans une application, fournir des informations et des services pertinents à

l'utilisateur, le système d'aide à la mobilité des voyageurs doit pouvoir prendre en compte la profession de l'utilisateur, son activité courante (loisir, travail), ses préférences et habitudes, etc.

Pour donner un exemple, les voyageurs pouvant provenir de différents pays et de différentes origines, les applications doivent être capables de s'adapter à la langue et à la culture de l'utilisateur. Ainsi, on pourrait imaginer une application qui adapte automatiquement sa charte graphique en fonction de l'origine géographique et culturelle de l'utilisateur.

Les applications doivent aussi être capables de s'adapter aux handicaps des usagers. Par exemple, au niveau des IHM, pour une personne possédant des problèmes de vue, les caractères affichés à l'écran pourraient être de dimension adaptée à ceux-ci.

En plus, le SIP doit respecter les préférences de l'utilisateur en ce qui concerne les couleurs (fond d'écran, texte, etc.) et les modes d'interaction préférés de l'utilisateur si plusieurs outils d'interaction sont disponibles sur la plate-forme cible (clavier, souris, joystick, etc.).

Tableau 4.1. Facettes du contexte relatif à l'utilisateur – exemples représentatifs

Age	Jeune, Adulte,...
Langue	Anglais, Français, Arabe,...
Expérience en informatique	Novice, moyen, avancé
Activité, Profession	Etudiant, Enseignant, ingénieur,...
But du voyage	Shopping, Business,...
Mode d'interaction préféré	Vocal, Tactile, Clavier,...
Couleurs préférées	Noir/blanc, Bleu/blanc...
Handicap	Aucun, Visuel, Auditif...

4.2.2. La prise en compte de l'environnement

Les caractéristiques de l'environnement sont variées et peuvent changer relativement fréquemment. L'IHM doit prendre en compte une gamme large de facettes liées au contexte relatif à l'environnement. Dans le domaine du transport, les voyageurs peuvent accéder à l'information tout en étant placés dans des environnements différents, par exemple :

- L'utilisation du système peut se faire dans un environnement bruyant (ex. dans la rue ou dans la gare) ou calme (ex. café peu fréquenté, dans le train, musée). Dans un environnement bruyant, il sera éventuellement nécessaire d'utiliser un niveau sonore élevé couplé à des alertes vibrantes. En revanche, dans un lieu calme, l'IHM devrait arrêter l'utilisation de messages sonores, et n'utiliser que des alertes vibrantes.

- La vitesse de déplacement de l'utilisateur peut aussi nécessiter une adaptation de l'interface pour en faciliter la lecture ; un exemple typique est lorsque l'utilisateur court, par exemple, lorsqu'il est en retard, et ceci afin d'attraper sa correspondance.

- En outre, parfois la luminosité augmente ; il peut donc s'avérer nécessaire d'éclairer l'arrière-plan et de foncer les textes),

- Le climat peut aussi jouer un rôle essentiel dans le système de guidage dans le domaine de transport ; par exemple, selon la situation de climat, le système peut offrir aux voyageurs des conseils concernant leur itinéraire. En plus, le choix d'un itinéraire par un utilisateur pourrait

être motivé par rapport aux conditions climatiques, par exemple. Un utilisateur pourrait préférer le vélo lorsqu'il fait beau et choisir le bus lorsqu'il pleut ou lorsqu'il fait très froid.

Tableau 4.2. Facettes du contexte relatif à l'environnement – exemples représentatifs

Lieu	Gare, Train,...
Luminosité	Aucune, Moyenne, élevée, ...
Bruit	Calme, Bruyant,...
Etat	Pressé, Normal,...
Température	Valeur numérique (ex. 37°)
Heure	10:00,13:00,...
Climat	sec, pluvieux, chaud...

4.2.3. La prise en compte de la plateforme

Le système peut être utilisé sur des supports d'interaction type (PC, PDA, Smartphone...), et même de différentes générations. Chaque plate-forme a ses propres caractéristiques. Le processus d'adaptation doit prendre en compte les différentes caractéristiques de la plate-forme cible.

Dans le domaine du transport, afin de faciliter et d'encourager les déplacements, les usagers doivent pouvoir accéder à l'information et aux applications avec tout type de plateformes de consultation (téléphone, PDA, ordinateur portable, etc.), par exemple :

- La taille de l'écran est un facteur essentiel pour l'interface graphique ; la taille de l'écran est variable selon la génération du terminal : 220*120 ou 320*240 pixels. L'espace disponible doit être respecté aussi, dans la mesure du possible (l'IHM ne doit pas dépasser l'espace disponible) et exploité (on doit procéder à un arrangement des composants de présentation dans cet espace).
- De plus, l'IHM doit respecter le style propre du support cible, prenant en compte ses caractéristiques. Le style inclut la structure de l'écran, les couleurs, les polices de texte acceptées, la taille des icônes, etc.
- L'IHM doit être aussi capable de s'adapter au mode d'interaction disponible (clavier classique, clavier avec joystick, écran tactile, etc.). Par exemple, de nombreux terminaux comportent un clavier physique ; d'autres utilisent seulement un clavier virtuel.
- En outre, la présentation de l'IHM peut être variée en fonction du système d'exploitation de la machine (par exemple, Linux et Windows peuvent être installés sur un PC, mais ils n'ont pas le même style d'affichage). Cela permet au processus d'adaptation de choisir et de configurer les composants de présentation en cohérence avec le guide de style.

Tableau 4.3. Facettes du contexte relatif à la plate-forme – exemples représentatifs

Type	PC, PDA,...
Espace d'affichage	220*300, 640*480, ...
Modes d'interaction disponibles	Ecran tactile, clavier,...
Style	Menu, fenêtre,...
Couleurs	5075, 3967,...
Système d'exploitation	Linux, Windows mobile, ...
Mode d'affichage	Horizontal, Vertical,...

4.3. Premier cas d'étude : application de la méthode MetQus

Nous allons dans cette partie mettre en application la méthode MetQus (METHode d'évaluation basée sur le QUestionnaire de Satisfaction), proposée dans le troisième chapitre, sur un système de gestion des déplacements entre rendez-vous d'un étudiant.

4.3.1. Présentation de l'application

Le système a pour mission d'inciter à l'utilisation des transports en commun ou des modes dits doux en présentant des solutions de déplacement personnalisées. La sensibilisation à la mobilité durable sera omniprésente. Personnaliser les informations consiste à présenter certaines informations à bon escient, au bon moment, de façon ciblée pour l'utilisateur. Il s'agit de prendre en compte ses contraintes, besoins et préférences dès lors qu'un choix peut être fait parmi un ensemble de possibilités de modes de transports et d'horaires. Cela consiste alors à mettre en œuvre des services d'accompagnement personnalisés autour de la mobilité quotidienne des étudiants (cf. Figure 4.1).

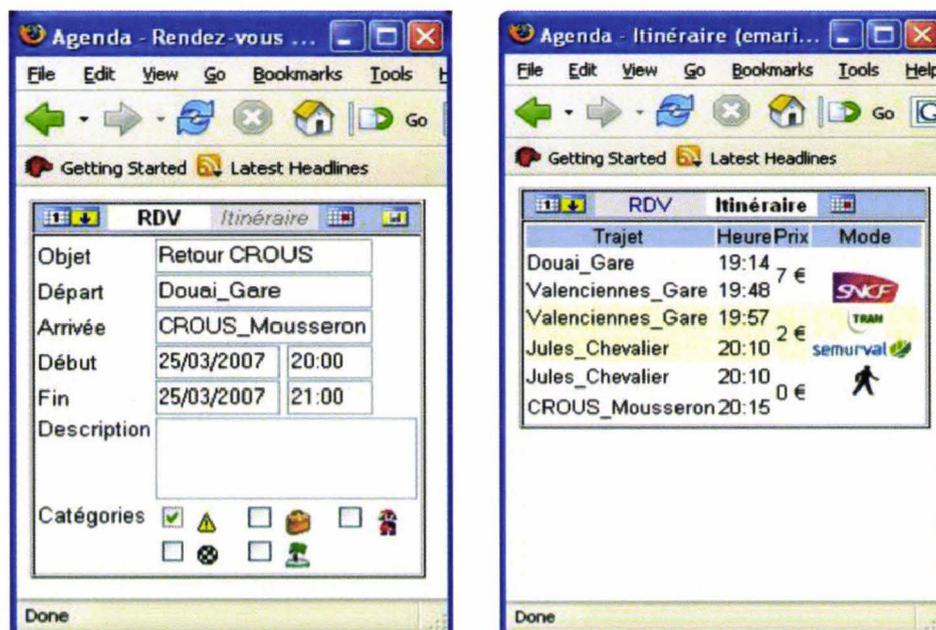


Figure 4.1. Exemple des interfaces de l'application à évaluer

Les étudiants disposent d'un emploi du temps initialement "statique" (mais qui évolue au cours du temps). Autour de cet emploi du temps et des activités sous-jacentes (ex : révisions), ils occupent leur temps libre de différentes manières : sports et loisirs, shopping, culture, rencontres familiales...

Il s'agit, selon une approche originale est d'accompagner les étudiants tout au long de leurs occupations de manière personnalisée :

- en termes de déplacements (moyen de transport préféré, coût du déplacement, etc.)
- en termes d'organisation des journées (par rapport aux préférences des étudiants et en fonction de l'offre de transport).

L'emploi du temps des étudiants enregistrés via leurs agendas devient alors "dynamique". Un agenda personnel peut alors être défini comme étant un outil personnalisé de gestion du temps

d'un utilisateur donné. Il se présente comme tout agenda, mais sa particularité réside dans ses fonctionnalités. Ainsi il doit permettre à l'utilisateur de :

1. visualiser le contenu de son agenda de manière journalière, hebdomadaire.
2. ajouter ou supprimer des rendez-vous via une interface homme-machine de saisie.

Mais, un agenda personnel, tel qu'il est envisagé dans le projet MoverPerso, doit posséder des fonctionnalités supplémentaires :

3. organiser (ou ré-organiser) dynamiquement la planification horaire des rendez-vous en fonction de l'utilisateur ;
4. gérer les déplacements entre les différents rendez-vous ;
5. être accessible à distance ;
6. utiliser éventuellement des données de géolocalisation ;
7. s'interfacer éventuellement avec d'autres agendas.

Parmi ces points d'étude, nous nous sommes principalement intéressés à la gestion des déplacements entre rendez-vous, ce qui constitue l'originalité des travaux dans le cadre de ce projet. Ce système a été développé au LAMIH dans le cadre de travaux sur la conception de systèmes à base d'agents pour la personnalisation des informations [Anli et al., 05 a][Anli et al., 05 b][Petit-Rozé et al., 04b].

La partie suivante décrit les différentes étapes fournies par la méthode proposée dans le but d'évaluer ce système de gestion des déplacements entre rendez-vous pour un étudiant.

4.3.2. Mise en oeuvre de MetQus

Nous détaillons maintenant la mise en pratique de la première méthode que nous proposons, intitulée MetQus (METHode d'évaluation basée sur le QUEStionnaire de Satisfaction), dédiée pour les systèmes d'information personnalisés.

4.3.2.1. Préparation de l'évaluation (Protocole de test)

L'objectif de cette expérience était d'évaluer le démonstrateur MoverPerso en se basant sur la méthode MetQus ainsi que sur les critères que nous avons définis pour l'évaluation d'un SIP.

- **Tâche évaluée** : nous nous sommes focalisés sur une tâche représentative de l'application : la recherche d'itinéraire après l'ajout d'un rendez-vous. Cette tâche présente la particularité de nécessiter la consultation d'au minimum trois pages-écrans de l'application. La dynamique de cette tâche est représentée par le diagramme d'états-transitions visible en figure 4.2.

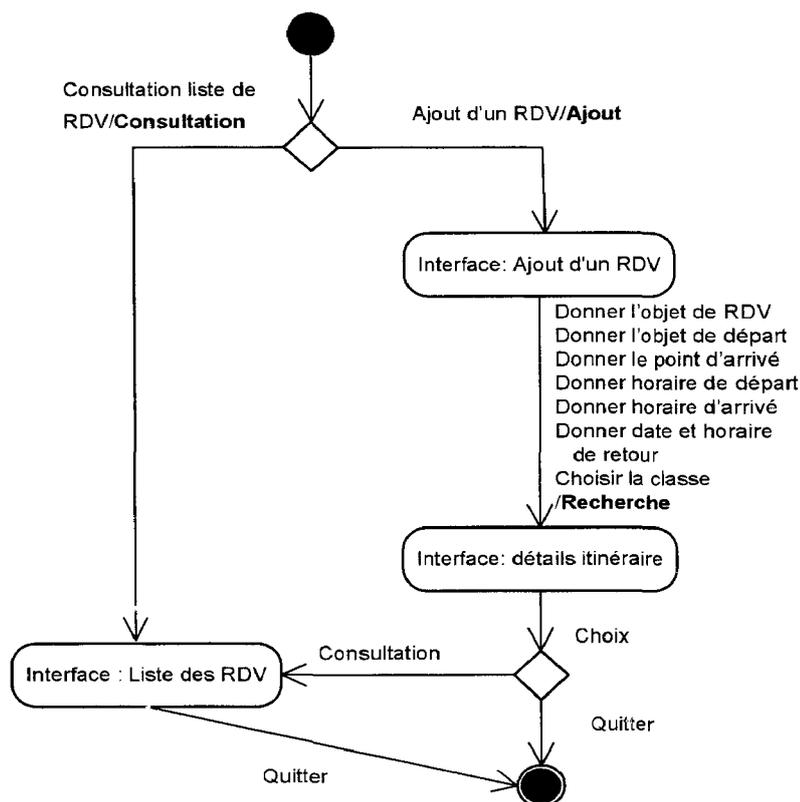


Figure 4.2. Dynamique de la tâche d'ajout d'un RDV

Les participants : Vingt sujets ont participé à cette évaluation (étudiants en informatique en mastère 1 ou doctorat). Ils ont une moyenne d'âge de 25 ans (cf. Tableau 4.4).

Tableau 4.4. Détail des sujets participant à l'évaluation

Utilisateurs	Age	Sexe	Profil	Utilisateur de transports collectifs
Sujet1	22	F	Mastère 1	Oui
Sujet2	23	F	Mastère 1	Non
Sujet3	23	F	Mastère 1	Oui
Sujet4	23	M	Mastère 1	Oui
Sujet5	21	M	Mastère 1	Oui
Sujet6	23	M	Mastère 1	Oui
Sujet7	21	M	Mastère 1	Non
Sujet8	24	M	Mastère 1	Non
Sujet9	47	M	Doctorant	Oui
Sujet10	27	M	Doctorant	Oui
Sujet11	23	M	Mastère 1	Oui
Sujet12	27	M	Doctorant	Oui
Sujet13	27	M	Mastère 1	Oui
Sujet14	32	M	Doctorant	Oui
Sujet15	27	M	Mastère 1	Oui
Sujet16	29	M	Doctorant	Oui
Sujet17	26	M	Mastère 1	Oui
Sujet18	28	M	Doctorant	Oui
Sujet19	23	M	Mastère 1	Oui
Sujet20	28	F	Doctorant	Oui

- Les outils et techniques utilisés

Dans cette expérimentation, plusieurs outils ont été utilisés, entre autres :

- **un questionnaire général** : l'évaluateur invite l'utilisateur à remplir ce questionnaire qui contient des informations générales telles que nom, prénom, email, âge, sexe, etc. (cf. Figure 4.3). Ce questionnaire intervient durant l'étape pré-expérimentale.

Figure 4.3. Questionnaire préalable



Figure 4.4. Evaluation de l'utilisation selon les critères proposés

- **Les fiches de critères** : chaque sujet utilise ces fiches pour juger le système (cf. Figure 4.4). Par exemple, il peut attribuer des poids et des notes, indiquer les problèmes et/ou les remarques signalées au cours de l'expérience et proposer des améliorations à apporter au système (cf. Tableau 4.5).
- **PDA (Portable Digital Assistant)**: assistant personnel prenant la forme d'un appareil numérique portable. Il s'agit d'un petit boîtier de la taille d'une calculatrice, qui tient

dans la main, abritant un véritable ordinateur et doté d'un écran tactile et d'un stylet (cf. Figure 4.5). Des PDA sont exploités pour cette expérimentation.

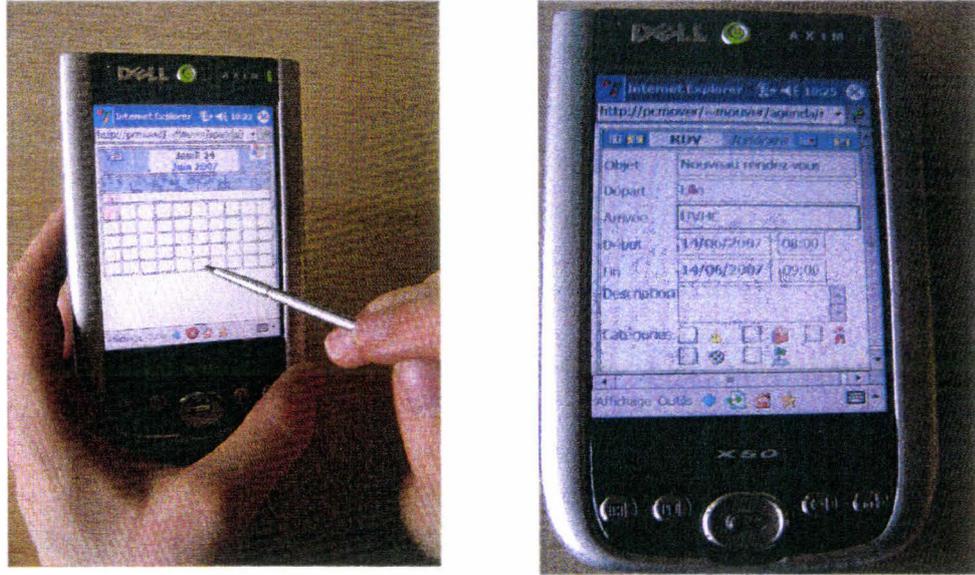


Figure 4.5. Interfaces sur PDA

4.3.2.2. Test de la maquette

Dans cette phase, les utilisateurs évaluent le système au fur à mesure qu'ils exécutent les tâches en fonction des critères déjà définis [Soui et al., 07]. Pour cela l'utilisateur affecte à chaque critère, une note selon son niveau de satisfaction et ceci suivant une échelle bien prédéfinie. En outre, dès qu'ils rencontrent un problème (correspondant au non respect d'un critère) au niveau d'une action d'une tâche, ils les notent en parallèle sur les fiches des problèmes ainsi que ses remarques. La durée d'une passation d'évaluation était de 3h par sujet (cf. Figure 4.6).

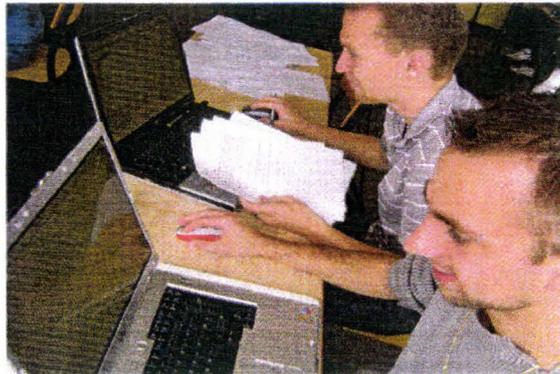


Figure 4.6. Test de la maquette

Un exemple de fiche d'évaluation remplie est donné en Figure 4.7. Chaque critère fait l'objet d'une fiche distincte.

a. L'adaptation à l'expérience : Nous entendons par expérience de l'utilisateur, toute connaissance acquise par une longue pratique jointe à l'observation. Ceci comprend, également, la familiarité de l'utilisateur avec le système et l'expérience liée au domaine d'application qui regroupe les concepts que l'utilisateur peut connaître. L'expérience est souvent modélisée par un stéréotype (débutant, initié et expert). Selon le niveau d'expertise de l'utilisateur pour une tâche donnée, le système procédera par étape (guidage du l'utilisateur novice) ou, pour l'expert, présentera un dialogue moins explicite que pour le novice (prévoir le raccourcis, des choix d'entrées simples ou multiples selon l'expérience, etc.). Cette dimension est liée à la personnalisation du contenant (adaptation de l'IHM) et à la personnalisation du contenu. Notons que Bastien et Scapin considèrent l'adaptation à l'expérience comme l'ensemble des moyens mis en oeuvre pour permettre au système de respecter le niveau d'expertise de l'utilisateur. Ces derniers ont déjà étudié ce critère dans l'évaluation de système interactif mais pas d'une manière explicite [Bastien, Scapin, 01].

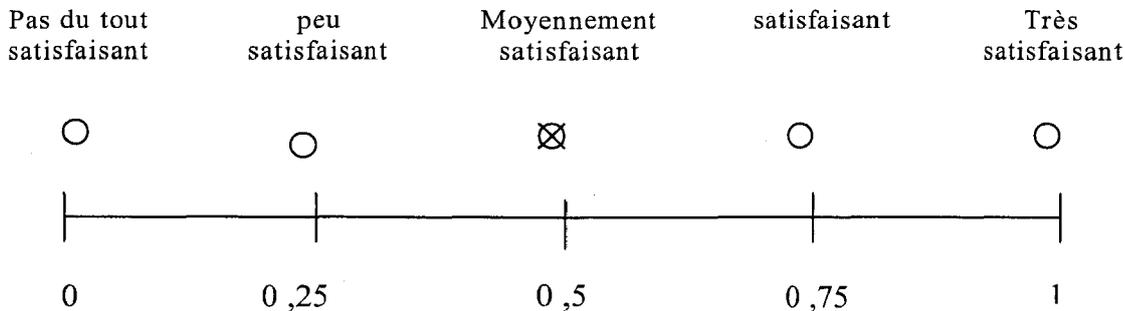
b. Est-ce que vous êtes intéressé par ce critère ?

a. Si vous êtes intéressé par ce critère, affectez un poids représentant l'importance que vous portez à ce critère (le poids est borné à la valeur dix) :

Oui **Non**

Le poids associé à ce critère est : 5

b. Selon votre niveau de satisfaction, affectez une note à ce critère, selon l'échelle suivante :



c. Le ou les problème(s) rencontré(s)?...guidage insuffisant pour un utilisateur novice.....

- Tâche concernée :recherche d'itinéraire.....
- Action(s) qui génère (nt) le problème :.....
- Remarques.....

Figure 4.7. Un exemple de fiche d'évaluation

4.3.2.3. Résultats et analyse

Cette partie présente les principaux résultats issus de ce test. Ceux-ci concernent les dimensions liées à personnalisation de contenu et de contenant.

- Personnalisation de contenu

La personnalisation de contenu regroupe trois critères : les préférences, les centres d'intérêts et l'expérience du domaine de l'utilisateur.

Comme on le voit dans la figure 4.8, 47% des utilisateurs sont très satisfaits, 41% sont satisfaits, 6% sont peu satisfaits et 6% des utilisateurs ne sont pas du tout satisfaits.

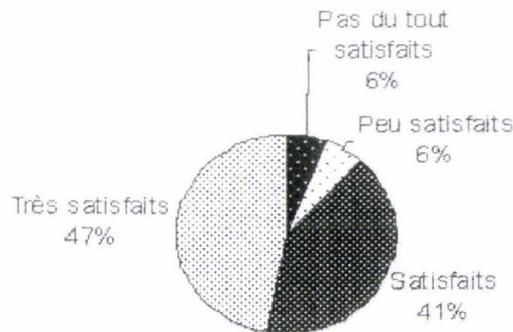


Figure 4.8. Avis des sujets vis-à-vis de la personnalisation liée au contenu

Afin de détailler les avis des sujets en ce qui concerne la personnalisation liée au contenu, nous avons comparé leur niveau moyen de satisfaction en fonction de trois critères liés à cette dimension. La figure 4.9 montre que les sujets ont un avis favorable relativement à la qualité de personnalisation liée aux préférences et à l'expérience : les moyennes des niveaux de satisfaction relatives à ces critères dépassent 0,6. Par contre la prise en compte des centres d'intérêts est moins satisfaisante.

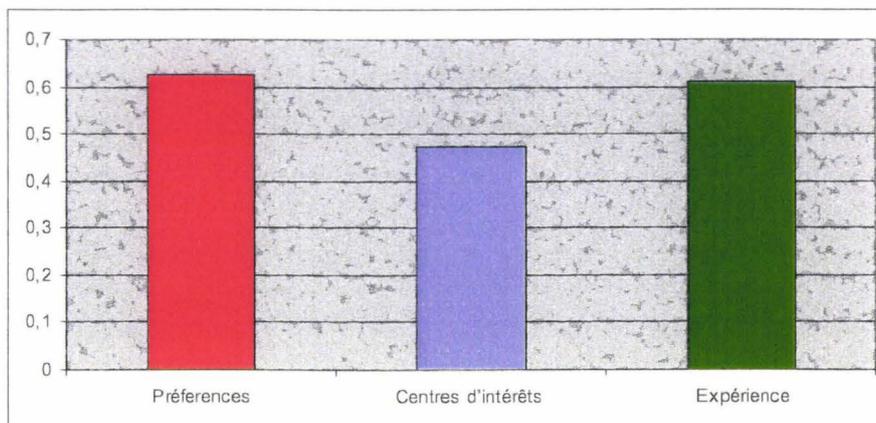


Figure 4.9. Moyennes, par critères de personnalisation liée au contenu, des niveaux de satisfaction des étudiants

- Personnalisation de contenant

La personnalisation liée au contenant regroupe quatre critères : l'adaptation à la plate-forme interactive, l'adaptation à l'environnement, l'adaptation au comportement de l'utilisateur et l'adaptation à la capacité physique de l'utilisateur (accessibilité). Le taux de satisfaction est de 33% pour les sujets très satisfaits, 17% pour les sujets satisfaits et 50% pour les sujets peu satisfaits (cf. Figure 4.10).

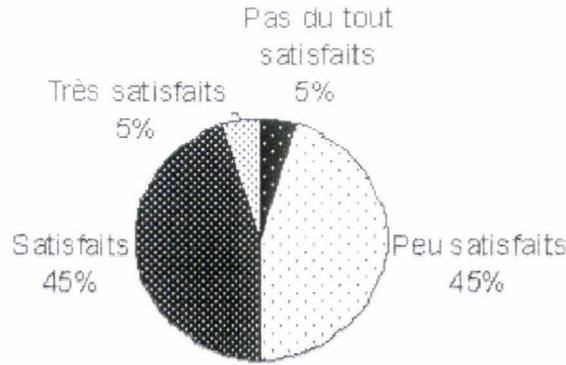


Figure 4.10. Avis des étudiants vis-à-vis de la personnalisation liée au contenu

Afin de détailler les avis des étudiants en ce qui concerne la personnalisation liée au contenu, nous avons comparé les moyennes de leurs niveaux de satisfaction en fonction des quatre critères liés à cette dimension.



Figure 4.11. Test avec PDA



Figure 4.12. Test avec PC

Après avoir testé l'application en utilisant deux supports interactifs différents (PC, PDA) (cf. Figures 4.11 et 4.12), les sujets ont donné un avis favorable pour l'adaptation du système à la plate-forme interactive. Pour le critère d'adaptation au comportement de l'utilisateur, les avis des étudiants sont centrés autour de la réponse neutre (moyenne = 5). En revanche, comme le montre la figure 4.13, les étudiants ont un avis défavorable concernant l'adaptation de l'application à la capacité physique de l'utilisateur et à l'environnement (moyenne <4).

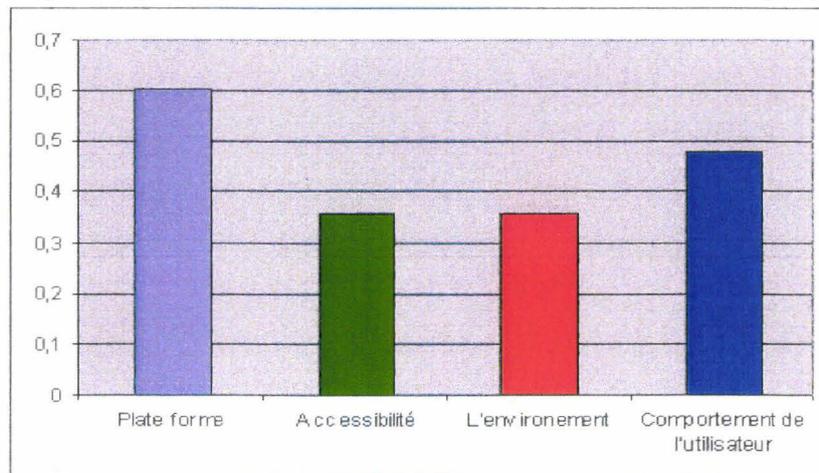


Figure 4.13. Moyennes, par critère de personnalisation liée au contenu, des niveaux de satisfaction des sujets

- Problèmes signalés par les sujets

Le graphique de la figure 4.14 montre que la majorité des sujets a signalé des problèmes liés à l'expérience, à l'accessibilité et au comportement de l'utilisateur. Concernant l'expérience, les sujets ont rapporté le manque d'aide lors de la première utilisation. Concernant l'accessibilité, un problème cité plusieurs fois est la petite taille des caractères qui entraîne une mauvaise lisibilité. Concernant le comportement, un problème récurrent est le manque de "signal de retour" lors des actions de validation.

