



HAL
open science

Modèle d'Interface Intelligente pour Terminaux de Communication

Olivier Sandel

► **To cite this version:**

Olivier Sandel. Modèle d'Interface Intelligente pour Terminaux de Communication. Informatique [cs]. Université Louis Pasteur - Strasbourg I, 2002. Français. NNT: . tel-00453013

HAL Id: tel-00453013

<https://theses.hal.science/tel-00453013>

Submitted on 3 Feb 2010

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.



UNIVERSITÉ STRASBOURG 1 - LOUIS PASTEUR

THÈSE

en vue de l'obtention du titre de

DOCTEUR DE L'UNIVERSITÉ LOUIS PASTEUR

Discipline : **Sciences**

Spécialité : **Informatique**

***Modèle d'Interface Intelligente
pour Terminaux de Communication***

Olivier SANDEL

présentée et soutenue publiquement

le 7 Juin 2002

- Unité de Recherche*** : Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT , UMR 7005 CNRS - ULP)
Equipe de Recherche Technologique en Informatique (ERTI)
- Directeur de Thèse*** : Professeur Jerzy KORCZAK, LSIIT, Université Louis Pasteur
Tuteur Industriel : Monsieur Daniel DOS-SANTOS, ESD, Alcatel Business Systems
- Rapporteur Interne*** : Professeur Jean-Jacques PANSIOT, LSIIT, Université Louis Pasteur
Rapporteurs Externes : Professeur Pierre VINCENT, LOR, Institut National des Télécommunications
Professeur Stan MATWIN, KAML, University of Ottawa
- Invité*** : Professeur Jean CHRETIEN, ENSPS, Alcatel Research and Innovation

A Stéphanie, ma Merveilleuse Epouse

Remerciements

A l'image de toutes mes investigations, recherches avancées, vastes approfondissements et multiples explicitations inhérentes que nous étudierons ensemble tout au long de ce mémoire, cette Thèse a été pour moi, entre-autres, une très importante et enrichissante expérience professionnelle, de loin la plus longue et la plus fournie parmi celles que j'ai déjà eu la chance d'effectuer jusqu'à ce jour. Mais ces nombreux enseignements n'auraient jamais pu m'être révélés sans l'aide et toute l'attention que mes partenaires et mon entourage en général n'ont pas hésité à mettre en permanence à ma disposition.

C'est pourquoi, je tiens vraiment à débiter ce mémoire par l'expression de mes plus sincères remerciements à tous ceux qui, par leur gentillesse, leur disponibilité et leur générosité, m'ont permis de me perfectionner et de mener à bien mes travaux de Thèse. Un très grand merci à :

Monsieur Jerzy KORCZAK, Professeur à l'Université Louis Pasteur (ULP) et Directeur de l'Equipe Apprentissage et Fouille de Données (AFD) du Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT), pour avoir si rapidement accepté de me prendre sous son aile, pour avoir été un Directeur de Thèse toujours à l'écoute, de très bon conseil, exigeant, précis et enthousiaste, pour la confiance qu'il m'a accordée, notamment dans le cadre des divers cours qu'il m'a demandé de donner au sein de l'ULP ainsi que lors de la création en 1999 de notre nouvelle Equipe de Recherche Technologique en Informatique (ERTI) dont il a également pris la Direction, pour l'intérêt et l'attirance aussi qu'il a su, depuis déjà plusieurs années avant la Thèse, éveiller en moi envers le monde de la recherche scientifique, pour l'amitié et le respect enfin qui ont jalonné l'ensemble de nos séminaires et réunions de travail.

Monsieur Jean CHRÉTIEN, *Project Manager* au sein du Groupe *ALCATEL* et Responsable du Département Recherche (*Corporate Research Center, Private Networks Department*) d'*ALCATEL Business Systems*, pour m'avoir proposé d'être mon Tuteur Industriel pendant la première moitié de la Thèse, pour l'accueil et les possibilités qu'il m'a d'emblée réservés au sein de l'entreprise, pour son grand intérêt, ses nombreuses idées innovantes et son enthousiasme permanent envers les progrès de mes recherches, pour son soutien et ses encouragements quant à la poursuite et l'achèvement de mes travaux, notamment suite à la dissolution du Département sur le site d'Illkirch (pour raisons stratégiques et économiques) à la fin de l'année 1999.

Monsieur Daniel DOS-SANTOS, *Project Manager* au sein d'*ALCATEL* et Responsable du Département Applications (*Enterprise Solutions Division, Business Applications Department*) d'*ALCATEL Business Systems*, pour avoir accepté d'être, dès le début de l'année 2000, mon Tuteur Industriel durant la seconde moitié de la Thèse, pour l'accueil et les possibilités qu'il m'a d'emblée réservés au sein de son service, pour sa compréhension et sa patience au niveau des multiples contraintes administratives inhérentes aux changements de statuts du milieu de Thèse, pour sa participation avisée et ses conseils remarquables quant à l'ensemble de la validation de nos prototypes logiciels, pour son soutien au niveau de l'achèvement de mes travaux.

Monsieur Stan MATWIN, Professeur à l'Université d'Ottawa (Canada) et Membre du Laboratoire "Extraction des Connaissances et Apprentissage Automatique" (*KAML, Knowledge Acquisition and Machine Learning*), pour avoir accepté d'être Rapporteur Externe de la Thèse, pour son prompt intérêt, ses questions éclairées et son soutien bien chaleureux pour mes travaux.

Monsieur Pierre VINCENT, Professeur et Responsable du Département Logiciels-Réseaux (LOR) de l'Institut National des Télécommunications (INT) d'Evry, pour avoir accepté d'être Rapporteur Externe de la Thèse, pour ses remarques et ses interrogations très constructives.

Monsieur Jean-Jacques PANSIOT, Professeur à l'ULP et Directeur de l'Equipe Réseaux et Protocoles du LSIT, pour avoir accepté d'être Rapporteur Interne de la Thèse, pour ses conseils approfondis en matière de rédaction scientifique, pour son humour et son respect également.

Monsieur Claude GEYER, Ingénieur au sein du Département Recherche d'ALCATEL et Chef de l'Equipe ayant en charge l'ergonomie et les interfaces de nouvelle génération appliquées aux terminaux de communication de la Gamme "WebTouch", pour ses importantes connaissances de la recherche en entreprise qu'il m'a permis de partager avec lui, pour nos agréables discussions et "brainstormings" très enrichissants qu'il n'a jamais hésité à conduire au sujet de mes travaux.

Monsieur Jean-Philippe MARCHAND, Ingénieur et Responsable "Logiciels" de l'Equipe de Messagerie d'ALCATEL, pour ses nombreuses idées et remarques au regard de mes recherches.

Madame Nicole LEVY, Maître de Conférences à l'Université de Versailles Saint-Quentin, pour sa gentillesse, ses informations et ses conseils approfondis en matière de Spécifications Formelles orientées vers le domaine des télécommunications, et notamment au sujet de "LOTOS".

Tous les membres des Départements "Recherche" et "Applications" d'ALCATEL, Docteurs, Ingénieurs, Techniciens et Secrétaires, pour leurs précieux conseils, leurs remarques pertinentes et leurs indispensables informations, pour leurs différentes lectures avisées et commentées de mes rapports et publications, mais aussi pour les pots, inévitablement suivis de discussions aussi passionnées qu'enrichissantes, pour leur complète et consciencieuse immersion dans mes travaux lors de la validation exhaustive de mes prototypes, enfin pour l'ambiance de travail très agréable.

Tous les membres du LSIT, pour leurs aides respectives, le respect et leurs intérêts pour mes recherches, pour leurs remarques pertinentes en séminaires et réunions de travail, et en particulier à l'ensemble de mes professeurs successifs, pour la grande qualité de leurs enseignements tout au long de mes études supérieures qui m'a permis d'évoluer efficacement durant cette Thèse.

Toutes les autres personnes que j'ai eu la chance de côtoyer au cours de la Thèse, au sein du Groupe ALCATEL comme de l'ULP, et qui m'ont apporté leur soutien toujours très appréciable.

Mon petit-frère Arnaud, Professeur Agrégé et Doctorant en Mécanique à l'INSA de Lyon depuis Septembre 2001, et son Epouse Blandine, Ingénieur, pour leur grande confiance et leurs constants encouragements, pour leurs apports réguliers "d'air pur montagnard", pour toutes nos discussions très instructives. Bon courage et bonne chance pour la poursuite et la fin de ta Thèse.

Mes Parents, Marie-Odile et Jean-Luc, qui m'ont donné le goût du travail bien fait, n'ont jamais cessé de croire en mes capacités, à persévérer dans mes études, dans la vie et au cours de cette Thèse en particulier, à traverser aussi les moments difficiles inévitables dans une entreprise de cette ampleur, et à rebondir ensuite. Merci à vous deux pour votre présence, vos soutiens et encouragements continus qui m'ont permis de me consacrer, sans retenue aucune, à cette Thèse.

Mon Epouse Stéphanie bien évidemment, pour son amour sans limites, son calme et sa sérénité très réconfortants tout au long de la Thèse, pour sa gaieté et sa joie de vivre permanentes, pour ses idées remarquables aussi et notre complicité primordiale. Merci Stéphanie de m'avoir pleinement "secondé" dans cette grande aventure, entamée dès le lendemain de notre Mariage...

Famille, amis et connaissances enfin, pour tous vos petits mots toujours très encourageants.

Table des Matières

INTRODUCTION -----	1
CHAPITRE 1 :	
TERMINAUX DE COMMUNICATION ET INTERFACES INTELLIGENTES -----	6
1.1. LES TERMINAUX DE COMMUNICATION -----	7
1.1.1. INTRODUCTION-----	7
1.1.1.1. Définition du concept-----	7
1.1.1.2. Etat des marchés et contexte évolutif actuel-----	8
1.1.1.3. Justifications de l'ensemble de nos orientations initiales de travail-----	10
1.1.2. DÉCOUVERTE ET ÉTAT DE L'ART DU "WEBPHONE"-----	12
1.1.2.1. Contexte historique, caractéristiques et propriétés générales-----	12
1.1.2.2. Les Terminaux Internet : récapitulatif et premières conséquences-----	13
1.1.3. UN SECOND DISPOSITIF POUR NOS RECHERCHES : LA "MESSAGERIE UNIFIÉE"-----	16
1.1.3.1. Introduction et principales caractéristiques-----	16
1.1.3.2. Contexte général du marché de la "Messagerie Unifiée"-----	18
1.1.3.3. La "Messagerie Unifiée" : récapitulatif et nouvelles perspectives-----	19
1.2. LES INTERFACES INTELLIGENTES -----	21
1.2.1. INTRODUCTION ET SITUATION DE LA PROBLÉMATIQUE-----	21
1.2.2. DÉFINITION QUALITATIVE D'UNE INTERFACE INTELLIGENTE-----	22
1.2.2.1. Description de toutes les caractéristiques remarquables-----	22
1.2.2.2. Dégagement d'une définition exhaustive d'interface intelligente-----	25
1.2.2.3. Vers une hiérarchisation des interfaces intelligentes-----	26
1.2.3. DÉFINITION TAXINOMIQUE D'UNE INTERFACE INTELLIGENTE-----	27
1.2.4. ETUDE DES RÉFÉRENCES REMARQUABLES D'INTERFACES INTELLIGENTES-----	30
1.2.4.1. Introduction et justification de telles études de cas-----	30
1.2.4.2. L'aide automatique et les tutoriels informatiques-----	31
1.2.4.3. Implication massive des techniques d'exploitation de connaissances-----	33
1.2.4.4. L'Enseignement Assisté par Ordinateur-----	37
1.2.4.5. L'avènement des agents intelligents-----	41
1.2.4.6. Application des interfaces intelligentes en entreprise-----	47
1.3. CONCLUSION ET ORIENTATION DE NOTRE PROJET DE RECHERCHE -----	52
1.3.1. RÉCAPITULATIF ET DISCUSSION DES CARACTÉRISTIQUES RECENSÉES-----	52
1.3.2. DÉGAGEMENT DE NOS AXES STRATÉGIQUES DE TRAVAIL-----	57
CHAPITRE 2 :	
SPÉCIFICATIONS DE NOUVEAUX CONCEPTS D'ASSISTANTS INTELLIGENTS -----	59
2.1. LE CONCEPT « D'ASSISTANT CONVERSATIONNEL » -----	61
2.1.1. ENONCÉ ET DÉFINITIONS-----	61
2.1.1.1. Intitulé du procédé-----	61
2.1.1.2. Premières règles de base-----	61
2.1.1.3. Adjonction de premières notions d'intelligence-----	62
2.1.2. PREMIÈRES PROPOSITIONS ADAPTÉES D'AGENTS INTELLIGENTS-----	63
2.1.2.1. Corrélations avec les règles de fonctionnement établies-----	63
2.1.2.2. Vers un nouveau concept d'assistant auto-adaptable-----	65

2.2. « L' ASSISTANT INDIVIDUEL DE TÉLÉCOMMUNICATION »	65
2.2.1. INTRODUCTION ET CARACTÉRISTIQUES GÉNÉRALES	65
2.2.2. LA NOTION " D' AUTO-ADAPTATION "	66
2.2.2.1. Définition précise du concept	66
2.2.2.2. Proposition d' un schéma architectural auto - adaptable	67
2.3. DÉFINITIONS STRUCTURÉES DES DIVERS SCÉNARIOS ENVISAGÉS	68
2.3.1. « L' ASSISTANT CONVERSATIONNEL AUTO- ADAPTABLE »	68
2.3.2. LA « CLASSIFICATION AUTO- ADAPTABLE » DES INFORMATIONS	74
2.3.3. ENSEIGNEMENTS ET PERSPECTIVES DE TRAVAIL	76
2.3.4. SCHÉMATISATION RÉCAPITULATIVE DES DIFFÉRENTS CONCEPTS DÉFINIS	76
2.3.5. VERS UNE INDISPENSABLE SPÉCIFICATION FORMELLE DE NOS MODÈLES	83
2.4. SPÉCIFICATIONS FORMELLES DES MODÈLES	84
2.4.1. INTRODUCTION	84
2.4.1.1. Définitions de base	84
2.4.1.2. Justification de l' approche préalable orientée scénarios	84
2.4.1.3. Justification du besoin inhérent de formalisation	85
2.4.2. DESCRIPTION DES DIVERSES CARACTÉRISTIQUES QUALITATIVES	86
2.4.2.1. <i>Estelle</i>	86
2.4.2.2. <i>LOTOS</i>	86
2.4.2.3. <i>PRS</i>	87
2.4.2.4. <i>SDL</i>	87
2.4.2.5. <i>UML</i>	87
2.4.3. DÉGAGEMENT DE LA MÉTHODE DE FORMALISATION ADÉQUATE	88
2.4.4. SPÉCIFICATION FORMELLE EFFECTIVE DE NOS ACTIONS AUTO- ADAPTABLES	89
2.5. CONCLUSION : ENSEIGNEMENTS ET PERSPECTIVES	89
CHAPITRE 3 :	
<i>MODÈLE AUTO-ADAPTABLE DE L' UTILISATEUR</i>	91
3.1. THÉORIE DES PROCÉDÉS ACTUELS DÉDIÉS À L' UTILISATEUR	92
3.1.1. INTRODUCTION	92
3.1.2. DÉFINITION D' UN MODÈLE DE CONNAISSANCES DE L' UTILISATEUR	93
3.1.3. TECHNIQUES DE CONCEPTION DES MODÈLES D' UTILISATEURS	93
3.1.4. CAPACITÉS DES AGENTS DE REPRÉSENTATION CONTEXTUELLE DE L' UTILISATEUR	95
3.1.5. RÉCAPITULATIF DES PRINCIPAUX ENSEIGNEMENTS ÉTABLIS	96
3.2. DÉFINITION " INTELLIGENTE " D' UN NOUVEAU " MODÈLE D' UTILISATEUR "	97
3.2.1. DESCRIPTION DES CARACTÉRISTIQUES QUALITATIVES	97
3.2.2. DÉGAGEMENT DE LA DÉFINITION INITIALE DE NOTRE MODÈLE	98
3.2.3. MISE EN OEUVRE ENVISAGÉE	99
3.2.3.1. Schéma global de fonctionnement	99
3.2.3.2. Analyse du système mis en avant et nouvelle définition qualitative	100
3.3. MÉTHODE D' ANALYSE DES COMPORTEMENTS AU SEIN D' ALCATEL	101
3.3.1. DÉFINITION DU CONCEPT	101
3.3.2. ADAPTATION À L' INTERFAÇAGE INTELLIGENT " GRAND PUBLIC "	102
3.3.3. CONCEPTION D' UN PROTOTYPE « D' INTERVIEW COMPORTEMENTALE »	105
3.3.3.1. Introduction	105
3.3.3.2. Première approche exhaustive	105
3.3.3.3. Vers la recherche d' une approche moins intrusive	107
3.4. CONCLUSION	108

CHAPITRE 4 :	
MÉCANISMES D'APPRENTISSAGE AUTOMATIQUE -----	110
4.1. DÉGAGEMENT DES CONCEPTS OPÉRATOIRES -----	111
4.1.1. MODE EXPÉRIMENTAL-----	111
4.1.2. INTRODUCTION DE LA THÉORIE DES SÉQUENCEMENTS D' ACTIONS-----	112
4.2. EXTRACTION DES ACTIONS DE L'UTILISATEUR -----	115
4.2.1. REPRÉSENTATION DES CONNAISSANCES CORRESPONDANT AUX SÉQUENCES-----	115
4.2.2. COMMENTAIRES ET ANALYSE DES INFORMATIONS EXTRAITES-----	116
4.2.3. REPRÉSENTATIONS CANONIQUES DES ACTIONS ET SÉQUENCES D' ACTIONS-----	118
4.3. TRAITEMENT TEMPOREL ET GESTION DES DONNÉES -----	120
4.3.1. INTRODUCTION-----	120
4.3.2. DÉFINITION DE LA BASE DE DONNÉES DES ACTIONS MANIPULÉES-----	121
4.3.3. TRADUCTION DE LA BASE DE DONNÉES POUR L' EXTRACTION DES RÉPÉTITIONS-----	122
4.3.3.1. Récapitulatif par utilisateur-----	122
4.3.3.2. Décompositions de séquences d'actions-----	122
4.3.3.3. Extraction et comptabilisation des répétitions-----	123
4.4. MISE EN OEUVRE DE NOUVELLES DÉFINITIONS -----	125
4.4.1. DÉGAGEMENT D' ACTIONS REMARQUABLES-----	125
4.4.2. ACTIONS DÉPENDANTES OU INDÉPENDANTES-----	126
4.4.3. HIÉRARCHIE DE REPRÉSENTATION DES SÉQUENCES D' ACTIONS EFFECTUÉES-----	129
4.4.4. DÉGAGEMENT DE L' ALGORITHME D' EXTRACTION DES RÉPÉTITIONS D' ACTIONS-----	130
4.4.5. GÉNÉRALISATION DU SÉQUENCEMENT D' ACTIONS OBTENU-----	131
4.5. ELABORATION DÉTAILLÉE DES MÉCANISMES D' APPRENTISSAGE -----	132
4.5.1. SPÉCIFICATIONS PRÉCISES-----	132
4.5.1.1. Introduction-----	132
4.5.1.2. Répétition de séquences d' actions-----	133
4.5.1.3. Caractéristiques d' écriture du courrier-----	133
4.5.1.4. Tri des messages-----	134
4.5.1.5. Simplification et correction automatiques-----	136
4.5.2. ALGORITHMIQUE DES MÉCANISMES D' INFÉRENCE-----	137
4.5.2.1. Introduction-----	137
4.5.2.2. Principales orientations issues de l' apprentissage automatique-----	137
4.5.2.3. Justification des structures d' implémentation choisies-----	141
4.5.2.4. Approche symbolique-----	143
4.5.2.5. Approche statistique-----	147
4.5.3. CONSTRUCTION FINALE D' UN MODÈLE COMPORTEMENTAL DE L' UTILISATEUR-----	151
4.5.3.1. Introduction de nos mécanismes de prédiction et d' anticipation-----	151
4.5.3.2. Modélisation dynamique des seuils comportementaux-----	151
4.5.3.3. Définition fonctionnelle des mécanismes de prédiction et d' anticipation-----	153
4.5.3.4. Procédé de simplifications auto- adaptables-----	156
4.5.3.5. Mise en œuvre de la modélisation du comportement des utilisateurs-----	157
4.5.3.6. Ajustement dynamique final de la relation homme/machine-----	157
4.6. CONCLUSION -----	159
CHAPITRE 5 :	
IMPLÉMENTATION ET VALIDATION DES PROTOTYPES -----	161
5.1. CONTRAINTES TECHNOLOGIQUES DE L' IMPLÉMENTATION -----	161
5.1.1. RETOUR SUR LES MODÈLES D' APPLICATION ACCUEILLANT NOS PROTOTYPES-----	161
5.1.2. LES LANGAGES <i>COM</i> ET <i>MAPI</i> -----	162
5.1.3. DÉCOMPOSITION DE NOS MODÈLES EN AGENTS INTELLIGENTS-----	164

5.2. DÉTAILS DE NOS PROCÉDURES D'IMPLÉMENTATION -----	168
5.2.1. INTRODUCTION-----	168
5.2.2. UN PREMIER AGENT INTELLIGENT : L'AGENT D'EXÉCUTION-----	168
5.2.3. L'AGENT DE RÉPÉTITION-----	174
5.2.4. L'AGENT DE SÉQUENÇAGE-----	175
5.2.5. L'AGENT DE COMPORTEMENT-----	176
5.2.6. CONCLUSION ET ENSEIGNEMENTS-----	181
5.3. VALIDATION DE NOS MODÈLES AUTO-ADAPTABLES -----	184
5.3.1. INTRODUCTION-----	184
5.3.2. INITIALISATION DE LA VALIDATION DU PROTOTYPE DÉVELOPPÉ-----	184
5.3.2.1. Détermination des contraintes d'environnement-----	184
5.3.2.2. Intégration du cycle de validation d'ALCATEL-----	185
5.3.3. ÉTUDE SUIVIE SUR UN PANEL D'UTILISATEURS REPRÉSENTATIFS-----	186
5.3.3.1. Descriptif et mise en œuvre des caractéristiques de l'étude-----	186
5.3.3.2. Dépouillement des données recueillies et analyse des résultats-----	187
5.4. CONCLUSION -----	191
 CONCLUSION ET PERSPECTIVES -----	 193
 BIBLIOGRAPHIE -----	 193
 <i>Liste des figures et copies d'écrans</i> -----	 217
<i>Liste des tableaux</i> -----	221
<i>Liste des définitions</i> -----	223
<i>Liste des scénarios</i> -----	224
 ANNEXES -----	 225
ANNEXE 1 :	
ÉTUDE DÉTAILLÉE DES TERMINAUX ET LOGICIELS DE COMMUNICATION ; DÉGAGEMENT ET DÉFINITION DES DIFFÉRENTS MODÈLES D'APPLICATION -----	226
A1.1. DÉCOUVERTE ET ÉTAT DE L'ART DU "WEBPHONE" -----	226
A1.1.1. LE "WEBPHONE" SELON ALCATEL-----	226
A1.1.2. LE "WEBPHONE @NET" DE SIEMENS-----	228
A1.1.3. LE "TEL@PHONE" DE MATRA NORTEL COMMUNICATIONS-----	228
A1.1.4. LE "WEB AUDIO PHONE" DE SAMSUNG-----	229
A1.1.5. LE "@MAX" DE COM ONE-----	230
A1.1.6. "L'INTERNET SCREENPHONE" D'UNIVERSAL MICROELECTRONICS COMPANY-----	231
A1.2. LA "MESSAGERIE UNIFIÉE", SECOND OUTIL POUR NOS RECHERCHES -----	232
A1.2.1. LA "MESSAGERIE UNIFIÉE" SELON ALCATEL : "L'AUM"-----	232
A1.2.2. "SMARTPHONE MESSAGERIE UNIFIÉE" DE NOVAVOX-----	233
A1.2.3. "DAVID PROFESSIONAL" DE TOBIT SOFTWARE-----	234
A1.2.4. "ENTERPRISE INTERACTION CENTER" D'INTERACTIVE INTELLIGENCE-----	235
A1.2.5. LA "MESSAGERIE UNIFIÉE" SELON FREELANNING-----	236

A1.3. DESCRIPTIONS AVANCÉES DES DIFFÉRENTS MODÈLES D'APPLICATION	237
A1.3.1. SPÉCIFICATION DES PROCESSUS "D'ENTRÉES/SORTIES"	237
A1.3.1.1. Introduction	237
A1.3.1.2. "Entrées/Sorties" en JAVA	237
A1.3.1.3. Recensement des processus d'entrées	238
A1.3.1.4. Recensement des processus de sorties	238
A1.3.2. SCHÉMAS DESCRIPTIFS DU MODÈLE DE L'APPLICATION "E-MAIL"	239
A1.4. TABLEAUX RÉCAPITULATIFS	244
ANNEXE 2 :	
SPÉCIFICATION FORMELLE DE NOS MODÈLES AUTO-ADAPTABLES	248
A2.1. RÉCAPITULATIF DES DIFFÉRENTS OPÉRATEURS DE LOTOS	248
A2.2. SPÉCIFICATION FORMELLE DE NOS ACTIONS AUTO-ADAPTABLES	249
A2.3. SPÉCIFICATION DU CONCEPT DE TRI : EXEMPLE DE "L'E-MAIL"	256
ANNEXE 3 :	
DÉTAILS D'IMPLÉMENTATION DE TOUS NOS MODÈLES AUTO-ADAPTABLES	261
A3.1. PRÉLIMINAIRES TECHNIQUES SUR LES LANGAGES COM ET MAPI	261
A3.1.1. COM	261
A3.1.2. MAPI	262
A3.2. DÉCLARATIONS ET INITIALISATION DES PROCÉDURES IMPLÉMENTÉES	265
A3.2.1. PROTOTYPAGES DES FONCTIONS ET MÉTHODES INHÉRENTES	265
A3.2.2. DÉCLARATIONS DES PRINCIPALES CLASSES ET MÉTHODES "OBJET"	267
A3.2.3. FONCTIONS ET MÉTHODES POUR LE TRAITEMENT GLOBAL DES ACTIONS	269
A3.3. LES PROCÉDÉS TERMINAUX DE NOTRE "COUCHE IA"	273
A3.3.1. LES AFFICHAGES	273
A3.3.2. L'INTERFACE D'ACCÈS AUX APPLICATIONS	275
ANNEXE 4 :	
VALIDATION MÉTHODIQUE ET EXHAUSTIVE DE NOS PROTOTYPES	277
A4.1. INTÉGRATION DES MÉTHODES DE VALIDATION D'ALCATEL	277
A4.2. MISE EN OEUVRE DES SCÉNARIOS DÉDIÉS À NOS MODÈLES	280
A4.2.1. LANCEMENT ET INITIALISATION DE LA "COUCHE IA"	280
A4.2.2. DÉCOUVERTE ET GESTION DES ACTIONS RÉPÉTÉES	283
A4.2.3. APPRENTISSAGE AUTOMATIQUE DES RÉPÉTITIONS DE SÉQUENCES D'ACTIONS	284
A4.2.4. PRÉDICTION AUTO-ADAPTABLE DES SÉQUENCES D'ACTIONS RÉPÉTÉES	287
A4.2.5. ANTICIPATION AUTO-ADAPTABLE DES SÉQUENCES D'ACTIONS FRÉQUENTES	290
A4.2.6. NOUVEAU DÉMARRAGE DE LA "COUCHE IA"	294
A4.2.7. SIMPLIFICATION AUTO-ADAPTABLE DES SÉQUENCES D'ACTIONS EFFECTUÉES	296
A4.2.8. VALIDATION COMPLÈTE DE LA MÉTHODE D'ANALYSE DES COMPORTEMENTS	300

Introduction

*Pourquoi les hommes auraient-ils à s'adapter aux systèmes,
alors que les systèmes devraient s'adapter aux hommes ?*

Kristina Höök

[Höök, 1997]

En découvrant cette citation bien équilibrée, symétrique et donc aisée à retenir de Kristina Höök, beaucoup doivent se dire que le domaine de l'Intelligence Artificielle ainsi mis en avant peut apparaître simple et complexe à la fois.

Certains diront, par exemple, que si ce sont les hommes qui conçoivent les systèmes et qu'ils ont justement beaucoup de mal à s'adapter à leurs semblables, il leur est sûrement difficile de développer un système artificiel qui aide les hommes dans leurs tâches quotidiennes. D'autres s'interrogeront sur la distance grandissante qui s'installe entre les interlocuteurs lorsque l'on passe d'un contexte homme-homme à un contexte homme-machine nettement plus... artificiel.

Mais, la tâche des ergonomistes et autres concepteurs d'interfaces homme-machine se complique lorsque les besoins dans ce domaine deviennent, comme à l'heure actuelle, bien plus exigeants. Les opérateurs et les fournisseurs de services de télécommunication ont en effet besoin d'offrir de nouveaux services à leur clientèle et d'augmenter la flexibilité et l'efficacité de leurs systèmes. Les terminaux intelligents deviennent par conséquent une plate-forme adéquate pour atteindre un rôle-clé de l'intelligence dans les futurs réseaux. Ces terminaux se doivent ainsi d'obtenir la capacité de comprendre des commandes d'utilisateur, d'apprendre automatiquement pour s'adapter au profil individuel de l'utilisateur et de savoir résoudre différents problèmes dans le domaine de la télécommunication.

Dans ce contexte, *ALCATEL* apparaît aujourd'hui en leader mondial des systèmes et équipements de télécommunication. Le groupe fournit des services complets aux opérateurs, fournisseurs de services *Internet*, entreprises et consommateurs. Sa gamme de solutions va des systèmes de réseaux pour les autoroutes de l'information aux systèmes de transport et d'accès pour les communications de voix, données, textes, images et multimédias en général, en passant également par des produits pour les entreprises et les usagers "grand public". L'entreprise propose aussi des solutions clés en main, depuis le simple service téléphonique jusqu'aux infrastructures multimédias les plus sophistiquées et les terminaux de communication fixes ou mobiles.

Par ailleurs, *ALCATEL* a anticipé l'évolution des besoins des entreprises en développant de nouvelles solutions logicielles : applications, gestion de réseaux, portails *Internet* multi-terminaux, logiciels de gestion de la relation clients. Ainsi, dans le cadre de la présente Thèse, s'est adjoint l'ensemble des multiples projets de recherche et de développement menés, au sein d'*ALCATEL*, par le *Business Applications Department (Enterprise Solutions Division)* et qui ont notamment permis à l'*ALCATEL Unified Messaging (AUM)* de voir le jour. Cette application logicielle, directement dédiée à la messagerie électronique sur PC et poste téléphonique via un réseau TCP/IP, est dite "unifiée" du fait qu'elle regroupe trois médias distincts (*e-mail*, *voice-mail* et *fax*) et manipule l'ensemble des messages, dont le format a été uniformisé, en une seule et même interface-utilisateur.

Cependant, la suppression des barrières entre la voix et les données, conjuguée en outre à l'explosion d'*Internet*, a suscité l'apparition d'outils de communication plus rapides et plus puissants qui révolutionnent aujourd'hui tout le système de communication mondial. L'ensemble des nouveaux terminaux et applications associées d'*ALCATEL* ne possèdent encore, en 1998, aucune faculté de raisonnement, ni même aucune capacité de représentation des connaissances, ce type d'avantages s'avérant pourtant de plus en plus appréciables et demandés.

C'est pourquoi, nos présents travaux d'interface intelligente auto-adaptable sont rapidement devenu un programme attractif d'*ALCATEL*, dont le "Département Recherche" ainsi que celui des "Applications" se sont trouvés fortement intéressés. Ceci concerne notamment les mécanismes d'apprentissage automatique inhérents et aujourd'hui réellement indispensables à des terminaux de communication avancés, ou encore qualifiés "d'intelligents".

Les nouvelles technologies de l'information et de la communication sont par conséquent indubitablement entrées dans nos vies. Par le biais de nos lignes téléphoniques, de nos réseaux câblés, de nos agendas intelligents, de nos écrans d'ordinateur, et de nos appareils ménagers à mémoire, le vingt-et-unième siècle a mis à profit les dernières années du précédent pour poser ses jalons dans nos villes, nos campagnes, nos maisons, nos bureaux et nos poches.

Nous attendons donc tous aujourd'hui beaucoup de compréhension, de rapidité et de soutien de la part de toute interaction "homme-machine", à savoir de la communication bilatérale en général entre une personne et chacun des appareils relevant de l'informatique. Cet interfaçage ne peut se passer d'un dialogue réel et performant à double-sens, non plus basé sur un simple contrôle de l'objet de communication, mais vraiment construit à partir d'une importante personnalisation de chaque terminal considéré. Si ce dernier demeure bien autonome en pratique, il est désormais chargé de mettre en confiance, de dialoguer voire de renseigner ses utilisateurs de la manière la plus sereine et la plus explicite possible.

Ainsi, une telle convivialité appelle le domaine de l'Intelligence Artificielle et plus particulièrement celui des systèmes intelligents. Pour cela, il nous faut inévitablement mieux cerner les différentes façons d'agir, les réactions prévisibles ou non de l'utilisateur face à la machine, ainsi que toutes ses demandes éventuelles, ce en vue d'une gestion fiable des différents agents de comportement...

Nous nous rendons bien compte alors que la communication ne peut s'effectuer autrement qu'en temps réel, puisqu'elle a lieu entre deux individus placés strictement sur le même plan : l'utilisateur et son interlocuteur artificiel. Reste également à approfondir l'indispensable interface "homme-information" qui autorise non seulement la navigation de l'utilisateur (sur le réseau *Internet* notamment), mais aussi la représentation et le tri des données recueillies, de même que le travail de l'information au cours du temps (mémorisation, rappel des dernières opérations, ...).

En conséquence, si l'interactivité est la capacité pour l'utilisateur de cheminer librement dans l'ensemble des connaissances qui lui sont présentées [Verroust, 1997], ce domaine se doit aussi de travailler pour permettre à l'homme d'intervenir et dialoguer de bien diverses manières avec chacun des systèmes informatiques abordés.

Le problème principal à résoudre durant cette Thèse s'est donc défini exactement par la modélisation et la conception d'une interface intelligente, personnalisée et praticable par tous les publics sur les terminaux de communication, ces derniers ayant été pris en compte en qualité de machines spécialisables accédant à *Internet*.

Nous nous sommes alors fixés plusieurs objectifs différents, en nous focalisant sur les acquisitions et les apprentissages automatiques des connaissances au travers de l'élaboration d'un innovant modèle de comportement de l'utilisateur. Ce procédé est ainsi chargé, entre-autres, de rendre les opérations courantes moins fastidieuses pour l'utilisateur, tout d'abord par l'intermédiaire d'agents logiciels dédiés à l'apprentissage en temps réel, systèmes dits aussi "intelligents" ou encore "comportementaux", puis par leurs complets traitements auto-adaptables.

Ainsi, nous avons décomposé le problème général de la Thèse en un certain nombre de travaux spécifiques, donnant lieu à l'élaboration de chapitres dont nous nous proposons à présent d'en exposer un plan global.

Le **Chapitre 1**, intitulé "Terminaux de Communication et Interfaces Intelligentes", entame ce mémoire par un état de l'art des divers terminaux de communication disponibles sur le marché mondial actuel, ainsi que des principaux systèmes concurrentiels du groupe *ALCATEL* en matière de messagerie électronique unifiée. Axés sur des analyses et spécifications de toutes les fonctionnalités et contraintes définissant les machines de communication existantes à ce jour, nous nous sommes en effet tenus à établir un récapitulatif de chacun des détails de ces appareils. Correspondant en quelque sorte à des "cartes d'identité" très précises de leurs avantages et inconvénients respectifs, c'est sur ces comparaisons que nous nous appuyerons tout au long de la Thèse afin de pouvoir définir ce que nous pouvons faire évoluer, voire surtout comment nous pouvons réussir à passer au-delà des dernières limites matérielles et logicielles.

Cette entrée en matière se voit suivie d'une complète classification taxinomique, de même que de la mise en œuvre d'une hiérarchisation nouvelle des interfaces intelligentes. Grâce en outre à plusieurs définitions innovantes au sujet de la qualité "d'intelligence", primordiale au sein de ce mémoire, les différentes facettes de ces interfaçages avancés justifient finalement les principales orientations envisagées pour la progression de nos travaux de recherche.

Le **Chapitre 2** des "Spécifications de Nouveaux Concepts d'Assistants Intelligents" se charge, au travers de notre premier concept nommé "Assistant Conversationnel", d'introduire la notion d'apprentissage par agents intelligents dans une interface-utilisateur dédiée aux terminaux de communication. Ces spécifications, associées bien entendu à des méthodes d'analyse formelle basées sur de solides fondations théoriques, vérifient que nos systèmes possèdent bien toutes les propriétés désirées avant leurs mises en œuvre. Nous réussissons alors à confirmer dès ce niveau que, pour éviter de trop lourds apprentissages de protocoles d'interaction artificiels et offrir au contraire des accès directs aux commandes logicielles, il convient d'adapter les capacités de la machine, et non plus celles de l'utilisateur [Wolff, 1999].

Le langage *LOTOS* en particulier nous amène à caractériser et entièrement valider, après comparaison des caractéristiques qualitatives de plusieurs méthodes de spécification envisagées, un "Assistant Individuel de Télécommunication". Celui-ci, finalement encore étendu à notre dernière et primordiale notion "d'Auto-Adaptation", se compose d'ensembles de fonctionnalités détaillées et de multiples scénarios donc auto-adaptables. Tous destinés à la création, à l'analyse et à la mise en œuvre d'une assistance personnalisée à l'utilisateur, ils s'avèrent alors capables de le soutenir dans ses communications avec un interlocuteur, de même que de classifier de manière automatique toutes les informations qu'il est amené à manipuler.

Le **Chapitre 3**, titré "Modèle Auto-Adaptable de l'Utilisateur", se focalise ensuite sur la définition de l'entité principale que représente pour nous l'utilisateur. Précédée d'un état de l'art permettant d'extraire et de caractériser les mécanismes nécessaires à une application efficace d'un tel modèle, nous développons plusieurs définitions qualitatives et intégrables aux travaux issus des différentes actions propres à notre "Assistant Individuel de Télécommunication".

Nous poursuivons par l'appréhension d'une méthode d'analyse des comportements humains. Nommé précisément "Persona" [Julhiet, 1993], nous élargissons ce dispositif à tout utilisateur pouvant être qualifié de "grand public", et donc susceptible d'avoir un jour à opérer sur les divers terminaux et logiciels d'ALCATEL. Intégrée au sein du concept visé de l'interaction homme-machine, nous associons finalement cette méthode, élément inévitable de notre nouveau modèle de l'utilisateur, à nos premiers prototypes "d'Interview Comportementale".

Nous aboutissons ainsi au **Chapitre 4** des "Mécanismes d'Apprentissage Automatique". Par lui, nous abordons un autre domaine de l'Intelligence Artificielle, à savoir l'étude des structures et des algorithmes d'apprentissage automatique qui vont, en définitive, nous permettre de reconnaître, de traiter et de répondre aux multiples actions de l'utilisateur, et ce de manière de plus en plus approfondie et personnalisée.

Dans le cadre de ces divers objectifs, nous avons tout d'abord assimilé les trois facultés essentielles décrivant les systèmes intelligents : la compréhension, l'apprentissage ainsi que la résolution de problèmes. Cette contribution à l'interfaçage homme-machine actuel permet de proposer, modes opératoires et graphes d'actions à l'appui, de nouvelles méthodes d'extraction, de représentation, de tri et de gestion en temps réel des connaissances, toutes réellement "auto-adaptées" aux manipulations habituelles de l'utilisateur.

En outre, l'étude d'une première application concrète, à savoir la fonctionnalité "e-mail" extraite de différents logiciels dédiés à la gamme des *WebTouch* d'ALCATEL, met en évidence les qualités d'apports intelligents tels que nos mécanismes d'inférence. Ceux-ci apparaissent en effet capables d'exploiter les données issues de l'apprentissage des séquences d'actions en fonction du profil de chaque utilisateur. De même, la modélisation des fonctions "Tell Me" et "Do It" [Lieberman, 1997] correspond à la mise en œuvre finale d'un modèle comportemental de l'utilisateur, à la fois incrémental et évolutif.

Par extension, nous avons alors précisé la conception d'un premier prototype d'interface auto-adaptable. Cela signifie, en d'autres termes, la définition d'un apprentissage automatique aboutissant à un modèle du couple "utilisateur-application" par la prise en compte de notre précédent "Modèle Auto-Adaptable de l'Utilisateur", conceptuel et "grand public", dans le traitement des actions caractérisant nos applications de test. Nous avons élaboré et comparé plusieurs algorithmes nouveaux, propres en particulier aux critères issus du domaine de l'apprentissage symbolique ainsi que de celui des concepts statistiques. Nous avons pu ensuite construire un modèle d'interface intelligente dont la principale qualité est de traiter toutes les séquences d'actions effectuées de manière répétitive sur une machine de communication (extraction et gestion des connaissances, apprentissage automatique et comportemental, mais aussi interaction, classification, prédiction, anticipation et simplification, toutes auto-adaptables).

Le **Chapitre 5** est intitulé "Implémentation et Validation des Prototypes" et nous permet finalement d'explicitier nos implémentations successives et toutes nos validations inhérentes. Soutenue par plusieurs annexes, cette partie débute par la définition des contraintes de plateforme et d'environnement pouvant être associées à nos travaux de développement, de même que par une nécessaire formation sur les matériels logiciels *COM* et *MAPI* directement reliés au monde des télécommunications.

Par conséquent, nous nous sommes penchés sur les détails de l'implémentation et de l'expérimentation, par agents auto-adaptables et indépendants, de toutes nos investigations explicitées jusqu'ici. Celles-ci ont été mises en application sur un outil logiciel dédié au moyen, en devenir, de la messagerie électronique unifiée : l'*ALCATEL Unified Messaging (AUM)*.

En achevant ainsi nos travaux de recherche, d'élaboration et de développement complet de notre nouvel outil informatique, intelligent et autonome, nous avons pris conscience de notre besoin d'évaluer, de valider et de tester l'ensemble des composantes de notre logiciel.

Aussi, suite à d'ultimes justifications et vérifications de scénarios de tests documentés au sujet des multiples phases d'implémentations, nous avons réussi à entériner la programmation de tous nos modèles auto-adaptables. Ceci a été réalisé, non seulement par la formalisation et la validation des critères d'évaluation et de performances des caractéristiques de nos prototypes développés, mais encore au travers de la mise en œuvre complète d'une étude suivie de nos concepts comportementaux sur un panel d'utilisateurs aussi variés et représentatifs que possible.

Facilitant ainsi amplement l'ensemble du travail quotidien de chaque individu, néophyte comme également très expérimenté, et ce en fournissant notamment un soutien convivial, rapide, fiable, efficace et toujours discret si nécessaire, nous aboutissons au prototype d'un complet système intelligent. Tout-à-fait autonome et portable, celui-ci demeure en outre capable de proposer à chaque utilisateur, et même d'opérer par suite seul avec une importante appropriation, l'exécution de tous types d'actions que le système prédit en s'appuyant en permanence sur notre modèle auto-adaptable de l'utilisateur.

En conclusion, ce mémoire de Thèse récapitule les divers constats et résultats obtenus tout au long de nos travaux de recherche. De plus, en tirant un maximum d'enseignements de nos investigations, nous tenterons également de projeter ces dernières vers de nouvelles perspectives de travail, qui nous apparaissent intéressantes et utiles pour l'avenir des interfaces intelligentes dédiées aux terminaux de communication.

Chapitre 1 :

Terminaux de Communication et Interfaces Intelligentes

*L'homme n'est pas encore remplacé,
mais nul ne sait mesurer l'impact exact des machines
sur la collectivité et sur les relations entre individus.*

Jean-Gabriel Ganascia

[Ganascia, 1993]

Cette citation de Jean-Gabriel Ganascia introduit à merveille ce premier chapitre puisque nous allons nous attacher, entre-autres, à y démontrer très précisément le besoin qu'éprouvent les concepteurs de terminaux de communication à présenter une interface orientée vers une prise en compte "intelligente" des utilisateurs potentiels. De plus, nous découvrons également par ces mots, et ce dans une seule et même phrase, tous les quatre principaux concepts ou entités qui vont nous accaparer tout au long de ce mémoire, à savoir l'homme opposé aux machines, et la collectivité ou le "grand public" qui se distingue des individus en particulier.

Ainsi, s'il existe aujourd'hui une très grande diversité de machines, de systèmes d'exploitation, de langages de programmation, d'applications logicielles dédiées, mais aussi et surtout d'utilisateurs, ce qui pose problème, désormais, ce n'est plus tant la capacité de calcul ou de mémorisation de ces machines que notre capacité à leur fournir des instructions cohérentes et pertinentes, puis encore à interpréter les résultats qu'elles nous donnent [Ganascia, 1993].

Point absolument essentiel, voire même "charnière" de ce problème général : le concept fondamental de la "communication" qui asseoit à lui seul les bases de notre présente Thèse.

Une communication de l'homme à la machine et de la machine à l'homme, où nous voyons déjà apparaître des difficultés de programmation et de génie logiciel, de compréhension mutuelle, de convivialité et d'attrayante humanisation aussi. La communication entre machines ensuite, avec de réels soucis de portabilité et de compatibilité de logiciels que nous aurons à résoudre au cours de nos implémentations. Enfin, la communication entre les hommes eux-mêmes, par machines interposées cette fois-ci, est caractérisée par ses problèmes liés à l'impact des récentes technologies de l'information.