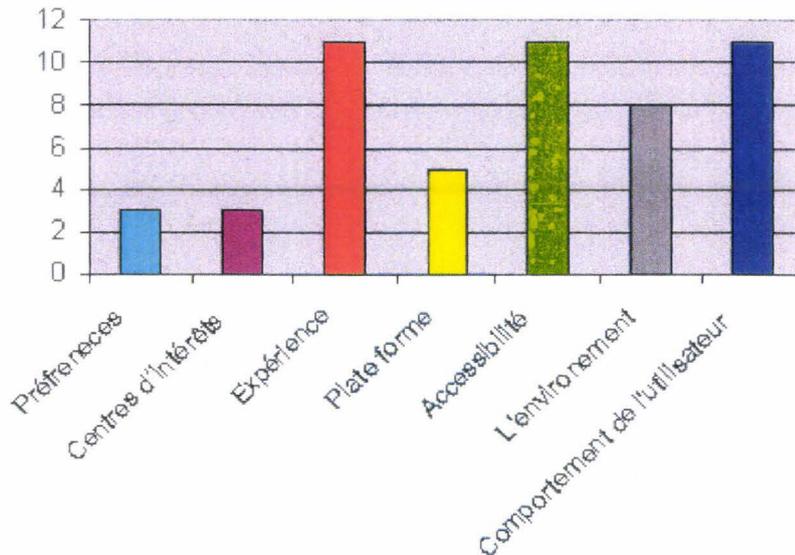


Figure 4.14. Répartition des problèmes en fonction du critère évalué

Les sujets ont signalé peu de problèmes liés aux préférences, aux centres d'intérêts et à la plateforme interactive. Ceci s'explique par l'importance qu'ont porté les concepteurs de ce système aux dimensions liées à la personnalisation de contenu et l'adaptation à la plateforme interactive en regard des autres critères.

4.3.3 Synthèse et discussion

Cette évaluation mettant en œuvre MetQus est basée sur l'intervention explicite de l'utilisateur et l'existence d'un système réel (modèle, maquette ou système final). Il s'agit de questionner l'utilisateur sur les incohérences de l'interface, les points forts ou faibles qu'il a ressentis lors de l'utilisation d'un SIP et éventuellement ses suggestions sur la manière de les corriger. C'est donc une méthode utilisable *a posteriori*. MetQus permet surtout de déterminer la satisfaction ou les problèmes de l'utilisateur qu'il est difficile ou impossible de mesurer autrement qu'avec le questionnaire.

Cette méthode permet de prendre en compte tous les critères d'évaluation liés au contenu ainsi que ceux liés au contenant. Bien que cette technique semble être facile à mettre en place, et permet de récupérer des informations utiles pour l'évaluation, elle nécessite un traitement lourd et un coût important. En effet, pour trouver un nombre satisfaisant de problèmes, de nombreux tests doivent être conduits. De plus, effectuer un seul test prend déjà un temps considérable (environ 3h par sujet, hors préparation et analyse des données), et nécessite une monopolisation importante de l'évaluateur.

4.4. Second cas d'étude : application de la méthode MetTra

Dans cette seconde étude de cas, nous expliquons l'application de la méthode MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage) [Soui et al., 08b], proposée dans le troisième chapitre, sur un système de diffusion d'informations et de services au profit des usagers des transports collectifs.

4.4.1. Présentation de l'application

Cette application porte sur l'évaluation d'un système d'information personnalisé. Le service étudié concerne la recherche d'itinéraire. A l'aide du système, l'utilisateur fournit le lieu de départ, le lieu d'arrivée et l'heure de départ. Le système lui propose un itinéraire suivant ses préférences. Les préférences de l'utilisateur peuvent correspondre à critères transport comme le moins de marche, le plus rapide, le moins coûteux, etc. L'historique de l'utilisateur est lié à des couples (requête, itinéraire choisi). Avec notre méthode, il est possible de prédire l'itinéraire susceptible d'être préféré par l'utilisateur pour une requête donnée. L'utilisateur a la possibilité de valider la proposition ou de choisir un autre itinéraire [Grislin-Le Strugeon et al., 07].

Rappelons ici que cette évaluation est basée sur l'exploitation d'un système de trace qui doit être intégré durant la phase de conception pour générer le modèle réel de l'utilisateur dans un contexte bien défini. En effet, ce modèle permet à l'évaluateur de détecter un premier ensemble représentatif de problèmes de personnalisation inhérents à l'exploitation d'un SIP. Les parties suivantes décrivent tour à tour, et pour le service de recherche d'itinéraire, la mise en pratique de la méthode proposée.

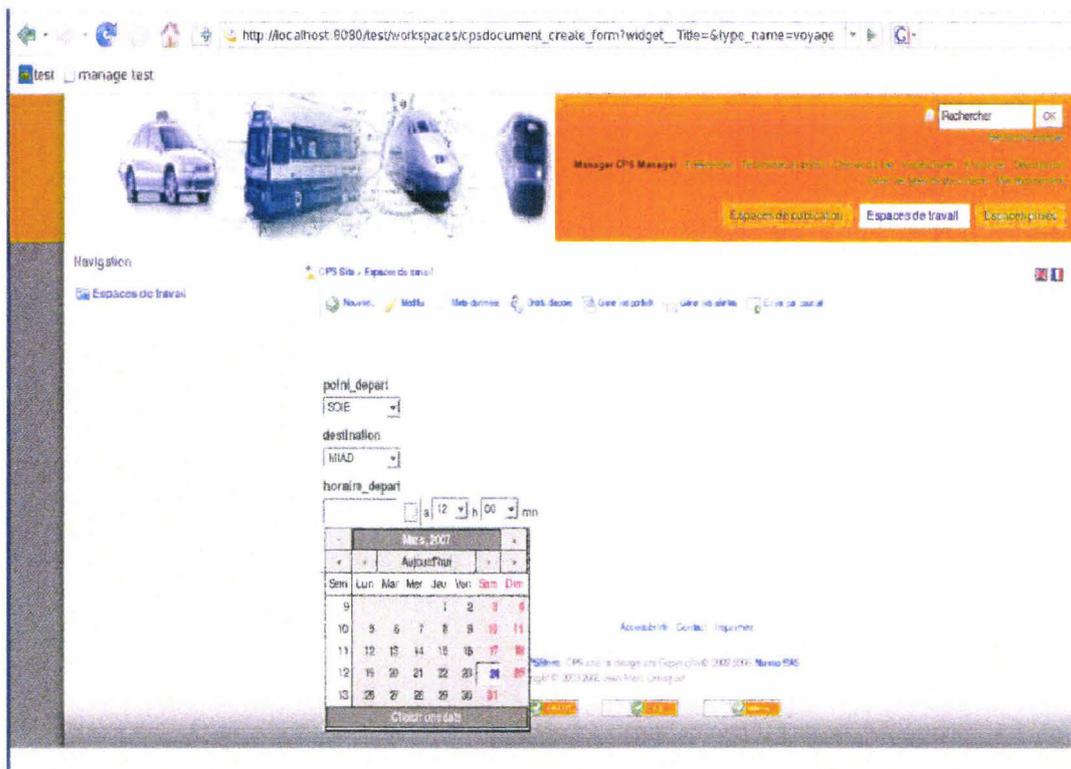


Figure 4.15. Exemple d'interface associé au service de recherche d'itinéraire

4.4.2. Mise en oeuvre de la méthode MetTra

Notre méthode repose sur quatre phases : (1) l'intégration du mécanisme de trace pendant la phase de conception du système, (2) le recueil des données relatives au comportement et aux réactions des interfaces du système, (3) l'identification et la description des problèmes rencontrés par l'utilisateur, (4) l'identification et l'explication des recommandations.

Phase 1 : L'intégration d'un mécanisme de trace pendant la conception du système

Cette phase se compose de trois étapes (analyse, conception et implémentation du service).

Analyse du service

Cette étape d'analyse comprend six étapes successivement décrites : (1) capture des besoins fonctionnels, (2) modèle Statique de la Tâche, (3) analyse fonctionnelle du service, (4) capture des besoins techniques, (5) conception générique, (6) modèle du dialogue homme-machine lié à la tâche.

- E1 : Capture des besoins fonctionnels

Les besoins fonctionnels liés au service sont modélisés au travers de diagrammes de cas d'utilisation. La Figure 4.16 présente les besoins fonctionnels pour le service de recherche d'itinéraire. Les cas d'utilisation peuvent se structurer en deux catégories :

- la première catégorie concerne l'identification de l'utilisateur (cas d'utilisation « S'inscrire » et « S'authentifier »).
- la deuxième catégorie regroupe les fonctionnalités pour la recherche d'itinéraire personnalisé (cas d'utilisation « Rechercher un itinéraire », « Afficher le détail d'un itinéraire » et « Choisir un itinéraire »).

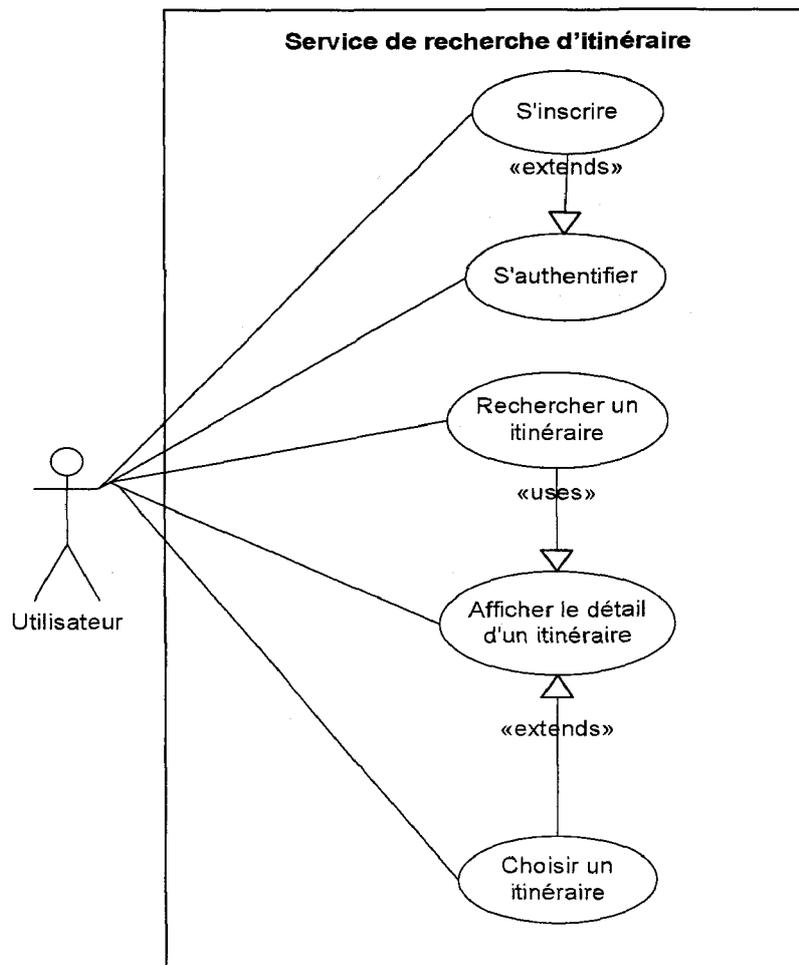


Figure 4.16. Diagramme des cas d'utilisation du service de recherche d'itinéraire personnalisé

Chaque cas d'utilisation est suivi d'une fiche de description textuelle. La figure 4.17 présente la fiche de description textuelle pour le cas d'utilisation « Rechercher un itinéraire ».

<p>1. Sommaire d'identification : inclut titre, but, résumé, acteur...</p> <p>Titre : Recherche d'itinéraire</p> <p>But : fournir un itinéraire à un usager de transports collectifs en tenant compte de ses préférences.</p> <p>Acteurs : utilisateur, système</p> <p>Date de création : 18 /05/2008 Date de mise à jour : 20/05/2008</p>
<p>2. Description des enchaînements :</p> <p>Préconditions :</p> <ul style="list-style-type: none"> -l'utilisateur est authentifié <p>Scénarios nominaux :</p> <p>S1 : Lancer requête</p> <ul style="list-style-type: none"> - 1. l'utilisateur décide de chercher un itinéraire (tâche utilisateur). - 2. l'utilisateur formule et valide sa requête (tâche interactive). - 3. le système envoie la requête de l'utilisateur au SP (tâche système). - 4. le système de personnalisation fournit le résultat (tâche système). - 5. le système d'information affiche le résultat (les itinéraires) (tâche système). - 6. l'utilisateur cherche le détail d'un itinéraire (tâche interactive). - 7. l'utilisateur choisit un itinéraire (tâche interactive). <p>Scénarios alternatifs :</p> <p>A1 : le lieu de départ non fourni.</p> <p>Ce scénario commence après l'étape 1 du scénario S1</p>
<p>3. Besoin d'IHM : Pour la formulation de la requête (la tâche interactive 2) : l'utilisateur doit pouvoir formuler sa requête en précisant le point de départ, le point d'arrivée, l'heure de départ et l'heure d'arrivée.</p>
<p>4. Contraintes non-fonctionnelles : Fréquence, temps de réponse, quitter.</p>

Figure 4.17. Fiche de description textuelle pour le cas d'utilisation "Rechercher un itinéraire"

- E2 : Modèle statique de la tâche (MST)

Ce modèle représente la tâche que l'utilisateur doit pouvoir effectuer au travers de l'interface. Cette tâche déjà définie dans l'étape de capture de besoins fonctionnels sous la forme d'un cas d'utilisation global, sera décomposée en une hiérarchie de tâches et de sous-tâches. La construction de chaque MST associée à un cas d'utilisation global est guidée par la seconde partie (description des enchaînements) de sa fiche de description déjà décrite dans l'étape précédente. Ce modèle est modélisé au travers d'un diagramme d'activité étendu. La Figure 4.18 présente le modèle statique pour la tâche « recherche d'itinéraire ».

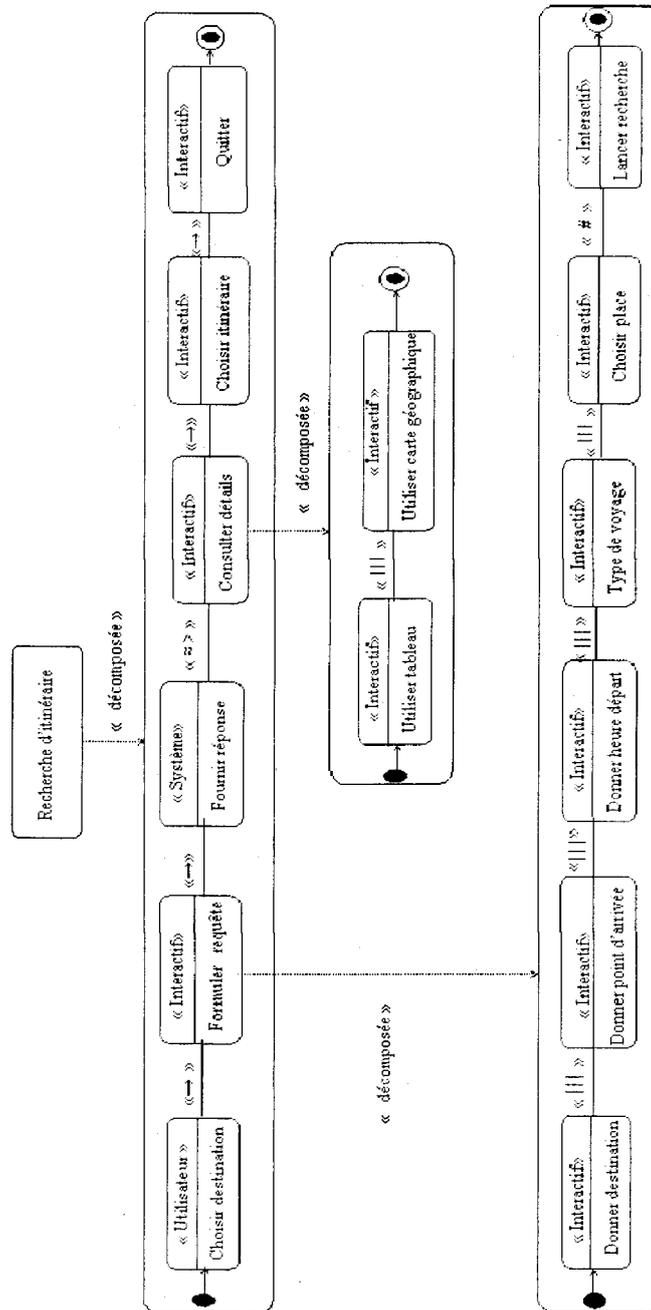


Figure 4.18. Le modèle statique pour la tâche « recherche d'itinéraire »

- E3 : Analyse fonctionnelle du service

La structure du service est modélisée au travers de diagrammes de classes. Les concepts connus des utilisateurs (objets métiers) et les concepts applicatifs liés à l'informatisation permettent d'identifier les classes candidates. La figure 4.18 présente la modélisation statique du service de recherche d'itinéraire. La Figure 4.19 présente un scénario pour la recherche d'itinéraire en détaillant les interactions entre les différents objets du service. L'utilisateur

lance sa requête au travers du service (recherche) de recherche d'itinéraire personnalisé. *Recherche* envoie à *Réponse* tous les itinéraires possibles répondant à la requête et recommande un itinéraire susceptible d'intéresser l'utilisateur. Cette réponse est associée à un choix (*Choix*) qui permet de référencer l'itinéraire choisie par l'utilisateur.

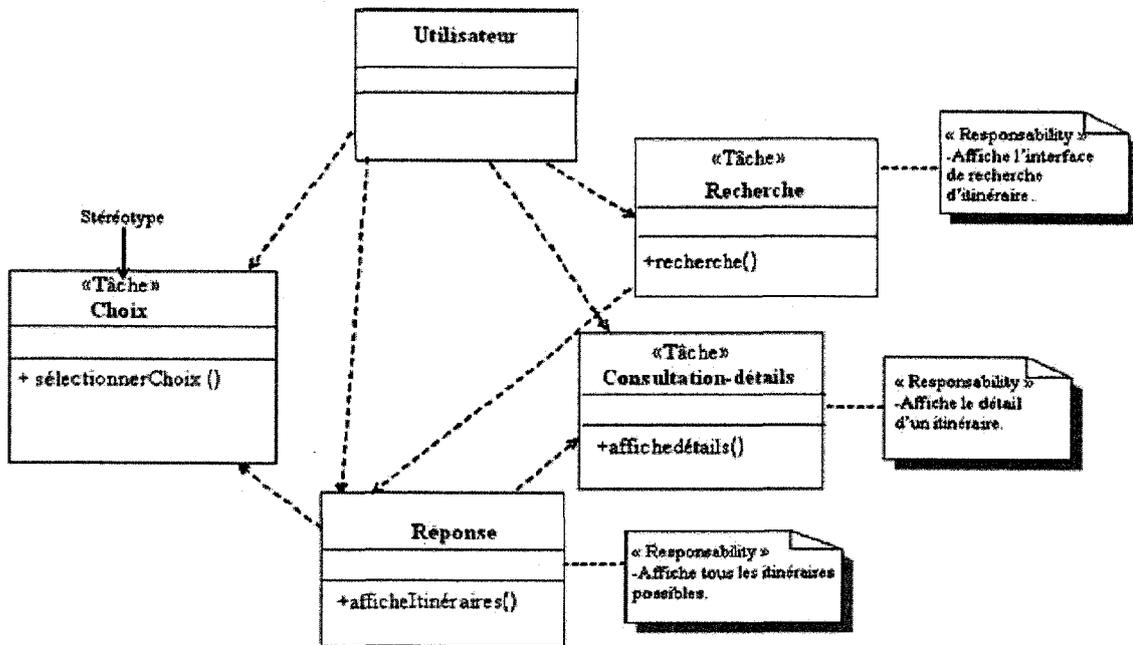


Figure 4.19. Diagramme de classe pour le service de « recherche d'itinéraire »

Nous avons aussi défini les modèles des objets interactifs pour la tâche recherche d'itinéraire (cf. Figure 4.20). En effet, en partant du modèle de tâche statique, on définit pour chaque tâche interactive possédant des sous tâches élémentaires, les objets interactifs nécessaires pour la réalisation des actions élémentaires de l'utilisateur. L'identification de ces objets est primordiale afin de générer des traces et vérifier la conformité de l'utilisation du système avec le modèle de tâches qui reste l'objectif de notre travail.

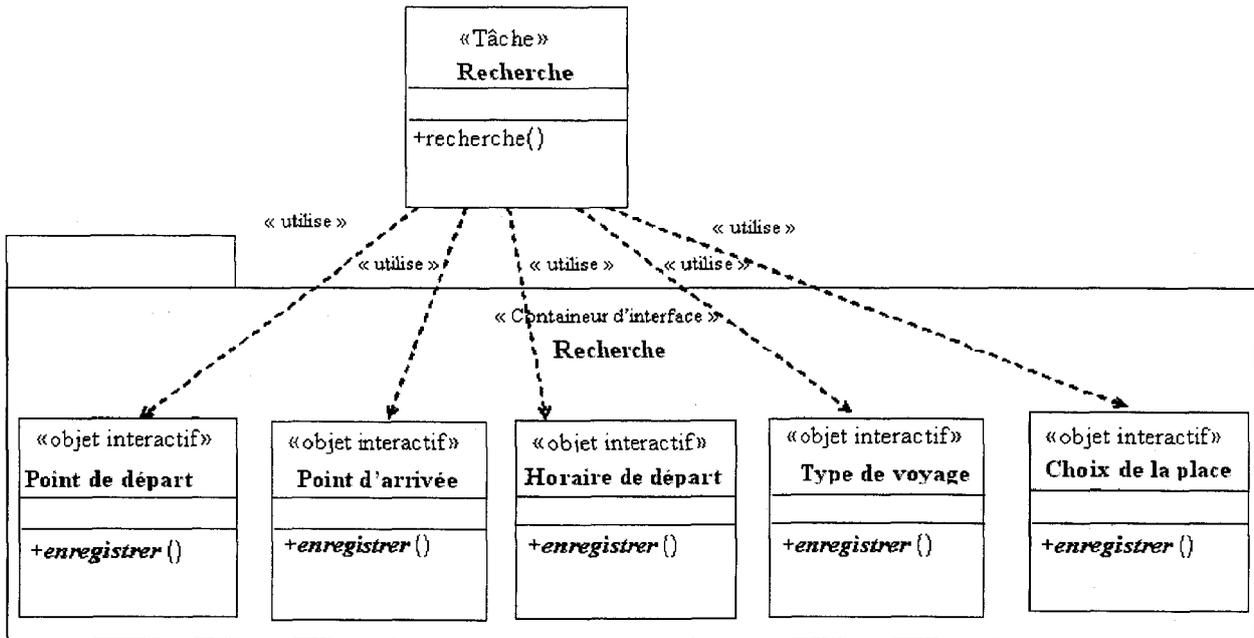


Figure 4.20. Modèle des objets interactifs pour la tâche de « recherche d'itinéraire »

- E4 : Capture des besoins techniques

Nous disposons d'un poste de travail sur lequel est installé Windows XP. Tous les composants logiciels seront installés sur ce poste. L'utilisateur accède au service de « recherche d'itinéraire » à partir de son poste de travail en passant par le réseau internet.

- E5 : Conception générique

La conception générique définit les éléments nécessaires à la construction de l'architecture technique indépendamment des aspects fonctionnels issus de la capture des besoins fonctionnels et de l'analyse fonctionnelle (cf. Figure 4.21).

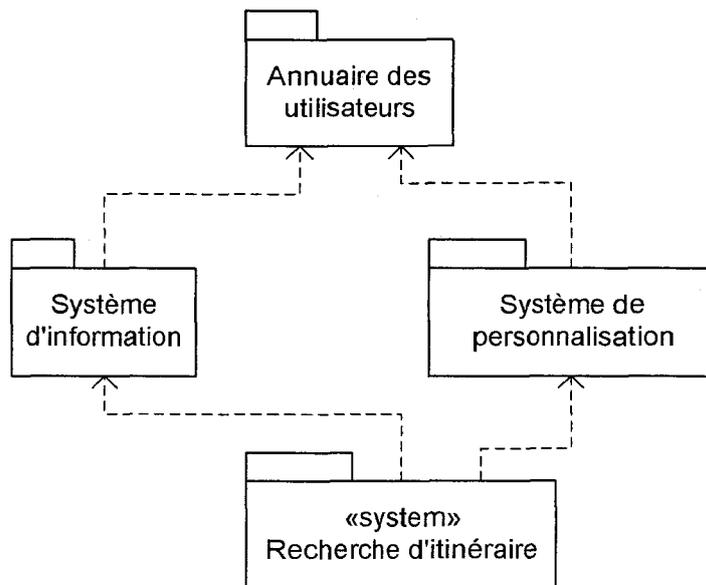


Figure 4.21. Conception générique pour le service de « recherche d'itinéraire »

E6 : Modèle du dialogue homme-machine lié à la tâche

Le modèle de dialogue de la tâche est modélisé au travers d'un diagramme d'états-transitions étendu. La figure 4.22 présente le modèle de dialogue pour la tâche recherche d'itinéraire.

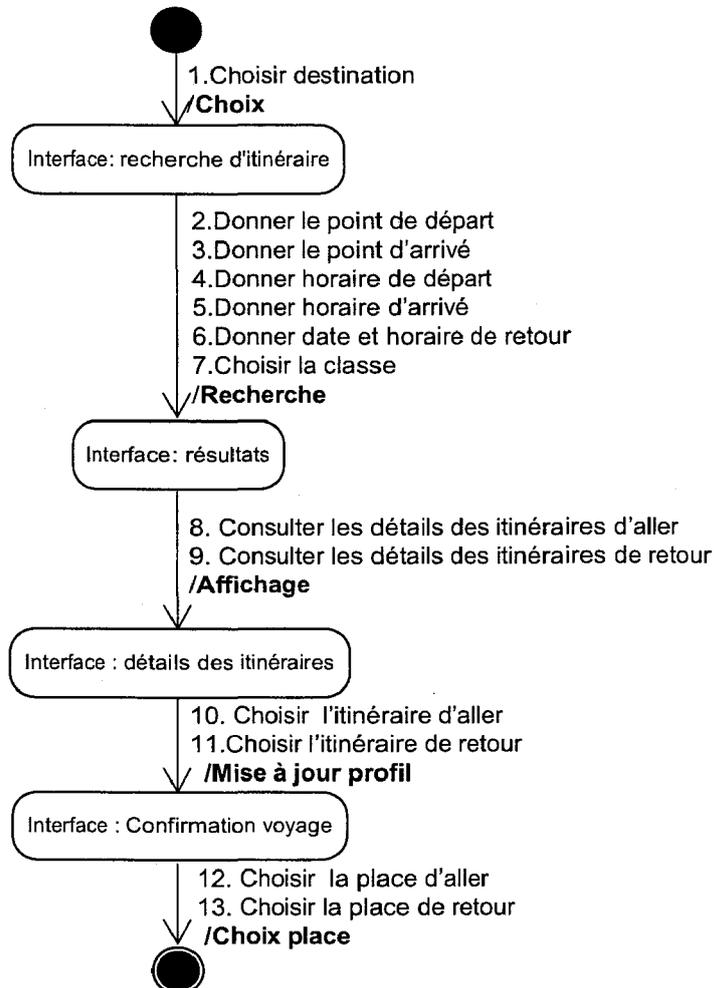


Figure 4.22. Modèle de dialogue du service « recherche d'itinéraire »

Conception du service

Dans cette étape, le diagramme de classe du service « recherche d'itinéraire » est développé en précisant les types des attributs de classe et les types de paramètres d'entrées/sorties des méthodes. C'est à ce niveau aussi que le diagramme de séquence est utilisé pour décrire les interactions entre les différentes classes du service (cf. Figure 4.23).

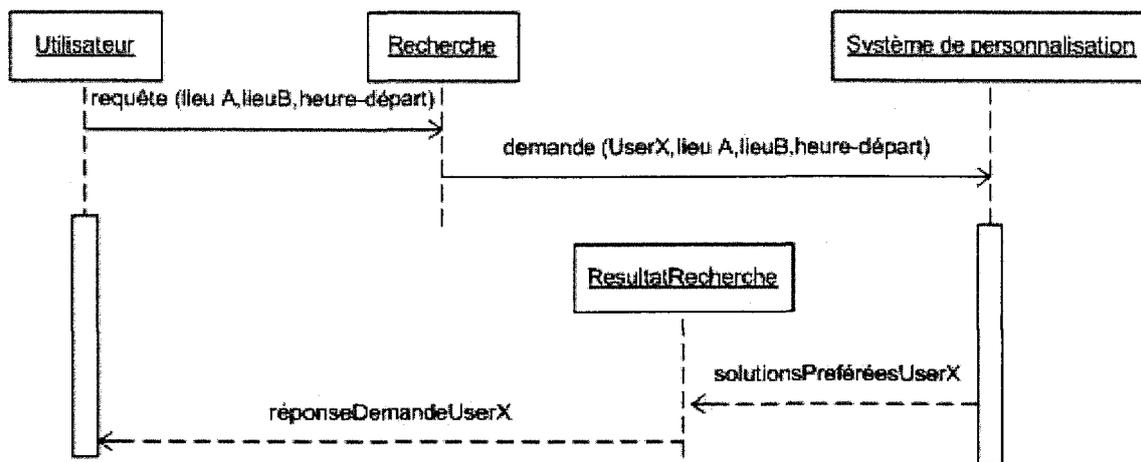


Figure 4.23. Diagramme de séquences pour la recherche d'itinéraire personnalisé

Implémentation du service

L'application a été implémentée en utilisant l'environnement Eclipse. C'est à ce niveau que le mécanisme de traces est inséré en se basant sur la bibliothèque Swing du langage Java. En effet, la boîte à outils Swing définit un ensemble de composants graphiques appelés widgets permettant de composer l'interface : boutons, champs de texte, barres de défilement, etc. L'action d'un utilisateur sur un widget, par exemple un clic sur un bouton, génère un évènement au sein du système. Des objets spécifiques appelés *listeners* peuvent être associés aux widgets afin d'écouter ces évènements. Pour notre étude de cas, nous avons associé par exemple un listener *KeyListener* à un widget *JTextFieldDestination* (champ de texte). Une fonction enregistrer () est également définie pour stocker les évènements de l'utilisateur et les réactions des interfaces de système d'une manière séquentielle dans un fichier de trace (cf. Figure 4.24).

```

Private JTextField getJTextFieldDestination ()
{
    if (jTextFieldDestination == null)
    {
        JTextFieldDestination = new JTextField (); // définition de champs de texte
        jTextFieldDestination.addKeyListener (new java.awt.event.KeyAdapter ()
        {
            Public void keyTyped (java.awt.event.KeyEvent e)
            {
                String v;
                If (jTextFieldDestination.getText ().length ()==0){
                    jTextFieldDestination.setName ("destination");
                    v= jTextFieldDestination.getName ();
                    Enregistrer (v); // La fonction permettant d'enregistrer les actions de l'utilisateur
                }
            }
        }
    }
    return jTextFieldDestination;
}
    
```

Figure 4.24. Une méthode de définition du champ de texte destination

Phase 2 : Recueil des données relatives au comportement de l'utilisateur et aux réactions des IHM du système

Nous présentons maintenant deux exemples d'application de la méthode afin de vérifier l'adaptation du système à l'expérience de l'utilisateur et à la plateforme interactive. Cette phase débute par l'étape de définition du contexte d'évaluation.

- Définition du contexte d'évaluation

Cette étape consiste à définir le contexte d'évaluation. Ainsi, selon le critère à vérifier, l'évaluateur doit choisir la catégorie des sujets qui seront impliqués dans l'évaluation (novices, experts, etc.), le support avec lequel les sujets interagissent avec le système pendant l'évaluation (PDA, PC, etc.) et l'environnement accueillant l'interaction (cf. Tableau 4.6).

Tableau 4.6. Définition du contexte d'évaluation

Les critères	Adaptation du système à l'expérience de l'utilisateur	Adaptation du système à la plate-forme interactive
Contexte d'évaluation proposé	Pour évaluer l'adaptation du système à l'expérience de l'utilisateur, l'évaluateur propose d'observer l'utilisation du système auprès de cinq sujets (novices) qui utilisent un système de recherche des itinéraires pour la première fois.	Pour évaluer l'adaptation du système à la plate-forme interactive, l'évaluateur propose d'observer l'utilisation du système par cinq sujets qui interagissent avec le système en utilisant un PDA.

- Enregistrement des actions de l'utilisateur et des réactions des IHM du système

Cette étape assure la capture des actions élémentaires de l'utilisateur et des réactions des IHM qui sont enregistrés dans un fichier de trace. Ce fichier peut être structuré sous format texte ou XML. La DTD⁵ (Document Type Definition) du fichier de trace structuré sous le format XML commence avec la balise <TraceFile>. Cette balise est composée de deux parties principales :

- La balise <ObjectInterface> décrit l'interface du système en termes des services fournis et des objets interactifs manipulés par l'utilisateur.
- La balise <contextModel> définit le contexte d'usage selon le triplet <plate-forme, utilisateur, et environnement>. La figure 4.25 donne un exemple d'extrait d'un fichier de trace décrit en XML

⁵ La DTD (Document Type Definition) définit une hiérarchisation d'éléments avec des typages de ces éléments via des notions d'attribut. Le document XML se conforme à la structure syntaxique de la DTD [OMG, 2000].