En conséquence, nous entamons nos travaux de Thèse par des définitions quant à notre domaine principal des terminaux et applications de communication disponibles à ce stade de nos recherches. Forts de ces indispensables données initiales, nous enchaînons par la découverte et la caractérisation du concept des dernières interfaces les plus évoluées, dites "intelligentes" et avant tout tournées vers la prise en compte permanente de la multitude des préférences de chaque utilisateur.

En effet, après un état de l'art des outils à considérer dans le cadre de notre partenariat avec la société *ALCATEL*, mais également et surtout des marchés concurrentiels concernant ces derniers, nous établissons plusieurs récapitulatifs de leurs qualités et contraintes d'utilisation les situant les uns par rapport aux autres. Ces catégorisations nous permettent alors de constater que ces nouveaux terminaux de communication et applications associées ne peuvent prétendre, à ce niveau, à une quelconque aptitude à l'apprentissage automatique ou au raisonnement approfondi, ni même auparavant à aucune compétence en matière de représentation des connaissances. Ce type d'avantages ou de "capacités" s'avère cependant aujourd'hui de plus en plus sollicités.

Après avoir initié le secteur des "interfaces intelligentes", nous apportons en second lieu un panel conséquent de nouvelles définitions destinées à nous permettre ensuite de concevoir un état de l'art fondamental de ce domaine. Insistant principalement à ce niveau sur l'intégration de notre concept de classification hiérarchique des interfaces intelligentes, nos travaux de recherche exposés dans ce chapitre justifient l'élaboration de différents procédés dédiés au traitement plus convivial, personnalisé et automatisé des habitudes répétitives des utilisateurs.

Combinaison avancée entre l'utilisation réelle des applications "grand public" et l'apprentissage implicite d'un outil dédié à la communication, nos modèles auto-adaptables prendront ainsi l'initiative d'aborder la discussion sur certains concepts que l'utilisateur semble ignorer et dont il pourrait cependant tirer profit dans la situation où il se trouve. En outre, les principaux développeurs ont aujourd'hui tous réalisé que, pour pouvoir vendre un grand nombre de copies de leurs logiciels et faire un maximum de bénéfices, ils doivent également rendre leurs applications facilement utilisables par des personnes n'ayant pas forcément de connaissances très approfondies en informatique [Ibrahim, 1995].

Il ne faudra par conséquent jamais perdre de vue, ce en total accord d'ailleurs avec les remarques de Kristina Höök déjà citées [Höök, 1997], que le surcroît de puissance des ordinateurs n'est pas destiné à asservir l'homme, mais plutôt à véritablement le servir.

1.1. Les terminaux de communication

1.1.1. Introduction

1.1.1.1. Définition du concept

A l'époque déjà bien évoluée qui est la nôtre, dès que nous abordons la notion de « terminal », nous assimilons presque immédiatement cette dernière au concept, général, du transport. Un terminal correspond alors non seulement à un endroit précis où aboutit une ligne de train, d'avion, de métro, de bateau ou encore de bus, mais aussi à un point unique où conduit une ligne d'informations, de données ou de renseignements.

Issu de l'anglicisme, le terminal considéré prend en conséquence toute sa dimension, au sens principal d'une fin ou d'une extrémité de quelque chose, d'une ligne de transport ou de... « communication » en l'occurrence pour nous. Plus spécifiquement, dans le domaine des techniques informatiques, un terminal se définit par un organe d'entrée et/ou de sortie relié à un ou plusieurs ordinateurs grâce à une ligne de transmission de données.

Comme nous venons de le rappeler, la communication demeure liée à la définition, aux fonctionnements successifs et surtout à l'utilité d'un tel terminal, qu'il soit extrait du domaine informatique ou non. Ce moyen de liaison entre deux points, ou encore cet accès à un lieu existant, permet de mettre en relation deux entités, humaines ou matérielles d'ailleurs.

Ainsi, par extension et en prévision des multiples facettes que nous allons nous attacher à étudier, nous pouvons retenir qu'un « terminal de communication » se définit, avant tout, par un processus matériel capable de traiter les phénomènes concernant la possibilité, pour un sujet, de transmettre une information à un autre sujet, ce par un langage naturel articulé ou par d'autres codes formalisés auparavant.

Ensuite, nous regardons plus précisément les diverses machines qui nous entourent et notons que le domaine des terminaux de communication, aussi complexe et déjà bien fourni soit-il, englobe aujourd'hui essentiellement deux grandes catégories de produits : le matériel multi-média et les téléphones [Ikonicoff, 1999].

La première classe de machines énoncée ici se décompose en plusieurs sous-groupes bien distincts comme les ordinateurs, fixes mais également portables, les nouveaux assistants numériques personnels, ou encore tout récemment les télévisions interactives. L'architecture et les composants matériels de ces terminaux demeurent de tout premier ordre pour permettre à ces derniers et aux hommes de communiquer. Ceci avancé, nous verrons maintes fois que c'est aussi grâce à des applications de haut niveau et à des logiciels de qualité que ces appareils vont progressivement réussir, en matière de communication, à s'imposer aux côtés de la téléphonie.

Les téléphones, ou plus généralement le domaine des télécommunications, retient davantage l'ensemble des réalisations mises en œuvre au sein du Groupe *ALCATEL*. A l'image de ce vaste secteur et à l'origine de la fabrication du fameux *Minitel*, *ALCATEL* se place dans les tous premiers rangs mondiaux en matière d'équipements dédiés à la téléphonie. Nous retrouvons trois nouvelles parties qui composent cette catégorie des terminaux de communication, à savoir les appareils fixes (filaire ou non), les téléphones mobiles ainsi que, depuis peu, les terminaux *Internet* ou « *WebPhone* ». A remarquer également que, en accord avec la précédente catégorie du matériel multi-média et comme nous nous sommes attachés à le développer au cours de nos présents travaux de recherche, la richesse et les capacités de la couche applicative apparaissent à ce niveau primordiales pour la conservation d'un caractère évolutif à ce type de machines.

Il faut également spécifier que les deux catégories mises en avant pour une description du domaine des terminaux de communication tendent aujourd'hui à se rapprocher et même à se fondre en de nouvelles classifications davantage orientées vers les progressions les plus récentes du marché. Ainsi est né par exemple, issu de la dernière conjoncture caractérisant nos terminaux de communication, le vaste marché des "terminaux mobiles" [Leclercq, 2001] englobant celui des téléphones mobiles ou GSM (*Global System for Mobile communications*) et celui des assistants numériques personnels (*Personal Digital Assistant* ou PDA).

1.1.1.2. Etat des marchés et contexte évolutif actuel

Face à un commerce des téléphones portables qui s'essouffle après avoir pourtant joué le rôle de locomotive sur le marché de la mobilité, les constructeurs de PDA font aujourd'hui preuve d'une santé insolente [Leclercq, 2001]. Un phénomène étroitement lié au succès récent de

ces derniers terminaux, à la fois légers, puissants, communicants et maintenant en couleur, qui s'introduisent facilement là où une solution à base de PC ne se justifie pas pour des raisons de coût ou d'encombrement.

En déployant une série d'applications très spécifiques, les capacités de communication de ces derniers PDA sont en effet très évoluées [Nemec-Poncik, 2001-2] : synchronisation des données avec un PC, compatibilité des applications et des bases de données avec celles de l'entreprise, transfert de données entre deux PDA (par l'intermédiaire de ports infrarouges), sans oublier la consultation de *mails*. Les assistants personnels de nouvelle génération s'intègrent ainsi pleinement dans le réseau de communication de chaque entreprise.

Cependant, les constructeurs s'attachent encore à étoffer la capacité mémoire de leur offre PDA, par ajout de cartes ou de modules notamment, par la taille et le prix des écrans aussi, ainsi que par le nombre de fonctionnalités : carnet d'adresses, agenda avec une alarme, recherche de contacts, connexion à *Internet*, calculatrice, lecture de *mails*, ...

Les assistants numériques personnels sont très simples à utiliser (pas de menu déroulant interminable pour trouver une application, mais des icônes explicites sur lesquelles il suffit de cliquer). *ALCATEL* en prend bonne note et lance, dès le mois de Septembre 1998, un téléphone mobile doté de nouvelles fonctionnalités d'agenda électronique : le "One Touch Com" [Derouet & Flores, 1998]. Les fonctions "calendrier", "bloc-notes", "répertoire", "calculatrice" et "horloge planétaire" sont regroupées au sein de ce terminal de communication ; au total, ce sont près de mille données qui peuvent y être stockées, soit une importante capacité de 350 Ko pour l'époque. Compatible avec les protocoles *SMTP* et *POP3*, l'accès à la messagerie *Internet* est aussi déjà au nombre des fonctions offertes avec la réception et l'émission de messages électroniques.

En concurrence avec le "Personal Mobile Communicator" de *Sharp* mais aussi le "Communicator" de *Nokia* (cependant déjà pourvu d'un navigateur *Internet*), ces trois téléphones portables correspondent à un nouveau genre dénommé *SmartPhone* : outre leur vocation première de téléphone GSM, ils font également office de *Personal Digital Assistant* ou PDA.

Le rapprochement annoncé entre les deux principales catégories qui composent les terminaux de communication, à savoir les télécommunications et le domaine de l'informatique en général, ne fait que s'amorcer [Finance, 2000]. Comparés aux PC, les PDA ouvrent une nouvelle ère de la "micro-informatique" portable, basée essentiellement sur un succès grandissant dans les entreprises ainsi qu'auprès du grand public [Nemec-Poncik, 2001-2]. Du côté de l'informatique, plusieurs exemples bien avangardistes nous sont également proposés en ce sens.

Ainsi, nous pouvons déjà noter le récent lancement aux Etats-Unis du terminal "eVilla" de *Sony* [Vladyslav, 2001-1], à l'origine d'un autre concept de terminaux de communication, uniquement dédiés à l'utilisation d'*Internet*. En effet, à l'instar d'une majorité des terminaux évoqués, ce dernier modèle s'oriente vers les secteurs les plus innovants du domaine des télécommunications en se focalisant sur la découverte et la manipulation du réseau des réseaux. Avec un système d'exploitation signé *Be*, *Opera Version 4.0* comme navigateur et surtout un écran *Trinitron* 15 pouces orienté en format "portrait", *Sony* intègre le nouveau domaine des "*Internet Appliances*", ou "terminaux d'accès *Internet*", et joue l'alternative de machines simples d'emploi pour les utilisateurs qui ne sont pas mordus d'informatique. Logée dans un écran "retourné" résolument original, l'unité centrale de ces PC simplifiés est dotée en définitive de 64 Mo de mémoire vive et présente, en l'absence de disque dur, deux ports "série" qui supportent le branchement d'un lecteur *Zip* de 250 Mo et d'une imprimante.

Ceci exposé, nous découvrons un second exemple de ce nouveau concept des terminaux d'accès *Internet* : la *Dot.Station* d'*Intel* fonctionnant sous *Linux*. Entièrement configurée à

l'avance et administrable à distance, la machine est composée d'un écran 15 pouces renfermant l'unité centrale et possède, au contraire de la précédente, un disque dur de 4 Go. Si l'idée principale demeure de permettre un accès *Internet* le plus simple possible, l'accent est mis sur un contrôle maximal par le fournisseur d'accès, garant d'une sécurité très appréciable de nos jours pour les données de chacun des utilisateurs.

Les photographies que nous présentons ci-dessous nous amène à prendre visuellement connaissance de ces derniers terminaux de communication mis sur le marché de l'informatique et des télécommunications (**Figure 1.1.1.2**).



Figure 1.1.1.2 : Photographies des derniers terminaux de communication de Sony (à gauche) et Intel, pleinement dédiés au réseau Internet [Vladyslav, 2001-1]

1.1.1.3. Justifications de l'ensemble de nos orientations initiales de travail

Le monde des télécommunications, comme celui de l'informatique qui lui est de plus en plus fortement lié, commence progressivement à assimiler que les multiples appareils, machines et terminaux de communication ne peuvent plus se passer d'une convergence des données vers les désirs de l'homme et ses besoins toujours plus personnalisés.

Comme le soulignait récemment le cabinet d'études *Arthur Andersen*, « le marché de la téléphonie mobile affronte aujourd'hui les difficultés connues, un an plus tôt, par le commerce électronique : celui d'un emballement autour d'une technologie révolutionnaire en perdant les caractéristiques vraiment utiles, à savoir celles qui débouchent sur des applications localisées, personnalisées, accessibles au bon moment, conviviales et peu coûteuses. » [Leclercq, 2001]

Ainsi, à brève échéance déjà, la croissance du marché de la téléphonie mobile va, dans un premier temps, avant tout reposer sur les services de messagerie, de courrier électronique et de publication d'information, puis sur les applications multi-médias et le commerce électronique. Un point de vue une nouvelle fois défendu par le cabinet d'études *Arthur Andersen* pour qui « la valeur ajoutée, en téléphonie mobile, sera apportée par les applications ».

Pour en tirer des enseignements de base visant à initier nos divers travaux de recherche, nous retenons à ce niveau que les acteurs qui participeront le plus à cette nouvelle chaîne de valeur seront, d'une part, les apporteurs de contenus, mais aussi et surtout, d'autre part, ceux qui seront en contact direct avec le client. Ils s'attacheront ainsi à remédier à ses propres problèmes, à ses souhaits quotidiens, personnalisés et ciblés.

Dans cette optique, *ALCATEL* lance, au début de l'année 1994, une gamme complète de standards téléphoniques qui permettent aux entreprises d'intégrer, dans un même système, toutes les formes de communication [Chabbert, 1994].

Grâce au développement d'une architecture de commutation spécifique *ALCATEL 4400* et à l'intégration d'un système d'exploitation temps réel conforme au standard *Unix*, *ALCATEL* espère couvrir, avec une seule gamme, tous les besoins du marché européen dans ce domaine. A la fois autocommutateur, multiplexeur, frontal de communication multi-média (voix, données, images) et plate-forme applicative, la nouvelle gamme de standards téléphoniques numérisés *ALCATEL 4400* permet aujourd'hui d'intégrer, au sein d'un même système, la communication de l'entreprise sous toutes ses formes. Des photographies précises, correspondant aux dernières versions de ces différents produits, nous sont proposées ci-dessous (**Figure 1.1.1.3**).

De plus, il est encore à noter que cet outil facilite aussi le portage de logiciels issus du commerce et la communication avec des outils informatiques (ordinateurs, réseaux locaux, etc).



Figure 1.1.1.3 : *Derniers serveurs de communication et standards téléphoniques numérisés "ALCATEL 4400"* [Alcatel, 2000]

Face au développement phénoménal de la bureautique au cours de la dernière décennie, une intégration des outils informatiques et de télécommunication devenait en effet indispensable. Le standard téléphonique numérisé *ALCATEL 4400* met ainsi pleinement le réseau et les fonctions du téléphone au service de l'informatique. Sur ces bases solides, nous allons pouvoir définir, concevoir, implémenter et valider nos divers modèles auto-adaptables.

Ce contexte de recherche à présent spécifié, nous avons aussi été amenés à prendre part à la naissance et au lancement, sur un marché toujours dédié au "grand public", d'un autre terminal de communication de tout premier plan en ces années 1998 et 1999 de début de Thèse. Nous avons ainsi nommé le « WebPhone » ou « terminal téléphonique *Internet* » d'*ALCATEL*. L'avènement extrêmement rapide de tels terminaux de communication a été avant tout catalysé par la découverte des possibilités incommensurables pour le "grand public" du réseau *Internet*.

Concernant ces travaux sur les terminaux de communication, sur les nouvelles générations de réseau aussi, de services professionnels et résidentiels enfin, les équipes de recherche d'ALCATEL s'impliquent en définitive pleinement dans l'étude des services multi-médias les plus avancés, ainsi que sur l'optimisation de leurs récents produits. Cette dernière nous a été alors confiée, en partie bien sûr, dans le cadre de la Thèse.

Nous précisons en outre que, si ce premier chapitre de notre mémoire constitue la base préalable de notre travail de recherche, il demeure néanmoins tout aussi manifeste qu'il nous a constamment fallu ne pas perdre de vue que ce domaine des terminaux de communication est sujet à une évolution de tous les instants. Le contexte de nos recherches s'est ainsi beaucoup modifié depuis trois ans, et continue encore d'évoluer, même quotidiennement parfois !...

1.1.2. Découverte et état de l'art du "Webphone"

1.1.2.1. Contexte historique , caractéristiques et propriétés générales

Avec la création de la télématique en 1976, et le lancement ensuite du *Minitel* en 1983, apparaît, en France, un marché véritablement très attirant autour de ce dernier et tout nouveau terminal de communication [Belot, 1999]. Près de vingt ans après, une réelle confirmation nous en est apportée, puisque nous pouvons notamment constater que l'on compte de nos jours pas moins de 15 millions d'utilisateurs du *Minitel* pour 6 millions de terminaux.

Afin d'attirer vers eux tous ces utilisateurs intéressés, plusieurs acteurs principaux du marché des télécommunications décident alors de lancer des boîtiers de communication aux fonctions multiples. Ces "tout-en-un", appelés "ScreenPhone" à l'origine, offrent ainsi à la fois un téléphone aux fonctions avancées, les services *Minitel* en couleurs, et des services *Internet* comme la messagerie électronique, le *Web* et le commerce électronique. Le "Webphone" est né.

Cependant, devant un potentiel aussi prometteur, la concurrence entre les différents protagonistes devient immédiatement très vive. En témoigne d'ailleurs le nombre et la richesse des membres de l'ISRF, ou « Internet Screenphone Reference Forum », créé au niveau mondial par *France Télécom* dès le début de l'année 1998. Nous relevons par exemple, et par ordre alphabétique, l'implication de sociétés comme *ACER*, *ALCATEL Business Systems*, *BELGACOM*, *DEUTSCHE TELEKOM*, *ERICSSON*, *France Télécom* bien sûr, *IBM*, *INFOGEAR*, *LOTUS Development Corporation*, *MATRA NORTEL Communications*, *PHILIPS Consumer Communications*, *LUCENT Technologies*, *SAGEM*, *SIEMENS*, *SUN Microsystems*, *SWISSCOM*, *TELIA* et même *US WEST*. Ce moyen de concertation dédié au nouveau "Screenphone" s'installe comme un organisme ouvert, mis en place pour établir un cahier des charges de normes destinées à soulager le développement d'un marché grand public mondial focalisé sur un terminal de base facile à utiliser [ISRF, 1999]. Avec un accès à *Internet* basé essentiellement sur des technologies *Java*, l'ensemble des caractéristiques ainsi déterminées adresseront des standards d'interfaces entre le terminal et l'opérateur de réseau ou encore les fournisseurs de service.

Indispensable à l'établissement de prototypes pour un interfaçage intelligent concernant ce type de terminal de communication, nous nous proposons de nous attarder sur un état de l'art complet des dernières avancées technologiques inhérentes au "Webphone", ce par la description détaillée des principaux terminaux disponibles sur le marché depuis 1998.

L' **Annexe 1** propose une étude approfondie des caractéristiques du "Web Touch One" d'ALCATEL, avec accès facile à *Internet* pour le "grand public" et mis en œuvre en partenariat avec la société *THOMSON Multimedia*. Nous nous attachons également à découvrir et à spécifier les principaux concurrents du précédent terminal de communication, que ceux-ci soient :

- le "WebPhone @net" de *Siemens*, tout premier rival déclaré,
- le "Tel@phone" de *Matra Nortel Communications*, d'origine et de conception entièrement françaises,
- le "Web Audio Phone" de *Samsung*,
- le "@Max" de *COM One*,
- ou encore "l'*Internet* Screenphone" d'*Universal Microelectronics Company (UMEC)*, réalisé en coopération avec l'imposant constructeur d'équipements et de logiciels de télécommunication qu'est *LUCENT Technologies (Bell Labs)*.

Nous évoluerons alors vers l'intéressant constat que, convivial, souvent doté d'un écran tactile couleur, d'un clavier rétractable aussi ainsi que d'un combiné téléphonique intégré, ces terminaux *Web* permettent finalement à tout particulier d'accéder rapidement et plus aisément que jamais au réseau des réseaux que définit *Internet* [Alcatel, 2001].

1.1.2.2. Les Terminaux *Internet* : récapitulatif et premières conséquences

Oscar du meilleur produit lors du salon du *CeBIT* à Hanovre en 1998 [Bessières, 1999], nous avons constaté que le "Webphone" d'ALCATEL, rebaptisé donc "Web Touch One" depuis, est resté une des vedettes au salon *Telecom* de Genève à la mi-October 1999. En outre, l'engouement pour ce type de terminaux de communication dédiés à *Internet* ne diminue pas avec les années. Au contraire, il ne fait qu'accroître avec la succession des présentations qui lui sont régulièrement réservées, organisées pour les diverses professions de spécialistes issus des domaines scientifiques concernés, mais aussi à l'intention du "grand public" [CeBIT, 2001].

Un terminal téléphonique et télématique, relativement bon marché et conçu aussi pour un accès direct à *Internet*, est assurément séduisant. Tout le problème réside ensuite dans son positionnement sur le marché mondial, notamment dans un contexte où les ordinateurs eux-mêmes deviennent de plus en plus accessibles. En plus d'ALCATEL qui se trouve à l'origine du concept du "Webphone", nous avons relevé que *Siemens*, *Matra Nortel Communications*, *Samsung*, *COM One*, *Universal Microelectronics Company*, et ce même en allant jusqu'à *Sony*, ont de fortes ambitions sur ce domaine, quoique plus modestes pour certains groupes comme nous le rappelons plus loin (**Tableau 1.1.2.2**). Mais tous doivent orienter les capacités de ces nouvelles machines de communication vers une proximité, une "personnalisation" et une réelle automatisation dédiées aux besoins d'un utilisateur "grand public". Ce dernier constat revêt alors le rôle primordial des fondements attachés aux théories "d'intelligences" qui nous intéressent.

ALCATEL poursuit aujourd'hui ses recherches dans ces directions, au sein des structures d'*Atlinks*, société partagée (*joint-venture* en anglais) entre ALCATEL et *THOMSON Multimedia*, et qui se place en tête sur le marché mondial actuel des terminaux *Internet* [Alcatel, 2001]. La gamme "Web Touch" commence à répondre à une demande de plus en plus évoluée pour de nouvelles applications destinées au "grand public" : commerce électronique, prévisions météorologiques,

circulation routière, etc. Cette même gamme "Web Touch" constitue aussi une des meilleures alternatives pour satisfaire la demande des utilisateurs sur les marchés verticaux à fort potentiel comme la banque électronique, les assurances, l'hôtellerie ou encore l'industrie automobile. La conjoncture du marché mondial des "Webphone" justifie donc les orientations d'auto-adaptation et d'interfaçage intelligent que nous avons choisies de modéliser.