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Traces>
3 <Services>
4 <Service id="S-search- itinerary" name= "Recherche d'itinéraire" />
5 <Object id="TextField-departure" name="Saisie du point de depart"/>
6 <Object id="TextField-Arrival" name="Saisie du point d'arrivée"/>
7 <Object id="TextField- date and time of departure" name="Saisie du date et heure de départ"/>
8 </Services>
9 <Services>
10 <Service id="S-book - ticket " name= " réserver mon voyage"/>
11 <Object id="TextField-departure" name="choisir itinéraire"/>
12 <Object id="TextField-choiseitinerary " name="choisir itinéraire"/>
13 <Object id="TextField- choiseplace" name="choisir une place"/>
14 </Services>
15
16 <Context>
17 <context id="context_default"/>
18 <platform id="plate-formel" type="any"/>
19 <environment id="environnement1" type="any"/>
20 <user id="user1" language="EN" experience="1" workstate="hurry"/>
21 </Context>
22 </Traces>

```

Figure 4.25. Exemple d'extrait d'un fichier de trace sous format XML

Phase 3 : Identification des problèmes

Cette phase comporte trois catégories d'évaluation : l'évaluation statique, l'évaluation multi-contextuelle et l'évaluation dynamique.

- L'évaluation statique

Rappelons que l'évaluation statique consiste à évaluer l'interface dans un contexte bien défini. Cette évaluation repose sur les règles d'évaluation définies par l'évaluateur pour chaque critère, de même que sur les attributs d'évaluation collectés par le système de traces (le nombre des services traités par interface, le nombre des objets interactifs utilisés par service, etc.). En fait, l'évaluateur vérifie le respect ou non des ces règles. Soulignons que ces règles sont proposées par l'évaluateur en se basant sur son expérience ou des connaissances issues de la littérature. A ce niveau, les données capturées (cf. Tableau 4.7, partie gauche) sont récupérées pour les analyser à l'aide des règles d'évaluation (cf. Tableau 4.7, partie centrale).

Ensuite, les résultats d'analyse sont affichés sous des formes compréhensibles pour le concepteur. Ces résultats d'analyse peuvent être interprétés par l'évaluateur pour critiquer le système (partie droite) et suggérer des améliorations nécessaires au concepteur.

La partie suivante présente un exemple d'évaluation de l'adaptation d'une interface par rapport à l'expérience d'un utilisateur novice.

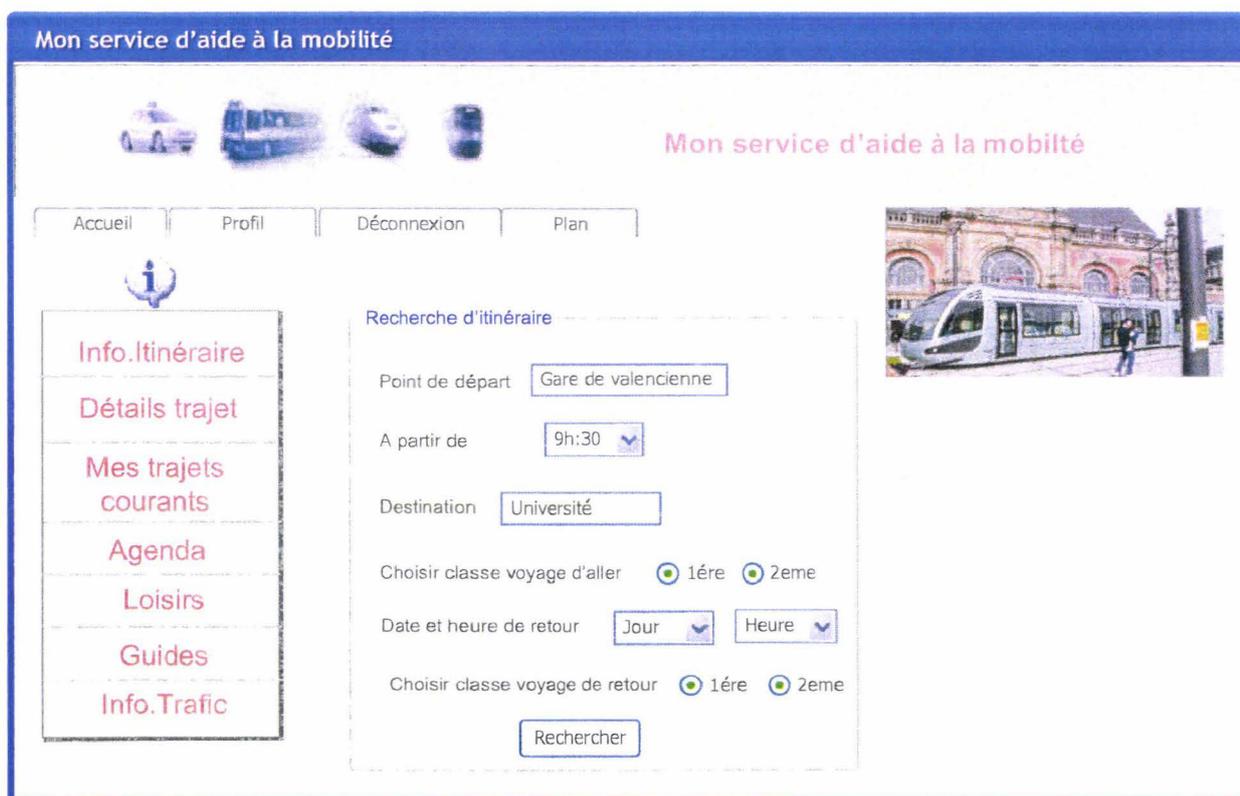


Figure 4.26. Un exemple d'une interface à évaluer associée au service de recherche d'itinéraire

Exemple illustrant le critère : adaptation à l'expérience de l'utilisateur

Pour s'assurer qu'un utilisateur novice réussit facilement à interagir avec une interface de recherche d'itinéraire, on suppose qu'une évaluation de l'utilisation du système auprès de cinq sujets novices est décidée. Cette évaluation permet de juger l'adaptation du système au niveau d'expertise des sujets novices en appliquant un ensemble de règles d'évaluation.

Tableau 4.7. Les attributs et les règles d'évaluation exploités dans le cadre de l'évaluation statique

Attributs d'évaluation	Règles d'évaluation	Observation de l'évaluateur
Nombre de services traités	Le nombre de services traités par l'interface pour un utilisateur novice doit être égal au maximum à un.	L'interface évaluée fournit à la fois deux services différents : la recherche d'itinéraire pour l'aller et pour le retour. Ceci peut désorienter un sujet novice.
Nombre des objets interactifs utilisés par service	Le nombre d'objets interactifs utilisés par service doit être égal à cinq au maximum.	L'interface évaluée est chargée car elle présente sept objets interactifs, ce qui peut engendrer une difficulté d'utilisation de l'interface pour un utilisateur novice. Ceci peut s'expliquer par le fait que cette interface ne fournit pas à un utilisateur novice une procédure claire et un guidage suffisant pour effectuer la tâche de recherche d'itinéraire.

Pour décrire les problèmes détectés, l'évaluateur place dans un tableau le modèle de dialogue de la tâche prescrite en spécifiant les problèmes détectés (cf. 4.27, partie gauche du tableau) et le modèle de dialogue intégrant les recommandations (cf. 4.27, partie à droite). Cette recommandation est proposée au concepteur par l'évaluateur afin de l'aider à résoudre les problèmes concernés.

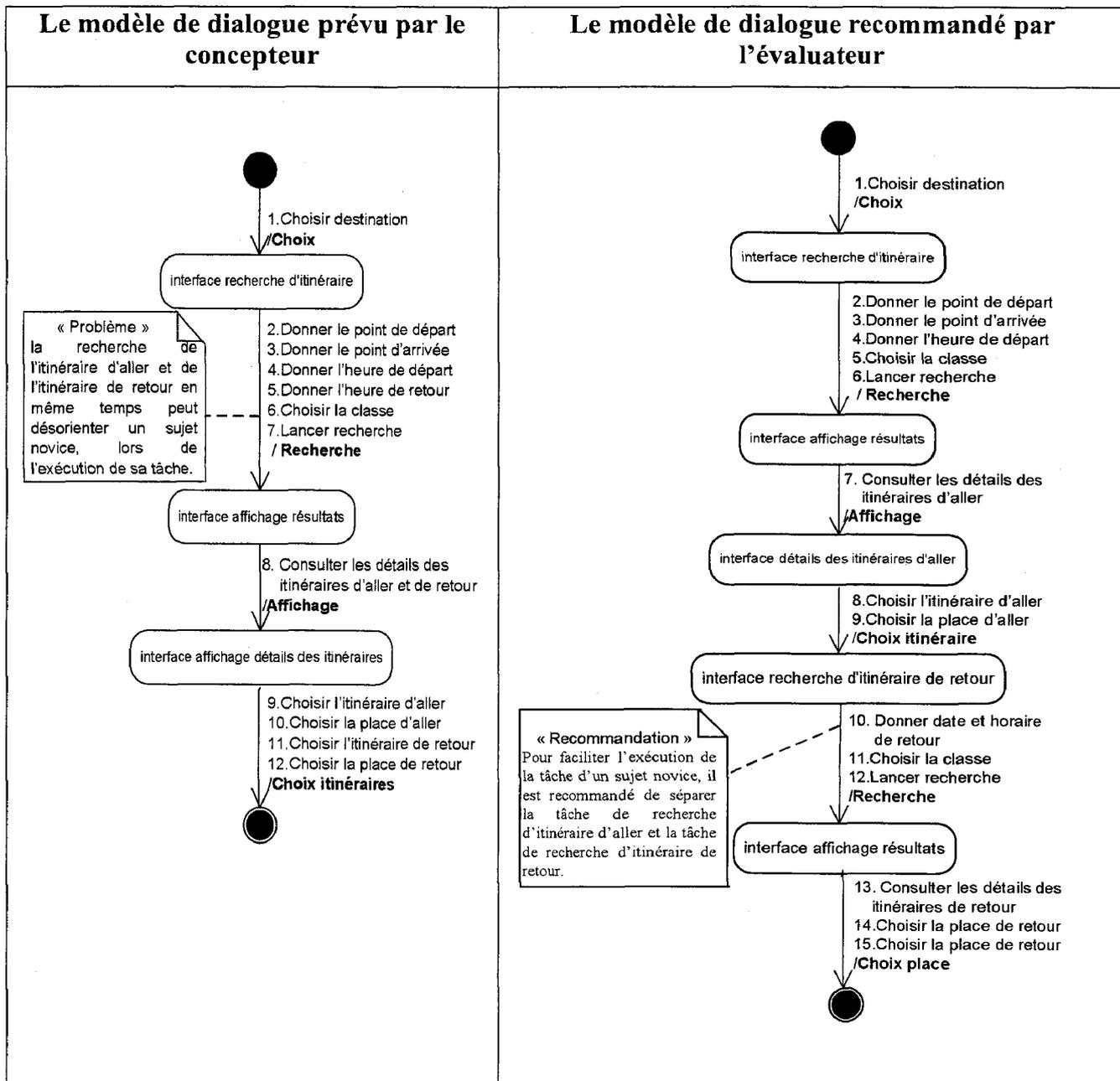


Figure 4.27. Modélisation avec UML de la tâche prescrite et de la procédure proposée par l'évaluateur

- L'évaluation multi-contextuelle

Rappelons que cette évaluation permet de vérifier la réaction des interfaces suite aux changements de contextes. Ainsi, l'interface peut posséder plusieurs facettes de présentation utilisables dans différents contextes. Cette évaluation se base sur une confrontation des modèles de dialogue d'une même tâche dans des contextes différents. Ces modèles de dialogue sont déterminés à partir des enregistrements effectués par le système des traces.

Cette confrontation peut être suivie d'un tableau comparatif contenant les divers types des objets interactifs par service dans les différents contextes. La partie suivante présente un exemple d'évaluation de l'adaptation multi-contextuelle d'une interface à la plate-forme interactive.

Exemple illustrant le critère : l'adaptation du système à la plate-forme interactive

Pour s'assurer qu'une interface d'un SIP s'adapte aux supports avec lesquels l'utilisateur interagit avec le système, nous proposons de tracer l'activité d'utilisateurs utilisant un PDA et un PC. Puis nous comparons les modèles de dialogue observées correspondant à ces contextes différents. Nous prenons comme exemple la tâche de recherche d'itinéraire visible en figure 4.28 qui montre le modèle de dialogue de l'utilisateur lors de l'utilisation d'un PC (partie gauche) et le modèle de dialogue relatif à l'utilisation d'un PDA (partie droite).

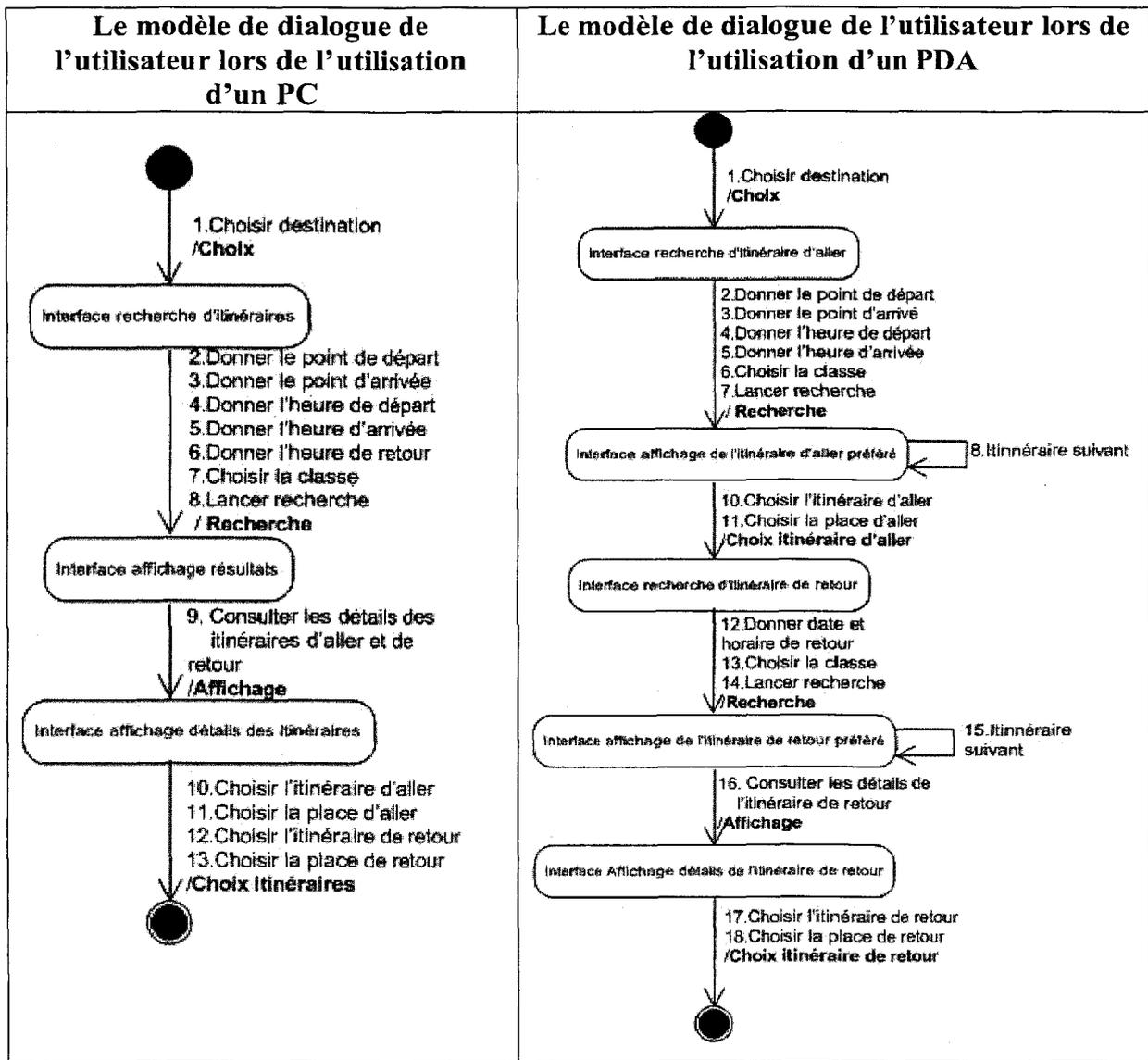


Figure 4.28. Les modèles de dialogue de l'utilisateur lors de l'utilisation d'un PC et PDA

La figure 4.28 montre que les deux modèles de dialogue sont différents. Ainsi, les sujets ont réussi à accomplir leur tâche mais par deux manières différentes. En effet, vue la capacité d'affichage limitée d'un écran d'un PDA, l'affichage des détails des itinéraires sera par itinéraire dans ce cas, au lieu de les présenter sur la même interface en utilisant un PC. Ceci

s'explique par la capacité d'adaptation du système qui consiste à affecter pour chaque support candidat le séquençement de tâches de navigation convenable.

L'évaluation d'adaptation navigationnelle peut être suivie d'une évaluation d'adaptation représentationnelle. Cette dernière évaluation repose sur la structure de l'interface en termes des objets de la boîte à outils présents sur la plate-forme cible. Pour cela, l'évaluateur place dans un tableau les différents types des objets interactifs par service pour montrer la différence d'adaptation de l'interface dans les différents contextes (cf. Tableau 4.8).

Tableau 4.8. Exemples d'adaptations représentationnelles lors de l'utilisation d'un PC et d'un PDA

Support d'interaction		PC	PDA
Objets interactifs pour l'interface de recherche d'itinéraire	-	Types	Types
	Départ	Textfield	Combobox
	Destination	Textfield	Combobox
	Heuredépart	Textfield	Combobox
	Rechercher	Jbutton	NSButton

- L'évaluation dynamique (ou de contenu)

Cette évaluation permet d'estimer si la personnalisation proposée par le système est capable de prendre en compte les préférences de l'utilisateur telles que son niveau professionnel, ses domaines d'intérêts, etc. Par exemple, dans le domaine du transport, lorsqu'un utilisateur choisit un itinéraire parmi d'autres, il ne le fait pas au hasard. Son choix est basé sur certains critères de sélection qui lui sont propres (et ceci est valable quel que soit le domaine dans lequel un choix s'impose). Ainsi, l'idée de cette évaluation est de comparer les services fournis à l'utilisateur par rapports à ceux qu'ils choisissent en se basant sur le coefficient de corrélation de Pearson et le calcul de taux d'erreur moyen du SIP (cf. 3.3.1.5.).

Une série de tests a été réalisée avec dix sujets (étudiants en informatique ayant une moyenne d'âge de 25 ans) et nous avons fait varier le nombre des requêtes de recherche d'itinéraire. Le tableau 4.9 montre le taux d'erreur moyen obtenu pour le système testé en fonction du nombre de requêtes. Dans notre travail, si l'utilisateur a choisi, parmi les itinéraires proposés, l'itinéraire 1, la prédiction est considérée comme bonne.

Tableau 4.9. Erreur de prédiction moyenne en fonction du nombre requêtes

Nombre des requêtes	1	2	3	4	5	6	7	8	9	10
taux d'erreur moyen	34.312%	8.291%	3.573%	0.951%	0.433%	0.069%	0.075%	0.079%	0%	0%

La figure 4.29 présente aussi l'évolution du taux d'erreur moyen en fonction du nombre de requêtes. L'évaluateur a constaté que lorsque le nombre des requêtes augmente, le taux d'erreur moyen diminue.

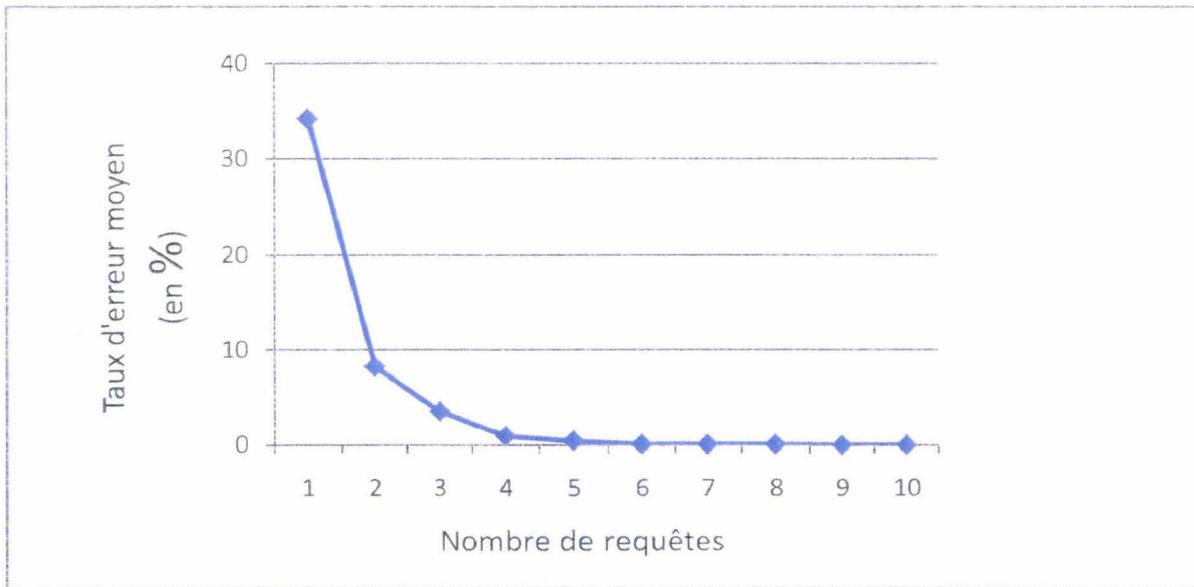


Figure 4.29. Evolution de taux d'erreur moyen en fonction de nombres de requêtes

A travers une autre expérimentation avec les dix sujets, en calculant pour chaque utilisateur, le coefficient de corrélation de Pearson, et comme précédemment, nous avons fait varier le nombre des requêtes. Le tableau 4.10 montre les résultats obtenus.

Tableau 4.10. Coefficient de corrélation de Pearson en fonction du nombre requêtes

Nombre de requêtes	1	2	3	4	5	6	7	8	9	10
Coefficient de corrélation de Pearson en %	0.12	0.23	0.21	0.45	0.06	0.06	0.75	0.79	0.91	0.98

La figure 4.30 montre que, plus le nombre des requêtes augmente, plus le coefficient de corrélation de Pearson diminue. Le nombre de requêtes influe sur les prédictions. Ce qui est logique car avec un grand nombre de requêtes, le système arrive à connaître les préférences des utilisateurs et devient capable de leur fournir des itinéraires qui correspondent à leurs besoins et préférences.



Figure 4.30. Evolution du coefficient de corrélation de Pearson en fonction du nombre de requêtes

4.4.3. Synthèse et discussion

Dans cette partie nous nous sommes intéressés particulièrement à l'évaluation des SIP. Notre approche, qui se base sur l'intégration de mécanismes de trace dès la conception, présente potentiellement, dans l'état actuel de nos travaux de recherche, plusieurs avantages.

En premier lieu, elle possède l'avantage de la prise en compte de la réalité de l'utilisation du système dans le cadre de tâches (réelles ou **représentatives**), **tout en impliquant l'utilisateur (réel ou représentatif)**. En effet, le système de trace produit des traces « orientées usage » qui fournissent rapidement des informations exploitables lors de la phase d'évaluation.

Notre approche propose aussi un ensemble de critères selon lesquels un SIP peut être évalué. Ces critères représentent un moyen pour aider l'évaluateur à identifier les problèmes rencontrés par les utilisateurs.

Un autre avantage de notre approche est la représentation graphique de la dynamique de la tâche (modèle de la tâche prescrite, modèle de la tâche réelle de l'utilisateur et modèle intégrant des recommandations de l'évaluateur) en se basant sur le diagramme d'états-transitions étendu issu du langage UML. Cette modélisation permet une prise en main relativement facile (puisque une grande majorité de développeurs ont adopté UML), dans la mesure aussi où il existe de nombreux ateliers de génie logiciel dédiés. Cependant, MetTra possède aussi des inconvénients, et nécessite d'être amélioré. Dans le chapitre suivant, nous allons présenter ces inconvénients et proposer de futures améliorations prenant la forme de perspectives de recherche.

4.5. Troisième cas d'étude : Application de la Méthode MetSim

La partie suivante présente la mise en pratique de la troisième méthode que nous proposons, intitulée MetSim (METHode d'évaluation par SIMulation), dédiée pour les systèmes d'information personnalisés.

4.5.1. Présentation de l'application

Dans le domaine ferroviaire, une des premières préoccupations d'un usager des transports collectifs est de se renseigner sur les prochains départs afin de savoir quand il pourra prendre son train, de prendre connaissance du quai concerné et, de réserver son voyage. Pour répondre à cette problématique, on va créer une application permettant d'afficher les prochains départs des trains, accessibles à travers une interface web (cf. Figure 4.31).

Mon service d'aide à la mobilité

Accueil Profil Déconnexion Plan

Point de départ

A partir de

Prochains départ de la gare de valenciennes

Num	Heure	Destination	Mode de transport	Durée
1567	9h :45	Université	tramway	20 min
7685	10 :05	Chemin vert	tramway	15 min
8754	10 :15	Compus Mont Hout	Bus	30 min
2375	10 :30	Université	tramway	20 min
7669	10 :34	Porte de Paris	Bus	18 min
4595	11 :00	Université	Bus	34 min
9745	11 :10	Douai	train	38 min
9527	11 :15	Poirier université	train	10 min

Figure 4.31. Exemple d'interface relative à l'affichage des prochains déplacements

4.5.2. Mise en oeuvre de la méthode MetSim

Rappelons que le principe de l'évaluation par simulation est de tester l'application dans des contextes d'usage différents. Dans cette partie, nous décrivons deux exemples d'évaluation de l'application dans deux contextes d'usage différents.

4.5.2.1. Premier exemple d'évaluation

L'évaluation débute par la définition du contexte d'usage. Le contexte choisi dans cette évaluation est le suivant : un usager des transports est doté d'un PDA et il est en état pressé (en effet, il arrive en gare juste quelques minutes avant son départ).

Suite à l'important changement au niveau du contexte lié à la plate-forme, l'adaptation permet de changer la facette de présentation de l'interface en adéquation avec les nouvelles caractéristiques de la plate-forme (PDA, possédant un espace d'affichage limité de 360x240). En outre, par la connaissance de l'agenda de l'utilisateur, le système doit pouvoir adapter l'interface à l'utilisateur de manière à ne lui présenter que les informations utiles correspondant uniquement à sa destination.

Afin d'évaluer cette adaptation, les conteneurs et les composants qui réagissent et leurs caractéristiques seront identifiés par l'évaluateur en comparant la spécification concrète de l'interface avant et après l'adaptation. Chaque conteneur ou composant adapté sera décrit en UsiXML (User Interface eXtensible Markup Language)⁶ [Limbourg et al., 04]. La figure 4.32 présente la spécification concrète de l'interface après le processus d'adaptation en la simulant sur un PDA.

The screenshot shows a software interface titled "Système d'évaluation" (Evaluation System). It is divided into two main sections: "Simulation sur un PDA" (Simulation on a PDA) and "Spécification concrète de l'interface" (Concrete interface specification).

The "Simulation sur un PDA" section displays a virtual HTC PDA screen. The screen shows a "Vision Test" application with a "Prochains départs" (Next departures) section. Below this, there is a table titled "Itinéraires préférés" (Preferred routes) with the following data:

Titre	Heure	Description	Moyen de transport	Quartier
1569	8 h 45	Unionville	tramway	20 min
2979	10 30	Unionville	tramway	20 min
9673	10 31	Pointe St-Louis	tram	8 min
4392	11 00	Unionville	Bus	34 min
9523	11 15	Pointe St-Louis	tram	10 min
3487	11 35	Unionville	Bus	37 min

The "Spécification concrète de l'interface" section shows the corresponding UsiXML code. The code defines the UI components and their constraints, such as grid layouts, text inputs, and output labels. Key elements include:

- Generated by GrafXML 1.2.0 build id : 200612141408
- Warning: AUI Model save is a work in progress. Use it at your own risk
- UI Model definitions for "projet1-aui_30" and "projet1-cui_30"
- Window definition for "window_component_0" with width="200" and height="150"
- GridBagBox definition for "grid_bag_box_1" with gridHeight="17" and gridWidth="20"
- Constraint definition for "constraint gridx="10" gridy="2" gridwidth="6" gridheight="1" weightx="1.0" weighty="1.0" fill="both" insets="0,0,0,0"
- InputText definition for "input_text_component_2" with name="input_text_component_2", isVisible="true", isEnabled="true", textColor="#000000", maxLength="50", and numberOfColumns="15"
- Constraint definition for "constraint gridx="2" gridy="2" gridwidth="4" gridheight="1" weightx="1.0" weighty="1.0" fill="both" insets="0,0,0,0"
- OutputText definition for "output_text_component_4" with name="output_text_component_4", content="/uiModel/resourceModel/cioRef[@cioId='output_text_component_4']", defaultContent="point de départ", isVisible="true", isEnabled="true", fgColor="#000000", isBold="true", and textColor="#000000"

Figure 4.32. La spécification concrète de l'interface après l'adaptation

Ensuite, le système à base de cas utilise ces caractéristiques extraites afin de comparer la réaction de l'IHM du nouveau cas à celle du cas antérieur (cf. figure 4.33).

⁶ Une description complète de Usixml est disponible à <http://www.usixml.org>

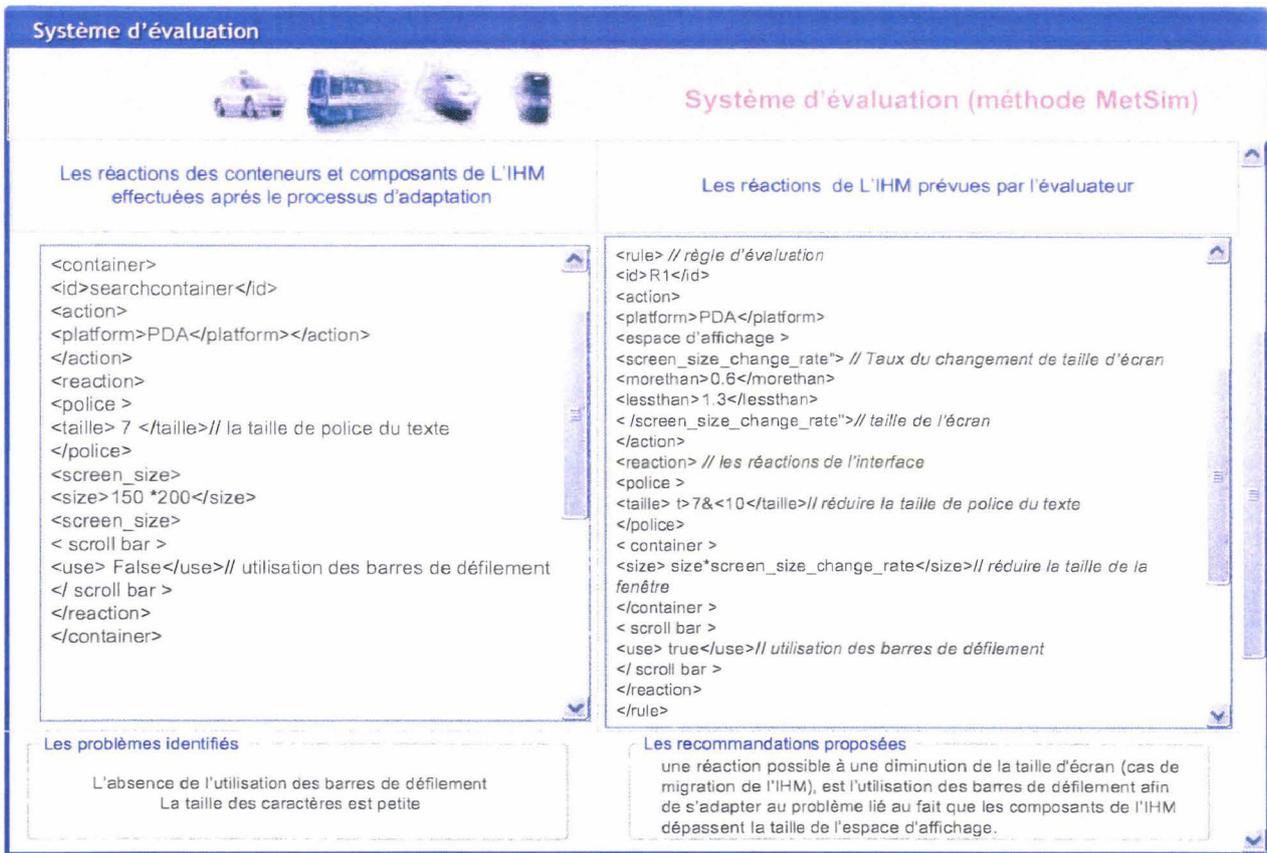


Figure 4.33. Les réactions réelles (partie gauche) de l'interface du nouveau cas et les réactions du cas similaire antérieur (partie droite)

En se remémorant des expériences passées voisines (similaires), l'évaluateur extrait le problème du cas le plus pertinent à des fins de réutilisation ; elle devient un problème proposé. Elle subit un processus de révision afin de devenir compatible avec le cas actuel. Les résultats d'évaluation peuvent être communiqués sous forme de fiche problème (cf. Tableau 4.11).

Tableau 4.11. Description des problèmes détectés

Contexte d'usage	Description du problème	Conséquence possible
Utilisation d'un PDA Utilisateur pressé	La taille des caractères est petite. Les informations affichées par l'interface ne sont pas claires (peu visibles).	En utilisant l'interface affichée par le PDA, l'utilisateur peut rencontrer des difficultés à interpréter les informations fournies, les caractères étant trop petits.

Enfin, l'évaluateur propose des recommandations pour les problèmes détectés. Des fiches contenant des descriptions textuelles des recommandations seront préparées par l'évaluateur (cf. Tableau 4.12).

Tableau 4.12. Exemple de description d'une recommandation

Contexte d'usage	Description des recommandations
Utilisation d'un PDA Utilisateur pressé	une réaction possible à une diminution de la taille d'écran (cas de migration de l'IHM), est l'utilisation des barres de défilement afin de s'adapter au problème lié au fait que les composants de l'IHM dépassent la taille de l'espace d'affichage.

4.5.2.1. Second exemple d'évaluation

Le contexte choisi dans cette évaluation est le suivant : un usager des transports est en train de courir : il est en retard et doit attraper sa correspondance. En plus, le système est utilisé dans un environnement fermé (par exemple à l'intérieur de la gare) où le niveau de luminosité peut être relativement faible. L'utilisateur interagit avec le système en utilisant un PDA.

Suite au changement au niveau du contexte, la vitesse de déplacement de l'utilisateur en train de courir peut nécessiter une adaptation de l'interface pour en faciliter la lecture. La luminosité ambiante peut aussi entraîner un changement au niveau des couleurs utilisées pour l'interface comme, par exemple, un passage des caractères en rouge sur fond noir pour faciliter la vision nocturne de l'utilisateur, etc. (cf. Figure 3.34).



Figure 4.34. Exemple d'adaptation d'une application à la vitesse de déplacement d'un utilisateur : déplacement normal, luminosité normale (a) et déplacement rapide, luminosité faible (b)

En se basant sur les spécifications concrètes de l'interface avant et après l'adaptation, le système identifie la réaction de l'IHM effectuée après le processus d'adaptation (cf. Figure 4.35).

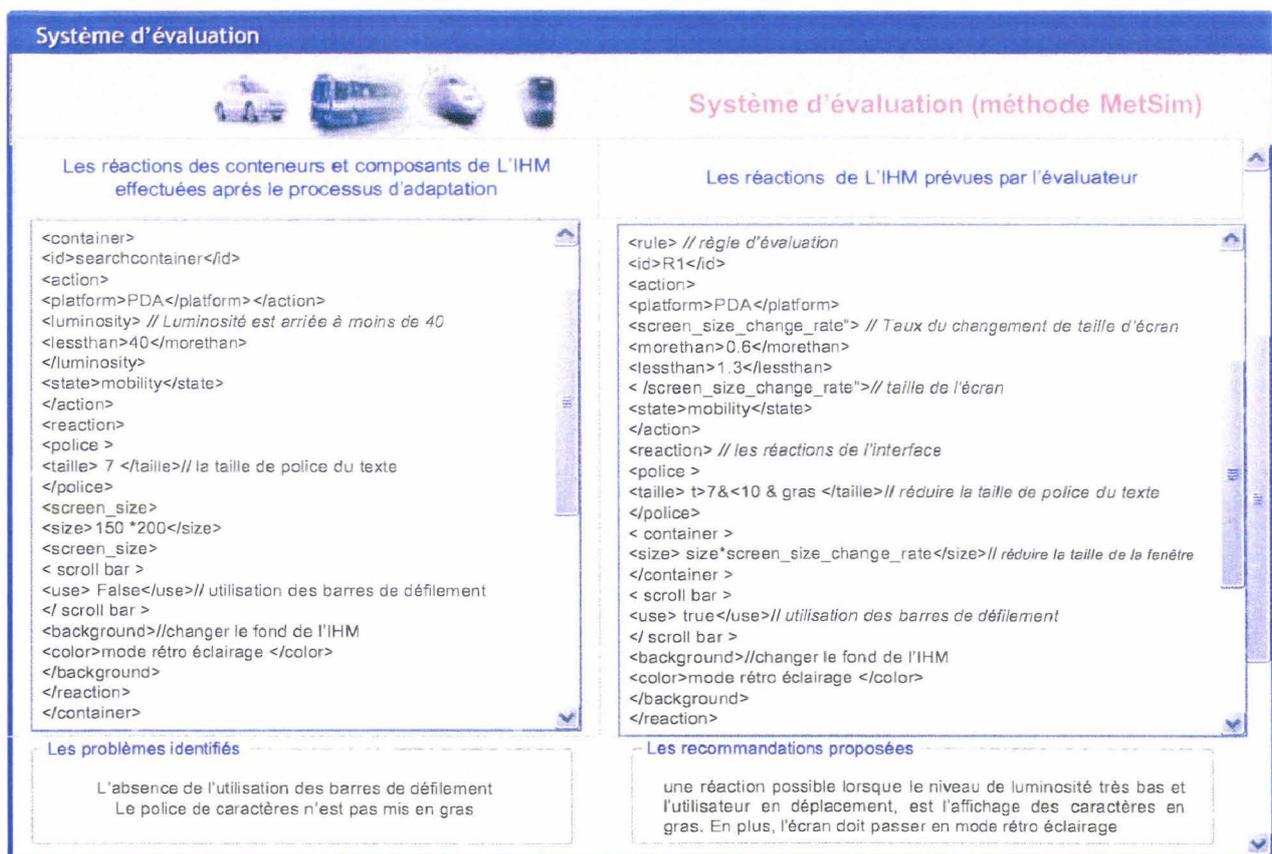


Figure 4.35. Les réactions réelles (partie à gauche) de l'interface et les réactions prévues par l'évaluateur (partie à droite)

Le changement du contexte a entraîné quelques modifications au niveau de la facette de présentation de l'interface. Parmi ces réactions, on peut citer le passage de l'écran en mode rétro éclairage, affichage avec une taille d'écran réduite (diminution de la taille d'écran).

Après la comparaison entre la réaction d'IHM du nouveau cas à celle du cas antérieur similaire, l'évaluateur a pu détecter les problèmes suivants : le style de police des caractères n'est pas mis en gras, il y a l'absence de barres de défilement permettant à l'utilisateur de visualiser facilement les informations affichées sur l'écran de PDA (cf. Tableau 4.13).

Tableau 4.13. Description des problèmes détectés

Contexte d'usage	Description de problème	Conséquence possible
- Niveau de luminosité faible - Vitesse de déplacement de l'utilisateur élevée	La visualisation des informations affichées par l'interface n'est pas claire.	Mauvaise lisibilité des informations. L'utilisateur peut éprouver des difficultés à lire les informations.

Pour remédier à ces problèmes, des recommandations sont proposées par l'évaluateur. Elles sont décrites dans le tableau 4.14.

Tableau 4.14. Exemple de description des recommandations

Contexte d'usage	Description des recommandations
<ul style="list-style-type: none"> - Niveau de luminosité faible - Vitesse de déplacement de l'utilisateur élevée 	Une réaction possible à une baisse du niveau de luminosité et à la vitesse de déplacement de l'utilisateur est de mettre les caractères en gras, avec agrandissement de la dimension des lettres, et plus grand espacement entre les mots, laisser un espacement entre les lignes, réduire les longueurs des lignes, et aussi l'utiliser des barres de défilement si nécessaire.

4.5.3. Synthèse et discussion

Nous venons illustrer succinctement les principes de fonctionnement de la méthode MetSim dans deux cas d'étude. A travers les différents modèles utilisés, elle permet un prototypage interactif d'une IHM adaptée en fonction d'un scénario d'usage.

Cette méthode de prototypage interactif ne s'appuyant pas sur la réalisation d'un prototype fonctionnel de l'IHM, elle peut être utilisée pour effectuer des tests prospectifs, étudier de nouveaux modes d'interaction induits par une future technologie, et envisager différents environnements d'utilisation du système. Cette méthode peut ainsi être intégrée au sein du cycle de conception d'un système.

L'intérêt des simulations est de pouvoir considérer différents scénarios en changeant le contexte d'usage (Plate-forme, Utilisateur, et Environnement). Dans MetSim l'IHM est considérée comme un ensemble de composants de présentation. Cette IHM est présentée en UsiXML. Grâce au langage à balises utilisé pour décrire l'abstraction de l'IHM, il est possible d'accéder à cette description et de changer le contexte d'usage. Il est par exemple intéressant de simuler l'appropriation de l'interface par l'utilisateur, de tester un ensemble d'hypothèses fonctionnelles sur cette interface dans des contextes différents et de procéder à des analyses statistiques. Etant donné que nous avons testé notre approche seulement dans le cadre limité d'une application de recherche des itinéraires personnalisés dans le domaine du transport, il serait aussi intéressant de généraliser l'approche à d'autres domaines d'application.

Conclusion

Ce chapitre de mise en oeuvre des méthodes d'évaluation proposées dans le cadre du transport, a été scindé en quatre parties.

La première a été consacrée à la présentation du projet MOUVER.PERSO, auquel nous avons contribué, et qui nous a fourni de nombreux éléments de réflexion. Puis nous avons décrit des éléments de contextes d'usage envisagés dans le cadre de l'évaluation des systèmes d'information personnalisés.

Une application de la première méthode proposée dans le cadre de notre travail (MetQus) a fait l'objet de la deuxième partie. L'objectif de cette expérience était d'évaluer le démonstrateur MOUVER.PERSO. Nous nous sommes focalisés sur une tâche représentative de l'application : la recherche d'itinéraire après l'ajout d'un rendez-vous pour un étudiant. Vingt sujets ont participé à l'expérimentation.

Dans la troisième partie, à partir d'une étude de cas, nous avons appliqué notre méthode MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage) sur un système de diffusion d'informations au profit des usagers des transports collectifs. L'étude de ce système a suivi aussi les différentes étapes de notre méthode.

La quatrième partie a présenté l'application de notre méthode MetSim étape par étape sur un système de recherche d'itinéraire personnalisé. S'appuyant sur les spécifications concrètes de l'interface avant et après l'adaptation et les règles d'évaluation, l'évaluateur peut identifier les problèmes rencontrés.

Pour conclure, il est important de rappeler que le développement des SIP, nécessite de disposer de méthodes d'évaluation de la qualité de personnalisation de ces systèmes. Ce mémoire a apporté de premiers éléments de solution et a concerné trois propositions de méthodes d'évaluation assurant l'intégration de recommandations visant à garantir leur capacité de personnalisation.

Ces méthodes sont pour l'instant dans une version qu'on pourrait qualifier de préliminaire et demandant des études et développements complémentaires.

Les perspectives de recherche à ce sujet font l'objet du dernier chapitre.

Chapitre 5

Perspectives de recherche

Sommaire

Introduction	136
5.1. Perspectives relatives à la méthode MetQus	137
5.2. Perspectives relatives à la méthode MetTra	139
5.3. Perspectives relatives à la méthode MetSim	144
Conclusion	145

Introduction

Dans les chapitres précédents de ce mémoire, après nous être intéressé à la personnalisation dans les systèmes d'information avoir dressé un état de l'art des méthodes d'évaluation des systèmes interactifs et des systèmes personnalisés. Puis, trois démarches d'évaluation proposés et développés ont été présentés. De premières applications ont aussi expliquées. Bien que ces méthodes d'évaluation permettent de fournir quelques fonctionnalités originales, ils présentent encore un ensemble de défauts et inconvénients. Dans ce chapitre, nous déterminons plusieurs pistes pour des recherches futures, visant à perfectionner les approches d'évaluations proposées, augmenter leur automatisation et les intégrer dans un environnement global.

5.1. Perspectives relatives à la méthode MetQus

Rappelons que l'objectif de la méthode MetQus (METHode d'évaluation basée sur le QUestionnaire de Satisfaction) est de recueillir, à l'aide de documents en principe structurés, un ensemble d'appréciations et d'opinions, de l'utilisateur, et de sa satisfaction après qu'il ait interagi avec le système. Bien que cette technique possède l'avantage d'être facile à mettre en place, elle présente encore un ensemble de limites. Comme perspectives concernant cette méthode, quelques futurs travaux sont proposés (cf. Figure 5.1) :

- Pour ancrer plus l'évaluation au contexte d'usage, il serait possible de proposer un questionnaire en ligne pour juger la qualité de personnalisation du système tout en interagissant avec celui-ci. Par exemple, les utilisateurs seraient invités à répondre à un ensemble des questions ou à indiquer, pour chaque critère, s'ils sont en désaccord ou en accord sur une échelle en 5, 7 ou 10 points.
- L'efficacité de l'aide apportée par les critères proposés dans le cadre de notre travail pour l'évaluation des SIP peut être améliorée en enrichissant le contenu de chaque critère c-à-d. proposer des sous-critères associés à chaque critère. Par exemple, le critère adaptation du système à la plate-forme interactive peut être subdivisé en deux sous-critères : l'adaptation du système à la plate-forme logicielle et l'adaptation du système à la plate-forme matérielle. En plus, Pour l'instant, les associations entre les critères d'évaluation de SIP et les événements d'interaction de l'utilisateur ne sont pas encore formalisées. En conséquence, l'activité d'analyse de données fournit seulement à l'évaluateur des indications pour l'aider à décrire les problèmes identifiés, à détecter les tâches concerné par ces problèmes et suggérer des améliorations au concepteur (calcul du niveau de satisfaction de l'utilisateur, degrés de personnalisation du SIP, etc.). Il s'agirait donc, de proposer un outil logiciel d'aide à l'évaluation dans un but de détection de problèmes et les tâches que les concerne et de critiquer du SIP. L'automatisation de la tâche d'analyse présente plusieurs avantages. En effet, l'automatisation de l'évaluation permet de :
 - Réduire le coût de l'évaluation : les méthodes qui automatisent l'analyse, ou la critique (proposition de commentaires et d'alternatives) peuvent réduire le temps de l'évaluation et par conséquent le coût ;

- Réduire le besoin de recours à des experts en évaluation : automatiser l'activité d'analyse ou de critique, apporte en principe une aide première aux concepteurs qui n'ont pas d'expérience approfondie en évaluation ;
- Faciliter la détection des erreurs : en se basant sur la spécification de l'interface, il devient possible de détecter automatiquement des incohérences et des erreurs, au niveau de l'interface à évaluer.

L'architecture de la méthode MetQus étendue est illustrée en figure 5.1.

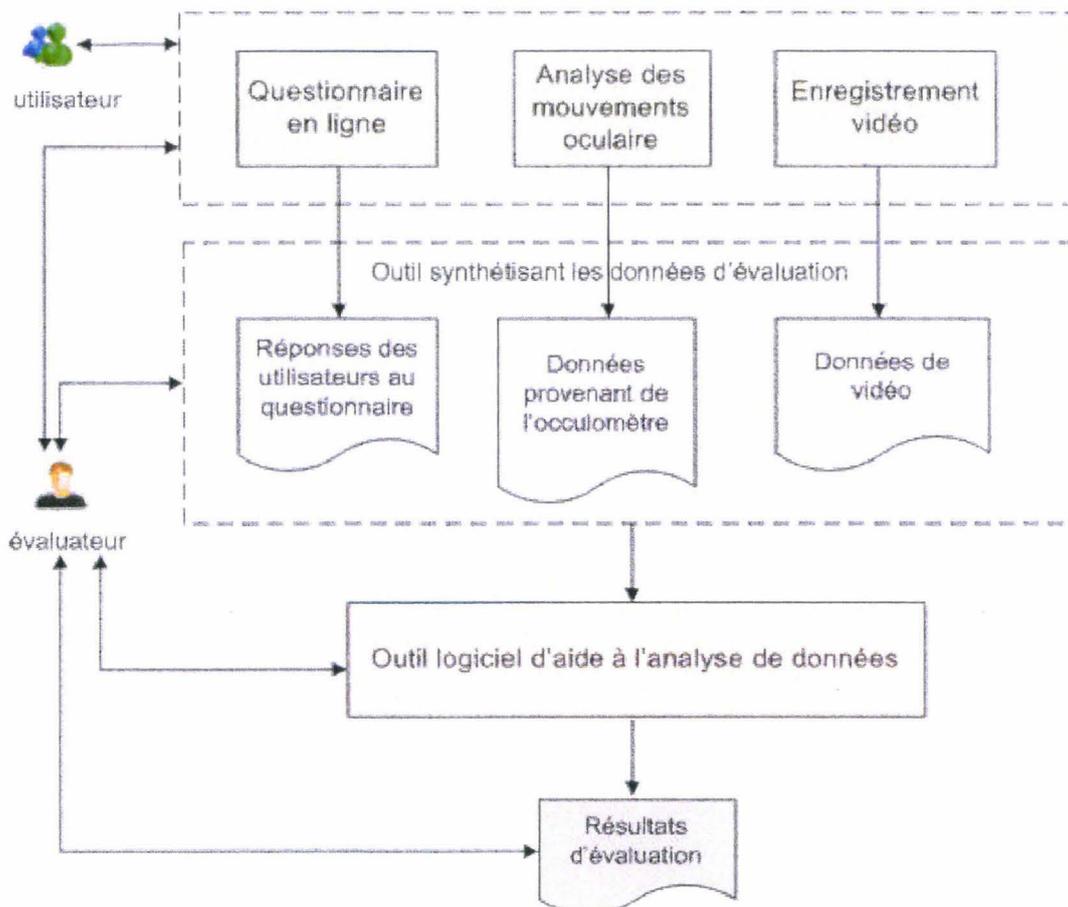


Figure 5.1. Architecture de la méthode MetQus étendue

5.2. Perspectives relatives à la méthode MetTra

Rappelons ici que la méthode MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage) est basée sur l'exploitation d'un système de trace qui doit être intégré durant la phase de conception du SIP pour générer le modèle réel de l'utilisateur dans un contexte bien défini. En effet, ce modèle permet à l'évaluateur de détecter un premier ensemble représentatif de problèmes de personnalisation inhérents à l'exploitation d'un SIP. En termes de perspectives concernant la méthode MetTra, plusieurs travaux pourraient être envisagés. En effet, cette méthode possède encore un ensemble de limitations relatives à la phase d'identification de problèmes. L'amélioration de cette activité pourrait faciliter à l'évaluateur la réalisation de ses tâches et augmenter l'automatisation de la méthode MetTra. Pour cela, nous proposons un outil d'identification automatique de problèmes. L'outil proposé

serait par exemple composé de trois modules correspondant aux trois catégories d'évaluation : l'évaluation statique, multi-contextuelle, et dynamique.

- L'évaluation statique consiste à évaluer l'interface dans un contexte bien défini. Cette évaluation repose sur les règles d'évaluation définies par l'évaluateur pour chaque critère, de même que sur les attributs d'évaluation collectés par le système de traces (le nombre de services traités par l'interface, le nombre des objets interactifs utilisés par service, etc.). En fait, l'évaluateur vérifie le respect ou non de ces règles. Soulignons que ces règles sont proposées par l'évaluateur en se basant sur son expérience ou des connaissances issues de la littérature. Pour faciliter son travail, nous proposons un premier module qui permettrait d'ajouter et/ou de mettre à jour la base de règles (selon l'évolution rapide des technologies en IHM et les nouveaux résultats de recherche dans le domaine de personnalisation). En conséquence, il nous faudrait aussi proposer une description formelle de ces règles afin qu'elles puissent être utilisées le plus automatiquement possible par l'outil d'identification automatique de problèmes proposé.
- Rappelons aussi que dans l'évaluation multi-contextuelle (cf. Figure 4.4.2), les modèles de dialogue relatifs à l'utilisation de l'interface par l'utilisateur dans les différents contextes sont confrontés pour montrer la différence d'adaptation de l'interface. Pour l'instant, la fonction de confrontation se base sur la visualisation des modèles de dialogues relatifs aux différents contextes. Dans l'avenir, il faudrait améliorer cette fonction. Pour cela, nous proposons un nouveau module permettant d'aider l'évaluateur à analyser automatiquement ces modèles de dialogue générés par le système de trace. Cette amélioration a pour objectif de détecter pour chaque adaptation les transformations effectuées suite au changement contextuel (services fournis par l'interface, la structure de l'interface en termes d'objets interactifs après le processus d'adaptation, etc.). Ces informations aident l'évaluateur à vérifier l'adaptation de l'interface aux différents contextes d'usages. Une telle amélioration faciliterait à l'évaluateur la réalisation de ses tâches. La figure 5.2 illustre notre vision de ce futur module.

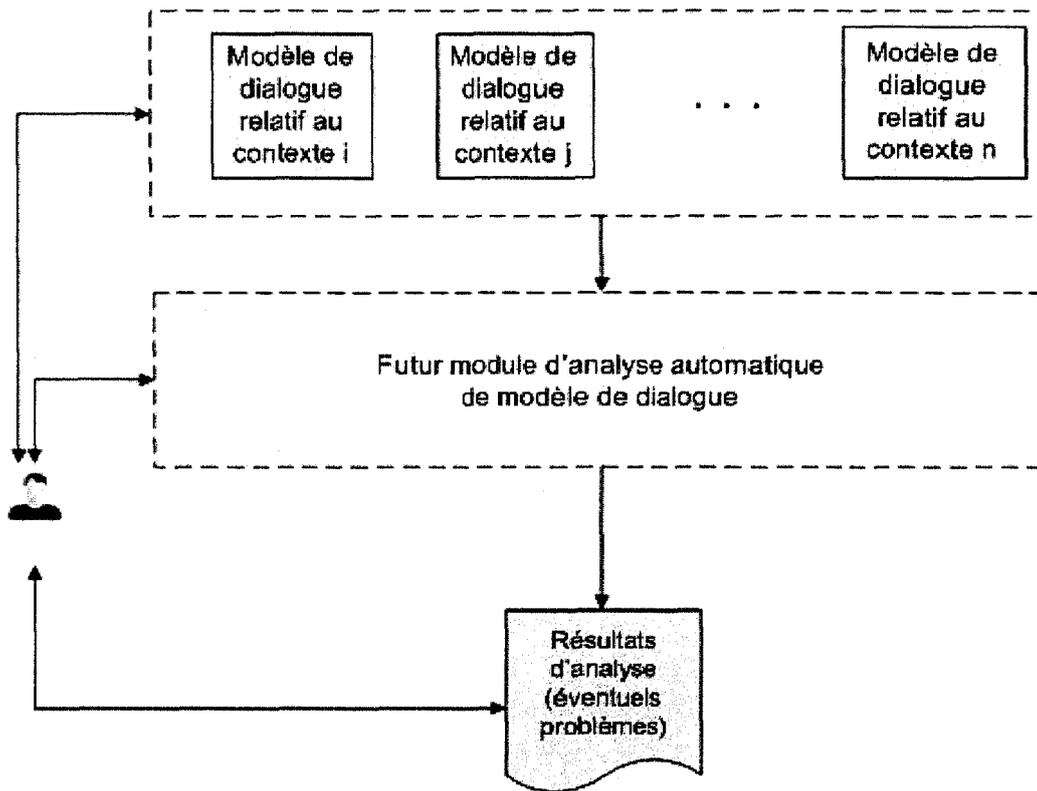


Figure 5.2. Fonctionnement du futur module d'analyse automatique de modèle de dialogue

- Concernant, l'évaluation dynamique (ou de contenu), elle consiste à évaluer la capacité d'apprentissage du système en se basant sur le coefficient de corrélation de Pearson et le calcul de taux d'erreur moyen du SIP (cf. 3.3.1.5.). Elle concerne principalement l'évaluation de la personnalisation liée au contenu et s'effectue pendant l'exécution (les critères d'évaluation ne sont pas connus dès la conception). Elle permet de savoir si la personnalisation proposée par le système est capable de prendre en compte les préférences de l'utilisateur telles que son niveau professionnel, ses domaines d'intérêts, etc. Dans l'avenir, il faudrait améliorer cette fonction. Pour cela, nous proposons un nouveau module d'évaluation à base d'agents logiciels permettant d'évaluer les solutions proposées par le système de personnalisation. L'idée du module proposé est d'associer chaque préférence ou critère à un agent *Critère*. Ainsi, chaque agent coopère dont le but d'estimer la qualité de personnalisation des solutions proposées par le SIP. Dans la perspective de fournir le meilleur compromis pour les différents critères, nous proposons un processus de négociation entre les agents *Critère*.

Le module proposé serait composé de trois catégories d'agents aux rôles différents et aux compétences correspondantes :

- les agents *Interface* s'exécutent sur les plates-formes d'interaction des utilisateurs. Un agent de ce type est capable de reconnaître la plate-forme à partir de laquelle l'utilisateur s'est connecté et par laquelle les informations doivent s'afficher. Ces agents permettent aussi de comprendre les requêtes de chaque utilisateur et de les aider à exprimer leurs demandes, leurs besoins.
- les agents *Critère* créent et mettent à jour les informations que le système possède sur les utilisateurs afin d'assurer l'évaluation des résultats.
- Enfin, les agents *d'évaluation* (ou superviseur) sont dotés de compétences nécessaires à l'évaluation de la qualité de personnalisation lié au contenu.

Dans ce qui suit nous expliquons principalement le comportement des agents *Critère* et des agents évaluateurs.

Comportement de l'agent Critère

Le diagramme de la figure 5.3 définit le modèle comportemental de l'agent critère. Durant la première étape, l'agent *Critère* affecte des notes e_{ij} aux différents solutions s_j suivant le critère C_i par la formule :

$$e_{ij} = \text{note}(C_i) = \frac{1}{\text{rang}(C_i)} * \delta$$

Puis, l'agent *Critère* détermine la note d'évaluation minimale e_{mini} et la note d'évaluation maximale e_{maxi} des solutions proposées par le système tels que $e_{\text{mini}} = \text{Min}(e_{ij})$ et $e_{\text{maxi}} = \text{Max}(e_{ij})$ avec $i = \{1, \dots, n\}$, $j = \{1, \dots, k\}$ (n étant le nombre de critères et k le nombre de solutions possibles pour la requête R).

Durant la deuxième étape, e_{mini} et e_{maxi} sont utilisés par l'agent *Critère* pour calculer son niveau de satisfaction ns_j . Ainsi le but de chaque agent est de s'éloigner de son e_{mini} pour se rapprocher le plus possible de son idéal e_{maxi} et ce par le biais de ce ns_j défini par : $ns_j = e_{\text{mini}} + (e_{\text{maxi}} - e_{\text{mini}}) * \varepsilon$ où $\varepsilon = \frac{1}{\text{moy}(e_i)}$ est un paramètre d'ajustement fixé dans $]0,1[$. Notons que e_i est la note du critère i pour la solution choisie par l'utilisateur u .

Ensuite l'agent *Critère* partage l'ensemble des solutions E en un ensemble des solutions préférées E_{Pref} et un ensemble de solutions moins préférés E_{moisp} tels que : $E_{\text{Pref}} = \{s \in E / e_{ij} \geq ns_j\}$ et $E_{\text{moisp}} = \{s \in E / e_{ij} < ns_j\}$

Par la suite, l'agent *Critère* qui possède le poids le plus supérieur w_{ui} ou l'agent d'évaluation envoie E_{moisp} initial à ses accointances et leur demande d'établir leur vote pour chacune des solutions de E_{moisp} . Une fois ce message reçu, chaque agent *Critère* se charge d'examiner l'ensemble des solutions de E_{moisp} , si une solution figure dans son ensemble de solution moins préférés, il vote pour sa suppression de E . Puis, il envoie la réponse à l'agent demandeur.

Lorsque toutes les réponses sont reçues, l'agent d'évaluation sera également chargé de calculer pour chaque solution de E_{moisp} le paramètre O_j représentant la somme de poids des critères qui sont pour l'élimination de la solution j et le paramètre N_j représentant la somme de poids des critères qui sont contre l'élimination de la solution j tels que $O_j = \sum w_i$ et $N_j = \sum w_i$. Lorsque O_j est supérieur à N_j , la solution j sera automatiquement supprimée de l'ensemble des solutions E (cf. Figure 5.4).

En revanche, il se pourrait que E devienne vide c-à-d que le niveau de satisfaction ns_j de la dernière itération trop élevé, alors l'agent *Critère* effectue un processus de relaxation à ce niveau en appliquant la formule : $ns_j = ns_j - (e_{\text{maxi}} - ns_j) * \frac{\varepsilon}{2}$.

Suite aux décisions prises par les différents agents *Critère* pour la suppression des solutions les moins préférés, si cardinal (E) > 1, l'agent superviseur demande aux

accointances d'augmenter leur niveau de satisfaction en appliquant la formule : $ns_i = e_{\min i} + (e_{\max i} - e_{\min i}) * 2\varepsilon$ et de démarrer un nouveau processus de négociation.

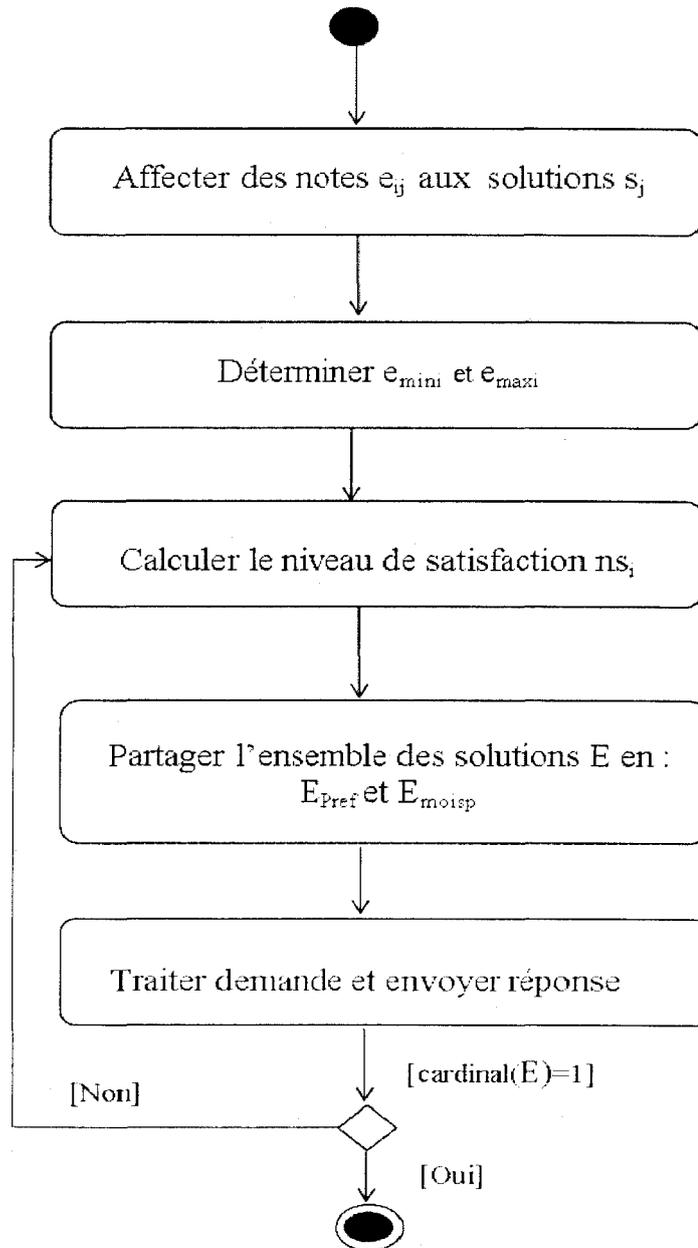


Figure 5.3. Comportement de l'agent *Critère*

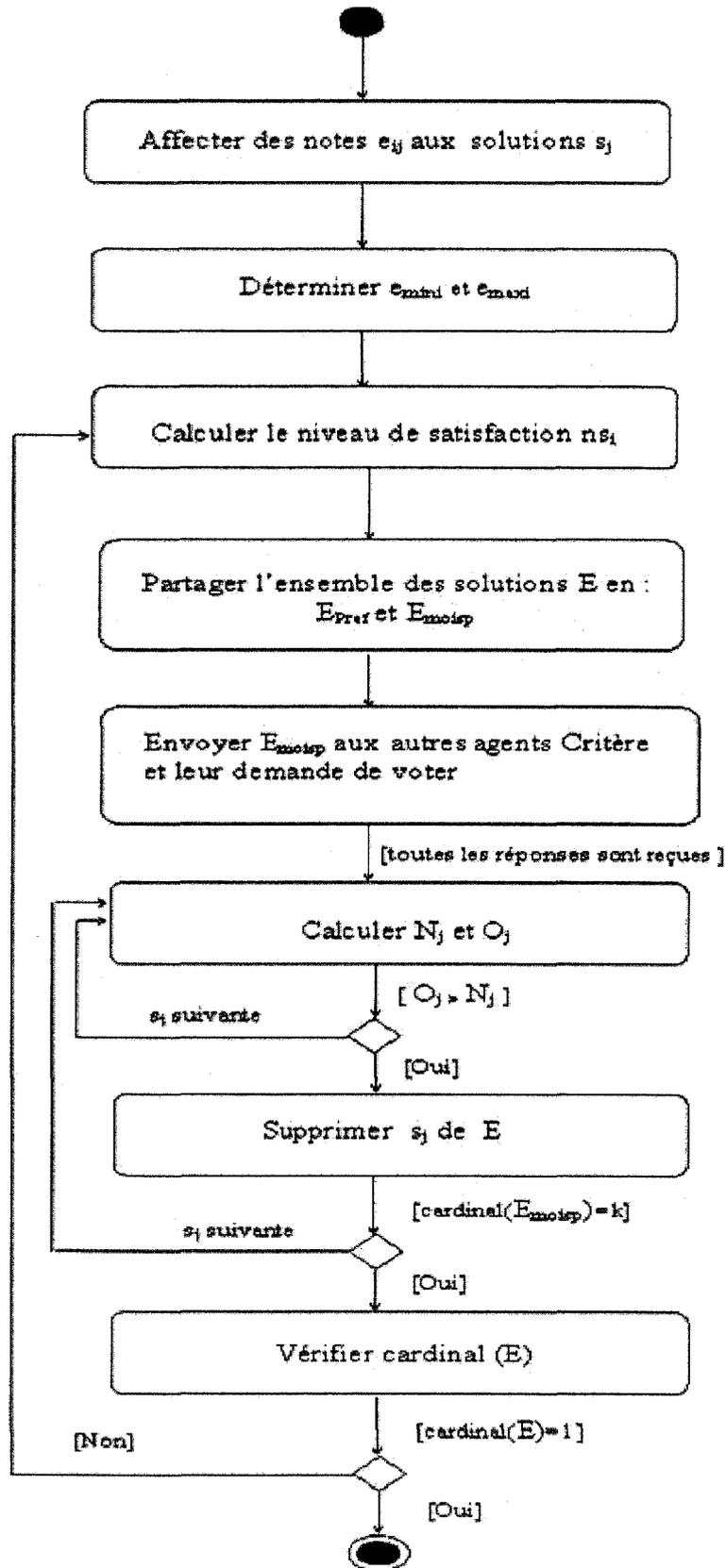


Figure 5.4. Comportement de l'agent *Evalueateur*

5.3. Perspectives relatives à la méthode MetSim

Comme perspectives concernant la méthode (METHode d'évaluation par SIMulation), quelques futurs travaux sont proposés :

- Pour l'instant, nous avons appliqué la méthode MetSim pour évaluer une IHM décrite en UsiXML. Dans l'avenir, il faudrait rendre notre méthode plus générique. Pour cela, nous proposons un outil permettant de récupérer les paramètres des éléments de l'interface (polices, couleurs, tailles, positions, etc.) à travers la description de l'interface du système interactif à évaluer qui pourrait être développé à l'aide de différents langages et environnements de programmation (UIDL - User Interface Description Language) comme UIML (User Interface Markup Language), UsiXML (USer Interface eXtensible Markup Language), XiML (eXtensible Interface Markup Language), XUL (XML-based User interface Language), etc. Cet outil serait capable de fournir des informations relatives aux parties composant une interface (par exemple, des labels, des boîtes de texte, des boîtes de liste, etc.), au style de présentation, au contenu et aux comportements de chaque composant (par exemple, aux position, couleur et police d'un label, au contenu d'un boîte de texte, à la couleur de fond la fenêtre, aux événements susceptibles de se produire sur chaque composant, etc.).
- Actuellement, dans notre méthode, la tâche d'identification des problèmes se limite à un rôle de fournisseur des indications nécessaires à l'évaluateur pour l'aider à interpréter les résultats d'analyse afin de critiquer le système et proposer des améliorations au concepteur. Dans l'avenir, il faudrait formaliser cette tâche le plus possible. Cette amélioration servirait à augmenter l'automatisation de la méthode MetSim. Pour cela, nous proposons un nouveau module permettant d'identifier des problèmes et effectuer de telles critiques. Ensuite, l'évaluateur pourrait modifier et/ou ajouter ses propres critiques si nécessaire. Enfin, ce nouveau module proposé permettrait de générer un document imprimable décrivant le résultat d'évaluation.
- Dans l'avenir, il serait aussi intéressant de proposer des supports de travail communs, visant à améliorer à la fois la collaboration et la représentation des données des intervenants de projet. La représentation des données joue un rôle important au niveau de la collaboration et un langage commun entre les acteurs de projet pourrait contribuer à renforcer les échanges. Ces supports communs pourront enrichir les aspects collaboratifs durant les projets et précisément dans l'évaluation des SIP. Ces solutions de modélisation sont issues de l'exploitation des techniques de modélisation du GL et de l'IHM qui offrent un potentiel considérable en termes de modélisation [Kolski 97], [Jambon et al., 01], [Morley et al., 07], [Bernonville et al., 06].

Enfin, des perspectives de recherche pourraient s'appuyer sur des études de cas plus complexes que ceux traités dans le chapitre IV (ex. système de diffusion d'information personnalisée, etc.). Ceci nous permettrait de continuer à tester les méthodes proposées dans le cadre de notre travail.

Conclusion

Dans ce chapitre, nous avons proposé différentes perspectives de recherche nécessitant un ensemble de travaux à effectuer à l'avenir. En effet, pour étendre MetQus, un environnement d'évaluation intégré pourrait être proposé : il permettrait de combiner des résultats d'évaluation provenant des différentes méthodes complémentaires (questionnaire en ligne, de vidéo et d'oculométrie, etc.). En outre, MetTra possède encore un ensemble de limitations relatives à la phase d'identification de problèmes. Pour cela, nous avons proposé un outil d'identification automatique de problèmes. L'outil proposé serait par exemple composé de trois modules correspondant aux trois catégories d'évaluation (l'évaluation statique, l'évaluation multi-contextuelle et l'évaluation dynamique). MetSim pourrait lui-même être amélioré en proposant un outil permettant de récupérer les paramètres des éléments de l'interface à évaluer et des supports de travail communs, visant à améliorer à la fois la collaboration et la représentation des données des intervenants de projet.

Conclusion générale

Les travaux de recherche présentés dans ce mémoire se situent dans la thématique de la personnalisation dans les systèmes d'information. Ils contribuent à l'évaluation des systèmes d'information personnalisés. Cette contribution comporte deux principaux aspects : l'analyse bibliographique et la critique globale de l'existant, et la proposition de trois méthodes complémentaires d'évaluation permettant d'évaluer et d'améliorer la qualité de personnalisation de SIP. Ces méthodes ont pour but de s'assurer qu'un SIP fournit à l'utilisateur une information pertinente via une interface adaptée, l'évaluation étant basée sur le contexte d'usage. Ainsi, différents contextes d'usage sont envisagés pour le SIP ; ces contextes devant être considérés afin d'évaluer ces systèmes.

Après une introduction des principaux termes rencontrés dans la littérature autour de la personnalisation, nous avons étudié sans souci d'exhaustivité mais plutôt de représentativité, des approches et méthodes existantes pour la personnalisation. Trois processus sont récurrents dans une activité de personnalisation : le processus de collecte d'information sur l'utilisateur pour connaître ses préférences et ses besoins, le processus de modélisation de ces préférences (modèle utilisateur) et le processus d'exploitation du modèle utilisateur pour la personnalisation (méthode de personnalisation). Différentes approches sont fournies dans la littérature pour réaliser chacun de ces processus. Dans ce premier chapitre, nous avons aussi présenté différents systèmes de personnalisation existants étudiés sous l'angle de l'évaluation.

Le deuxième chapitre a concerné l'évaluation dans les systèmes interactifs tout en ayant en visée celle des systèmes d'information personnalisés. Ce chapitre a illustré d'abord le principe de l'évaluation ainsi que les critères d'évaluation existants relativement aux systèmes interactifs. Ensuite, nous avons présenté une étude de modèles de développement enrichis sous l'angle de l'évaluation des SIP. Nous avons aussi présenté une étude des méthodes d'évaluation existantes en fonction des critères de personnalisation que nous proposons pour l'évaluation de SIP. En effet, pour présenter ces méthodes, nous avons retenu la classification de [Huart, Kolski et Bastien 08] qui s'inspire de [Whitefield 91]. Celle-ci classe les méthodes d'évaluation en plusieurs types de méthodes, selon que l'utilisateur et le système sont réels ou représentés. Cette étude nous a permis de montrer qu'aujourd'hui, il n'existait pas de méthode ou d'outils permettant d'évaluer les SIP. Nous avons constaté que les méthodes étudiées se focalisent sur l'utilité et l'utilisabilité du système. Par contre, la notion de personnalisation est généralement négligée, voire bâclée. Il devient donc indispensable de disposer de démarches rigoureuses permettant d'évaluer ces systèmes.

Ce constat a conduit au troisième chapitre qui a proposé trois démarches complémentaires d'évaluation. En effet, ces méthodes peuvent prendre en compte les spécificités des SIP et fournir aussi des solutions originales par rapport aux approches traditionnelles d'évaluation. Nous avons présenté dans ce mémoire trois contributions tentant d'y apporter des solutions :

- La première contribution a concerné une méthode, intitulée MetQus (METHode d'évaluation basée sur le QUestionnaire de Satisfaction). L'objectif de cette technique est de recueillir un ensemble d'appréciations, d'opinions et d'attitudes, de l'utilisateur après qu'il ait interagi avec un système d'information personnalisé. Elle est basée sur l'intervention explicite de l'utilisateur et l'existence d'un système réel (maquette, système final). Après la collecte d'opinions et d'attitudes, de l'utilisateur, cette méthode est capable d'effectuer des analyses comme des statistiques, des calculs liés à différentes mesures concernant la satisfaction de l'utilisateur et d'afficher les résultats d'analyse sous une forme textuelle ou graphique.

- La deuxième contribution a pris la forme d'une méthode d'évaluation précoce appelée MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage). Partant du principe que la meilleure façon de détecter les problèmes liés à un système d'information personnalisé est d'observer l'utilisateur réel lorsqu'il réalise sa tâche, le principe de MetTra consiste à prévoir l'évaluation dès les premières étapes de développement d'une application personnalisée en intégrant un système de trace. Celui-ci produit des traces « orientées usage » qui fournissent des informations exploitables lors de la phase d'évaluation. Ensuite, des modèles réels relatifs à l'utilisation de l'interface par l'utilisateur dans différents contextes sont générés pour étudier le SIP. Ils fournissent des indications utiles à l'évaluateur afin d'effectuer des analyses des événements d'interaction capturés, de critiquer le système et proposer des améliorations au concepteur.

- La troisième contribution correspond à une méthode d'évaluation par simulation basée sur un système à base de cas, intitulée MetSim (METHode d'évaluation par SIMulation). Elle consiste à évaluer une IHM personnalisée en faisant varier des hypothèses d'usages liés à des contextes différents. L'intérêt des simulations est de pouvoir considérer différents scénarios en changeant le contexte d'usage (Plate-forme, Utilisateur, et environnement). Le principe de l'évaluation par simulation est donc de tester l'application dans des contextes d'usage différents. Elle fournit à l'évaluateur les indications nécessaires pour l'aider à critiquer l'IHM et proposer des améliorations au concepteur. Elle aide aussi à représenter les problèmes détectés lors des évaluations et à reformuler des recommandations qui en résultent.

Le quatrième chapitre a permis de présenter les résultats issus des applications des trois méthodes d'évaluation proposées. Nous avons commencé par présenter le projet MOUVER.PERSO que nous avons eu la chance d'intégrer pendant la durée de notre thèse, et auquel nous avons contribué ; ce projet nous a fourni de nombreux éléments de réflexion⁷.

Puis, nous avons exposé un premier cas d'étude en suivant les étapes de la première méthode proposée dans le cadre de notre travail. L'objectif de cette expérience était d'évaluer le démonstrateur MOUVER.PERSO en se basant sur la méthode MetQus ainsi que sur les critères que nous avons définis pour l'évaluation d'un SIP. Nous nous sommes focalisés sur une tâche représentative de l'application : la recherche d'itinéraire après l'ajout d'un rendez-vous.

Ensuite, nous avons étudié, à travers un deuxième cas d'étude, comment, à l'aide d'une méthode d'évaluation, qualifiée de précoce (c'est-à-dire utilisée lors des premiers étapes d'un cycle de développement logiciel), basée sur les traces d'usage, il est possible d'évaluer un SIP dans le domaine du transport de manière semi-automatique. Dans ce but, une expérimentation a été réalisée au sein de notre laboratoire LAMIH avec différents sujets. Pendant cette expérimentation, MetTra a été appliquée pour évaluer un prototype d'un SIP dans le domaine du transport. Le service évalué concerne la recherche d'itinéraire. La particularité de ce service se situe au niveau de la fourniture des résultats qui sont personnalisés par rapport aux préférences de l'utilisateur.

Enfin, dans le quatrième chapitre, nous avons décrit quelques exemples d'évaluation impliquant la méthode MetSim. Cette méthode a été appliquée pour évaluer un système

⁷ Le projet ANR VIATIC.MOBILITE portant également sur l'information voyageur a constitué aussi pour nous une source de réflexion importante. Ce projet coordonné par l'INRETS-ESTAS (G.Uster) a impliqué 5 laboratoires de recherche et 10 entreprises.

permettant d'afficher les prochains départs dans une gare ou une station, accessibles à travers une interface web. Ceci nous a permis de montrer que si MetSim apportait de nombreuses solutions intéressantes pour un grand nombre de problématiques d'évaluation des SIP elle était loin d'être finalisée et que de nombreux travaux doivent encore être menés pour la rendre générique et surtout pour lui permettre de prendre en compte toutes les dimensions de personnalisation.

L'application des méthodes proposées sur des cas d'étude différents nous a permis de valider les approches proposées dans leur globalité surtout d'en appréhender certaines limites et contraintes.

Enfin, dans le dernier chapitre, quelques perspectives de recherche ont été envisagées. Ainsi, pour étendre MetQus, un environnement d'évaluation intégré pourrait être proposé : il permettrait de combiner des résultats d'évaluation provenant de différentes méthodes complémentaires (questionnaire en ligne, de vidéo et d'oculométrie, etc.). En outre, MetTra possède encore un ensemble de limitations relatives à la phase d'identification de problèmes. Pour cela, un outil d'identification automatique de problèmes a été proposé. Cet outil serait par exemple composé de trois modules correspondant aux trois catégories d'évaluation (l'évaluation statique, l'évaluation multi-contextuelle et l'évaluation dynamique). MetSim pourrait lui-même être amélioré en proposant un outil permettant de récupérer les paramètres des éléments de l'interface à évaluer et des supports de travail communs, visant à améliorer à la fois la collaboration et la représentation des données des intervenants de projet.

Un long chemin reste encore à parcourir dans le domaine de l'évaluation des systèmes d'information personnalisés, pour parvenir à une batterie de méthodes complémentaires qui pourrait être mise à la disposition des équipes de conception et d'évaluation.

Bibliographie générale

- [Aas 97] Aas K. A Survey on Personalised Information Filtering Systems for the World Wide Web; Research Report n° 922, Oslo, Norway, Norwegian Computing Center, 1997.
- [Abed et al., 98] Abed M., Ezzedine H. Vers une démarche intégrée de conception-évaluation des systèmes Homme-Machine, *Journal of Decision Systems*, volume 7, p. 147-175, 1998.
- [Abowd et al., 99] Abowd G.D., Dey A.K., Brown P.J., Smith M., Steggles P. Towards a Better Understanding of Context and Context-Awareness, *Lecture Note In Computer Science*, Vol. 1707. Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, pp. 304-307, 1999.
- [Aggarwal et al., 99] Aggarwal C.C., Wolf J.L., Wu K-L. et Yu P.S. Horting Hatches an Egg : A New Graph-Theoretic Approach to Collaborative Filtering, In *Proceedings of KDD'99, 5th International Conference on Knowledge Discovery and Data Mining*, pp. 201-212, San Diego, USA, 1999.
- [Albrecht et Zukerman 07] Albrecht D. and Zukerman I., Introduction to the special issue on statistical and probabilistic methods for user modeling, *User Modeling and User-Adapted Interaction*, pp. 1-215, 2007.
- [Allen 79] Allen R.B. Mental Models and User Models, In Helander M., Landauer T.K. et Prabhu P., (EDS) *Handbook of Human-Computer Interaction*. Elsevier Science B.V., 1997.
- [Ambrosini et al., 97] Ambrosini L., Cirillo V. and Micarelli A. A Hybrid Architecture for User- Adapted Information Filtering on the World Wide Web. In Jameson, A., Paris, C. and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, New York: Springer Wien, pp. 59-61, 1997.
- [Anderson 93] Anderson, J.R. *Rules of the Mind*. Lawrence Erlbaum Associates, Hilldale, NJ. 1993.
- [Ang 99] Ang C.L. Enactment of IDEF0 Modes, *International Journal of Production Research*, 37(15), pp. 3383-3397, 1999.
- [Anli 05a] Anli A, Grislin-Le Strugeon, E. and Abed, M., Une plate-forme de personnalisation basée sur une architecture multi-agents. *Revue des Nouvelles Technologies de l'Information (RNTI-E)*, 5, pp. 95-100, 2005.
- [Anli 05b] Anli A., Grislin-Le Strugeon E., Abed M. A Generic Personalization Tool based on a Multi-agent Architecture. In C. Stephanidis (Ed.), *Proceedings HCI International (Las Vegas, Nevada, July 22-27, 2005)*, Volume 7 - Universal Access in HCI: Exploring New Interaction Environments, Lawrence Erlbaum Associates, Mahwah, New Jersey, pp. 1-8, 2005.
- [Anli 05c] Anli A., Kolski C., Abed M. Principes et architecture pour la personnalisation d'information en interaction homme-machine, application à l'information transport. In *Proceedings of IHM 2005, International Conference Proceedings Series*, ACM Press, Toulouse, pp. 123-130, 2005.

[Anli 06d] Anli A. Méthodologie de développement des systèmes d'information personnalisés - Application à un système d'information au service des usagers des transports terrestres de personnes. Thèse de Doctorat, Université de Valenciennes et du Hainaut-Cambrésis, Valenciennes, 2006.

[Anli 06e] Anli A., Abed M. PerSyst : Un Système de Personnalisation de l'information transport multimodale. E-STA (Sciences et Technologies pour l'Automatique), 3, pp. 1-6, 2006.

[Anli et Abed 05] Anli A., Abed M. PerSyst : Un Système de Personnalisation générique, Application à l'information voyageur, Workshop Méthodologies et Heuristiques pour l'Optimisation des Systèmes Industriels (MHOSI'05). Hammamet, Tunisie, 2005.

[APP 92] Apple Computer Inc. Macintosh human interface guidelines, Reading, MA: Addison Wesley, 1992.

[Apple 92] Apple Computer Inc., Macintosh human interface guidelines. Reading, MA: Addison Wesley, 1992.

[Baccino et al., 05] Baccino T., Bellino C., Colombi T. Mesure de l'utilisabilité des interfaces. TIC et Sciences Cognitives. Hermes. ISBN : 2-7462-1026-6, 2005.

[Baraglia et Silvestri 07] Baraglia R. Silvestri F, Dynamic Personalization of Web Sites Without User Intervention. Communications of the ACM, vol. 50, n° 2, pp. 63-67, 2007.

[Baron et al., 06] Baron, M., Lucquiaud, V., Autard. D. et Scapi D., K-MADE : un environnement pour le noyau du modèle de description de l'activité., 18ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'2006), ACM Press, Montréal, 2006, pp. 287-288.

[Barthet 94] Barthet M.-F., Liberati V., Ponamale M., Ergoval. Outil d'aide à l'évaluation ergonomique des logiciels", Actes du congrès Ergo-IA'94, Bayonne, France, pp. 482-490. 1994.

[Bastien et Scapin 02] Bastien J.M.C., Scapin D.L. Les méthodes ergonomiques : de l'analyse à la conception et à l'évaluation, in : Ergonomie et Informatique Avancée, Ergo-IA'2002, A. Drouin, J. Robert (Eds), I.D.L.S., Biarritz, 8-10 2002.

[Bayardo et al., 97] Bayardo Jr. R. J., Bohrer W., Brice R., Cichocki A., Fowler J., Helal A., Kashyap V., Ksiezyk T., Martin G., Nodine M., Rashid M., Rusinkiewicz M., Shea R., Unnikrishnan C., Unruh A. and Woelk D. InfoSleuth: agent-based semantic integration of information in open and dynamic environments, In Proceedings of the 1997 ACM SIGMOD international conference on Management of data, pp.195-206, May 11-15, Tucson, Arizona, United States, 1997.

[Bazsalicza et Naïm 01] Bazsalicza M. et Naïm P. Data mining pour le web : Profiling – Filtrage collaboratif – Personnalisation client. Paris : Editions Eyrolles, 2001.

[Bernonville et al., 06] Bernonville S., Leroy N., Kolski C., Beuscart-Zéphir M., Association des réseaux de Petri et des critères d'ergonomie des logiciels pour la modélisation et la réingénierie de systèmes interactifs, cas de la prescription thérapeutique en milieu hospitalier. ErgoIA 2006, "L'humain comme facteur de performance des systèmes complexes" (Biarritz, France, 11-13 Octobre 2006), ESTIA & ESTIA.Innovation, Biarritz, pp. 55-62, 2006.

- [Berrut et Denos 03] Berrut C., Denos N. Filtrage collaboratif. In Gaussier E., Stéfanini M.H., Assistance intelligente à la recherche d'informations, Hermès-Lavoisier, 2003, pp. 241-269, 2003.
- [Billsus et al., 99] Billsus D. and Pazzani M. A Hybrid User Model for News Story Classification. In J. Kay (Ed.), *User Modeling : Proceedings of the 7th International Conference (UM'99 – Banff, Canada, June 20-24)*, pp. 99–108, Springer-Verlag, 1999.
- [Bisseret et al., 99] Bisseret A., Sebillotte S., Falzon P., Techniques pratiques pour l'étude des activités expertes, Octares édition, 1999.
- [Boidin 05] Boidin E. Apprentissage automatique pour la personnalisation: Application aux transports terrestres de personnes. Rapport de DEA. Université de Valenciennes et du Hainaut-Cambrésis, 2005.
- [Booch et al., 98] Booch, G., Rumbaugh, J. and Jacobsen, I. Unified Modeling Language User Guide. Addison Wesley, 1998.
- [Bradley 00] K. Bradley, R. Rafter, B. Smyth, Case-Based User Profiling for Content Personalisation, Lecture Notes in Computer Science, Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pp. 62–72, Volume 1892/2000, 2000.
- [Breese et al., 98] Breese J.S. Heckerman D. and Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering, In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Madison, Morgan Kaufmann Publisher. pp. 43-52 1998.
- [Breese et al., 98] Breese J.S., Heckerman D. et Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Madison, Morgan Kaufmann Publisher, 1998.
- [Brossard 08] Brossard A. PERCOMOM : une méthode de modélisation des applications interactives personnalisées appliquée à l'information voyageur dans le domaine des transports collectifs. Mémoire de Doctorat, Université de Valenciennes et du Hainaut-Cambrésis, Valenciennes, 2008.
- [Buisse 99] Buisse, A., Vers une démarche industrielle pour le développement d'interfaces Hommes-Machine. Thèse en Informatique, Université de Rouen, 1999.
- [Byrne et al., 98] Byrne, M. D., Anderson, J. R. Perception and Action. In J. R. Anderson & C. Lebiere (Eds.), *Atomic Components of Thought*, pp. 167-200, d'édition, 1998.
- [Calvary 05] Calvary G., Dâassi O., Coutaz J., Demeure A., Des Widgets aux Comets pour la Plasticité des Systèmes Interactifs. Revue d'Interaction Homme Machine, Europa, Vol 6 N°1, pp. 33-53, 2005.
- [Calvary et al., 03a] Calvary G., Coutaz J., Daassi O., Balme L., Demeure A., Towards a new Generation of widgets for supporting software plasticity: the comet, In Proceedings of 9th IFIP Working Conference on Engineering for Human-Computer Interaction, Hambourg, Allemagne, 2003.
- [Calvary et al., 05b] Calvary G., Dâassi O., Coutaz J. and Demeure A. Des Widgets aux Comets pour la Plasticité des Systèmes Interactifs. Revue d'Interaction Homme-Machine (RIHM), 6(1), pp. 33–53, 2005.

- [Candillier 01] Candillier L., Apprentissage automatique de profils de lecteurs. Mémoire de DEA, Lille 3 – Laboratoire d'Informatique Fondamentale de Lille, Equipe GRAPPA, 2001.
- [Card 83] Card S., Moran T., Newell A., The psychology of human-computer interaction, Lawrence Erlbaum Associates Publishers, London, 1983.
- [Chi et al., 00a] Chi, E., Pirolli, P., Pitkow, J. The scent of a site : a system for analysing and predicting information scent, usage, and usability of a web site, Proceedings of the conference on Human Factors in computers systems, pp. 161-168, ACM Press, 2000.
- [Chi et al., 03b] Chi Hi, Rosien A., Supattanasiri G., Williams A., Royer C., Chow C., Robles E., Dalal B., Chen J., Cousins S. The Bloodhound Project: Automating Discovery of Web, Usability Issues using the InfoScent Simulator. Accessible in citeseer.ist.psu.edu/655522.html, 2003.
- [Chi et al., 88] Chin J.P., Diehl V.A., Norman K.L. Development of an instrument measuring user satisfaction of the human-computer interface, Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems, ACM Press, pp. 213-218, 1988.
- [Cinquin et al., 02] Cinquin L., Lalande P. A et Moreau N. Le projet ECRM : Relation client et Internet. Editions Eyrolle, Paris, 2002.
- [Coutaz, 90] Coutaz. J. Interface homme-ordinateur, conception et réalisation. Dunod, Paris, 1990.
- [Coyle et Cunningham 02] Coyle L. et Cunningham P., A Case-Based Personal Travel Assistant for Elaborating User Requirements and Assessing Offers, Proceedings of the 6th European Conference, ECCBR 2002, S. Craw, A. Preece (Eds.). LNAI Vol. 2416, pp. 505-518, Springer-Verlag, 2002.
- [Cram et al., 07] Cram D., Jouvin D., Mille A., Visualizing Interaction Traces to improve Reflexivity in Synchronous Collaborative e-Learning Activities. 6th European Conference on e-Learning, Academic Conferences Limited ed. Copenhaguen, pp. 147-158. ISBN 978-1 905305-57, 2007.
- [Diaper et Stanton, 03] Diaper, D., Stanton, N.A. (Eds.), The handbook of task analysis for human-computer interaction. Mahwah, New Jersey: Lawrence Erlbaum Associates, 2003.
- [Diaz et al., 02] Diaz, M. (Ed.). Réseaux de Pétri, Modèles Avancés. Hermes, Paris, 2002.
- [Drug 90] Drury C.G., Methods for direct observation of performance, In : Evaluation of Human works : a practical ergonomics methodology, Wilson J.R. and Corlett E.N. (Eds.), Taylor & Francis, 1990.
- [Eason 84] Eason K. D. Towards the Experimental Study of Usability, Behaviour and Information Technology, 3, pp. 133-143, 1984.
- [Enrique et al., 07] Enrique F., Sherry Y., Robert D. Macredie. L. The role of human factors in stereotyping behaviour and perception of digital library users: a robust clustering approach, User Modeling and User-Adapted Interaction, 17, pp. 305-337, 2007.
- [Ezzedine et Trabelsi, 05] Ezzedine H., Trabelsi. A. From the design to the evaluation of an agent based human-machine interface, Application to supervision for urban transport system. In P. Borne, M. Benrejeb, N. Dangoumeau, L. Lorimier (Ed.), IMACS World Congress

"Scientific Computation, Applied Mathematics and Simulation" (July 11-15, Paris), ECL, pp. 717-725, 2005.

[Farenc 97] Farenc C. Ergoval : une méthode de structuration des règles ergonomiques permettant l'évaluation automatique d'interfaces graphiques, thèse de doctorat, Université de Toulouse 1, 1997.

[Fisher 96] Fisher D., Iterative Optimization and Simplification of Hierarchical Clusterings, Journal of Artificial Intelligence Research, n° 4, 1996, pp. 147-179.

[Fortes et al., 97] Fortes R., Nicoletti M.C., LiOS: a tool for supporting evaluation in the WWW, Hypertextes et Hypermédias H2PTM'97, Volume 1 n°2-3-4/1997, Hermès, Paris, 1997.

[Francisco et al., 08] Francisco M. Trindade F.M. Pimenta M.S, RenderXML – A Multi-plateforme Software Development Tool, In 7th International Conference on Computer-Aided Design of User Interfaces (CADUI'2008), Albacete, Spania, 2008.

[Frasincar 02] Frasinca F., Houben G-J. Hypermedia Presentation Adaptation on the Semantic Web. In de Bra P., Brusilovsky P. & Conejo R., (Eds.), Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web- Based Systems, Malaga, Spain, LNCS 2347, pp. 133-142, 2002.

[Gamboa 98] Gamboa Rodríguez F., Spécification et implémentation d'ALACIE : Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques, Thèse de Doctorat, Paris XI, 1998.

[Ganneau et al., 08] Ganneau V., Demumieux, R. Calvary G., Demumieux R. EMMA : Modèle Utilisateur pour la Plasticité des Interfaces Homme Machine en Mobilité, Collecter les usages réels des clients de téléphonie mobile (un outil embarqué). Actes des quatrième journées francophones Mobilité et Ubiquité 2008, UBIMOB'08, Institut Universitaire de Technologie, Saint Malo, France, ACM International Conference Proceeding Series 277 ACM Press, 2008.

[Garzotto et Matesa, 97] Garzotto F., Matesa M., A systematic method for hypermedia usability inspection, The new review of hypermedia, Vol. 3, pp. 39-65, 1997.

[Gauch et al., 03] Gauch S., Chaffe J., Pretschner A., Ontology-Based User Profiles for Search and Browsing, To appear in J. User Modeling and User-Adapted Interaction, the Journal of Personalization Research, Special Issue on User Modeling for Web and Hypermedia Information Retrieval, 2003.

[Gena et al., 04] Gena C., Torre I., The Importance of Adaptivity to Provide On-Board Services. A Preliminary Evaluation of an Adaptive Tourist Information Service on Board Vehicles, Applied Artificial Intelligence, Special Issue on Mobile A.I, 2004.