Nous présentons, sur la page suivante, un tableau bien récapitulatif des caractéristiques significatives, contraintes techniques et capacités d'utilisation extraites des appareils les plus remarquables composant le marché des terminaux de communication *Internet* (**Tableau 1.1.2.2**). A noter qu'une case de ce tableau comportant un tiret signifie que l'information correspondante n'est pas disponible à l'heure actuelle, ou que la société concernée n'a pas souhaité nous divulguer de détails à ce sujet, souvent pour des raisons évidentes de forte concurrence.

Ainsi, au moment où l'ordinateur reste l'outil numéro un des hommes pour se connecter à l'*Internet*, il apparaît de plus en plus sûr que les "Webphones" ont des avantages indéniables, en termes de convivialité comme de simplicité d'accès [Thorel, 1999].

Cependant, nous retenons ici aussi qu'une des principales difficultés de nos recherches se résume par le fait que les machines ne cessent d'évoluer à grande vitesse, en possédant une durée de vie très courte. En conséquence, nous avons constamment dû nous adapter et chercher à établir nombre de concepts évolutifs, incrémentaux et surtout capables de résister aux rapides changements matériels annoncés pour le futur. De plus, nous relevons, à l'image des dernières années que nous venons tous de vivre, que l'avenir nous réserve sûrement des surprises dans ce domaine des terminaux de communication et, par extension, des télécommunications. En effet, nous pouvons espérer que notre soif permanente de technologie, d'innovation, d'automatisation, d'assistanat implicite, de relations humaines et de curiosité interactive aussi, sera toujours comblée et satisfaite. Pour n'en citer qu'un, voici un exemple choisi parmi les plus récents et qui nous entraîne vers cette vision d'un futur surprenant lié aux communications (**Figure 1.1.2.2**).

Figure 1.1.2.2 : Photographie du terminal Internet "iCOM" de la société Interactive Imaging Systems (IIS) [Courrier, 2001]



Ce récent terminal de communication, qui s'apparente très nettement à un simple objet de la vie courante, a la taille d'une grosse télécommande. Au milieu, une dépression en plastique mou sert de pointeur à l'utilisateur. Equipé d'un microphone, d'un haut-parleur et d'une prise de casque, "iCOM" possède une sorte d'oeilleton protubérant [Courrier, 2001]. L'internaute qui y colle son œil perçoit alors une image équivalente à celle d'un écran couleur de 21 pouces, soit pas moins de 60 centimètres de diagonale !

Caractérisée par une mémoire vive de 32 Mo, cette machine futuriste d'*Interactive Imaging Systems* se voit attribuée une autonomie de fonctionnement de plus de 12 heures grâce à une batterie en lithium de dernière génération. Et si le système comprend enfin un navigateur, un lecteur de fichiers musicaux et un gestionnaire de courrier électronique, le tout est connecté à l'*Internet* par un modem, avec ou sans fil.

Tableau 1.1.2.2 : *Récapitulatif des caractéristiques techniques et contraintes d'utilisation émanant des principaux terminaux de communication de type "WebPhone"*

Modèles Composants	<i>ALCATEL</i> Web Touch One	<i>Siemens</i> WebPhone @net	<i>Matra Nortel</i> Tel@phone	<i>Samsung</i> Web Audio Phone	<i>COM One</i> @Max	<i>UMEC</i> Internet Screenphone
Date de Sortie	Automne 1999	-	Décembre 1999	Début 2000	Printemps 2000	Début 2001
Système d'Exploitation	PersonalJava	Windows CE	Windows CE 2.1	pSOS PersonalJava	QNX	-
Control Process Unit	Motorola Power PC 823 66 MHz	-	X86	StrongArm (ISA-1100)	NS Geode Media Gx1 266 MHz	-
Disque Dur	2 Mo (1000 adresses)	100 mails ou 20 fax	-	-	250 entrées	-
Mémoire Vive	32 Mo	-	20 Mo	16 Mo	32 Mo	-
Bouton dédié au Minitel	oui	-	oui	non	oui	oui
Lecteur de Carte à Puce	oui	oui	oui	oui	oui	non
Web	HTTP 1.1 HTML 3.2 SSL 3.0 bouton dédié	JavaScript HTML 3.2 SSL 3.0 bouton dédié	HTTP 1.1 HTML 3.2 SSL 3.0	HTTP 1.1 HTML 3.2 SSL 3.0	JavaScript HTTP 1.1 HTML 3.2 SSL 3.0	JavaScript HTTP 1.0 HTML 3.2 SSL 3.0 bouton dédié
Connexion	33,6 Kbps	33,6 Kbps	33,6 Kbps	56 Kbps Ethernet 10 base T	56 Kbps Ethernet 10/100 base T	56 Kbps
Messagerie	SMTP POP3 IMAP4 diode clignotante	SMTP POP3	POP3 IMAP4	POP3 IMAP4	SMTP POP3 voyant lumineux	SMTP POP3 indicateur lumineux
Ecran	tactile VGA 256 couleurs 640 x 480 7,5 pouces	tactile monochrome 640 x 480	tactile 256 couleurs 8 pouces	tactile 256 couleurs 8,2 pouces	LCD VGA 16 millions de couleurs 640 x 480 8,2 pouces	tactile VGA couleur 7,4 pouces
Clavier	alpha-numérique	externe	"azerty"	"azerty"	alpha-numérique TouchPad	alpha-numérique

Nous relevons de ce tableau récapitulatif que les terminaux de communication de type "WebPhone" et dédiés à "l'accès facile" à *Internet* sont des outils informatiques et télématiques très récents. Les choix effectués par les constructeurs au niveau des systèmes d'exploitation respectivement mis en œuvre se partagent en deux tendances principales : *PersonalJava* de *Sun* et *Windows CE* de *Microsoft*. Cependant, les informations que nous avons pu obtenir au sujet des caractéristiques attenantes aux cœurs-mêmes de ces terminaux, à savoir les puces électroniques (*CPU* ou *Control Process Unit*) et les disques durs utilisés, sont aussi disparates que difficiles à déchiffrer et toujours jalousement protégées. La concurrence entre fabricants demeure féroce et tout-à-fait palpable, mais bien compréhensible au vu des énormes enjeux financiers : le paragraphe 1.1.1.2 abordé précédemment a permis de nous en fournir un premier aperçu.

Ceci dit, une grande majorité des composants du **Tableau 1.1.2.2** montre toutefois que les orientations des terminaux de communication évoluent à grands pas vers davantage de capacités techniques, de commodités et d'aisances pour l'utilisateur. Rares sont les appareils qui ne dépassent pas 16 Mo de mémoire vive, qui ne mettent en avant des boutons dédiés à des fonctionnalités uniques, précises et ciblées, ou même ne font valoir un avanguardiste lecteur de carte à puce. Cette focalisation sur l'entité finale que représente l'utilisateur se traduit même jusqu'aux constantes améliorations apportées aux périphériques classiques tels que l'écran (tactile pour tous, avec une taille et un panel de couleurs aussi étendus que possible) et le clavier (inévitablement alpha-numérique et au rangement pratique pour la plupart).

Enfin, et c'est bien là que tous ces terminaux *Internet* se rejoignent inexorablement, tout a été fait lors de la conception de ceux-ci pour aboutir à de véritables et efficaces moyens de communications pluri-directionnelles : aussi bien au niveau de la rapidité de transfert des données (33,6 Kbps pour les plus anciens, 56 Kbps avec carte *Ethernet* pour les derniers en date) et de la richesse des logiciels de gestion du *Web*, qu'en ce qui concerne plus spécifiquement les qualités et les différents protocoles de messagerie mises à disposition de l'utilisateur. Ainsi, tous les travaux inhérents aux présents et futurs terminaux de communication se voient orientés vers un interfaçage homme-machine de plus en plus évolué et complet, choisi de manière nettement plus judicieuse que dans le cas des divers outils de communication ayant déjà jalonné le passé.

1.1.3. Un second dispositif pour nos recherches : la "Messagerie Unifiée"

1.1.3.1. Introduction et principales caractéristiques

Nous avons déjà maintes fois constaté, au cours de la partie précédente, que les usagers d'*Internet* croient beaucoup en l'avenir de ce média. Cela établi, nous découvrons que, pour eux, l'activité *Internet* est aussi très majoritairement associée à la pratique du courrier électronique (91%) [Cohu-Weill, 2001]. Et c'est seulement loin derrière qu'ils font appel au *Net* dans le cadre de la recherche d'informations, que ce soit pour le domaine professionnel (59%), les loisirs (74%) ou encore la vie pratique (93%). Si les usages plus spécialisés (discussions, forums, jeux en réseaux, achats en ligne) sont cependant moins courants, le téléchargement de fichiers, programmes et musique, fait de son côté un bon score avec 47%.

En outre, nous pouvons noter que le jugement que portent les internautes sur le réseau des réseaux est plutôt positif : en effet, ils apprécient beaucoup l'utilisation pédagogique qu'ils peuvent en faire (75%), ainsi que les possibilités de simplification de la vie quotidienne (70%).

Ils soulignent de plus que le *Web* peut apporter à la création littéraire et artistique, de même qu'aider au développement de l'activité professionnelle à domicile.

Il ressort donc, dès à présent, que la messagerie électronique caracole nettement en tête des diverses applications et manipulations opérées de nos jours sur le réseau *Internet*. Toutefois, si du côté du "grand public" la masse de publicités estampillées "www" dans les magazines et les médias tendrait à nous faire croire que tous les Français sont connectés à l'*Internet*, on navigue là encore entre douce illusion et vaste intoxication [CCI, 2001]. En effet, d'importants besoins en développement et en formation, mais aussi en ergonomie, en convivialité, en proximité, en personnalisation et en auto-adaptation semblent aujourd'hui nécessaires quant à la découverte et à l'utilisation des capacités de communication électronique inhérente à un tel outil planétaire.

Nous ne pouvons que constater, alors, que la messagerie électronique impose une réelle gestion de tous les instants, stricte et rigoureuse. Elle devient tous les jours plus professionnelle et de nombreux organismes cherchent actuellement à rendre l'offre qui lui correspond plus réactive et plus fiable [Bordage & Saiz, 2001].

L'utilisation des *e-mails* est devenue monnaie courante pour les sociétés. Et si la plupart d'entre-elles ne l'utilisent pas encore comme un outil de communication et de "marketing" à part entière, chaque message envoyé véhicule néanmoins l'image et les valeurs de l'entreprise. Si cela induit que le traitement de ces *e-mails*, entrants comme sortants, se doit d'être efficace, il tend également à garantir un service de qualité, dédié à l'ensemble de la clientèle, et constitue le meilleur vecteur pour bâtir une relation forte et durable avec tous les partenaires d'une entité. De nombreux modules, parfois nouveaux, peuvent venir se greffer sur le processus de messagerie : l'infrastructure de gestion des *e-mails* s'enrichit ainsi au fur et à mesure des besoins de l'entreprise. Au départ simple outil de transmission au format "texte", la messagerie assume désormais beaucoup de fonctions commerciales, gère la sécurité ainsi que la confidentialité des transactions, etc. Pour exemple, les anti-virus et autres filtres ont été les tous premiers modules adjoints aux différents serveurs chargés de la réception et de l'émission des messages électroniques. Plus récemment, des plates-formes de "marketing", dédiées à l'envoi en nombre d'*e-mails* et à l'amorce du traitement des *e-mails* entrants, sont venues renforcer ce dispositif.

Mais nous reviendrons sur tout ceci lors de nos prochaines investigations attachées à rendre "intelligente" la messagerie électronique "unifiée" mise sur le marché par *ALCATEL*.

En attendant, il nous reste à définir ce qu'est la notion de "Messagerie Unifiée", seconde catégorie de produits orientés vers le monde de la communication et mis à notre disposition par *ALCATEL* pour accueillir nos travaux successifs d'analyse, de modélisation, de conception et, en définitive, d'implémentation et de validation.

Nous retiendrons alors que le domaine de la messagerie unifiée se trouve être tout aussi récent que le précédent concept relevant des "Webphone", et présente une nouvelle fois de très intéressantes caractéristiques avangardistes dans le cadre de nos recherches. Plus techniquement, ce deuxième secteur issu des catalogues d'*ALCATEL* correspond à une application logicielle dédiée à la messagerie électronique, sur ordinateur comme sur poste téléphonique, et via le réseau *Internet* bien entendu. En outre, elle est qualifiée "d'unifiée" du fait qu'elle regroupe, en un unique produit, le traitement de plusieurs médias distincts comme les *e-mails*, les *voice-mails*, les *fax* ou encore les *SMS* ("Short Messages Services") [Lichtner, 2000], tout en manipulant ces différents messages en une seule et même interface-utilisateur.

D'un autre côté, nous nous devons de signaler qu'il ne faut absolument pas confondre le concept de "Messagerie Unifiée" avec celui de "Messagerie Intégrée". En effet, si ces procédés font tous deux appel à une seule interface-utilisateur, l'architecture de la messagerie unifiée surclasse son homologue puisqu'elle se compose aussi d'un unique lieu de stockage (ou serveur),

d'un unique annuaire ou répertoire pour chaque utilisateur, ainsi que d'un seul administrateur pour tout le système. La messagerie intégrée, au contraire, a elle toujours besoin de plusieurs lieux de stockages différents, de plusieurs répertoires aussi pour un même utilisateur, et surtout d'une administration sur terminaux au moins dédoublée [Alcatel, 2000]. Un inconvénient de taille apparaît pour ce principe de messagerie intégrée, à savoir que ce concept requiert une capacité très élevée au niveau des réseaux locaux d'une entreprise, notamment car il demeure nécessaire de copier l'ensemble des messages et des annuaires entre différents serveurs dédiés.

A présent, nous nous proposons de prendre connaissance des diverses propriétés de l'application de "Messagerie Unifiée" d'ALCATEL, appelée "AUM", de même que de toutes celles composant aujourd'hui le panel de la concurrence d'un tel outil de communication.

1.1.3.2. Contexte général du marché de la "Messagerie Unifiée"

Selon le cabinet d'études *F&H*, le marché des applications de messagerie unifiée devrait atteindre 5 milliards de Dollars en 2005, contre seulement 549 millions de Dollars aujourd'hui [Lichtner, 2000]. Il s'agit là d'une progression pouvant être qualifiée de fulgurante, d'autant plus que ce secteur apparaît dopé par les nouvelles technologies, dont *Internet*, mais surtout par l'explosion des applications sans fil, associant les assistants numériques personnels (PDA) et la téléphonie mobile (GSM) que nous avons déjà étudiés. La personnalisation et la mobilité sont en conséquence les premiers facteurs pour faire décoller ce marché.

Le marché de la messagerie unifiée, solutions destinées aussi bien aux entreprises qu'au domaine du "grand public", demeure en outre très hétérogène. Les offres de messagerie unifiée regroupent des acteurs issus d'univers très variés. Nous y trouvons des éditeurs tels que *Lotus*, *Microsoft* ou *Novell*, des opérateurs comme *France Télécom* et *Cegetel*, des constructeurs (*IBM*, *Sun*), des spécialistes du réseau avec *Lucent*, et des fournisseurs de GSM (*Nokia*, *Ericsson*, *Siemens*, *ALCATEL*). Aucun domaine n'échappe à cette nouvelle vague, et nous pouvons même trouver certains fournisseurs d'accès *Internet* qui allouent aujourd'hui ce type de service.

Fort de ce constat, le cabinet d'études *Ferris Research*, spécialisé dans les nouvelles technologies, distingue deux catégories de fournisseurs [Lichtner, 2000]. Il s'agit alors, d'une part, de ceux qui proposent des solutions sur mesure (*IBM*, *Lotus*, *Siemens*, *Ericsson*, *Lucent*, *Sun*, *Microsoft*, etc) visant surtout les grandes entreprises, et, d'autre part, des éditeurs qui offrent des solutions "clé-en-main" destinées aux plus petites sociétés, mais aussi aux particuliers séduits par les avantages de la messagerie unifiée. Ces éditeurs avancent des plates-formes intégrées pour fédérer la voix, les *fax* et les messages électroniques. Quant aux entreprises concernées, ce sont plutôt celles qui génèrent un important trafic d'informations (services "marketing", consultants, relations publiques, communications, etc). Mais cela peut être également des petites ou des moyennes entreprises qui ont des collaborateurs itinérants...

En conclusion à cette étude de marché, remarquons qu'*ALCATEL* se place sérieusement dans chacune de ces deux précédentes catégories, et fournit des solutions pour les entreprises, sans exception, qui possèdent déjà un système *e-mail* complet et/ou des postes téléphoniques de type "4400" (se référer au paragraphe 1.1.1.3 pour davantage de précisions à ce sujet).

L'*Annexe 1* nous apporte une nouvelle fois le détail des diverses caractéristiques issues de cet innovant domaine, non seulement selon *ALCATEL* au travers bien entendu de "l'*ALCATEL Unified Messaging*" (*AUM*), mais également d'après ses principaux concurrents listés ci-après :

- “SmartPhone Messagerie Unifiée” (*SMU*) de *Novavox*,
- “David Professional” (*Version 6.0*) de *Tobit Software*, éditeur allemand,
- “Enterprise Interaction Center” (ou encore *EIC*), développé par la société américaine *Interactive Intelligence*,
- “Freeplanning” produit par la société française du même nom, créée à Paris par une équipe de quatre professionnels du *Web* et partenaire de *SUN Microsystems*.

1.1.3.3. La "Messagerie Unifiée" : récapitulatif et nouvelles perspectives

Nous pouvons d'emblée retenir à ce niveau que l'offre et les technologies inhérentes au concept de la messagerie unifiée commencent à affluer. Devant ce jeune domaine d'outils de communication électronique qui ne fait encore que poser ses premières bases, se profile une des principales difficultés de nos recherches, à savoir une indispensable adaptation de notre part, associée à une anticipation et à un contexte de travail en permanente évolution.

Par conséquent, à l'image des conclusions établies précédemment sur les “Webphone”, nous nous sommes constamment attelés à intégrer dans nos propres travaux les modifications successives apportées par les protagonistes de la messagerie unifiée. Sans oublier que nous nous sommes aussi régulièrement demandés dans quelles mesures nous pouvions prévoir, avec un maximum de précisions, les avancées des différentes sociétés dans ce domaine en devenir.

Ainsi, malgré un environnement en perpétuelle progression, nous réussissons à extraire plusieurs critères importants et communs aux diverses applications étudiées. En effet, aussi bien *ALCATEL* que *Novavox*, *Tobit Software*, *Interactive Intelligence* et *Freeplanning* s'attachent tous à rechercher, en premier lieu, une réelle facilité d'intégration quant à leurs produits respectifs, notamment en implémentant des applications modulaires sur des architectures “client-serveur” existantes. Nous noterons en plus à ce propos que la majorité des plates-formes que nous avons abordées font appel à *Microsoft Windows* et/ou *Exchange*.

Par suite, les sociétés ciblées présentent au moins un module destiné à la fonction d'administration du système. Avec des options de gestion et d'archivage, les messages multimédias sont traités, qu'ils soient réceptionnés ou envoyés, en visant une forte efficacité et une centralisation générale, comme pour l'uniformisation des messages dans “l'*ALCATEL Unified Messaging*”. D'ailleurs, *ALCATEL* et *Novavox* proposent, de manière conviviale, la possibilité de configurer la messagerie : taille de chaque boîte aux lettres, temps de validité des messages, etc.

Mais la richesse de ces différents logiciels ne s'arrête pas là, puisque nous avons également constaté que nombre d'extensions, toutes plus intéressantes les unes que les autres, nous sont proposées par les sociétés successives. A noter que c'est essentiellement ainsi, avec un cœur théorique et des principes de conception presque identiques, que la distinction entre les applications de messagerie unifiée s'effectuent sur le marché actuel.

A l'image du “Webphone”, le domaine de la “Messagerie Unifiée” se trouve donc devant un avenir très prometteur et capable d'apporter à celle-ci d'importantes améliorations. Dans ce contexte rendu difficile du fait d'une rude concurrence entre les entreprises concernées, il faut souligner que “l'*AUM*” se situe très nettement dans le peloton de tête, grâce aux capacités reconnues et à la grande expérience d'*ALCATEL* en matière d'équipements de télécommunication.

En tête du marché mondial de la mise en œuvre des réseaux télématiques, *ALCATEL* se distingue également de ses poursuivants par la prise en compte dans "l'*AUM*" d'un unique réseau d'administration et d'une architecture d'interfaçage orientée vers les besoins de l'utilisateur.

Dans cette optique, primordiale à notre époque, nous allons chercher à approfondir et à innover les caractéristiques propres au travail répétitif et quotidien de nos utilisateurs. Conforme à la démonstration apportée au niveau des terminaux *Internet*, force est de constater qu'un réel vide s'avère aujourd'hui indispensable à combler quant à une véritable assistance automatique et adaptable aux objectifs de chaque utilisateur, salarié d'une entreprise comme particulier œuvrant en privé. Cette dernière remarque introduit ainsi instantanément tous les principaux fondements qui qualifieront nos futurs apports "d'intelligence" aux derniers logiciels d'*ALCATEL*. Et c'est également dans ce même but que nous concevrons ensuite un ensemble de nouveaux procédés d'apprentissage automatique, capables de remédier en définitive à un manque d'applications de messagerie électronique dédiées à nos comportements d'humains.

Enfin, nous ne pouvons clore cette transition avec l'univers des interfaces intelligentes sans présenter un tableau récapitulatif de toutes les informations remarquables, contraintes et autres capacités techniques issues des applications les plus significatives composant aujourd'hui le domaine de la messagerie unifiée (**Tableau 1.1.3.3**).

Tableau 1.1.3.3 : *Récapitulatif des caractéristiques et contraintes techniques émanant des applications de "Messagerie Unifiée"*

Modèles Composants	<i>ALCATEL</i> AUM	<i>Novavox</i> SMU	<i>Tobit Software</i> David Professional	<i>Interactive Intelligence</i> EIC	<i>Freeplanning</i> Freeplanning
Architecture	Client-Serveur (<i>ALCATEL</i> 4400 PBX)	Client-Serveur	Client-Serveur	Client-Serveur (Java et PBX)	Client-Serveur
Outil(s) d'Exploitation ("Serveur")	MS Windows NT, MS Exchange Server 5.5	MS Exchange	MS Exchange, Novell Groupwise	MS Windows NT	MS Windows
Application de Messagerie ("Client")	MS Outlook	MS Outlook	Lotus Notes SMTP	MS Exchange	MS Outlook, Lotus Notes
Médias Traités	e-mails, fax, messages vocaux	e-mails, fax, messages vocaux	e-mails, SMS, messages vocaux, télécopies	tous types de messages, télécopies	mails, SMS, fax
Modules Logiciels Principaux	Réseau administrateur, Interface- utilisateur, Opératrice Automatique	Serveur, Administrateur, Agent	Fax, Messagerie vocale, SMS, Administration, <i>WebBox</i>	Serveur d'interaction, Interaction administrateur, <i>Interaction Designer</i> , Serveur Vocal Interactif	Partage, Synchroniseur, Visualiseur, Coordinateur, Alerte

1.2. Les interfaces intelligentes

1.2.1. Introduction et situation de la problématique

La question de savoir si une machine pense ou est "intelligente" peut paraître futile. Après tout, il ne s'agit jamais que d'un assemblage matériel juste capable d'obéir aux ordres. En outre, il paraît difficile d'accorder la pensée ou "l'intelligence" à un objet qui n'est pas vivant et qui n'éprouve aucun sentiment. A cette position de bon sens, Alan Turing a répondu, dès 1950, en faisant observer que nous supposons tous l'intelligence chez une personne à partir des quelques phrases que nous échangeons avec elle. De plus, nous ne connaissons les sentiments de cette personne qu'à partir de ce qu'elle nous en dit : nous ne savons pas ce qui se passe à l'intérieur de sa tête. Alan Turing conclut qu'une "entité" qui parviendrait à se faire passer pour un être humain aurait un comportement "intelligent" et devrait donc être considérée comme tel.