[Goldberg et al., 01] Goldberg K., Roeder T., Gupta D. and Perkins C., Eigentaste : A Constant Time Collaborative Filtering Algorithm, Information Retrieval Journal, vol. 4(2), pp. 133-151, 2001.

[Gowen 03] Gowan JP M: A multiple model Approach to personalised information access Master of Master in computer science, faculty of science, University College Dublin, 2003

[Grislin et Kolski, 96] Grislin M., Kolski, C. Evaluation des interfaces homme-machine lors

du développement de système interactif, *Technique et Science Informatiques (TSI)*, 15 (3), p. 265-296, 1996.

[Grislin-Le Strugeon et al., 07] Grislin-Le Strugeon E., anli A., Petit-Rozé C., Soui M., Conreur G., Dos Santos P., Abed M., Kolski C. MOUVER.PERSO, MObilité et mUltimodalité Voyageurs Etudiants, Système d'information multimodale personnalisée. Rapport intermédiaire (mai 2006) et rapport final de contrat de recherche PREDIT DTT 216-75-01, LAMIH, Valenciennes, 2007.

[Hammond 84] Hammond N., Hinton G., Barnard P., MacLean A., Long J. et Whitefield A. Evaluating the interface of a document processor: a comparison of expert judgement and user observation, *Proceedings of the First IFIP Conference on Human-Computer Interaction: Interact'84*, London, UK, pp. 725-729, 1984.

[Hammond et al., 84] Hammond N., Hinton G.E., Barnard P., Long J., Whitefield. A. Evaluating the interface of a document processor: A comparison of expert judgement and user observation. In *Human-Comuter Interaction, Interact'84*, B.Schackel (Ed.), Elsevier Science Publishers B.V. : North-Holland, IFIP, 725-729, 1984.

[Hariri et al., 08] Hariri M-A., Tabary D., Lepreux S., Kolski C. Context aware Business adaptation toward User Interface adaptation, *Communications of SIWN*, 3, pp. 46-52, 2008.

[Hartson et al., 89] Hartson H.R., Hix D. Human computer interface development: concepts and systems for its management. *ACM : computer surveys*, vol 21(1), pp. 5-92, 1989.

[Hawarah 06] Hawarah L., Simonet A. and Simonet M.: Évaluation d'une approche probabiliste pour le classement d'objets incomplètement connus dans un arbre de décision, Troisième atelier sur la "Fouille de données complexes dans un processus d'extraction de connaissances 2006.

[Hayes & Cunningham 01] Hayes C., Cunningham P. Smart Radio—community based music radio. *Knowledge-Based Systems*, n° 14, pp. 197-201, 2001.

[Hayes & Cunningham 04] Hayes C. Cunningham P.. Context boosting collaborative recommendations. *Knowledge-Based Systems*, n° 17, pp. 131-138, 2004.

[Henze et Nejd1 99] Henze, N. et Nejd1, W. Adaptivity in the KBS Hyperbook System., In 2nd Workshop on Adaptive Systems and User Modeling on the WWW, Toronto, Canada, 1999.

[Herlocker 00] Herlocker, J.; Konstan, J.; and Reidl, J. 2000. Explaining collaborative filtering recommendations, In *ACM Conference on Computer Supported Cooperative Work*, 241–250, 2000.

[Herlocker et al., 99] Herlocker J. L, Konstant J. A., Brochers A. and Riedl J. A algorithmic framework for performring collaborative filtering. In *proc. 1999 Conf. Research and Developpement in Information retrieval*, pp. 230-237, Berkeley, CA, Août 1999.

[Hinz et Fiala, 2004] Hinz, M., Fiala, Z., AMACONT: A System Architecture for Adaptive Multimedia Web Applications, In *Proceedings of the Berliner XML Tage 2004*, Berlin, Allemagne, pp. 65-74, 2004.

[Hoyer et Czogolla 02] Hoyer R. et Czogolla O. Approach to personalised information services to public transport. In *9th world congress on Intelligent transportation systems*, Chicago Illinois, october 14-18, 2002.

- [Huart, Kolski et Bastien 08] Huart J., Kolski C., Bastien C. (2008). L'évaluation de documents multimédias, état de l'art. In Merviel S. (Ed.), *Objectiver l'humain ? Volume 1, Qualification, quantification*, Hermes, Paris, pp. 211-250, ISBN 978-2-7462-1553-5, 2008.
- [Hulzebosch et Jameson, 96] Hulzebosch R., Jameson A., FACE: A Rapid Method for Evaluation of User Interfaces. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry*, pp. 195-204, London, 1996.
- [Jennings, 93] Jennings A. et Higuchi H. A User Model Neural Network for a Personal News Service. *User Modeling and User-Adapted Interaction*, 3(1): pp 1-25, 1993.
- [Jermann et al., 2001] Jermann P., Soller A., and Muehlenbrock M., From mirroring to guiding: A review state of the art technology for supporting collaborative learning, In *Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, 2001.
- [Jiang et al., 1992] Jiang J., Murphy E.D., S.C. Bailin, W.F. Truszkowski, M. Szczur. Prototyping a Knowledge based compliance checker for User-Interface. Evaluation on Motif development environments: Second Annual International Motif Users Meeting, Bethesda, MD: Open Systems, pp. 258-268, 1992.
- [John 95] John B.E., Packer H., Learning and using the cognitive walkthrough method: a case study approach, *Proceedings of CHI'95 (May 7-11, Denver)*, ACM, Denver, pp.429-436, 1995.
- [John et Kieras, 96] John, D.E. Kieras. Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction*, vol. 3, pp. 287-319, 1996.
- [Johnson et al., 93] Johnson, P., Wilson, S., Markopoulos, P., Pycock, J. ADEPT-Advanced Design Environment for Prototyping with Task Models. In *Proceeding, InterCHI'93*, ACM Press, Amsterdam, The Netherlands pp. 56-57, April. 1993.
- [Just et al., 92] Just, M. A., Carpenter, P.A. A capacity theory of comprehension: Individual differences in working memory. *Psychologie Revue*. 99 (1), pp. 122-149. 1992.
- [Kahn et al., 94] Kahn, J., M., Prail A., Formal Usability Inspection. In Nielsen, J., & Mack R., L., *Usability inspection method*, pp. 141-171. John Willey & son, 1994.
- [Kay, 01] User Modeling for Adaptation. In Stephanidis C. (ed.), *User Interface for All*. LEA, publishers London, pp. 271-294, 2001.
- [Kieras 95] Kieras D.E., Wood, S.D., Abotel, K., Hornof, A. GLEAN: A Computer-Based Tool for rapid GOMS Model Usability Evaluation of User Interface Designs. In *Proceedings of UIST, Pittsburgh, New-York*, ACM Press, pp. 91-100. 1995.
- [Kieras 97] Kieras, D. E. A guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction (2nd ed., pp.733-766)*, 1997.
- [Kirakowski 88] Kirakowski J., Corbett M. Measuring user satisfaction, *People and Computer IV, Proceedings of the Fourth Conference of the British Computer Society, Human-Computer Interaction Specialists Group*, Manchester, UK, pp. 329-338, 1988.

- [Kitajima et al., 96] Kitajima, M., Polson, P. G. A comprehension-based model of exploration, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground. M. J. Tauber (Ed.), pp.324-331, ACM Press. 1996.
- [Kolski et Millot 91] Kolski C., Millot P. A rule-based approach to the ergonomic "static" evaluation of man-machine graphic interface in industrial processes. *International Journal of Man-Machine Studies*, 35, pp. 657-674, 1991.
- [Kolski, 2001] Kolski C. Analyse et conception de l'IHM, Interaction Homme-Machine pour les Systèmes d'Information, vol. 1. Hermès, Paris, ISBN 2-7462-0239-5.
- [Kolski, 97] Kolski C. Interfaces Homme-Machine : Application aux systèmes industriels complexes. Hermès, vol 2. Hermès, Paris, ISBN 2-86601-578-9, 1997.
- [Korfhage 97] Korfhage R.R. Information storage and retrieval. Wiley Computer Publishing, ISBN 0-471-14-338-3, 1997.
- [Krulwitch et al., 97] Krulwich B., Burkey C. the InfoFinder agent : Learning User Interests Through Heuristic Phase Extraction. *IEEE Expert: Intelligent Systems and Their Applications*, 12(5), pp 22-25, 1997.
- [Kurki et al., 1999] Kurki, T., Jokela, S., Sulonen, R., and Turpeinen, M., Agents in delivering personalized content based on semantic metadata, In Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace, pp. 84-93, Stanford, USA, 1999.
- [Lajmi 09] Lajmi S., Ghedira C. , Ghedira K. CBR Method for Web Service Composition, , Lecture Notes in Computer Science 4879, pp. 314-326, Springer, ISBN 978-3-642-01349-2, 2009.
- [LaLomia et al., 90] LaLomia M.J., Sidowski J.B., Measurements of computer satisfaction, literacy, and aptitudes: A review, *International Journal of Human-Computer Interaction*, vol. 2, pp. 231-253, 1990.
- [Lam et Xie 02] Lam S. H. et Xie F. Provision of Personalised Transit Travel Information System and Architecture. In *9th world congress on Intelligent transportation systems*, Chicago Illinois, october 14-18, 2002.
- [Le Bodic et al., 03a] Le Bodic, L., De Loor, P., Kahn, J. Multimodal interfaces evaluation with virtual reality simulation, In Jacko, J. et Stephanidis, C., *Human Computer Interaction, Theory and practice (Part 1) - Proceeding of the HCI International Conference*, 1, 661 {665. Lawrence Erlbaum Associates, 2003.
- [Le Bodic et al., 04b] Le Bodic, L., De Loor, P., Calvet, G., et Kahn, J. SIHMM : Simulateur de l'Interaction Homme Machine Multimodale. In Nigay, L., *XVI conference sur l'interaction Homme Machine, Annexes des actes*, pp. 31-34, Namur. ACM press, 2004.
- [Le Bodic et al., 04c] Le Bodic, L., Favier, P., Calvet, G. A model for naturalistic virtual user simulation, *Proceeding of the IADIS International conference CELDA 2004*, 467-470, Lisboa, 2004.
- [Le Bodic et al., 05d] Le Bodic, L., De Loor, P., Kahn, J.. Umar : a modeling of multimodal artefact. Jacko, J. et Stephanidis, C., *Proceeding of HCI International 2005*, A paraitre, Las Vegas. Lawrence Erlbaum Associates, 2005.

- [Lebodic 05e] Lebodic L. Approche de l'évaluation des systèmes interactifs multimodaux par simulation comportementale située, Thèse de doctorat, Université de Bretagne Occidentale, juin, 2005.
- [Lecerof et Paterno 98] Lecerof A., Paterno F. Automatic support for usability evaluation. *IEEE Trans. on Software Engineering*, 24 (10), 1998.
- [Lekakos and Giaglis 07] Lekakos G. and Giaglis G.M. A hybrid approach for improving predictive accuracy of collaborative filtering algorithms, *User Modeling and User-Adapted Interaction*, 0924-1868 (Print) 1573-1391, Pages1-215,2007,
- [Lepreux et al., 03] Lepreux S., Abed M., Kolski, C., A human-centred methodology applied to decision support system design and evaluation in a railway network context, *Cognition, Technology & Work*, vol. 5, pp.248–271, 2003.
- [Lewis et al., 1990] Lewis C.H., Polson P.G., Wharton C., Rieman J. Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces. In: Carrasco, Jane, Whiteside, John (ed.): *Proceedings of the ACM CHI 90 Human Factors in Computing Systems Conference*, Seattle, Washington, USA. pp.235-242, 1990.
- [Lieberman 95] Lieberman H. Letizia, An Agent That Assists Web browsing. In *International Joint Conference on Artificial Intelligence (IJCAI'95 – Montreal, Canada, August 20-25)*, San Francisco, USA, Morgan Kaufman Publishers, 1995.
- [Limbourg et al., 04] Limbourg Q., Vanderdonckt J., Michotte B., Bouillon L. and Lopez V. (2004). UsiXML: a Language Supporting Multi-Path Development of User Interfaces. *Proc. of 9th IFIP Working Conf. on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems 199 DSV-IS'2004*, Hamburg, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, vol. 3425, pp. 200–220, 2004.
- [Lowgren et Nordqvist, 92] J. Lowgren, T. Nordqvist. Knowledge-based evaluation as design support for graphical user interfaces. *Proceeding of ACM conference on Human Factors in Computing Systems, CHI'92*, Bauersfeld, (Eds.), Monterey, California, pp.181-188. May 3-7, 1992.
- [Lucquiaud 05] Lucquiaud V. Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateurs, *Proceedings of the 17th international conference on Francophone sur l'Interaction Homme-Machine*, pp. 83 – 90, 2005.
- [Lumineau 03] Lumineau N., Un tour d'horizon du filtrage collaboratif. Travail réalisé dans le cadre de l'AS Personnalisation de l'Information, janvier – décembre 2003, disponible à l'adresse:www.prism.uvsq.fr/recherche/themes/sial/cnrs/Fichiers/Rapport/TourDHorizonDuFiltrageCollaboratif-LIP6.pdf
- [Magoulas et al., 03] Magoulas G. D., Chen S. Y. and Papanikolaou K. A. Integrating Layered and Heuristic Evaluation for Adaptive Learning Environments, In: Weibelzahl, S. and Paramythis, A. (eds.), *Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems*, held at the 9th International Conference on User Modeling UM, Pittsburgh, pp. 5-14, 2003.
- [Mahfoudi et al., 01] Mahfoudi, A, Abed, M., Tabary D., From the formal specifications of users tasks to the automatic generation of the HCI specification. In Blanford A.,

- Vanderdonckt J., Gray, P. (Eds), People and computer XV – Interaction without Frontiers – Joint Proceeding of HCI 2001 and IHM 2001, pp 331-347. London: Springer, 2001.
- [Manber et al., 00] Manber U., Patel A. et Robison J. Exeprience with personalization on Yahoo! Communications of the ACM, 43(8), pp 35-39, 2000.
- [Mariage, 05] Mariage C. MetroWeb : logiciel de support à l'évaluation de la qualité ergonomique des sites web. Thèse de Doctorat, UCL, Louvain-la-Neuve, 2005.
- [MIC 92] Microsoft Corporation, The Windows interface: an application design guide. Redmon, WA: Microsoft Press, 1992.
- [Miettinen et Oulasvirta, 07] Miettinen M., Oulasvirta A., Predicting time-sharing in mobile interaction, Proceedings of User Modeling and User-Adapted Interaction, pp. 475–510, 2007.
- [Millot et Roussillon 91] Millot P., Roussillon E. Man-machine cooperation in telerobotics: Problematics and methodologies, In Second Symposium on Robotics, Gif-sur-Yvette, 1991.
- [Moha et al., 05] Moha N., Gaffar A., Seffah A. Enquête sur les pratiques de tests d'utilisabilité. IHM'05, 17ème Conférence Francophone sur l'Interaction Homme-Machine, pp. 115-122, September 27-30, Toulouse, France, 2005.
- [Molich 90] Molich R., Nielsen J. Improving a human-computer dialogue, Communications of the ACM, 33, p. 338-348, 1990.
- [Mostert et al., 89] Mostert D.N.J., Eloff J.H.P. et von Solms S.H., A methodology for measuring user satisfaction, Information Processing & Management, vol. 25, n° 5, p. 545-556, 1989.
- [Murgue 05] Murgue T. De l'importance du pré-traitement des données pour l'utilisation de l'inférence grammaticale en Web Usage Mining. In Workshop Modélisation Utilisateurs et Personnalisation de l'Interaction Homme-Machine, Extraction et Gestion de Connaissances (EGC 2005), Paris, 2005.
- [Newell 90] Newell, A. Unified theories of cognition. Harvard University Press, 1990.
- [Newman 95] Newman W., Analysis and evaluation of multimedia systems, in Multimedia Systems and Applications, Earnshaw R.A, Vince J.A. (Eds.), Academic Press Ltd, pp. 115-129, 1995.
- [Nielsen 90] Nielsen J., Molich R. Heuristic Evaluation of User Interfaces, Proceedings of CHI '90, p. 249-256, 1990.
- [Nielsen 93] Nielsen J., Usability engineering. San Diego, CA: Academic Press, 1993.
- [Nielsen et Mack 94] J. Nielsen. Heuristic evaluation. In J. Nielsen et R.L. Mack (Eds), Usability inspection methods, New York, John Wiley & Sons, pp. 25-62, 1994.
- [Nielsen et Molich, 90] Nielsen J., Molich. R. Heuristic evaluation of user interfaces, CHI'90 Conf. Proc., Seatle, ACM Press, New York, pp. 349-356, 1990.
- [Norsham et al., 09] Norsham I, Norazah Y, and Puteh. J. Advance. Soft Comput. Appl., Adaptive Course Sequencing for Personalization of Learning Path Using Neural Network Vol. 1, No. 1, July 2009 ISSN 2074-8523, 2009.

[O'Grady et al., 05] O'Grady M.J., O'Hare G.M.P. and Sas C. Mobile agents for mobile tourists: a user evaluation of Gulliver's Genie. *Interacting with Computers*, 17, pp. 343-366, 2005.

[O'Hare et O'Grady 03] O'Hare G.M.P. and O'Grady M.J. Gulliver's Genie: a multi-agent system for ubiquitous and intelligent content delivery. *Computer Communications*, 26, pp. 1177- 1187, 2003.

[Paterno et al., 1997] Paterno F., Mancini C., Meniconi S. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models, *Proceedings Interact'97*, pp. 362-369, Sydney, Chapman&Hall,1997.

[Paul et al., 94] Paul R., Neophytos I., Mitesh S., Peter B., John R. GroupLens : an Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*. Chapel Hill, North Carolina, 1994.

[Pazzani et al., 96] Pazzani, M., Maramatsu J., Billsus, D., Syskill de Webert : Identifying interesting Web sites, In *Proceeding of the fourteenth national conference on artificial intelligence* , AAAI Press, 1996

[Pennock et al., 00] Pennock D., Horvitz E., Lawrence S., Giles C. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI-2000*, Morgan Kaufman, San Francisco, pp. 473-480, 2000.

[Petit-Rozé et al., 03a] Petit-Rozé C., Anli A., Grislin-Le Strugeon E., Abed M., Kolski C. AGENPERSO : Interface homme-machine à base d'AGENTS Logiciels PERSONNELS d'information aux usagers des TC. Rapport final de projet PREDIT, LAMIH, Valenciennes, 2003.

[Petit-Rozé et al., 03b] Petit-Rozé C., Kolski C., Grislin-Le Strugeon E., Anli A., Abed M., Uster G. AgenPerso : Interface homme-machine à base d'agents logiciels personnels d'information au service des usagers des transports collectifs. 15ème Conférence Francophone sur l'Interaction Homme-Machine. Caen, ACM Press, 2003.

[Petit-Rozé et al., 04d] Petit-Rozé C., Anli A., Grislin-Le Strugeon E., Abed M., Uster G., Kolski C. Système d'information transport personnalisée à base d'agents logiciels. *Génie Logiciel*, 70, pp. 29-38.

[Plos et Buisine 06] Plos, O., Buisine, S. Universal design of mobile phones: A case study. *CHI'06 International Conference on Human Factors in Computing Systems (work-in-progress)*, pp. 1229-1234, ACM Press, 2006.

[Pollier 92] Pollier A., Évaluation d'une interface par des ergonomes: diagnostics et stratégies, *Le Travail Humain*, vol. 55, pp. 71-96, 1992.

[Polson et al., 1992] P.G. Polson, C. Lewis, J. Rieman, et C. Wharton. Cognitive walkthroughs: A method for theory- based evaluation of user interfaces. *International Journal of Man-Machine Studies* 36, pp. 741-773.

[Power et al., 09] Power C., Petrie H., Mitchel R., A Framework for Remote User Evaluation of Accessibility and Usability of Websites, *Universal Access in Human-Computer Interaction. Addressing Diversity*, Lecture Notes in Computer Science, Volume 5614/2009, pp. 594-601, 2009.

- [Pretschner et Gauch 99] Pretschner A., Gauch S., Personnalisation on the web, Technical report ITTC-FY 2000, TR-13591-01, Information and communication technology center, Departement of electrical engineering and computer science, University of Kansas, 1999.
- [Pretschner et Gauch 99] Pretschner A., Gauch. S. Ontology Based Personalized Search. In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 1999.
- [Ragg et al., 02] Ragg T., Menzel W., Baum W., and Wigbers M. Bayesian learning for sales rate prediction for thousands of retailers. *Neurocomputing*, 43(1- 4), 127–144, 2002.
- [Rao et Georgeff 91] Rao A.S. and Georgeff M.P. Modeling Rational Agents within a BDI Architecture. In Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91), pp. 473-484, 1991.
- [Ravden et Johnson, 89] Radven S.J., Johnson G.I. Evaluating usability of human-computer interfaces : a practical method. Ellis Horwood, Chicester, 1989.
- [Reiman et al., 91] Reiman, J. Davies S., Hair, C. Esemplare M., Polson E., and Lewis C. An automated cognitive walkthrough. In Robertson, S.E, Olsen, G.M., and Olsen, J.S. (Eds.), *Human Factors in Computing Systems, CHI 91 Conference Proceedings*. ACM Press, New York, 1991.
- [Reisner 81] Reisner P. Formal Grammar and Human Factors Design of an Interactive Graphics System, *IEEE Transactions on Software Engineering*, 7 (2), pp. 229-240, March 81.
- [Resnick et al., 94] Resnick P., Iacovou, Suchak M., Bergstrom P., Riedl J., *GroupLens : An Open Architecture for Collaborative Filtering of Netnews*. Proceedings of ACM Conference on Computer Supported Cooperative Work, pp. 175-186, 1994.
- [Resnick et al., .94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl, *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*, In proc. of ACM 1994 Conference on Computer Supported Cooperative Work, Chapel Hill, Caroline du Nord, 1994.
- [Roques 05] Roques P. *UML 2 par la pratique – Etudes de cas et exercices corrigés*. Paris : Eyrolles, 2005.
- [Rosenberg, 01] Rosenberg M., The personalization story. (Disponible sur: <http://www.itworld.com/Man/2676/ITW010511rosenberg/>), 2001.
- [Ross 77] Ross D.T. Structured Analysis (SA): A Language for Communicating Ideas. *IEEE Transactions on Software Engineering*, 3(1), pp. 16–34. 1977.
- [Salton 83] Salton G. and McGill M. *Introduction to modern information retrieval*. New York : McGraw-Hill, 1983.
- [Samaan et al., 02] Samaan K., Delotte O., Tarpin-Bernard F., Processus de génération d'IHM multicibles pour applications interactives. In Proceedings of the 14th French-Speaking Conference on Human-Computer interaction (Poitiers, France, November 26 - 29, 2002). M. Beaudouin-Lafon, Ed. *IHM '02*, vol. 32. ACM Press, New York, NY, pp. 251-254, 2002.
- [Scapin et al., 89] Scapin D.L., Pierret-Golbreich C., “MAD : une méthode analytique de description des tâches”, *Colloque sur l'ingénierie des IHM*, Sophia-Antipolis, 1989.

- [Scapin et Bastien 01], Scapin et Bastien J.M.C., Analyse des tâches et aide ergonomique à la conception : l'approche MAD* (chapitre 3), in Analyse et Conception de l'I.H.M. / Interaction Homme-Machine pour les S.I vol.1, C. kolski, Editor. 2001, Hermès Science : Paris, France, 2001.
- [Scapin et Pierret-Golbreich 90] Scapin, DL. , Pierret-Golbreich C. Towards methods for task description: MAD. In Berlinguet, L.Berchelette, D. (Eds).Work with Display Unit'89, Elsevier Science Publisher, Amsterdam, The Nertherlands,pp. 371-380.
- [Schiaffino 00] S. Schiaffino, A. Amandi, User profiling with Case-Based Reasoning and Bayesian Networks, IBERAMIA-SBIA 2000 Open Discussion Track, 2000.
- [Schmidt et al., 02] Schmidt A., Van Laerhoven K., Strohbach M., Friday A. and Gellersen HW. Context Acquisition based on Load Sensing In Proceedings of UbiComp 2002, G. Boriello and L.E. Holmquist (Eds). Lecture Notes in Computer Science, Vol 2498, ISBN 3-540-44267-7; Springer Verlag, Göteborg, Sweden, pp. 333 – 351, 2002.
- [Scholtz et al., 98] Scholtz J., Laskowski S., Developing Usability Tools and Techniques for Designing and Testing Web Sites. National Institute of Standards and Technology (NIST), 1998.
- [Scine A, Kershberg L, 02] WebSifter: An ontology-based personnalisable search agent for the web, In proceedings of International conference on digital libraries. Reasearch and Practice, 2000.
- [Seffah 04] Seffah A., Javahery H. eds., Multiple User Interfaces: Cross Platform Applications and Context-Aware Interfaces, John Wiley & Sons, Chichester, England, 2004.
- [Senach 90] Senach B. Evaluation ergonomique des IHM : Une revue de la littérature. Rapport INRIA n°1180, 1990.
- [Shakhnarovich 06] Shakhnarovich, T. Darrell, and P. Indyk, eds. Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. MIT Press, 2006.
- [Shardanand et Maes 95] Shardanand U. and Maes P., Social Information Filtering : Algorithms for Automating "World of Mouth", In Proceeding of the CHI-95 Conference. Denver, USA, ACM Press, pp. 210-217, 1995.
- [Shin 95] Shiba Y., Conception à l'Ecoute du Marché, INSEP Editions, Mouvement Français pour la Qualité, 1995.
- [Shneiderman 92] Shneiderman B., Designing the user interface: strategies for effective human-computer interaction (2nd ed.), Reading, MA: Addison-Wesley, 1992.
- [Smith 86] Smith S.L., Mosier J.N., Guidelines for designing user interface software, Report EDS-TR-86-278, The MITRE Corporation, Bedford, MA, 1986.
- [Smith et Mosier, 1986] S. L. Smith and J. N. Mosier, Guidelines for Designing User Interface Software, The MITRE Coporation, Bedford ESD-TR-86-278 MTR-10090, 1986.
- [Sottet et al., 05a] Sottet J.S., Calvary G. and Favre J.M., Towards Model-Driven Engineering of Plastic User Interfaces, Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI'05) held in conjunction with the ACM/IEEE 8th International

Conference on Model Driven Engineering Languages and Systems (MoDELS'05), Montego Bay, Jamaica, 2005.

[Sottet et al., 06b] Sottet, J.S., Calvary, G., Favre, J.M. Towards Mappings and Models Transformations for Consistency of Plastic User Interfaces, The Many Faces of Consistency, Workshop CHI2006, Montréal, Québec, Canada, 2006.

[Soui et al., 07a] Soui, M., Abed, M., Kolski, C. and Ghedira, K. : Criteria devoted to evaluate personalized interactive systems, Nouvelles tendances technologiques en génie électrique & informatique, GEI'2007, Edition CPU, Sfax, pp. 119-125, mars, ISBN 978-9973-61-631-9, 2007.

[Soui, M et al., 08b] Soui, M., Abed, M., Kolski, C. and Ghedira, K., Uster G. Evaluation of Personalized Information Systems: Application in Intelligent Transport System. Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering, SEKE2008 (San Francisco, USA, July 1-3, 2008), pp. 877-880, ISBN 1-891706-22-5, 2008.

[Srisuwan et Srivihok 07] Srisuwan, P., Srivihok, A., Personalized Trip Information for E-Tourism Recommendation System Based on Bayes Theorem, in IFIP International Federation for Information Processing, Volume 255, Research and Practical Issues of Enterprise Information Systems II Volume 2, eds. L. Xu, Tjoa A., Chaudhry S. (Boston: Springer), pp. 1271-1275, 2007.

[Stephanidis et al., 01] Stephanidis C., Paramythis A, Sfyraakis M.et Savidis A. A case Study in Unified User Interface Development: The AVANTI Web Browser. In C. Stephanidis (Ed.), User Interfaces for All – concepts, methods and tools. pp. 525-568, NJ Mahwah : Lawrence Erlbaum Associates, 2001.

[Sweeney 93] Sweeney M., Maguire M., Shackel B., Evaluating user-computer interaction: a framework, International Journal of Man-Machine Studies, 38, p. 689-711, 1993.

[Tarby 93a] Tarby J.C, Gestion Automatique du dialogue Homme-Machine à partir de spécification conceptuelles. Thèse en Informatique, Université de Toulouse I, Septembre 1993.

[Tarby et al., 06b] Tarby J.C., evaluation précoce et conception orientée evaluation, Actes du congrès ERGO-IA'06, Biarritz, 11-13 2006.

[Tarby et al., 07c] Tarby J.C., Ezzedine, H., Tran, C-D., Laporte, P. and Kolski, C.: Traces using aspect oriented programming and interactive agent-based architecture for early usability evaluation: basic principles and comparison. J. Jacko (Ed.), Human-Computer Interaction, Part I, HCII 2007, Lecture Notes in Computer Science (LNCS), Springer-Verlag, pp. 632-641, 2007.

[Thevenin 99] Thevenin, D., Coutaz, J. Plasticity of User Interfaces: Framework and Research Agenda. In Proceedings of INTERACT'99, Edimbourg, UK, pp. 110-117, 1999.

[Tracey, 05] Tracey S., User Experience in Interactive Computer Game Development, Lecture Notes in Computer Science, Volume 3101/2004, 2005.

[Tyler et al., 07] Tyler N.A., Fujiyama T., Childs C. Evaluation Measure for Mobility and Accessibility, How do elderly pedestrians perceive hazards in the street?, A Laboratory for the assessment of pedestrian capabilities. Eleventh International Conference on Mobility and Transport for Elderly and Disabled People, Montreal, June 2007.

- [Ungar et Foster 98] Ungar L.H. and Foster D.P. Clustering Methods for Collaborative Filtering. AAAI Workshop on Recommendation Systems, M. Kaufman (Ed.) « Empirical analysis of predictive algorithms for collaborative filtering. Uncertainty in Artificial Intelligence », Proceedings of the 14th Conference, pp. 43-52, 1998.
- [Upendra et Pattie 95] Upendra S. et Pattie M. Social Information Filtering : algorithms for automating "word of mouth". In Proceedings of CHI'95, Human factors in computing systems. pp. 210-217, 1995.
- [Uster 98] Uster G. Information multimodale. Document interne. INRETS, 1998.
- [Uster, 00] Uster G., État des réflexions sur le développement de l'information multimodale en France. Congrès UITP, Hanovre, 2000.
- [Uster, 01] Uster G., Développement de l'information multimodale en France : quels leviers actionner ?, Annales des Ponts et Chaussées n°98, pp 22-26 Paris 2001.
- [Valentin et al., 93] Valentin A., Vallery G., Lucongsang R., L'évaluation ergonomique des logiciels : une démarche itérative de conception, Montrouge, édition ANACT, 1993.
- [Vallet et al., 05] Vallet, D., Mylonas, P., Corella, M. A., Fuentes, J. M., Castells, P., Avrithis, Y.: A Semantically-Enhanced Personalization Framework for Knowledge-Driven Media Services. IADIS WWW/Internet Conference (ICWI 2005), Lisbon, Portugal, October 2005.
- [Vanderdonckt 94] Vanderdonckt J. Guide ergonomique de la présentation des applications hautement interactives, Presses Universitaires de Namur, Belgique, 1994.
- [Vanderdonckt 99] Vanderdonckt J. Development Milestones towards a Tool for Working with Guidelines, Interacting with Computers, 12 (2), pp. 81-118, 1999.
- [Vanderdonckt et al., 05] Vanderdonckt J., Grolaux D., Van Roy P., Limbourg Q., Macq B., Michel B. A Design Space For Context-Sensitive User Interfaces, In Proceedings of IASSE, 2005.
- [Villanova-oliver 02] Villanova-oliver M , Adaptabilité dans les systèmes d'information sur le web : Modélisation et mise en oeuvre de l'accès progressif. Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2002.
- [Virvou et Kabassi 02] Virvou M. et Kabassi K. IFM: An Intelligent Graphical User Interface Offering Advice. In 2nd Hellenic Conf. on AI, SETN-2002, Thessaloniki, Greece, pp. 155-164, April, 2002.
- [Waern et al., 98] Waern A., Averman C., Tierney M., Rudström A. and Laaksolahti J. ConCall: an information service for researchers based on EdInfo. Research report T98:04, Swedish Institute of Computer Science, 1998.
- [Whitefield 91] A. Whitefield, F. Wilson, J. Dowell. A framework for human factors evaluation. Behaviour & Information Technology, Vol 10, n°1, pp. 65-79, 1991.
- [Young 90] Young R.M., Whittington J., A Knowledge analysis of interactivity, Proceedings of Interact'90, Diaper D., Cockton G., Shackel B. (Eds), Elsevier Scientific Publishers B.V., p. 207-212, Cambridge, UK, 27-31, 1990.

[Zimmermann et Lorenz, 08] Zimmermann A., Lorenz Andreas, LISTEN: a user-adaptive audio-augmented museum guide, Proceedings of User Modeling and User-Adapted Interaction, pp. 390-414, 2008.

ANNEXE A

Le projet Mover.Perso

(Cette annexe a pour objectif de compléter les explications du 4.1, chapitre 1)

Mover.Perso (MObilité et mUltimodalité Voyageurs Etudiants en Région Nord Pas de Calais) est un projet PREDIT (Programme national de Recherche et D'Innovation dans les Transports terrestres) financé par la PREDIM (Plate-forme de Recherche et d'Expérimentation pour le Développement de l'Information Multimodale). Ce projet a été réalisé au sein du LAMIH en collaboration avec l'INRETS et la société Archimed (fin de projet en 2007).

L'objectif de ce projet est la mise en œuvre concrète de services et à leur évaluation. La personnalisation de l'information en est l'axe central. Ces services ont pour objet d'être centrés sur la mobilité au quotidien, dans un but d'accompagnement du jeune dans son « univers » et ses activités. Les étudiants disposent d'un emploi du temps initialement "statique" (mais qui évolue au cours du temps). Autour de cet emploi du temps, ils occupent leurs temps libres de différentes manières : sports et loisirs, shopping, études, culture...

L'originalité de ce projet est d'accompagner les étudiants tout au long de leurs occupations de manière personnalisée :

- en termes de déplacements (moyen de transport préféré, coût du déplacement, etc.) ;
- en termes d'organisation des journées (par rapport aux préférences des étudiants et en fonction de l'offre de transport).

L'application s'appuie sur un agenda adapté pour PDA accessible au travers du web. Dans cette application, trois modèles d'agents ont été définis. Les agents de *Recherche* pour la recherche des itinéraires, les agents de *Gestion de profil* pour la recommandation des itinéraires et les agents *Agenda* qui s'occupent de la gestion du planning de l'étudiant. La figure A fournit deux illustrations des IHM de l'application (l'application a été développée en PHP).

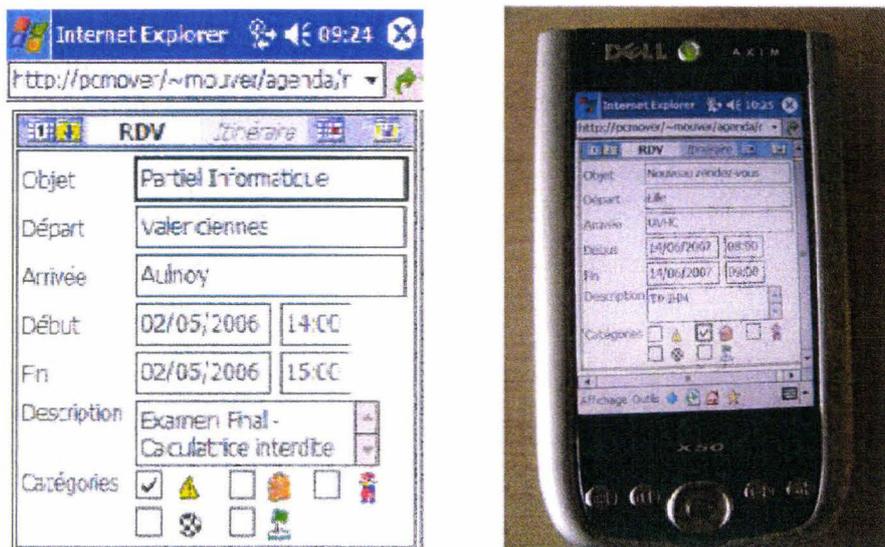


Figure A. Exemple d'interfaces homme-machine de l'application

ANNEXE B

Outils logiciels utilisés dans le cadre de notre travail

Dans cette partie nous allons présenter les outils et les langages utilisés dans le cadre de notre travail.

B1. NetBeans

Pour la mise en oeuvre de la méthode MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage), nous avons choisi le langage java comme langage de programmation vu sa puissance et sa modernité. Ainsi, l'application a été implémentée en utilisant l'environnement NetBeans. NetBeans est un logiciel de développement java complet et performant avec une facilité de prise en main. Son éditeur graphique de fenêtre intégré est très pratique pour maquetter des interfaces homme-machine rapidement et sans aucune ligne de code pour la génération de la présentation. La figure B montre l'interface utilisée par l'évaluateur lors de l'activité de définition de contexte d'évaluation (cf. 3.3.1.4.1).

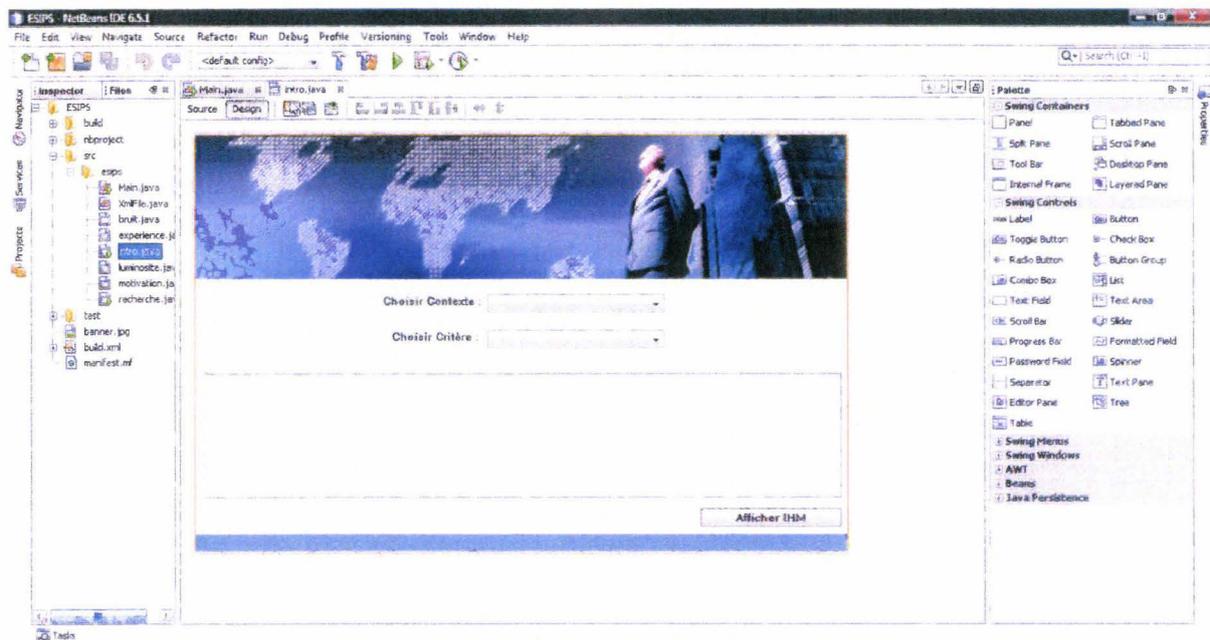


Figure B. Interface homme-machine de NetBeans

On trouve ci-dessous un exemple extrait du code de l'application développée.

```

public class ApplicationTransport extends JFrame {
private static final long serialVersionUID = 1L;
public File log = new File("log.txt"); // @jve:decl-index=0:
    private JPanel jContentPane = null; // Définition des objets interactifs
    private JMenuBar jMenuBar = null;
    private JMenu jMenuFile = null;
    private JMenuItem jMenuItemNouveau = null;
    private JMenuItem jMenuItemClose = null;
    private JPanel jPanel = null;
    private JPanel jPanel1 = null;
    private JPanel jPanel2 = null;
    private JLabel jLabel = null;
    private JLabel jLabel1 = null;
    private JLabel jLabel3 = null;
    private JLabel jLabel4 = null;
    private JTextField jTextFieldDepart = null;
    private JTextField jTextFieldDestination = null;
    private JTextField jTextFieldheured = null;
    private JButton jButtonRechercher = null;
    private JPanel jPanel3 = null;
/**
 * This method initializes jMenuBar
 *
 * @return javax.swing.JMenuBar
 */
private JMenuBar getJMenuBar() {
    if (jMenuBar == null) {
        jMenuBar = new JMenuBar();
        jMenuBar.add(getJMenuFile());
    }
    return jMenuBar;
}
/**
 * This method initializes jMenuFile
 *
 * @return javax.swing.JMenu
 */
private JMenu getJMenuFile() {
    if (jMenuFile == null) {
        jMenuFile = new JMenu();
        jMenuFile.setText("Fichier");
        jMenuFile.add(getJMenuItemNouveau());
        jMenuFile.add(getJMenuItemClose());
    }
    return jMenuFile;
}
    * This method initializes jMenuItemNouveau
    * @return javax.swing.JMenuItem
private JMenuItem getJMenuItemNouveau() {
if (jMenuItemNouveau == null) {
    jMenuItemNouveau = new JMenuItem();
    jMenuItemNouveau.setText("Nouveau");
}
return jMenuItemNouveau;
}
}

```

```

/**
 * This method initializes jMenuItemClose
 *
 * @return javax.swing.JMenuItem
 */
private JMenuItem getJMenuItemClose() {
    if (jMenuItemClose == null) {
        jMenuItemClose = new JMenuItem();
        jMenuItemClose.setText("Quitter");
    }
    return jMenuItemClose;
}