Ainsi, de nos jours, plutôt que de chercher si un ordinateur peut être intelligent, les chercheurs essaient de fabriquer une machine capable de se faire passer pour un être humain. En conséquence, il devient indispensable d'axer les travaux sur l'utilisateur. Nous faisons appel aux systèmes dits "auto-adaptatifs", destinés à traiter les besoins individuels et changeants des utilisateurs, voire même présentant assez "d'intelligence" pour agir sur le comportement de ceux-ci [Benyon, 1998]. Cette caractéristique entraîne la définition d'un indispensable processus nommé "prédiction d'actions".

Mais... quelles sont justement les motivations et les grands besoins actuels en matière de prédiction et d'anticipation intelligente des actions ? Comment sommes-nous prévisibles ?...

[Davison & Hirsh, 1998] nous apprennent à ce sujet que certains parmi nous affichent des modèles d'actions à travers tout ce qu'ils font, la plupart des actions arrivant sans pensée consciente. Quelques modèles sont déjà répandus, et sont enseignés en tant que règles, comme lire de gauche à droite, ou conduire du bon côté de la route. D'autres modèles sont une fonction de notre style de vie, comme acheter de la pizza sur le chemin du travail à la maison tous les vendredis, ou programmer le magnétoscope pour enregistrer notre comédie favorite chaque semaine. Beaucoup sont un résultat de la façon dont les interfaces sont conçues, comme le modèle du mouvement de notre doigt composant sur un téléphone un numéro que nous faisons souvent, ou comment nous pouvons nous connecter sur notre ordinateur, vérifier le courrier, lire les nouvelles, et visiter notre site *Web* favori pour les derniers résultats sportifs. Comme les ordinateurs pénètrent de plus en plus nos vies, le besoin d'un système capable de s'adapter à l'utilisateur, sans être explicitement programmé par le système du concepteur, devient évident.

Et [Davison & Hirsh, 1998] de renchérir qu'une voiture pouvant offrir un conseil sur la conduite des routes est utile. Une autre pouvant deviner notre destination (comme un magasin de pizzas parce qu'on est vendredi et que nous quittons le travail) est susceptible d'être considérée encore plus pratique, particulièrement si nous n'avons pas eu besoin de programmer cette information. La capacité à prévoir l'action suivante de l'utilisateur permet au système d'anticiper les besoins de celui-ci (peut-être à travers une exécution spéculative ou intelligente), de s'adapter et d'améliorer les habitudes de travail de l'utilisateur (comme automatiser les tâches répétitives).

Nous découvrons par conséquent que la gestion de notre vie de tous les jours se règle bien souvent suivant une multitude d'ordinateurs, très nombreux, mais également si discrets que nous remarquons à peine qu'ils existent [MacIntyre & Feiner, 1996]. C'est ce qui se caractérise en d'autres termes par une "informatique omniprésente" ("*ubiquitous computing*" en anglais).

Ainsi, si la terminologie “d’interface intelligente” présente *a priori* un concept de travail aussi complet que complexe, elle s’attache au propre domaine de l’intelligence artificielle. Cependant, s’il apparaît comme acquis qu’une interface se doit d’être à la fois attirante et intéressante pour les novices (apprentissage et patience, aides et soutiens multiples, etc) comme pour les habitués (rapidité d’exécution, raccourcis et compléments de travail, etc), il n’en est pas de même, loin s’en faut, de la définition de la notion d’intelligence. En effet, ce dernier concept demande et mérite à l’heure actuelle une description nettement plus fine, pointue et rigoureuse.

Nous allons donc chercher à établir une définition qualitative de l’interfaçage intelligent, suivi d’une nouvelle hiérarchisation et d’une classification taxinomique, à l’aide notamment d’un arbre hiérarchique d’informations et de données recueillies au niveau d’un état de l’art avancé.

1.2.2. Définition qualitative d’une interface intelligente

1.2.2.1. Description de toutes les caractéristiques remarquables

Il s’avère, en tout premier lieu, que le substantif “interface” corresponde à une vitrine physique, un support technique ou une architecture matérialisée et manipulable directement par l’utilisateur sur une machine définie. De son côté, l’adjectif “intelligent” fait davantage référence à une qualité reconnue à l’interface considérée, une caractéristique subjective qui demande une définition rigoureuse, tant elle reste aujourd’hui plus ou moins justement utilisée et implémentée.

De plus, l’amélioration d’une interface passe obligatoirement par l’intégration d’une intelligence la plus adaptative, voire auto-adaptative, possible [Benyon, 1993]. Il nous apparaît donc indispensable d’établir la définition détaillée d’une “interface intelligente”, caractérisation aussi pratique et orientée vers les utilisateurs “grand public” des terminaux de communication.

Des termes composant l’expression “interface intelligente”, c’est avant tout le second qui appelle une définition explicite. Cependant, nous devons garder à l’esprit que des propriétés importantes sont requises aujourd’hui pour une interface homme-machine [Palanque & Bastide, 1994]. Ces indéniables attributs regroupent, en plus d’une absence de situations insolubles (que l’on nomme plus précisément *deadlocks* en anglais) et d’une prédictabilité de chaque commande (traitement identique dans un même contexte donné), la réinitiability d’une interface (possibilité de toujours atteindre l’état initial ou un état prédéfini) et la disponibilité d’une commande, ce à tout moment (commande d’aide par exemple). En outre, la succession des commandes doit être fidèle aux actions de l’utilisateur (respect de l’ordre de lancement des commandes).

En pratique, toutes les interfaces graphiques de qualité sont du type *WIMP*, ce sigle anglophone signifiant *Windows, Icons, Menus* et *Pointing*. Les interfaces se caractérisent alors par la recherche d’une facilité d’utilisation, d’une concision, d’une cohérence et d’une flexibilité vis-à-vis de ses utilisateurs tendant à être améliorées. Toutes requièrent également un contrôle efficace, c’est-à-dire l’intégration d’applications relevant de la “programmation par événements” où l’utilisateur reste maître de l’interaction tout au long de la session de travail. Il demeure primordial pour cet utilisateur de pouvoir visualiser, à tout moment, l’ensemble des commandes mises à sa disposition par l’application concernée. Enfin, toute action licite doit rester activable et être présentée comme telle à l’utilisateur, ce qui confère une grande liberté d’action et un niveau de guidage important, les actions illicites étant clairement inactivables [Palanque & Bastide, 1995].

En laissant le travail sur l'architecture, la parole, la biométrie (empreintes digitales et oculaires, visages et physionomies de l'utilisateur, etc), la multi-modalité ainsi que la gestuelle, nous nous attèlerons aux mécanismes d'apprentissage automatique inhérents et indispensables à de telles interfaces intelligentes, et notamment en ce qui concernent les différentes habitudes et autres réactions de l'utilisateur en matière de communications télématiques.

En conséquence, une "interface" se traduit par un système spécialisé de représentation des connaissances [Benyon & Murray, 1993-1], un dispositif d'interaction entre l'homme et le produit [Bonner, 1997], un système interactif [Schlimmer & Hermens, 1993] ou encore une secrétaire de communication d'un utilisateur dans un environnement bureautique [Nangle, Cunningham & Evans, 1998]. Elle est en outre toujours chargée de mettre en place le dialogue entre l'utilisateur et le système [Höök, 1997], de présenter une maîtrise à toutes épreuves (avec une notion primordiale de robustesse) et de gérer de manière compétente les tâches de l'utilisateur [Shneiderman, 1997].

Cette interface devient alors "intelligente" et intègre une certaine mesure ou capacité d'intelligence [Benyon, 1993] dès l'instant où elle travaille avec des méthodes d'Intelligence Artificielle dans la perspective de l'utilisateur [Whitehead, 1998] et incorpore une ou de préférence plusieurs, voire un maximum des "fonctionnalités" que nous listons maintenant. Ces dernières permettent donc la caractérisation d'une intelligence mise en exergue par une interface :

- Adaptation (c'est-à-dire capacité à faire face aux changements [Mataric, 1994]) du système à son utilisateur [Höök, 1997], en agissant, en travaillant sur le comportement de l'utilisateur [Benyon & Murray, 1993-1] de manière *a priori* au vu d'une situation ou *a posteriori* au vu d'un résultat ou d'un comportement [Garbay & Maitre, 1997] ; mais aussi adaptation de tous les messages d'erreurs, de conseil et d'explication au niveau global de connaissance et de compétence de l'utilisateur [Whitehead, 1998].
- Automatisation et par conséquent auto-adaptation [Benyon, 1993], exécution de fonctions sélectionnées indépendamment du contrôle immédiat de l'utilisateur, bien que sous l'entière conduite des requêtes précédentes de l'utilisateur [Whitehead, 1998].
- Autonomie par compréhension de l'état courant de son environnement et action indépendante pour faire progresser les buts [Akoulchina & Ganascia, 1997], agissement simultané et sans intervention de l'utilisateur, que ce dernier soit occupé ou non [Lieberman, 1997].
- Modularité et appropriation [Benyon & Murray, 1993-1], en traitant les besoins individuels et changeants des utilisateurs [Benyon, 1998], en suggérant des solutions spécifiques pour des tâches particulières de l'utilisateur, avec les raisons pour lesquelles elles ont été considérées comme appropriées et en réalisant une réponse ou une action appropriée pour l'utilisateur selon un certain degré d'initiative [Whitehead, 1998].
- Interprétation des directives de l'utilisateur [Akoulchina & Ganascia, 1997], de ses buts ou de ses actions d'entrée dans un environnement ou contexte particulier, tout en fournissant aussi une réaction, une sortie compatible avec un modèle mental d'utilisateur [Bonner, 1997].
- Aide à l'utilisation de systèmes de plus en plus complexes [Höök, 1997], à la réduction des erreurs de l'utilisateur [Schlimmer & Hermens, 1993].

- Mise en valeur de l'expérience des utilisateurs face aux nouvelles possibilités logicielles offertes [Höök, 1997] par la facilité, l'aisance et la rapidité à comprendre puis apprendre les fonctionnalités de base du système initial, de même que le plaisir à assimiler les caractéristiques avancées [Shneiderman, 1997].
- Simplicité de l'application en temps réel [Höök, 1997], puisque les utilisateurs peuvent immédiatement voir si leurs actions ont atteint leurs buts [Shneiderman, 1997].
- Individualisation et personnalisation des systèmes [Höök, 1997], retrouvées notamment pour les experts dans la rapidité à exécuter un large éventail de tâches et par la possibilité de définition de nouvelles fonctions et caractéristiques [Shneiderman, 1997].
- Intégrité et compréhension de l'information [Höök, 1997], accessibilité universelle d'un support compréhensif et de haute qualité [Stephanidis, 1997].
- Prédiction active de ce que l'utilisateur va écrire, anticipation par présentation d'alternatives sans que l'utilisateur n'ait à les demander [Schlimmer & Hermens, 1993].
- Non-restriction et non-insistance [Akoulchina & Ganascia, 1997], contrôle et ajustement par l'utilisateur, en laissant ce dernier libre de choisir d'ignorer les actions prévisibles et de définir manuellement les paramètres du système [Schlimmer & Hermens, 1993]; confiance et maîtrise pour l'utilisateur puisqu'il est l'initiateur des actions, qu'il cherche à contrôler et que les réponses du système sont dans ce cas prévisibles [Shneiderman, 1997].
- Surveillance de l'interaction [Stephanidis, 1997] et identification de certaines actions de l'utilisateur comme des erreurs, protection de l'utilisateur des conséquences de ces erreurs, explications et conseils pour aider l'utilisateur à les éviter [Whitehead, 1998].
- Tolérance de la variation et de l'imperfection des entrées de l'utilisateur, pas de rejet de commandes ou de données du à des erreurs triviales [Whitehead, 1998].
- Conseil de l'utilisateur sur des méthodes appropriées pour atteindre ses buts, commentaire des méthodes adoptées par l'utilisateur dans le passé [Whitehead, 1998].
- Assistance active [Lieberman, 1997] et personnalisée qui "regarde par-dessus l'épaule de l'utilisateur", qui apprend ses centres d'intérêts afin d'agir sur son comportement [Akoulchina & Ganascia, 1997], par compréhension riche et incorporation d'informations sur la connaissance et l'ensemble des croyances de l'utilisateur [Ramscar, Pain & Lee, 1997].
- Apprentissage (c'est-à-dire adjonction de connaissances [Whitehead, 1998]) non-visible des habitudes et des préférences de l'utilisateur [Akoulchina & Ganascia, 1997], en prenant l'initiative d'aborder la discussion de certains concepts que l'utilisateur semble ignorer et dont il pourrait tirer profit dans la situation où il est [Ibrahim, 1995], et en profitant également du temps de réflexion et de l'information que l'utilisateur fournit librement, sans avoir recours à une interaction séparée [Lieberman, 1997].
- Convivialité par la facilité d'utilisation de l'interface, la qualité de la documentation accessible et la clarté des messages d'erreurs fournis par le système en cas de fausse manipulation ou de situations anormales [Ibrahim, 1995].
- Efficacité des opérations tout en dépendant au minimum des données provenant de l'utilisateur [Nangle, Cunningham & Evans, 1998].

1.2.2.2. Dégagement d'une définition exhaustive d'interface intelligente

Sous une image résumant nos concepts de manière déjà très intéressante, une interface intelligente peut être abordée comme un apprentissage des caractéristiques de communication de l'utilisateur et une construction d'un profil-utilisateur correspondant qui va automatiquement capturer les préférences immédiates de l'utilisateur [Nangle, Cunningham & Evans, 1998]. Nous retiendrons ainsi l'intitulé à suivre pour définir une "interface intelligente" (*Définition 1.2.2.2*).

Définition 1.2.2.2 : *Enumération des caractéristiques permettant la qualification de l'intelligence d'une interface*

- aide et assistance
(réduction des erreurs, messages de commentaires explicatifs et conseils)
- auto-adaptation comportementale
(adaptation automatique aux caractères et habitudes de l'utilisateur)
- auto-adaptation "informationnelle"
(apprentissage et assimilation des intérêts et préférences de l'utilisateur)
- automatisation
(adaptation et assistance permanentes, prédiction et surveillance actives)
- autonomie, initiative et efficacité de progression
(vers les buts courants)
- classification et parcours d'informations
- convivialité
(aisance d'utilisation, qualité et clarté des messages)
- création et maintenance d'un modèle d'utilisateur
- évolutivité individualisée
(rapidité de travail et d'apprentissage, personnalisation des outils)
- intégrité et accessibilité universelle de l'information
- interprétation et compréhension des actions de l'utilisateur
(explicites comme implicites)
- modularité et appropriation
(traitement des besoins individuels)
- non-restriction et non-insistance
(liberté de contrôler chaque automatisation)
- prédiction et anticipation par présentation d'alternatives
- réactivité et réponse à l'utilisateur
- suggestion de solutions spécifiques et personnalisées
- simplicité et humanisation de l'assimilation des fonctionnalités
(initiales et avancées)
- surveillance et sécurité des données
(protection des conséquences d'erreurs)
- temps réel
- tolérance
(variation et imperfection des commandes de l'utilisateur).

Nous parvenons maintenant à mieux saisir l'intérêt de premier plan que représente une implantation d'intelligence dans les terminaux de communication "grand public" d'ALCATEL. Il s'agit de "concerner" l'utilisateur de manière pertinente en cherchant à faire venir la machine vers l'être humain [Blandet & Bohrer, 1998]. Il est également intéressant, voire indispensable, que l'utilisateur sente la présence d'une aide implicite (et constante) provenant de la machine avec laquelle il communique.

Nous rejoignons l'idée intuitive de la problématique énoncée en introduction, à savoir la fameuse interrogation du "pourquoi les hommes auraient-ils à s'adapter aux systèmes, alors que les systèmes devraient s'adapter aux hommes ?" [Höök, 1997]. Il ne faudra, en effet, jamais perdre de vue qu'un système n'est pas intelligent si trop de perspicacité, de compréhension et d'efforts sont demandés à l'utilisateur [Nangle, Cunningham & Evans, 1998].

1.2.2.3. Vers une hiérarchisation des interfaces intelligentes

Une interface dite intelligente peut finalement présenter plus ou moins d'intelligence, de même qu'être nuancée au niveau de ses capacités conceptuelles. A présent, les différents critères caractérisant les interfaces intelligentes étudiées au cours de nos recherches sont au nombre de vingt. Or, rappelons que nous considérons qu'une interface intelligente se doit d'utiliser au moins une, ou, nettement mieux, un maximum de ces vingt "fonctionnalités" listées auparavant.

Par conséquent, nous proposons d'établir une hiérarchie des interfaces intelligentes selon qu'elles incorporent plus ou moins de fonctionnalités "intelligentes", plus ou moins de nos vingt critères d'intelligence, ce qui correspond alors à plus ou moins d'intelligence pour une interface quelconque donnée. La hiérarchie, que nous nommerons à l'avenir *HII* (Hiérarchie des Interfaces Intelligentes), répond donc à la règle de calcul suivante :

Définition 1.2.2.3 : *Spécification du calcul fonctionnel régulant notre concept d'hiérarchie des interfaces intelligentes*

Soient les 20 critères d'intelligence issus de notre précédente **Définition 1.2.2.2** ainsi que w une interface donnée quelconque.

Le *degré ou niveau d'intelligence* $H_{II}(w)$ est alors défini de manière unique par

$H_{II}(w)$ = nombre de critères d'intelligence complètement incorporés dans w .

Grâce à cette nouvelle hiérarchie, que l'on peut également qualifier d'échelle ou encore de classification, nous pourrions à présent aisément définir et distinguer les différentes interfaces par le simple calcul de leur "degré ou niveau d'intelligence", respectif et aussi unique. Nous obtenons une catégorisation applicable aux multiples interfaces intelligentes existantes et à venir.

Le tableau à suivre nous en donne un complet aperçu (**Tableau 1.2.2.3**).

Tableau 1.2.2.3 : *Détail et mise en valeur des catégories d'interfaces intelligentes issues de notre définition d'une hiérarchie adaptée à l'interfaçage homme-machine*

Valeurs de HII	Classes de l'interface
0	Intelligence inexistante
1-4	Intelligence très faible
5-8	Intelligence faible
9-12	Intelligence moyenne
13-16	Intelligence forte
17-19	Intelligence très forte
20	Intelligence "parfaite"

Nous possédons ainsi une méthode pour l'évaluation et la hiérarchisation de chacune de nos interfaces intelligentes, méthode à la fois fiable et pratique comme nous en apporterons de nouvelles preuves dans les diverses pages à venir. Nous allons en effet nous attacher à présent à l'appliquer aux interfaçages que nous aborderons, déjà existants de même qu'encore à l'étude.

De plus, une troisième de nos définitions relatives à nos interfaces intelligentes, à savoir une caractérisation cette fois-ci taxinomique, va nous permettre de compléter cette catégorisation des divers interfaçages issus de nos principales références bibliographiques. Le classement devra cependant être pleinement évolutif en prévision des nouvelles informations qui seront à prendre en considération au cours de nos recherches futures.

1.2.3. Définition taxinomique d'une interface intelligente

L'éventail de nos références bibliographiques concerne tous les types d'articles, livres, tutoriels informatiques, diaporamas, supports de cours, compte-rendus, rapports techniques et même parfois sources brutes de programmes. Toutes ces références ont été obtenues dans les différentes bibliothèques strasbourgeoises et alliées, mais aussi grâce aux nombreux moteurs de recherche mis à notre disposition par le réseau *Internet*.

Ainsi, nous avons mis en avant cinq critères entraînant une représentation taxinomique de notre bibliographie. Nous devons insister sur le fait qu'il s'est avéré impossible d'obtenir des intersections vides entre les différents concepts et sous-concepts détaillés dans les pages à suivre. Des caractéristiques communes apparaissent en effet de manière indélébile entre eux et nous ont poussées à parler de dominantes respectives, qu'elles soient principales ou encore secondaires.

Voici donc, au travers des prochains schémas que nous avons constamment ajustés au cours de la Thèse, l'ensemble des arbres hiérarchiques traduisant les informations que nous avons réussies à extraire lors de nos diverses et successives recherches sur le vaste domaine des interfaces intelligentes (**Figures 1.2.3.1 à 1.2.3.6**).