/**
 * This method initializes jPanel
 *
 * @return javax.swing.JPanel
 */
private JPanel getJPanel() {
    if (jPanel == null) {
        jPanel = new JPanel();
        jPanel.setLayout(new FlowLayout());
        jPanel.add(getJPanel1(), null);
        jPanel.add(getJPanel2(), null);
    }
    return jPanel;
}

/**
 * This method initializes jPanel1
 *
 * @return javax.swing.JPanel
 */
private JPanel getJPanel1() {
    if (jPanel1 == null) {
        jLabel1 = new JLabel();
        jLabel1.setText("Destination");// Définition d'un label "Destination"
        jLabel = new JLabel();
        jLabel.setText("Point de départ");//Définition d'un label " Point de départ "
        jLabel3 = new JLabel();
        jLabel3.setText("Heure de départ");//Définition d'un label " Heure de départ "
        jLabel4 = new JLabel();
        jLabel4.setText("Choisir classe");//Définition d'un label " Choisir classe"
        jLabel.setPreferredSize(new Dimension(86, 20));
        GridLayout gridLayout = new GridLayout();
        gridLayout.setRows(4);
        jPanel1 = new JPanel();
        jPanel1.setLayout(gridLayout);
        jPanel1.add(jLabel, null);
        jPanel1.add(jLabel1, null);
        jPanel1.add(jLabel3, null);
        jPanel1.add(jLabel4, null);
    }
    return jPanel1;
}

```

```

* This method initializes jPanel2
* @return javax.swing.JPanel
*/
private JPanel getJPanel2() {
    if (jPanel2 == null) {
        GridLayout gridLayout1 = new GridLayout();
        gridLayout1.setRows(3);
        jPanel2 = new JPanel();
        jPanel2.setLayout(gridLayout1);
        jPanel2.add(getJTextFieldDepart(), null);
        jPanel2.add(getJTextFieldDestination(), null);
        jPanel2.add(getJTextFieldheured(), null);
    }
    return jPanel2;
}

* This method initializes jTextFieldDepart
* @return javax.swing.JTextField
*/// Définition d'un Listener pour le champ de texte "point de départ"
private JTextField getJTextFieldDepart() {
    if (jTextFieldDepart == null) {
        jTextFieldDepart = new JTextField();
        jTextFieldDepart.setActionCommand("Depart");
        jTextFieldDepart.setPreferredSize(new Dimension(100, 20));
        jTextFieldDepart.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyTyped(java.awt.event.KeyEvent e) {
                String v ;
                if (jTextFieldDepart.getText().length()==0){
                    jTextFieldDepart.setName("depart");
                }
                v= jTextFieldDepart.getName();
                System.out.println(v);
                enregister(v);}
        });
    }
    return jTextFieldDepart;
}; Définition d'un Listener pour le champ de texte "heure de départ"
private JTextField getJTextFieldheured() {
    if (jTextFieldheured == null) {
        jTextFieldheured = new JTextField();
        jTextFieldheured.setActionCommand("heured");
        jTextFieldheured.setPreferredSize(new Dimension(100, 20));
        jTextFieldheured.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyTyped(java.awt.event.KeyEvent e) {
                String v ;
                if (jTextFieldheured.getText().length()==0){
                    jTextFieldheured.setName("heured");
                }
                v= jTextFieldheured.getName();
                System.out.println(v);
                enregister(v);}
        });
    }
    return jTextFieldheured;
}
}

```

```

* This method initializes jTextFieldDestination
* @return javax.swing.JTextField// Définition d'un Listener pour le champ de texte "heure de
départ"
private JTextField getJTextFieldDestination() {
    if (jTextFieldDestination == null) {
        jTextFieldDestination = new JTextField();
        jTextFieldDestination.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyTyped(java.awt.event.KeyEvent e) {
                String v ;
                if (jTextFieldDestination.getText().length()==0){
                    jTextFieldDestination.setName("destination");
                    v= jTextFieldDestination.getName();
                    System.out.println(v);
                    enregister(v);// TODO Auto-generated Event stub
                }
            }
        });
    }
    return jTextFieldDestination;
}
/** Définition d'un Listener pour le bouton "recherche d'itinéraire"
* This method initializes jButtonRechercher
* @return javax.swing.JButton
private JButton getJButtonRechercher() {
    if (jButtonRechercher == null) {
        jButtonRechercher = new JButton();
        jButtonRechercher.setText("Rechercher");
        jButtonRechercher.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                String v ;
                System.out.println("clique butna()");
                v=jButtonRechercher.getText();
                System.out.println(v);
                enregister(v);
                String req="Select * from trajet where
(pointd='"+jTextFieldDepart.getText()+"\'and
"+"\'"+jTextFieldDestination.getText()+"\'";
                connecction(req);// TODO Auto-generated Event stub
            }
        }); Enregistrement des événements des utilisateurs dans le fichier de trace
    }
    return jButtonRechercher;
}

public void enregister(String c){
    try {
        FileWriter fileWriter = new FileWriter("log.txt",true);//FileWriter(log);
        //fileWriter.write("J'ai cliqué sur le bouton\n");
        fileWriter.write(c);
        fileWriter.write(" \n");
        fileWriter.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

```

private JPanel getJPanel3() {
    if (jPanel3 == null) {
        jPanel3 = new JPanel();
        jPanel3.setLayout(new FlowLayout());
        jPanel3.add(getJButtonRechercher(), null);
    }
    return jPanel3;
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            FenetrePrincipale thisClass = new FenetrePrincipale();
            thisClass.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            thisClass.setVisible(true);
        }
    });
}

public FenetrePrincipale() {
    super();
    initialize();
}

private void initialize() {
    this.setSize(300, 200);
    this.setJMenuBar(getJMenuBar());
    this.setContentPane(getJContentPane());
    this.setTitle("JFrame");
}

public void connexion(String requete) {
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conn=DriverManager.getConnection("jdbc:odbc:bd1","","");
        System.err.println("connexion ok");
        Statement st=conn.createStatement();
        ResultSet rs=st.executeQuery(requete);
        while (rs.next()){
            System.err.println("depart : "+rs.getString("pointd"));
            System.err.println("arrivée : "+rs.getString("pointa"));
            System.err.println("prix: "+rs.getInt("prix"));
            System.err.println("duree: "+rs.getInt("duree"));
        }
    }
    catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        System.err.println("erreur dase");
        e.printStackTrace();
    }
    catch (SQLException e){System.err.println("connexion ko");}
}

private JPanel getJContentPane() {
    if (jContentPane == null) {
        jContentPane = new JPanel();
        jContentPane.setLayout(new BorderLayout());
        jContentPane.add(getJPanel(), BorderLayout.CENTER);
        jContentPane.add(getJPanel3(), BorderLayout.SOUTH);
    }
    return jContentPane;
}

```

B2. XML

XML (eXtensible Markup Language) est un langage de balises comme le Html mais il est extensible, évolutif. En XML, les balises ne sont pas prédéfinies. C'est l'utilisateur qui doit ou peut définir ses propres balises. Le XML est un métalangage, soit un langage pour écrire d'autres langages ; il a été créé particulièrement pour structurer de l'information.

Dans la méthode MetTra, l'étape de capture des actions élémentaires de l'utilisateur et des réactions des interfaces conduit à des données enregistrées par le système de trace dans un fichier XML. La DTD⁸ du fichier de ce fichier est structurée sous le format XML ; elle commence avec la balise <TraceFile>. Cette balise est composée de deux parties principales :

- La balise <ObjectInterface> décrit l'interface du système en termes de services fournis et objets interactifs manipulés par l'utilisateur.
- La balise <contextModel> définit le contexte d'usage selon le triplet <plate-forme, utilisateur, et environnement>.

La figure C donne un exemple d'extrait d'un fichier de trace décrit en XML.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Traces>
3  <Services>
4  <Service id="S-search- itinerary" name= "Recherche d'itinéraire" />
5  <Object id="TextField-departure" name="Saisie du point de depart"/>
6  <Object id="TextField-Arrival" name="Saisie du point d'arrivée"/>
7  <Object id="TextField- date and time of departure" name="Saisie du date et heure de départ"/>
8  </Services>
9  <Services>
10 <Service id="S-book - ticket " name= " réserver mon voyage"/>
11 <Object id="TextField-departure" name="choisir itinéraire"/>
12 <Object id="TextField-choiseitinerary " name="choisir itinéraire"/>
13 <Object id="TextField- choiseplace" name="choisir une place"/>
14 </Services>
15
16 <Context>
17 <context id="context_default"/>
18 <platform id="plate-forme1" type="any"/>
19 <environment id="environnement1" type="any"/>
20 <user id="user1" language="EN" experience="1" workstate="hurry"/>
21 </Context>
22 </Traces>

```

Figure C. Exemple d'extrait d'un fichier de trace décrit en XML

Notons aussi que dans la méthode MetSim, nous avons utilisé le langage XML pour représenter les règles d'évaluation.

B3. UsiXml

UsiXml (*USer Interface eXtensible Markup Language*) [Limbourg *et al.*, 04] est un langage basé XML qui décrit l'interface utilisateur pour des contextes d'usage multiples. Ce langage encapsule différents modèles et dispose d'un support explicite pour les relier entre eux, les transformer et en sélectionner des parties. Les transformations graphiques représentent le mécanisme central utilisable pour manipuler les différents modèles exprimés en UsiXML. UsiXML définit les modèles suivants : modèle de domaine, modèle de tâches, modèle d'interface utilisateur abstrait, modèle d'interface utilisateur concret, modèle de contexte,

⁸ La DTD (Document Type Definition) définit une hiérarchisation d'éléments avec des typages de ces éléments via des notions d'attribut. Le document XML se conforme à la structure syntaxique de la DTD [OMG, 2000].

modèle de transformation et modèle de correspondance. Chaque modèle défini dans UsiXML peut être utilisé indépendamment des autres modèles.

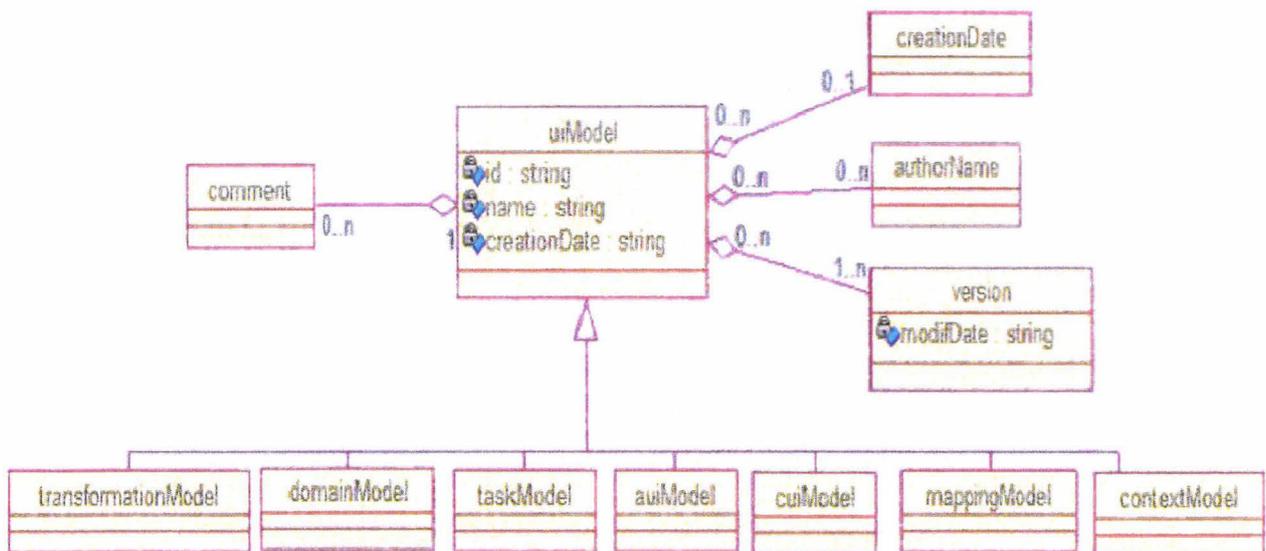


Figure D. Les différents modèles d'IHM dans UsiXML

Il existe différents types d'outils liés à UsiXML [Vanderdonckt 05]. Par exemple, IdealXML est un éditeur graphique pour la modélisation de tâches, de données et du modèle d'interface utilisateur abstrait. Cet outil gère aussi la création de modèle de correspondance entre ces trois premiers modèles. TransformiXML est un autre outil : il gère la définition, le stockage et la manipulation du modèle de transformation entre modèles. FlashiXML, et QtkXML sont des moteurs de rendu qui élaborent différents types d'interfaces finales. Pour une description plus précise d'UsiXML, on pourra se référer à la thèse de [Florins 06] et au site Internet <http://www.usixml.org>. Dans notre travail, trois de ces modèles ont été exploités dans nos études de faisabilité : *cuiModel* (Concret User Interface Model), *contextModel* et *resourceModel*.

CUIModel (Concret User Interface Model) : il fournit une spécification de l'interface utilisateur en termes d'objets concrets d'interaction et de relations concrètes. Les objets concrets et les relations fournissent un vocabulaire qui est aussi indépendant que possible de n'importe quel langage de programmation ou d'un toolkit utilisé pour implémenter l'UI (User Interface). Un CUI est une abstraction du FUI (Final User Interface). Un CUI définit des widgets, layout et interface de navigation et contient un comportement détaillé. Bien qu'un CUI rende explicitement un FUI, c'est toujours une maquette qui fonctionne seulement dans un environnement particulier. Un CUI peut également être considéré comme une concrétisation d'un AUI au niveau supérieur (voir l'exemple d'extrait d'une description d'une interface, ci-dessous, Partie 1).

Context Model : c'est un modèle décrivant les trois aspects du contexte d'utilisation dans lequel un utilisateur effectue une tâche interactive avec une station de travail spécifique et dans un environnement donné. Ce modèle peut contenir un ou plusieurs modèles de contextes. Un modèle de contexte se compose : d'un modèle d'utilisateur (*User model*), d'un modèle de plateforme (*Platform model*) et d'un modèle d'environnement (*Environment model*) ; (voir l'exemple d'extrait d'une description d'une interface ci-dessous, Partie 2).

ResourceModel : le Resource Model est hérité de l'*uiModel*. L'élément Resource décrit toutes les entités qui peuvent influencer la tâche interactive de l'utilisateur avec l'UI prévu. Cet élément comprend toutes les traductions possibles (textes, tool tips, ...) des composants

graphiques contenus dans le CUI. Le "Resource Model" contient un tag "cioRef" qui permet de spécifier les traductions concernant l'identificateur du widget. La relation qui permet de désigner un widget à son "cioRef" s'effectue grâce à l'identificateur du widget et de l'identificateur du "cioRef" nommé "cioId". (voir l'exemple d'extrait d'une description d'une interface ci-dessous, Partie 3).

On trouve ci-dessous un exemple extrait d'une description d'une interface à évaluer décrite en UsiXML.

```
<?xml version="1.0" encoding="UTF-8"?>
<uiModel xmlns="http://www.usixml.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.usixml.org/ http://www.usixml.org/spec/UsiXML-ui_model.xsd"
  id="transport_18" name="transport"
  creationDate="2009-10-18T20:24:04.137+01:00" schemaVersion="1.6.4">

  <head>
    <version modifDate="2009-10-18T20:24:04.138+01:00">1</version>
    <authorName>Makram</authorName>
    <comment>Generated by GrafiXML 1.2.0 build id : 200612141408</comment>
    <comment>WARNING : AUI Model save is a work in progress. Use it at your own
risk</comment>
  </head>

  <aiiModel id="transport-aii_18" name="transport-aii"/>
  <cuiModel id="transport-cui_18" name="transport-cui">
    <window id="window_component_22" name="window_component_22"
      width="458" height="350">
      <gridBagBox id="grid_bag_box_23" name="grid_bag_box_23"
        gridHeight="17" gridWidth="22">
        <constraint gridx="4" gridy="6" gridwidth="4"
          gridheight="1" weightx="1.0" weighty="1.0"
          fill="both" insets="0,0,0,0">
          <outputText id="output_text_component_24"
            name="output_text_component_24"
            content="/uiModel/resourceModel/cioRef[@cioId='output_text_component_24']/resource/@content"
            defaultContent="Heure départ" isVisible="true"
              isEnabled="true" isBold="true" textColor="#000000"/>
          </constraint>
          <constraint gridx="0" gridy="0" gridwidth="22"
            gridheight="5" weightx="1.0" weighty="1.0"
            fill="both" insets="0,0,0,0">

```

```

<imageComponent id="image_component_25"
  name="image_component_25"

content="/uiModel/resourceModel/cioRef[@ciId='image_component_25']/resource/@content"
  defaultContent="/resources/00/P6100194.jpg"
  isVisible="true" isEnabled="true" textColor="#000000"/>
</constraint>
<constraint gridx="4" gridy="8" gridwidth="4"
  gridheight="1" weightx="1.0" weighty="1.0"
  fill="both" insets="0,0,0,0">
  <outputText id="output_text_component_26"
    name="output_text_component_26"

content="/uiModel/resourceModel/cioRef[@ciId='output_text_component_27']/resource/@content"
  "
  defaultContent="Date départ" isVisible="true"
  isEnabled="true" isBold="true" textColor="#000000"/>
</constraint>
<constraint gridx="10" gridy="6" gridwidth="6"
  gridheight="1" weightx="1.0" weighty="1.0"
  fill="both" insets="0,0,0,0">
  <inputText id="input_text_component_28"
    name="input_text_component_28" isVisible="true"
    isEnabled="true" textColor="#000000"
    maxLength="50" numberOfColumns="15" isEditable="true"/>
</constraint>
<constraint gridx="10" gridy="8" gridwidth="6"
  gridheight="1" weightx="1.0" weighty="1.0"
  fill="both" insets="0,0,0,0">
  <inputText id="input_text_component_29"
    name="input_text_component_29" isVisible="true"
    isEnabled="true" textColor="#000000"
    maxLength="50" numberOfColumns="15" isEditable="true"/>
</constraint>
<constraint gridx="10" gridy="10" gridwidth="6"
  gridheight="1" weightx="1.0" weighty="1.0"
  fill="both" insets="0,0,0,0">
  <inputText id="input_text_component_30"
    name="input_text_component_30" isVisible="true"
    isEnabled="true" textColor="#000000"
    maxLength="50" numberOfColumns="15" isEditable="true"/>
</constraint>
content="/uiModel/resourceModel/cioRef[@ciId='output_text_component_26']/resource/@content"
  "
  defaultContent="Heure arrivée" isVisible="true"
  isEnabled="true" isBold="true" textColor="#000000"/>
</constraint>
<constraint gridx="4" gridy="10" gridwidth="4"
  gridheight="1" weightx="1.0" weighty="1.0"
  fill="both" insets="0,0,0,0">
  <outputText id="output_text_component_27"
    name="output_text_component_27"

```

```

</constraint>
  <constraint gridx="11" gridy="12" gridwidth="5"
    gridheight="1" weightx="1.0" weighty="1.0"
    fill="both" insets="0,0,0,0">
    <button id="button_component_31"
      name="button_component_31"

content="/uiModel/resourceModel/cioRef[@cioId='button_component_31']/resource/@conten
t"
      defaultContent="Recherche itinéraire"
      isVisible="true" isEnabled="true" textColor="#000000"/>
    </constraint>
  </gridBagBox>
</window>
</cuiModel>
<contextModel id="transport-contextModel_18" name="transport-contextModel"> //
(Partie 2)
  <context id="transport-context-fr_FR_18" name="transport-context-fr_FR">
    <userStereotype id="transport-stfr_FR_18" language="fr_FR"
stereotypeName="transport-stfr_FR"/>
    <platform id="transport-platform_18" name="transport-platform"/>
    <environment id="transport-env_18" name="transport-env"/>
  </context>
</contextModel>
<resourceModel id="transport-res_18" name="transport-res"> //(Partie 3)
  <cioRef cioId="image_component_17">
    <resource content="/resources/00/Snapshot_20090530.jpg" contextId="transport-
context-fr_FR_18"/>
  </cioRef>
  <cioRef cioId="image_component_25">
    <resource content="/resources/00/P6100194.jpg" contextId="transport-context-
fr_FR_18"/>
  </cioRef>
  <cioRef cioId="output_text_component_24">
    <resource content="Heure départ" contextId="transport-context-fr_FR_18"/>
  </cioRef>
  <cioRef cioId="output_text_component_26">
    <resource content="Heure arrivée" contextId="transport-context-fr_FR_18"/>
  </cioRef>
  <cioRef cioId="output_text_component_27">
    <resource content="Date départ" contextId="transport-context-fr_FR_18"/>
  </cioRef>
  <cioRef cioId="button_component_31">
    <resource content="Recherche itinéraire" contextId="transport-context-fr_FR_18"/>
  </cioRef>
</resourceModel>
</uiModel>

```

B4. GrafiXML

Dans le cadre de notre travail (Méthode MetSim), nous avons utilisé l'éditeur graphique GrafiXML. GrafiXML est un éditeur d'interface à niveau de fidélité élevée fondé sur le langage UsiXML. GrafiXML permet de représenter une interface graphique en positionnant

les objets interactifs souhaités, que ce soient des objets standards (un bouton radio, une liste de sélection, etc.) ou des objets ajoutés (la saisie d'une heure, un calendrier, un conteneur, etc.). GrafiXML produit automatiquement des spécifications fonctionnelles rédigées en UsiXML. Ces spécifications peuvent être liées à un contexte d'utilisation. Il est possible aussi de décliner différentes interfaces possibles pour plusieurs contextes d'utilisation associés au même projet [Vanderdonckt, 07]. Cet outil a été utilisé pour la mise en pratique de notre méthode MetSim (METHode d'évaluation par SIMulation). La figure E (a) montre la fenêtre de conception de l'outil GrafiXML utilisée par le concepteur pour générer une interface finale (cf. Figure E (b)).

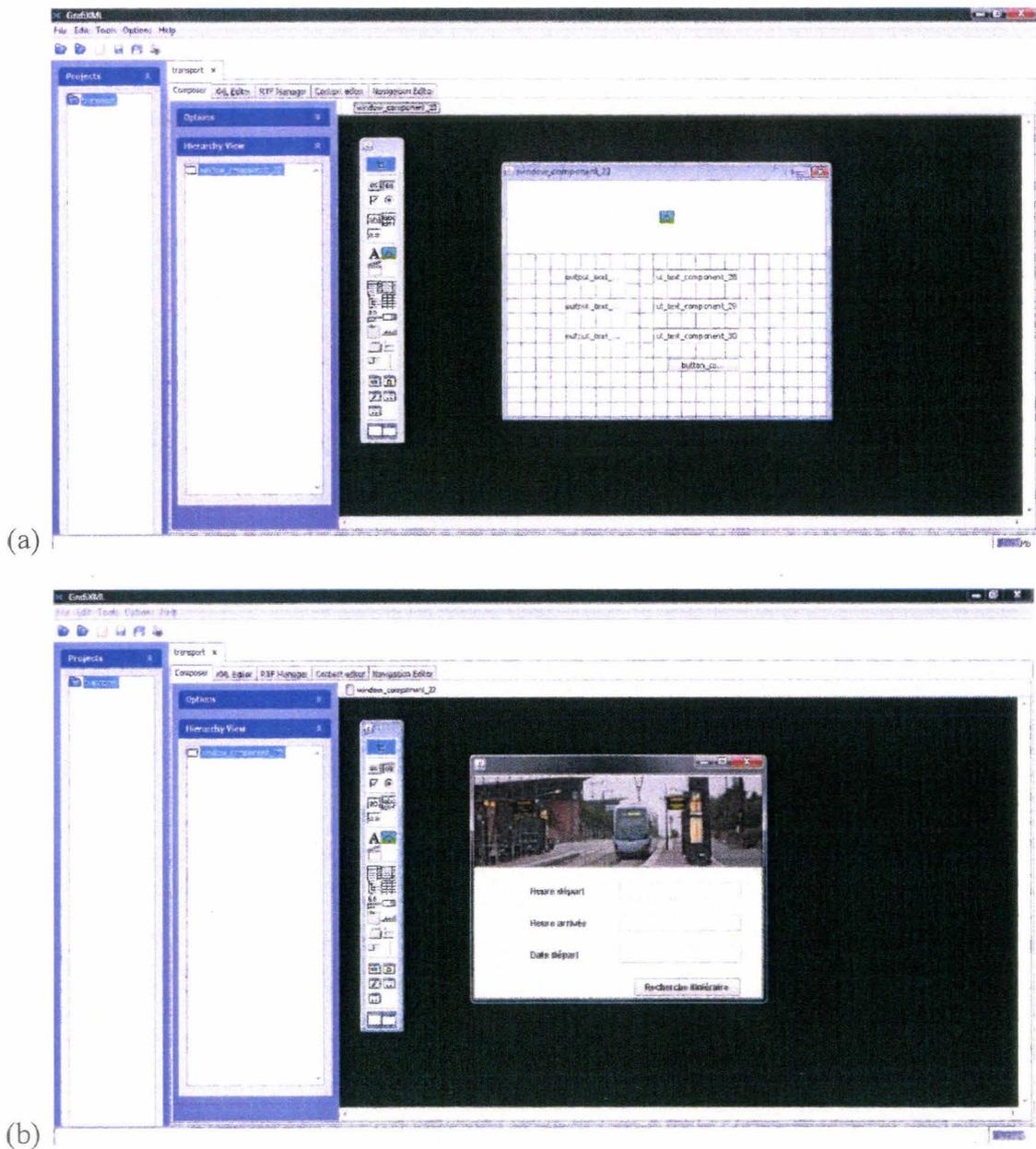


Figure E. Exemples d'interface homme-machine de l'outil GrafiXML

Résumé

Les systèmes d'information personnalisés (SIP) sont de plus en plus présents dans la vie de tous les jours, aussi bien chez soi que dans le cadre du travail. Ils possèdent de plus en plus différentes capacités d'adaptation, ils sont disséminés dans l'environnement de l'utilisateur. En effet, de nombreux travaux sont menés aussi bien dans le monde académique que dans celui des entreprises sur la modélisation de l'utilisateur, les méthodes de conception de SIP, les outils de construction d'interfaces adaptées, les algorithmes de personnalisation, etc. Cependant, l'évaluation de ces systèmes est négligée, voire ignorée par les spécialistes du domaine. Pour combler ce manque, il s'agit d'étudier et mettre en place une batterie de méthodes permettant d'évaluer et d'améliorer leur qualité de personnalisation.

Des verrous scientifiques liés à l'évaluation centrée contexte des SIP apparaissent. C'est dans ce contexte que nous proposons trois démarches complémentaires d'évaluation : la première est , intitulée MetQus (METHode d'évaluation basée sur le QUestionnaire de Satisfaction), la seconde prend la forme d'évaluation précoce qui consiste à prévoir l'évaluation dès les premières étapes de développement d'une application personnalisée en intégrant un système de trace, intitulée MetTra (METHode d'évaluation qualifiée de précoce basée sur les TRAcés d'usage) et la troisième correspond à une évaluation par simulation d'une maquette de SIP, intitulée MetSim (METHode d'évaluation par SIMulation). MetSim permet de détecter les problèmes engendrés suite au changement du contexte en se basant sur un système à base de cas. Le cadre applicatif concerne l'évaluation des systèmes d'information personnalisés dans le domaine des transports collectifs des personnes.

Mots clés : Système d'Information Personnalisé (SIP), Evaluation, Personnalisation, Interaction Homme-Machine.

Abstract

The personalized information systems (PIS) are more and more present everywhere (at home, at work, in public places, etc.). More they can have different adaptation capabilities, and can be disseminated in the user environment. In fact, several studies in research and in industry were set about user modelling, design methods and tools for HCI generation. However, the evaluation of such systems is neglected in the HCI (Human-Computer Interaction) and HMS (Human-Machine Systems) literatures. To fill up this lack, it is necessary to envisage new evaluation methods.

Difficulties concerning context-centred evaluation appear. In this research field, we propose three new complementary approaches dedicated to personalized interactive system evaluation. The first one is called MetQus consists in collecting behavioural information concerning the system use. This method actually necessitates explicitly the user intervention and the existence of a version of the PIS (model, mock-up, prototype or final system). The second one, is called MetTra, consists in foreseeing the evaluation from the early stages of application development by integrating a tracing system. The last one is called MetSim an evaluation by simulation of PIS mock-up. MetSim is also based on Case Based Reasoning system to identify problems. These evaluation approaches have been validated by applying them to evaluate PIS in transport field.

Keywords : Evaluation, Human-Computer Interaction (HCI), Personalized Information System (PIS), Personalization.



Bibliothèque Universitaire de Valenciennes



00900694