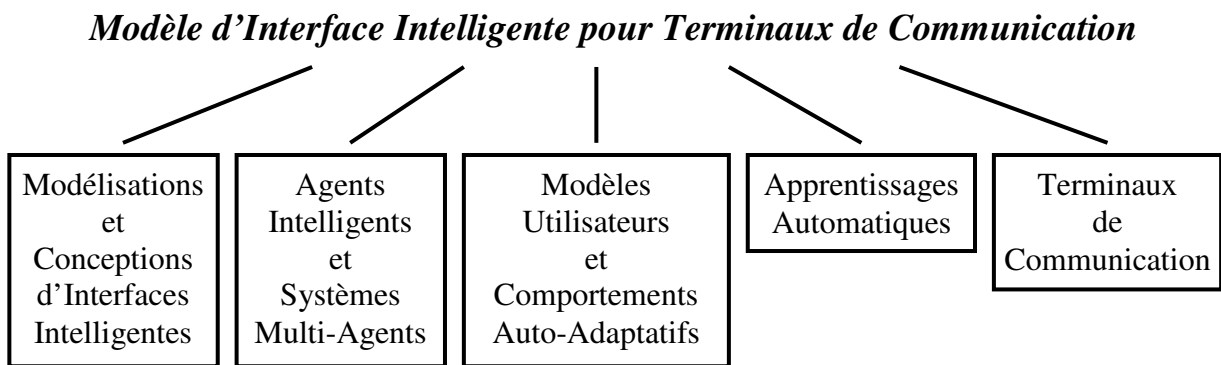


Figure 1.2.3.1 : *Arbre principal détaillant la racine et le premier niveau de la définition taxinomique dédiée aux interfaces intelligentes*

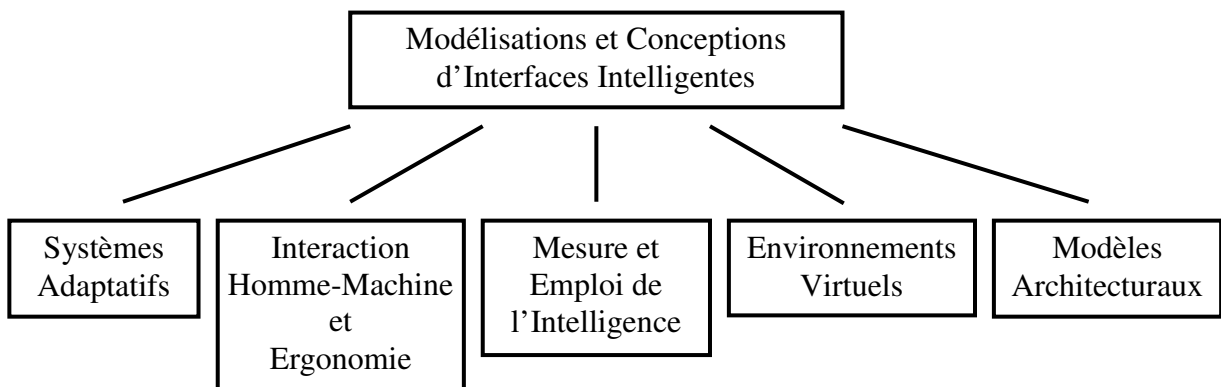


Figure 1.2.3.2 : *Premier arbre du second niveau de la définition taxinomique dédiée aux interfaces intelligentes et exposant leurs différentes modélisations et conceptions remarquables*

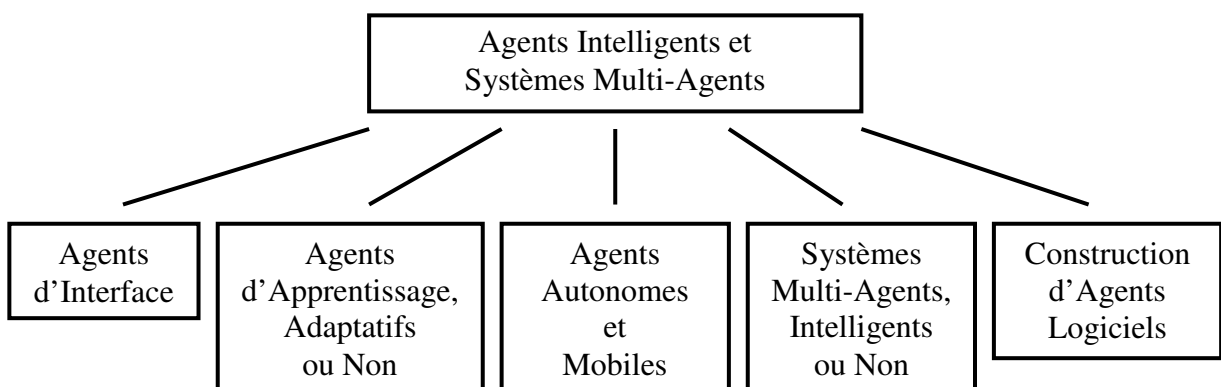


Figure 1.2.3.3 : *Arbre du second niveau de la définition taxinomique des interfaces intelligentes, détaillant les agents intelligents et systèmes multi-agents*

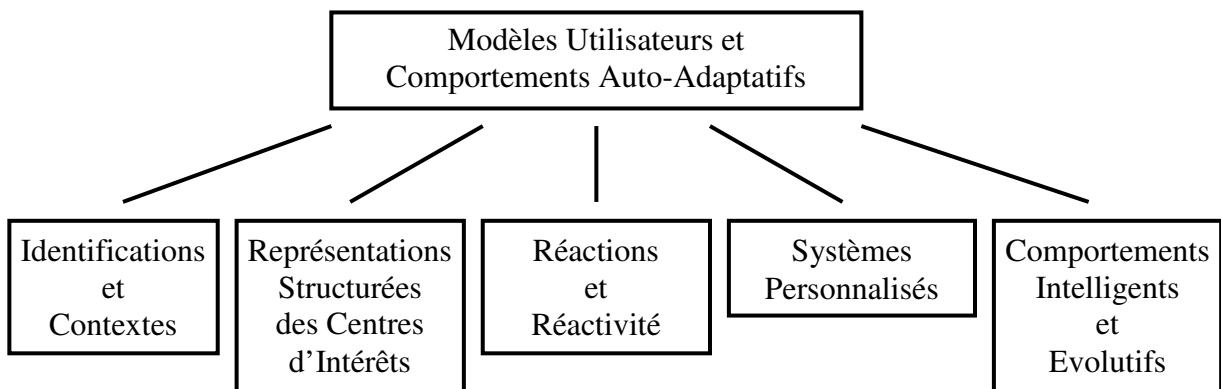


Figure 1.2.3.4 : *Troisième arbre du second niveau de la définition taxinomique et spécifiant les modèles d'utilisateurs et comportements auto-adaptatifs*

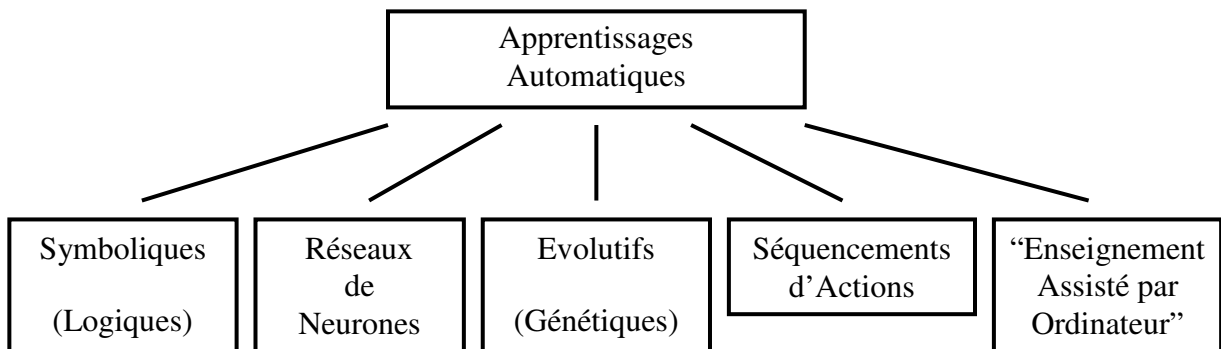


Figure 1.2.3.5 : *Quatrième arbre du second niveau de la définition taxinomique dédiée aux interfaces intelligentes et détaillant l'ensemble des concepts envisageables pour traiter l'apprentissage automatique des séquences d'actions émanant de chacun de nos utilisateurs*

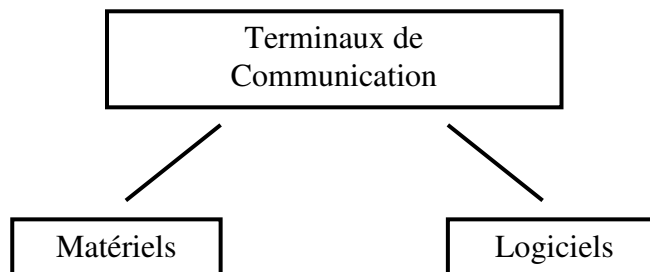


Figure 1.2.3.6 : *Dernier arbre de la définition taxinomique dédiée aux interfaces intelligentes faisant état des différentes informations apportées par les terminaux de communication, d'ALCATEL comme de la concurrence*

Pour conclure, nous noterons encore que nous avons préféré nous limiter à un nombre de niveaux égal à deux, ce essentiellement pour des raisons de généralité et de pertinence des données recueillies. Nous ne souhaitons en effet pas nous restreindre abusivement quant aux références prises en compte. Les documents considérés demeurent certes multiples à ce stade, mais sont tous susceptibles de fournir d'intéressantes informations pour nos travaux. Nous constatons également que, dès le premier niveau de notre définition, apparaît une quantité impressionnante de domaines disparates, théoriques et pratiques d'ailleurs, ayant rapport avec notre sujet de Thèse. Cela rejoint et confirme la forte nécessité que nous avons mise en avant de débiter nos recherches par la spécification rigoureuse d'une définition qualitative et hiérarchique de la notion d'interfaces dites "intelligentes".

Nous apercevons à présent l'intérêt que nous avons eu à décrire, approfondir et spécifier les principales interfaces intelligentes existant dans la littérature scientifique. C'est ce que nous allons développer, non sans omettre, pour chacune de nos références étudiées, de valoriser sa position dans notre système de classification hiérarchique des intelligences d'interfaces.

1.2.4. Etude des références remarquables d'interfaces intelligentes

1.2.4.1. Introduction et justification de telles études de cas

Un des plus importants volets de notre problématique de Thèse, grandement attendu par les différentes équipes d'ALCATEL qui ont soutenu nos recherches, peut être traduit et exprimé de la manière interrogative suivante : que peuvent apporter les divers mécanismes d'apprentissage automatique, les traitements de séquences d'actions et tous les autres agents intelligents aux applications actuelles de l'entreprise ?

Afin de répondre au mieux à cette question, nous allons étudier les références que nous avons estimées majeures. Ces publications sont aussi les plus significatives dans leurs domaines respectifs, ainsi qu'en constante relation avec les orientations de recherche de nos laboratoires associés (Départements d'ALCATEL, Groupe de Recherche en Intelligence Artificielle du LSIIIT).

Nos prochaines investigations ainsi définies, de même que développées en accord avec les différents concepts théoriques mis en avant au sein des références abordées, se verront toutes constamment évaluées et insérées dans notre processus d'Hiérarchie des Interfaces Intelligentes. Nous en résumerons ensuite les caractéristiques à l'aide d'un récapitulatif dédié à l'ensemble de nos références bibliographiques étudiées, ce qui nous permettra de conclure ce chapitre par l'analyse et l'extraction des diverses informations nécessaires à la poursuite de nos recherches.

Ainsi, par le biais de nos premiers états de l'art traitant du domaine des terminaux de communication et de celui des interfaces intelligentes, nous réussirons à spécifier toutes les grandes caractéristiques de notre principal sujet de recherche qu'est l'interaction homme-machine. Une analyse du réalisme (lui-même lié aux paradigmes d'interaction) ainsi que de la plasticité (concernant cette fois les aspects plus logiciels de l'interaction) nous entraîne enfin directement vers nos principaux buts d'offrir, à chacun de nos utilisateurs, une téléprésence plus vivante tout en veillant à la protection de leurs espaces privés respectifs [Coutaz, 1998].

1.2.4.2. L'aide automatique et les tutoriels informatiques

La diversité se situe comme une des principales caractéristiques en matière d'interaction homme-machine. En effet, elle peut se décrire selon quatre composants différents [Benyon, 1998] : descriptions des utilisateurs (connaissances, expériences, niveaux sociaux et culturels, états d'esprit, ...), du travail effectué (vitesse, efficacité, esthétique, précis, spécifique, large, ...), de l'environnement (encadré, isolé, confortable, hostile, ...) et des systèmes informatiques considérés (interfaces, matériels, ...). Nous découvrons ainsi la notion primordiale de "système adaptatif" capable d'effectuer une analyse implicite des connaissances et capacités intellectuelles de la population des utilisateurs, de l'environnement des opérations et de la modélisation du domaine, pour ce qui concerne l'utilisateur tout comme pour l'interaction d'ailleurs.

Le développement de tels systèmes adaptatifs nécessite alors trois principes distincts [Benyon & Murray, 1993-2], à savoir l'identification tout d'abord des caractéristiques du système à rendre adaptatif (domaine d'action, niveaux de détails, ...), la définition ensuite de la variété des utilisateurs et des environnements que le système est amené à rencontrer, et enfin, l'obtention des données pour le modèle de l'utilisateur (interrogation explicite des usagers, inférences implicites émanant de l'interaction, ...). Dans ce contexte, les systèmes d'aide automatique, définis aussi d'outils en mode actif, se caractérisent par l'observation de l'activité de l'utilisateur et la possibilité d'intervention courante ou sans requête de ce dernier. A force de suggestions notamment, ces procédés s'attachent à assister l'utilisateur vers la réalisation d'un but [Bruillard, 1997]. Quant aux tuteurs informatiques ou modes tutoriels, ils se distinguent en ayant la lourde tâche de faire découvrir un concept donné à tout utilisateur considéré comme néophyte.

[Benyon, 1993] nous propose une première méthode composée de cinq phases d'analyse, étapes exprimées en anglais et que nous sommes amenés à considérer pour la conception de systèmes adaptatifs et le développement proprement dit "d'Interfaces-Utilisateur Intelligentes" :

- *functional analysis* (établissement des fonctions principales du système).
- *data analysis* (compréhension et représentation des structures de données de l'application).
- *task knowledge analysis* (informations demandées à l'utilisateur par le système, à savoir la stratégie de recherche requise, l'entrée des données, le modèle mental affecté, ...).
- *user analysis* (connaissances et capacités intellectuelles de la population des utilisateurs à laquelle le système est sensé répondre ; la population initiale étant analysée et classifiée selon les aspects dérivés des quatre autres phases d'analyse de la méthode).
- *environment analysis* (environnement dans lequel le système doit opérer, incluant les aspects physiques et les caractéristiques logicielles comme la quantité et le type de supports-utilisateur nécessaires).

Des précisions qualitatives, au niveau notamment de l'architecture de base d'un tel système adaptatif, peuvent également rapidement y être associées [Benyon & Murray, 1993-1], comme la définition d'un "modèle du domaine" qui se doit d'être un modèle de l'application à rendre adaptative maintenu selon trois niveaux distincts (description des tâches, couche physique et couche logique). De son côté, le "modèle de l'interaction" correspond à un processus du système avec lequel ce dernier va interagir, tout comme le "modèle-utilisateur" s'attache à prendre en compte l'utilisateur par des mécanismes d'inférence, d'évaluation et d'adaptation.

L'interface devient ainsi intelligente, ne serait-ce que par une adaptation automatique à ses utilisateurs, à l'image de celle mise en avant par [Harrington, Banks & Santos, 1996] et qui va jusqu'à conseiller l'utilisateur selon la perception de ses besoins et la fréquence des activités pratiquées. Dédicée aux systèmes-experts génériques montrant des structures destinées à gérer et faire abstraction des complexités d'une interface à domaine indépendant, ces seconds travaux associés à un examen minutieux de l'architecture conceptuelle d'une interface-utilisateur intelligente fournissent à l'utilisateur un environnement de travail pratique et intuitif. Quatre catégories d'utilisateurs, propres au domaine des systèmes-experts, ressortent de ce contexte :

- *application user* (habilité à interroger le système-expert afin d'obtenir une donnée appropriée à l'élaboration d'un travail pratique).
- *application expert* (extrêmement bien informé dans le domaine où l'application est utilisée et également responsable de l'adjonction de nouvelles informations dans le système existant).
- *knowledge engineer* (spécialiste en acquisition des connaissances et bases de données, intermédiaire entre les experts et la machine).
- *computer scientist* (chargé du développement, de la conception et de la maintenance du système).

Et nous aboutissons finalement à un premier "modèle d'interface" capable de s'adapter aux diverses particularités de l'utilisateur regroupées en dynamiques d'interaction et en styles [Gavrilova & Voinov, 1997]. En plus de fournir à l'utilisateur une aide automatique et réactive par pointage, une explication et une correction de ses erreurs, ces systèmes d'aide adaptatifs, ou "tutoriels informatiques", commencent aussi à dépendre du professionnalisme et de la stratégie cognitive de l'utilisateur, que ce soit au niveau des buts, du détail et même de la navigation au travers de l'information. Ainsi, nous pouvons notamment y retenir une mesure de la productivité de l'utilisateur à l'intérieur d'une interface concrète. L'expérimentation est basée sur l'émulation d'une situation de stress graduellement renforcée par une réaction d'apprentissage négative, et figée ensuite par des investigations en psychologie expérimentale et éducative.

Nous achevons ici notre rencontre avec cette catégorie initiale d'interfaces intelligentes dédiées à toute aide adaptée aux particularités de chaque individualité, non sans avoir auparavant listé les composants les plus importants d'un tel modèle d'interface :

- contrôle (dialogues et dynamiques d'interaction comme la vitesse d'animation, l'expérience professionnelle, le rôle dans l'application, l'âge et l'habileté).
- conception (écran, nombre, apparence, placement et taille des caractères correspondant aux boutons ; formes et tailles des éléments visuels comme les styles de textes et de bordures ou encore les propriétés du système de fenêtres).
- navigation (niveau de responsabilité et mode d'accès aux périphériques contrôlés, chemins concrets de navigation et style d'opération).
- aide (instructions générées automatiquement à l'image du pointage, de l'explication claire et de la correction des erreurs ; de la fourniture aussi d'informations de référence comme la stratégie de présentation ou encore le langage dépendant du niveau professionnel de l'utilisateur).

En résumé, et en rapport avec les trois références plus particulièrement étudiées durant cette première approche des interfaces intelligentes, nous retiendrons finalement à ce niveau les caractéristiques appropriées à venir (**Tableau 1.2.4.2**). Toutes les données sont rapportées, comme explicité précédemment, à nos propres définitions du domaine. Ainsi, et comme en attestent

également les principaux qualificatifs extraits de chaque interface intelligente décrite, c'est en nous assistant de la **Définition 1.2.2.3** que nous pouvons calculer chaque valeur du degré **HII** correspondant au nombre des critères d'intelligence respectivement découverts.

L'accroissement alors relevé en ce qui concerne l'intelligence apparaît fonction des diverses avancées réalisées depuis une dizaine d'années en matière d'Intelligence Artificielle. Catalysé par un apport progressif de nouvelles techniques toujours plus développées, [Gavrilova & Voinov, 1997] présente ainsi une relative importante majorité des caractéristiques nécessaires à nos yeux pour spécifier une interface intelligente de qualité.

Tableau 1.2.4.2 : *Caractérisation hiérarchique des références d'interfaces intelligentes remarquables traitant des aides automatiques et tutoriels informatiques*

Référence abordée	Valeur obtenue pour HII	Classe de l'interface	Principaux qualificatifs de l'intelligence
[Benyon, 1993]	8	Intelligence faible	adaptation automatique - mesure - compréhension - demandes d'informations - classification - environnement - expériences et fréquences d'utilisation - inférences implicites
[Harrington, Banks & Santos, 1996]	11	Intelligence moyenne	domaine indépendant - environnement intuitif - interrogation appropriée - adjonction d'informations - contrôle - conseil - adaptations automatiques - perception des besoins - fréquence des activités
[Gavrilova & Voinov, 1997]	14	Intelligence forte	mesure de productivité - réaction d'apprentissage - classification - modèle d'utilisateur - adaptation aux particularités - contrôle - vitesse et expérience - navigation - éléments visuels - explication et correction des erreurs - fourniture d'informations - stratégie cognitive

1.2.4.3. Implication massive des techniques d'exploitation de connaissances

En parallèle aux travaux que nous venons d'étudier, nombreux sont les chercheurs qui s'accordent à souligner que la complexité croissante des nouveaux problèmes abordés limite considérablement l'applicabilité d'approches algorithmiques figées et déterministes, c'est-à-dire entièrement programmées à l'avance. Une description des problèmes ne saurait représenter la diversité des situations à traiter.

C'est dans cette optique que [Garbay & Maitre, 1997] met en place un projet d'acquisition interactive et dynamique de nouvelles connaissances, d'adaptation également aux conditions changeantes de l'environnement, et de construction de manière progressive de solutions adaptées au contexte potentiellement évolutif d'un problème à résoudre, que ce soit dans sa structure, ses comportements ou ses intentions. Cet outil, baptisé "Machine Perceptive Intelligente", se définit par une architecture fonctionnelle d'apprentissage, de reconnaissance et de manipulation de concepts symboliques. En outre, il possède la particularité de mêler apprentissage de classes et apprentissage de règles en un seul et même logiciel. Nous devons rappeler alors à ce niveau que, contrairement à l'apprentissage de classes qui fait appel à un apprentissage en continu permettant un ajustement dynamique des classes, une classification de même qu'une "fouille de données" en

apprentissage non supervisé, l'apprentissage de règles apporte, lui, de son côté et au vu d'un résultat ou d'un comportement précis, une estimation des erreurs, de la distance au but ainsi que de la rapidité de convergence d'un algorithme. Un tel apprentissage entraîne par conséquent l'acquisition de nouvelles règles et méthodes de modélisation, d'interprétation, d'évaluation, de contrôle et d'adaptation des comportements.

L'intégration de l'utilisateur dans la boucle d'apprentissage, par correction d'erreurs et pré-structuration du problème (à savoir transposition à des cas complexes des connaissances apprises sur des cas simples), est réalisée dans le cadre de trois problématiques essentielles, appréhendées en termes d'interactions entre systèmes et environnement. En plus de l'apprentissage, nous pouvons y noter l'interprétation ou la mise en relation de descriptions et de symboles, au terme de stratégies incrémentales de recherche et d'exploitation de l'information (construction progressive et contrôlée de la solution). Quant à l'adaptation, elle se traduit par le développement des organisations et des comportements adaptés à la situation à analyser et au but à atteindre, la capacité à formaliser et à exploiter les contraintes mutuelles entre situations, comportements, compétences et intentions. En effet, de façon à regrouper les compétences requises et mettre en oeuvre les comportements appropriés à une situation à analyser et à un but à atteindre, [Garbay & Maitre, 1997] va jusqu'à définir des buts réalistes selon les ressources disponibles, la complexité des situations à analyser et l'enjeu de la résolution. Nous retiendrons enfin la mise en oeuvre de différentes approches du type "essai / erreur" grâce à une évaluation concurrentielle de la qualité des résultats et la formalisation du couplage "erreur/réparation".

Davantage orienté vers l'applicatif "grand public", mais toujours en s'appuyant sur les techniques d'exploitation de connaissances, [Rhee, Kim & al., 1997] nous propose en second lieu un système intelligent destiné à l'accession et au répertoriage de données multimédias (*Intelligent Multimedia Repository*). Les utilisateurs, présentés comme débutants, ne sont pas dans ce cas des personnes entraînées à l'utilisation des systèmes informatiques, bien que le système puisse, bien entendu, être employé pour des utilisateurs à différents niveaux d'expertise.

Doté d'une capacité d'adaptation à l'utilisateur spécialisée dans le cadre de l'assistance à la recherche d'un accès efficace aux informations, ce nouvel outil se charge du stockage de la connaissance comportementale et de l'historique des interactions émanant de chaque utilisateur connecté. Basés sur un modèle en couches nommé *Human-COMputer Symmetry* ou *H-COS*, l'interface-utilisateur intelligente (IUI) adoptée ainsi que le modèle d'utilisateur associé font appel, simultanément, à des réseaux de neurones. Ceux-ci opèrent en rétro-propagation (pour une adaptation à court terme) et à l'aide d'un arbre de décision basé sur le *Case-Based Reasoning* (dans le but d'une adaptation à long terme cette fois-ci). Ainsi, le traitement permanent des diverses requêtes et réponses successives recueillies permet à terme de minimiser les erreurs de communication dues aux écarts sémantiques entre l'utilisateur et le système.

De nombreux avantages, directement applicables aux interfaces intelligentes, émanent des réseaux neuronaux, assemblages interconnectés de simples éléments de traitement, unités ou noeuds, et dont la fonctionnalité est basée sur le neurone animal. Rappelons que la capacité de traitement d'un réseau de neurones est stockée dans les forces des connexions inter-unités, ou poids, obtenues par un processus d'adaptation, ou d'apprentissage, à partir d'un ensemble d'exemples [Gurney, 1996]. De cette manière, les mécanismes neuronaux, dès lors qu'ils sont implémentés au sein de systèmes intelligents, supplantent nombre de procédés informatiques classiques, comme le démontre la liste ci-contre :

- le style de traitement des réseaux neuronaux est complètement différent de celui utilisé par les procédés informatiques classiques puisque plus apparenté au traitement du signal (la combinaison de signaux et la production de nouveaux contrastent avec l'exécution d'instructions stockées dans une mémoire).

- l'information est stockée dans un ensemble de poids plutôt que dans un programme, et les poids s'adaptent lorsqu'un ensemble d'exemples est présenté au réseau de neurones.
- les réseaux neuronaux sont robustes en présence de bruits (de petits changements dans un signal d'entrée ne vont pas rigoureusement affecter la sortie d'un noeud).
- les réseaux de neurones sont robustes en présence de défaillances techniques (un changement dans un poids n'affectera seulement que la sortie d'un petit nombre de formes d'entrée possibles).
- les concepts de haut niveau sont représentés comme forme d'activité concernant beaucoup de noeuds plutôt que par le contenu d'une faible portion d'une mémoire informatique.
- les réseaux neuronaux sont bons au niveau des tâches "perceptives" et rappels associatifs (l'approche symbolique a justement des difficultés avec ces tâches).

En résumé, nous garderons à l'esprit quatre grands points forts capables, à eux seuls, de caractériser toute la puissance du domaine des réseaux de neurones : la réaction aux bruits, le traitement des incertitudes, l'efficacité et la robustesse [Sandel, 1997]. Nous verrons cependant que de récentes techniques d'Intelligence Artificielle peuvent apporter davantage de qualités à une interface intelligente de dernière génération. Certains inconvénients des réseaux neuronaux arriveront de la sorte à être résorbés, comme par exemple la lourde architecture nécessaire au bon fonctionnement de tels procédés.

Cela dit, il subsiste tout-de-même des outils représentatifs de la mise en œuvre efficace des avantages issus des réseaux de neurones. En voici un exemple, réellement apprécié de nos jours : l'*Adaptive-Subspace Self-Organizing Map (ASSOM)* ou "Carte Adaptative de Kohonen" [Kohonen, Kaski & Lappalainen, 1997]. Implémenté sur l'architecture modulaire d'un réseau de neurones, dont les modules apprennent à identifier des formes entrées et sujettes à des transformations simples, ce système d'apprentissage non-supervisé présente une interface intelligente de cartographie de documents basée sur le *Web* au travers d'une visite de collections d'articles organisées automatiquement. Techniquement, chaque module neuronal devient, de manière adaptative, spécifique à une classe restreinte de transformations (translation, rotation ou encore changement d'échelle). Les modules, indépendants les uns des autres dans le réseau de neurones, s'accordent vers des caractéristiques similaires selon une manière méthodique. En outre, si différentes transformations existent dans les signaux d'entrée, différents sous-ensembles d'unités (ou modules) s'accordent en définitive à ces classes de transformation.

Mais le réseau de neurones peut également fonctionner ici comme un pré-processeur d'apprentissage pour l'extraction de caractéristiques invariantes : chaque module du réseau devient alors invariant à un seul type de transformation et décode une certaine classification de caractéristiques. S'en suit la reconnaissance invariante d'une forme comme reconnaissance invariante de tous les composants de ses caractéristiques, et un apprentissage adaptatif des classes d'invariance correspondant aux séquences courtes de formes d'entrée nommées "épisodes".

Ces travaux ont donné lieu à une implémentation que nous présentons dans ces lignes. Nommée *WEBSOM* [Kohonen, Kaski & Lappalainen, 1997], cette application permet une exploration ergonomique d'une carte de documents issue du *Web* par exemple (d'où son nom). En cliquant sur une zone de l'image de la carte correspondant à une sorte de "photographie instantanée" du réseau *Internet*, nous obtenons une description du matériel trouvé dans la zone sous-jacente par l'étiquetage de la carte à l'aide de mots-clés. La possibilité nous est donnée, soit d'obtenir une vue zoomée concernant le noeud courant ou d'opter pour une flèche à huit têtes dans le but de déplacer le centre du zoom d'une moitié de la largeur de la vue dans la direction sélectionnée, soit de cliquer sur la ligne d'un sujet de document pour permettre la lecture de l'article désiré.

Les noeuds de la carte constituant la structure de base de cet outil, nous pouvons relever que les documents en rapport les uns les autres sont souvent dans les mêmes ou tous proches noeuds de la carte. Par extension, les discussions les plus spécifiques se trouvent au sein des “groupes” les plus clairs, à savoir dans les régions lumineuses entourées par une couleur sombre. Il apparaît aussi que, près des arêtes de la carte, se trouvent typiquement les documents les plus “différents” et que les surfaces du milieu de carte correspondent aux discussions “typiques”, ou concernent plusieurs thèmes différents. La **Figure 1.2.4.3** ci-dessous nous apporte ainsi un aperçu que nous avons extrait de ce procédé de “cartographie zoomée *WEBSOM*”. Nous avons bien sûr choisi de travailler avec notre domaine de prédilection qu’est celui de l’Intelligence Artificielle.

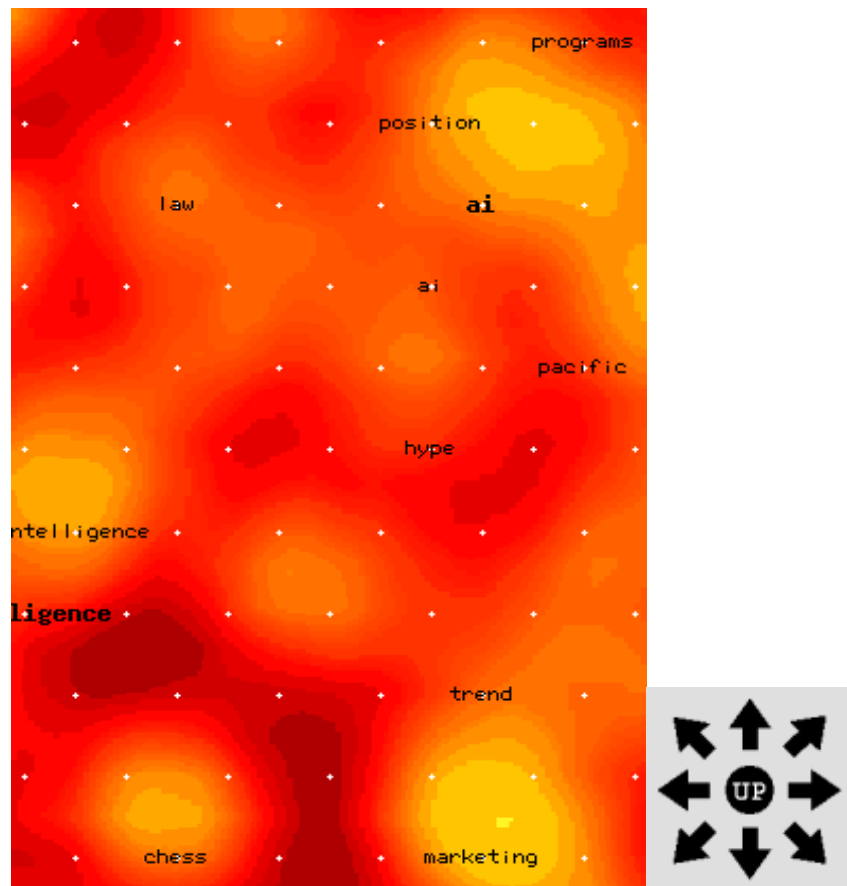


Figure 1.2.4.3 : *Exemple d’une carte WEBSOM [Kohonen, Kaski & Lappalainen, 1997], bien appliqué au réseau Internet et focalisé ici sur le domaine de l’intelligence artificielle (“ai” sur la présente carte)*

Pour conclure cette seconde partie de notre étude des catégories d’interfaces intelligentes, voici un état des diverses caractéristiques et niveaux d’intelligence relevés jusqu’ici, toujours en accord avec nos propres définitions du domaine (**Tableau 1.2.4.3** sur la page suivante).

Nous devons alors admettre que [Kohonen, Kaski & Lappalainen, 1997] supplante les deux autres contributions étudiées. En effet, ces derniers travaux nous mettent surtout en garde devant la grande nécessité de toujours prendre en considération les multiples allégations probables du ou des utilisateurs concernés par la manipulation future des logiciels. Ceci est avancé dans l’optique d’une quête d’efficacité, d’un développement et d’une réalisation d’interfaçage pertinents.

Par conséquent, nous retenons que, si les techniques d'exploration des connaissances et les autres procédés "intelligents" apparaissent indispensables à un modèle d'interface intelligente, l'entité primordiale d'une véritable relation entre l'homme et la machine demeure l'utilisateur potentiel. Les issues intéressantes pour nos divers travaux de recherche ne se limitent cependant pas ici à cette intégration de l'homme au sein des outils manipulés. En effet, la présente et nouvelle catégorie d'interfaces intelligentes nous propose aussi d'unir apprentissage de classes et de règles en une seule et même entité logicielle, de traiter plusieurs approches d'évaluation et de correction d'erreurs, tout comme de prendre en considération une importante liberté de découverte et d'initiative laissée à nos utilisateurs.

Tableau 1.2.4.3 : *Caractérisation hiérarchique des références d'interfaces intelligentes remarquables traitant de l'implication massive des techniques d'exploitation de connaissances*

Référence abordée	Valeur obtenue pour HII	Classe de l'interface	Principaux qualificatifs de l'intelligence
[Garbay & Maitre, 1997]	9	Intelligence moyenne	ajustement dynamique - stratégies incrémentales - formalisation - apprentissage non supervisé - acquisition de règles - méthodes de modélisation - contrôle - adaptation des comportements - correction d'erreurs - classification - manipulation de concepts symboliques - évaluation de qualité
[Rhee, Kim & al., 1997]	9	Intelligence moyenne	répertoire - capacité d'adaptation - assistance - niveaux d'expertise - modèle d'utilisateur - stockage - connaissance comportementale - historique des interactions - erreurs de communication - écarts sémantiques - arbre de décision - réseaux de neurones en rétro-propagation
[Kohonen, Kaski & Lappalainen, 1997]	12	Intelligence moyenne	architecture modulaire - réseau de neurones - identification de formes - apprentissage non supervisé - transformations adaptatives - extraction de caractéristiques - classification - reconnaissance - classes d'invariance - séquences - cartographie de documents - organisation automatique

1.2.4.4. L'Enseignement Assisté par Ordinateur

Vitrine de l'informatique très prisée et en constante amélioration ces dernières années, l'Enseignement Assisté par Ordinateur (EAO), ou encore l'Environnement d'Apprentissage avec Ordinateur, nous emmène au cœur scientifique de l'Intelligence Artificielle : conception de programmes donnant aux apprenants les moyens de réaliser des tâches, capables également d'observer et d'interpréter leur comportement, puis d'intervenir au moment opportun et de manière appropriée sur la base de leur analyse [Bruillard, 1997]. Notre domaine de l'exploration des connaissances prend alors bien toute sa dimension "d'ingénierie de la connaissance", nécessaire tant au niveau des processus de résolution suivis par les humains qu'au niveau de l'implantation en machine de tels processus.

Issu des idées et grands courants de recherche concernant les applications de l'informatique à la formation, le concept de l'EAO se résume à un partenariat entre un utilisateur (ou "élève") et une machine, à la coopération aussi entre agents humains et agents artificiels. Cependant, si un grand nombre de travaux en EAO visent une autonomie croissante avec des systèmes utilisables par des élèves sans maître directement accessible, des acteurs impliqués dans des dispositifs de formation s'appuyant massivement sur les technologies éducatives à distance font le constat que l'apprenant ne peut être isolé et que son autonomie ne peut être totale. Il s'agit donc aujourd'hui plutôt de concevoir des systèmes coopératifs d'apprentissage qui intègrent comme acteurs des enseignants ou formateurs et des apprenants, et qui offrent de bonnes conditions d'accès à des ressources formatives distribuées, humaines, et/ou médiatisées.

Un exemple opérant une transition appréciable entre la découverte des techniques d'Intelligence Artificielle et le présent domaine de l'Enseignement Assisté par Ordinateur nous est fourni par la méthodologie de conception centrée sur l'utilisateur (*User Centered Design* ou *UCD*) et développée dans le cadre de solutions logicielles pour la célèbre firme américaine IBM [Murray, Schell & Willis, 1997]. Application aidant tous les types d'utilisateurs à construire des règles pour manipuler la tâche universelle du courrier électronique, cette interface-utilisateur s'attache à rejoindre les besoins de l'utilisateur en se laissant conduire par tous les aspects de l'expérience de celui-ci et en intégrant ses diverses réactions dans le but de produire plans, priorités et décisions. Ce logiciel présente un "format d'interview" interrogeant les utilisateurs sur des critères de règles (expéditeur ou sujet) et des conditions (priorité) de manière amicale. Les utilisateurs sont guidés à travers une série de questions et de réponses pour construire autant de règles qu'ils en ont besoin. Des écrans de confirmation et d'autres éléments d'interface-utilisateur donnent à l'individu l'assurance que ses différentes règles opèrent comme souhaité.

Loin d'être réellement automatisée, cette approche *UCD* est néanmoins basée sur quatre principes intéressants. Si la compréhension des utilisateurs demeure la force motrice permettant toute conception, la considération de l'expérience totale de l'utilisateur, c'est-à-dire tout ce que voit et touche l'utilisateur, ainsi que le recueil fréquent, rigoureux et rapide de ses réactions, dirigent, par une estimation permanente de la compétitivité, le développement d'un produit. De plus, plusieurs objectifs atteints lors de l'élaboration pratique renforcent et mettent en valeur la réalité de ces procédés. Il est nécessaire de noter ici la fourniture d'un moyen enfin simple pour la construction de règles dédiées au courrier électronique. Dans beaucoup d'autres produits, les formats de règles sont compliqués, les expressions logiques (*et / ou / non*) et leurs multiples combinaisons sont confuses, et le véritable nombre de règles possibles est trop impressionnant. En conséquence, les utilisateurs trouvent la tâche d'établir des règles trop difficile, et donc les capacités de construction de règles de beaucoup d'applications ne sont pas utilisées.

Par suite, [Murray, Schell & Willis, 1997] a cherché à minimiser le temps de démarrage de l'application. Il s'avère en effet que, cette tâche faisant partie du "travail réel" d'un utilisateur, le courrier électronique doit permettre à l'utilisateur d'entrer et de sortir rapidement. L'adressage des règles les plus courantes et la limitation des tâches à celles les plus fréquemment utilisées sont alors basés sur l'analyse de tâches et les préférences de l'utilisateur. En outre, la création d'un ensemble limité de tâches pour lesquelles la plupart des personnes sont susceptibles d'établir des règles présente les prémisses d'une anticipation auto-adaptable des actions.

Nous arrivons ainsi à une description instructive de la distinction possible entre trois différents styles d'interaction au sein d'une interface. Une première approche pas-à-pas pour la construction de règles propose une série de questions à l'utilisateur, avec affichage d'une liste de critères et actions à choisir, puis construction à partir des données recueillies. La seconde version est caractérisée par l'emploi de modèles de règles sous forme de phrases à deux ou trois champs

d'entrée pour des attributs-clés. Toutes deux se différencient d'une ultime interaction structurée incluant des champs de conditions et d'actions désirées que l'utilisateur complète aisément pour construire une de ses propres règles.

A ce niveau, nous remarquons que ces mécanismes d'Intelligence Artificielle se rapprochent inéluctablement des quatre points majeurs du développement de l'EAO spécifié ici [Bruillard, 1997]. En effet, l'apport de l'Intelligence Artificielle à nos investigations concernant les interfaces intelligentes permet d'envisager la réalisation d'un tuteur idéal, adaptable à tous les élèves et tous les contextes de formation, en améliorant l'adaptabilité des "machines à enseigner", cette qualité demeurant le critère essentiel de telles machines. Les possibilités d'adaptation sont alors multiples, à l'image des modes inhérents de présentation, du filtrage et de la granularité des connaissances manipulées, ainsi que du choix des types de chemins à faire suivre à l'apprenant. Nous pouvons aussi maintenant espérer fournir aux machines des capacités de résolution, non seulement dans un contexte de formation, mais aussi dans un cadre de travail. Cependant, en plus de connaître ce que les machines sont susceptibles de faire, il est essentiel d'appréhender leurs limites : le seuil d'incompétence renseigne sur les compétences à développer chez les humains. La machine devient de cette manière un auxiliaire de résolution ou une prothèse, un partenaire et un amplificateur de la pensée de l'homme, voire un guide, un outil ou un instrument.

Il nous faut alors définir le statut de la notion de connaissance, celle-ci conditionnant les directions de recherche et les types de solutions préconisées dans la réalisation de tels systèmes informatiques. Le point délicat dans l'éducation est de savoir qui dispose des connaissances nécessaires : l'apprenant cherche à acquérir des connaissances et dispose d'instruments, mais ne détient pas toujours, contrairement à l'enseignant, des connaissances lui donnant l'autonomie nécessaire face aux instruments qu'on lui confie. Aussi, la compréhension du comportement de l'apprenant devient primordiale, notamment pour savoir ce que le destinataire de nos travaux connaît exactement. On passe alors de la simple analyse de la réponse ou du résultat à celle du processus-même de résolution ou de recherche d'une solution. En cherchant constamment à traiter la démarche de l'apprenant pour le comprendre ou lui donner les moyens de se comprendre et de modifier ses comportements et ses connaissances, nous découvrons finalement la difficulté à capturer et à représenter la connaissance individuelle de chaque élève, conduit en conséquence à se limiter à des domaines très procéduraires.

C'est pourquoi, nous aboutissons à l'élaboration d'un modèle dit "de l'utilisateur", que nous expliciterons et développerons plus avant au cours du **Chapitre 3**. Miroir du comportement de l'apprenant en tant qu'utilisateur d'un ordinateur, nous verrons que cette modélisation correspond entre-autres à la façon dont l'élève se sert de la machine et interagit avec ses programmes. Nous remarquons par ailleurs encore ici qu'une session EAO se déroule toujours de manière cyclique (présentation d'informations, assimilation et réponse de l'apprenant à une question posée, analyse et validation, continuation résultante ou branchement à une autre partie du cours).

Reconnue en tant qu'approche "utilisateur comme étudiant" [Brusilovsky & Schwarz, 1997], la seconde référence abordée dans ces lignes paraît alors plus fortement dédiée à l'enseignement auto-adaptatif recherché. Apportant grande assistance aux débutants, à l'aide d'un apprentissage incrémental, elle présente aussi surtout un procédé de séquençement adaptatif d'actions. En effet, en plus du développement d'interfaces auto-explicatives pour des applications avancées, cette approche implémente plusieurs adaptations destinées à aider les débutants dans l'utilisation d'applications complexes sur le *Web*. L'ordre d'explication, d'apprentissage ou d'habilitation des caractéristiques de l'interface est déterminé par un séquençement adaptatif, une technologie d'enseignement intelligent qui est basée sur un modèle pédagogique de l'interface et sur la connaissance de l'utilisateur.

Différentes caractéristiques des utilisateurs sont ainsi obligatoirement prises en compte, comme leurs capacités dans le domaine informatique, leurs expériences respectives du *Web*, les connaissances générales et la formation.

Enfin, nous découvrons, au sein des systèmes *CALL* (*Computer-Assisted Language Learning*), une autre interface orientée vers l'utilisateur, associée à la conception d'une application tenant compte cette fois des forces, des faiblesses, des préférences et du niveau de compétences de chaque individualité étudiante durant les cours [Murphy & McTear, 1997]. Appuyée sur une architecture purement fonctionnelle, cette interface intelligente se compose de modules fournissant respectivement une analyse linguistique détaillée de chaque réponse de l'étudiant aux exercices du programme, la création d'un modèle dynamique de tout étudiant, un contrôle des réactions du système face aux entrées des étudiants ainsi que de la structure du matériel offerte à chacun d'eux.

Un premier module dit "d'instruction" entretient le façonnage d'exercices de langage pour chaque étudiant et détermine le moment d'intercession dans la leçon en utilisant les approches d'instructions, les stratégies et les techniques incorporées dans le système. Ces dernières correspondent à la représentation explicite des concepts grammaticaux et linguistiques, à un ensemble d'exercices, à une librairie stockant le texte des réactions et une série de gabarits adaptatifs, à un ensemble de pages grammaticales disponibles à la demande de l'étudiant. Associé à ces différents mécanismes intelligents, nous retrouvons ici un modèle de l'utilisateur répertoriant notamment le stéréotype de l'étudiant, les règles d'acquisition implicite et le moteur d'inférence le caractérisant, toujours sous la forme de librairies dynamiques. Un troisième module, "de diagnostic" cette fois, vérifie l'exactitude de l'entrée de l'étudiant par analyse grammaticale et description détaillée de l'erreur linguistique, dans le cas d'une entrée erronée et le tout sur la base d'une hiérarchie de classes d'erreurs.

En définitive, un modèle général nommé "modèle du contexte" se charge de gérer un répertoire central de stockage de toutes les données pertinentes concernant l'opération courante du système, à savoir, à la fois, les informations du modèle de l'étudiant comme le moyen d'expression préféré de l'étudiant et les définitions de l'exercice courant, ainsi que la cause de l'erreur de l'étudiant, identifiée par le module de diagnostic.

En résumé, et pour reprendre une citation émanant de la Commission Européenne, nous constatons que nous progressons inexorablement vers une société de la connaissance [Bruillard, 1997] : "de l'éducation et de la formation tout au long de la vie". Aussi, nous retiendrons finalement à ce niveau les différentes informations à venir sur la page suivante (**Tableau 1.2.4.4**).

A noter alors que la valeur relativement faible obtenue par [Murray, Schell & Willis, 1997] au niveau de notre système hiérarchique des interfaces intelligentes est due essentiellement à un important vide propre à l'automatisation des procédés mis en avant au cours de ces travaux. Quant au même résultat attribué par nos définitions à [Brusilovsky & Schwarz, 1997], il résulte de son côté du fait que ces recherches sont focalisées sur l'étude de mécanismes d'auto-adaptation, au détriment malheureusement d'autres points pourtant nécessaires à une interface intelligente dédiée, qui plus est, à l'EAO. Nous noterons par exemple, à ce stade de nos réflexions, le manque de traitement d'erreurs éventuelles émanant d'un utilisateur, des modèles très intrusifs et insistants aussi vis-à-vis de celui-ci, ainsi qu'une absence de surveillance propre aux données recueillies, sécurité indispensable pourtant dans un outil opérant au niveau d'*Internet*.

En règle générale, les références d'interfaces intelligentes faisant état de travaux récents effectués dans le cadre de l'Enseignement Assisté par Ordinateur faillissent quelque peu au niveau de la liberté, de l'initiative laissée aux étudiants, voire encore en ce qui concernent des éventuelles suggestions de manipulations pouvant être portées à la connaissance des utilisateurs.

Cela provient, sans modification ou solution *a priori* nécessaires, du rôle et de la définition-même de l'EAO qui demeure un outil destiné à instruire et à simplement fournir des informations à l'utilisateur, certes de manière aussi automatique et adaptée que possible.

Tableau 1.2.4.4 : *Caractérisation hiérarchique des références d'interfaces intelligentes remarquables traitant de l'Enseignement Assisté par Ordinateur*

Référence abordée	Valeur obtenue pour HII	Classe de l'interface	Principaux qualificatifs de l'intelligence
[Murray, Schell & Willis, 1997]	8	Intelligence faible	besoins, expérience et réaction de l'utilisateur - construction de règles - format d'interview - assurance - compréhension des utilisateurs - compétitivité - expressions logiques - minimisation du démarrage - rapidité - limitation des tâches à celles les plus fréquemment utilisées - champs d'actions
[Brusilovsky & Schwarz, 1997]	8	Intelligence faible	interfaces auto-explicatives - aide aux débutants - apprentissage incrémental - séquençement adaptatif - enseignement intelligent - modèle pédagogique - connaissance de l'utilisateur - capacités - expérience
[Murphy & McTear, 1997]	12	Intelligence moyenne	modèle dynamique de l'individualité étudiante - stockage de données pertinentes - moyen d'expression préféré - diagnostic - vérification d'exactitude - hiérarchie de classes d'erreurs - moment d'intercession - série de gabarits adaptatifs - moteur d'inférence

1.2.4.5. L'avènement des agents intelligents

Depuis le début de l'histoire, les gens ont été fascinés par l'idée d'agencements non-humains tels que des androïdes, humanoïdes, robots et autres cyborgs [Bradshaw, 1997]. Bien que les automates aient existé pendant des siècles, c'est seulement récemment que quelque chose de ressemblant aux agents autonomes a commencé à apparaître. Les agents déployés diffèrent de façon importante des concepts précédents : le mouvement a aujourd'hui changé du matériel au logiciel, des atomes qui comprennent un robot mécanique aux bits qui composent un agent digital. Ces agents logiciels fonctionnent continuellement et de manière autonome dans un environnement particulier qui est souvent peuplé par d'autres agents et processus. Idéalement, ils apprennent de leurs expériences, communiquent et coopèrent avec d'autres agents, et, comme exigé, bougent dans des réseaux privés et sur l'*Internet*.

Des tendances récentes ont précisé que la complexité logicielle continuera à augmenter dramatiquement dans les décennies à venir. La nature dynamique et distribuée, tant des données que des applications, exige qu'un logiciel ne réponde plus simplement aux demandes d'information, mais prévoit intelligemment, s'adapte et cherche activement des façons de soutenir les utilisateurs. Non seulement ces systèmes doivent assister la coordination de tâches parmi les humains, ils doivent aussi aider à gérer la coopération au sein des programmes répartis.

En réponse à ces besoins, les efforts des chercheurs de plusieurs domaines différents ont commencé à s'unir autour d'un large agenda commun : le développement d'agents logiciels. D'une part, des chercheurs des domaines de l'interaction homme-machine, des interfaces-utilisateur intelligentes et adaptatives, de l'acquisition des connaissances et de la programmation dédié à l'utilisateur se sont intéressés à l'interaction entre humains et agents logiciels. D'autre part, des chercheurs des domaines de l'Intelligence Artificielle, de la robotique, de la vie artificielle aussi et de la programmation d'objets distribués ont progressé dans la compréhension des possibilités de développement, raisonnement et coopération des communautés d'agents. De telles lignes complémentaires de recherche sont motivées par deux soucis principaux : les limitations des interfaces à manipulation directe, et les complexités de l'informatique distribuée.

Ce contexte introduit ainsi les agents intelligents et leurs concepts de raisonnement, de représentation des connaissances et d'apprentissage. Pourtant, les applications pratiques d'agents ont des origines nettement plus pragmatiques [Caglayan & Harrison, 1997]. Les développeurs de logiciels ont commencé par créer différents outils pour automatiser des processus spécifiques ; les agents intelligents ont alors, ces dernières années, attiré l'attention pour plusieurs raisons :

- comme un moyen de mettre de l'intelligence dans des interfaces pour permettre aux utilisateurs non qualifiés d'obtenir davantage des applications informatiques.
- comme un moyen de personnaliser des applications et des services pour rencontrer les préférences, buts et désirs de l'utilisateur.
- comme un moyen de gérer la récupération, la dissémination et le filtrage des quantités énormes d'informations devenant disponibles sur les réseaux d'entreprise, réseaux à valeur ajoutée, et particulièrement l'*Internet*.

C'est pourquoi, nous aboutissons à une forte présence déjà de la technologie des agents intelligents dans de nombreuses applications d'entreprise et autres produits commerciaux aujourd'hui disponibles : agents de bureau, de recherche sur le *Web*, programmes de recherche et filtrage d'information personnalisée, agents de base de données, ... Par définition, un agent est alors une personne ou une chose qui agit, est capable d'agir ou est autorisé à agir pour une autre [Caglayan & Harrison, 1997]. Cette essence d'agent possède dès lors deux attributs-clés : un agent fait des choses, et un agent agit au nom de quelqu'un ou de quelque chose. Par extension, un agent logiciel peut être vu comme une entité informatique qui exécute de manière autonome des tâches déléguées par un utilisateur, mais également telle une pièce de logiciel pouvant opérer sur le comportement d'une autre entité [Carrez, 1998] ou encore un processus informatique qui sent son monde et agit sur lui [Mataric, 1994].

Ainsi, un tel agent logiciel devient intelligent dès lors qu'il se voit attribuer certaines qualités et capacités d'effectuer des actions autonomes ou automatiques telles que la délégation destinée à d'autres entités, une certaine habileté de communication, un contrôle ou une mise en action [Caglayan & Harrison, 1997]. De multiples compétences nouvelles ont ensuite vu le jour, toutes aussi intéressantes et bénéfiques les unes que les autres, afin de permettre aux agents intelligents de progresser : mobilité, interaction (sociabilité et comportement adaptatif), coopération mutuelle, apprentissage et même raisonnement [Carolan, Collins & al., 1997].

Il subsiste cependant un inconvénient de taille concernant ces récents outils logiciels, à savoir qu'il s'agit encore de les gratifier d'une réelle crédibilité et efficacité aux yeux des divers utilisateurs, que ceux-ci soient néophytes comme très expérimentés. En effet, de nombreuses personnes croient toujours qu'un agent intelligent ne va pas réellement exécuter ce qu'ils lui demandent, ou va se détériorer et faire quelque chose qu'ils n'ont pas désiré. De même, le contrôle par l'utilisateur, relativement fréquent encore, des activités d'un agent logiciel reste un point entravant l'utilisation d'agents pour beaucoup de personnes [Murray, Schell & Willis, 1997].

Mais, si un tel établissement de la présence et de la performance des agents logiciels demeure encore friable, il n'en est pas moins vrai que les agents intelligents évoluent chaque jour davantage. Ils deviennent plus capables, ingénieux et puissants, intelligents en somme, voire même incontournables pour les applications informatiques futures, et en particulier les interfaces intelligentes. C'est pourquoi, nous ne nous étonnons plus en découvrant un agent logiciel démontrant par exemple des facultés d'autonomie, de rationalité, de réactivité, de planification ou encore de modélisation, de l'environnement comme de l'utilisateur d'ailleurs [Carrez, 1998].

Enfin, nous aboutissons à l'implémentation proprement dite de nos agents intelligents, et découvrons les nombreuses doléances concernant leurs modèles d'apprentissage [Maes, 1994] :

- apprentissage non-supervisé (l'agent doit apprendre principalement sans professeur).
- apprentissage incrémental (l'agent doit apprendre de chaque expérience en remarquant qu'il ne peut y avoir un apprentissage et une phase d'exécution séparés).
- apprentissage dynamique et approprié, focalisé sur les connaissances permettant d'atteindre les buts (au vu des actuels environnements complexes et imprévisibles, un agent ne peut avoir les moyens d'apprendre chaque fait disponible).
- apprentissage capable de faire face aux bruits, aux défections de tous genres, aux environnements probabilistes également.
- apprentissage incorporant des connaissances initiales (pas besoin de tout apprendre en partant de rien, en particulier si la connaissance préalable est facilement disponible).

Une première référence remarquable à ce niveau est alors la présentation de "Letizia", agent d'interface-utilisateur chargé d'assister les déplacements d'un utilisateur sur le *Web* [Lieberman, 1995]. Réalisant en temps réel des suggestions de pages *Web* susceptibles d'intéresser l'utilisateur pendant son déplacement sur *Internet*, cet agent intelligent opère un suivi du comportement de l'utilisateur (liens, recherches initiales, requêtes d'aide) et tente d'anticiper ses centres d'intérêts en effectuant une exploration simultanée et autonome des liens à partir de la position de l'utilisateur. Arrivant à l'automatisation d'une stratégie de déplacement basée sur la "recherche du meilleur premier" (*best-first search* ou "avancement à partir du meilleur nœud ouvert, quelque soit l'endroit où ce nœud se situe dans l'arbre partiel développé"), l'efficacité de "Letizia" est accrue par des heuristiques déduites du propre comportement de l'utilisateur.

Nous découvrons, par suite, une définition de deux types d'agents déjà inséparables, à savoir les agents d'interface et les agents dits "autonomes" [Lieberman, 1997]. Si les premiers proposent à l'utilisateur, dans une interface intelligente, une assistance active sans jamais se maintenir en arrière-plan, la seconde catégorie de ces agents agit au contraire simultanément à l'utilisateur et sans intervention de celui-ci, qu'il soit occupé ou non. Pour ces dernières entités, aucune conversation continue avec les utilisateurs n'est nécessaire et leur travail demeure par conséquent non intrusif face aux interlocuteurs respectifs. De plus, différents aspects, certes encore parfois encore un peu simplistes mais intéressants, ressortent ici dans le but de concevoir une interface intelligente efficace :

- suggérer plutôt qu'agir,
- tirer avantage de l'information que l'utilisateur fournit librement à l'agent, sans avoir recours à une interaction séparée,
- tirer avantage du temps de réflexion de l'utilisateur,
- l'attention de l'utilisateur peut être irrégulière dans le temps,
- les agents d'interface autonomes peuvent avoir une hésitation entre réflexion et action,
- une interface autonome peut ne pas s'accorder avec tous les styles cognitifs d'utilisateurs.

Avouons de suite que la majorité des points précédents, et en particulier le tout dernier, a le mérite de nous laisser songeur et nous poussent à chercher à améliorer la présente situation. Cela s'éclaircit dès lors que [Maglio & Barrett, 1997] avance l'idée d'un rassemblement de données caractérisées par l'observation d'utilisateurs expérimentés du *Web* recherchant des informations spécifiques. Les individus répètent alors les mêmes formes de recherche et conceptualisent leurs diverses recherches en fonction de leurs formes standards. Cette nouvelle assistance en recherche d'informations, basée sur des agents intelligents personnels, entraîne l'adaptation de ces derniers à une collaboration fiable entre utilisateurs d'*Internet*. Nous aboutissons ainsi à la prise en compte d'un avantageux modèle d'utilisateurs, modèle défini par trois étapes successives :

- recueil de données sur le comportement de personnes cherchant de l'information sur le *Web*.
- analyse de ces données pour apprendre ce que ces chercheurs d'informations font et pensent.
- construction d'agents *Web* spécifiques pour supporter les comportements de recherche identifiés (fourniture d'un support personnel pour la recherche d'informations et le transfert efficace de la connaissance acquise par une personne à une autre).

Davantage orienté vers l'interfaçage homme-machine, [Maulsby & Witten, 1997] met aussi en évidence la description de l'élaboration, de l'implémentation et de l'évaluation d'agents apprenant à automatiser des tâches répétitives au sein d'une interface devenue intelligente. De plus, au contraire de la plupart des projets d'Intelligence Artificielle qui leur sont antérieurs en date, ces travaux se concentrent sur une étude d'utilisateur virtuelle, avec un agent simulant l'homme utilisé pour découvrir les interactions que les personnes trouvent naturelles.

Ainsi, en développant et testant différentes méthodes d'interaction pour instruire un agent informatique, avec tout-de-même, pour asseoir l'expérience dans la réalité, une limitation des types d'instructions comprises par l'agent, des caractéristiques textuelles qu'il observe et des généralisations qu'il peut apprendre, deux outils intelligents et complémentaires voient le jour. Il s'agit d'un agent entraîné par les utilisateurs à éditer une bibliographie et un système inhérent d'apprentissage automatique interactif. Simulé par un chercheur caché derrière un écran (rappelons que les concepteurs d'interface appellent cela un scénario de "Magicien d'Oz", ou "*Wizard of Oz*"), le nouvel agent s'éduque en scrutant les utilisateurs qui décrivent des tâches allant du trivial au difficile, et qui inventent également leurs propres méthodes d'enseignement et autres commandes vocales. Cette étude montre par conséquent les manières selon lesquelles les utilisateurs communiquent instinctivement via des "indications", ou, spécifié partiellement, des instructions ambiguës. Ces indications peuvent être communiquées en utilisant la parole, en pointant ou par sélection dans des menus. L'implémentation, associée à ce travail et qui a abouti à la mise en œuvre du système d'apprentissage, démontre que les ordinateurs peuvent apprendre à partir d'exemples et d'informations, même ambiguës. En outre, par l'acquisition de concepts dans un domaine riche à partir de peu d'exemples, en gardant les utilisateurs dans la boucle et en leur permettant d'influencer le cours de l'apprentissage, on obtient une intéressante production de règles comme des descriptions pour classer, trouver, modifier ou générer des données.

Autre point remarquable, les observations et résultats de l'étude-utilisateur démontrent la présence de deux principaux styles de dialogue entre l'homme et la machine, à savoir ceux de "bavard" et de "silencieux". Ils sont alors aisément assimilables à des personnalités qualifiées respectivement d'extravertie et d'introvertie. Un ensemble de commandes utilisées est aussi mis en avant, à la fois pour contrôler l'apprentissage ("montre ce que j'ai fait" ou "ignore cela") et la prédiction ("fais le suivant" ou encore "fais le reste"). D'autres enseignements apparaissent concernant notre présente notion d'agents intelligents. Nous pouvons noter que les utilisateurs apprécient et exploitent l'apprentissage incrémental : ils se contentent en effet d'enseigner les cas spéciaux lorsqu'ils surgissent, plutôt que de les anticiper. En outre, pour obtenir une indication

précise, il ressort qu'il vaut mieux proposer une conjecture plutôt que de demander brutalement à l'utilisateur : "de quoi s'agit-il ici ?"... Enfin, il apparaît également primordial qu'agents comme utilisateurs doivent être capables de se référer aux exemples et instructions passés.

Les "Assistants Personnels de Communication" [Nangle, Cunningham & Evans, 1998] apportent également une utilisation et une application intéressantes quant au présent domaine des agents logiciels intelligents : le filtrage d'informations ou de données spécialisées. Véritables "maîtres d'hôtel digitaux" ou "secrétaires virtuelles", ces agents exécutent, pour et surtout à la place de l'utilisateur, toutes sortes de tâches administratives comme l'organisation de voyages, le shopping sur le *Web* ou la recherche d'informations qui peuvent intéresser chacun de nous. Assimilé à une secrétaire dévouée dans un environnement bureautique, un tel assistant gère les différentes modalités de la communication personnelle, comme les appels téléphoniques, les *fax* et les *e-mails*. Il propose un filtrage d'appels intelligent empêchant le bombardement constant de son propriétaire par des appels téléphoniques non-souhaités pendant que celui-ci est engagé dans un travail de plus haute priorité (limite imminente d'un projet par exemple, ou réunion).

Plus précisément, l'apprentissage des caractéristiques de communication d'un utilisateur va aider les agents intelligents à filtrer automatiquement les appels selon ses préférences. Cela va permettre, non seulement la réduction de l'entrée de connaissances substantielles et de besoins de maintenance par les systèmes existants, mais aussi le refus de scénarios erronés et la capacité à opérer selon une manière plus autonome et "proactive". Nous aboutissons ainsi à l'apport fort d'une aide destinée à apprendre aux employés comment travailler de façon plus professionnelle et productive. En définitive, ces travaux ont mis en évidence un système multi-agents composé d'assistants personnels négociant et collaborant entre-eux afin de gérer intelligemment les appels. Garantissant l'efficacité des opérations tout en dépendant au minimum des données provenant de l'utilisateur considéré, [Nangle, Cunningham & Evans, 1998] avance également un apprentissage des caractéristiques de communication associé à la construction d'un profil d'utilisateur qui va tenter de capturer et de traiter automatiquement les diverses préférences immédiates de l'utilisateur.

Enfin, nous ne pouvons omettre d'aborder et d'exposer les dernières investigations américaines en matière d'intelligence et de raisonnement rapportés aux capacités de nos agents logiciels. En effet, rares sont les ouvrages qui n'ont pas salué la récente création des déjà fameux "Elfes Électroniques" pour "équipes débordées de Californie" [Pynadath, Tambe & al., 2000] ! Ces agents, capables de prendre certaines décisions à la place d'un être humain dépassé par les événements, ont en effet pour mission de gérer l'emploi du temps de leur maître, de le négocier éventuellement avec les autres Elfes, de gérer aussi le menu du déjeuner et même de désigner, en accord avec ses "collègues", le chercheur devant assurer la présentation hebdomadaire de leur laboratoire aux visiteurs. Chaque être humain de l'équipe possède ainsi son Elfe, installé sur son propre ordinateur et qui reste en contact avec lui via son téléphone portable ou son organisateur électronique. L'Elfe peut, de plus, aller jusqu'à localiser précisément son maître grâce à un capteur *GPS* (*Global Positioning System*) que le chercheur porte sur lui. Pour exemple, dans le cas où un des membres de l'équipe se trouve trop loin du bureau pour être à l'heure en réunion, son Elfe le rappelle à l'ordre, puis propose à ses congénères d'informer leurs "seigneurs" respectifs que la séance est reportée de vingt minutes... Il arrive cependant (déjà !) que ces agents intelligents prennent un peu trop de libertés. Ainsi, ayant compris que le chef de projet annulait habituellement ses rendez-vous s'il avait dix minutes de retard, son petit majordome prit l'initiative de décommander une entrevue avec le patron lui-même de l'institut ! Des réglages supplémentaires sont donc à l'étude pour éviter qu'une telle bévue ne se répète à l'avenir.

Par extension, il semble d'ailleurs que les futures organisations humaines, à grande échelle, seront encore plus hautement "agentisées" [Scerri, Pynadath & Tambe, 2001]. En effet, cette

vision apparaît d'ores-et-déjà bien plausible, à l'aide notamment d'agents logiciels supportant les tâches traditionnelles de recueil d'information, de planification et de surveillance d'exécution, mais aussi grâce à d'autres augmentant le contrôle des ressources et des périphériques (en matière de communication universelle par exemple). Toutefois, tant que ces agents hétérogènes accepteront davantage de ces activités, ils devront faire face aux tâches supplémentaires d'interfaçage avec des personnes, et même d'action en tant que procurations de ces dernières. L'association dynamique de tels agents intelligents va permettre aux organisations d'agir de manière cohérente, d'atteindre de manière robuste leurs buts de mission, de réagir rapidement aux crises, et de s'adapter dynamiquement aux événements. Les progrès réalisés dans cette "agentisation intelligente" pourraient par conséquent assister toutes les organisations, y compris les militaires, les organismes de réponse aux désastres civils, les sociétés et même les universités.

Afin de conclure cette partie dédiée aux mécanismes d'agents intelligents en matière d'interfaçage évolué, nous retiendrons finalement les diverses informations du **Tableau 1.2.4.5**. Les concepts et idées destinés aux terminaux de communication demeurent riches et nombreux à l'issue de ces références.

Tableau 1.2.4.5 : *Caractérisation hiérarchique des références d'interfaces intelligentes remarquables traitant de l'avènement des agents dits "intelligents"*

Référence abordée	Valeur obtenue pour HII	Classe de l'interface	Principaux qualificatifs de l'intelligence
[Lieberman, 1995]	13	Intelligence forte	agent d'interface-utilisateur - suivi du comportement - tentative d'anticipation - centres d'intérêts - exploration simultanée et autonome des liens - stratégie de déplacement - recherche du meilleur premier - heuristiques - assistance active - suggestions en temps réel - styles cognitifs
[Maglio & Barrett, 1997]	11	Intelligence moyenne	recherche d'informations - apprentissage - agents <i>Web</i> - comportements de recherche - support personnel - transfert de connaissances - observation d'utilisateurs expérimentés - assistance - identification et suggestion de formes standards similaires - historique - collaboration
[Maulsby & Witten, 1997]	12	Intelligence moyenne	agent apprenant et simulant l'homme - automatisation de tâches répétitives - caractéristiques textuelles - généralisations - scénario de "Magicien d'Oz" - indications ambiguës - apprentissage automatique interactif - génération de règles - styles de dialogue - degré de répétition
[Nangle, Cunningham & Evans, 1998]	14	Intelligence forte	filtrage d'appels - caractéristiques de communication - intentions - apprentissage - assistants personnels - préférences immédiates de l'utilisateur - connaissances et besoins - autonomie "proactive" - système multi-agents - négociation et collaboration - profil-utilisateur
[Pynadath, Tambe & al., 2000]	15	Intelligence forte	"Elfes Électroniques" - localisation - gestion et négociation - initiative - planification et surveillance d'exécution - contrôle de ressources - agents hétérogènes - procurations - buts de mission - rapidité - adaptation dynamique aux événements - résolution de problèmes - ordonnancement

Ainsi, si nous pouvons relever que les premiers principes mis en œuvre à la base des agents logiciels se caractérisent par la continuité de fonctionnement ainsi que par l'autonomie, nous devons également insister sur les facultés d'intelligence, très utiles et de plus en plus variées, attachées à ces outils. Non seulement, ils apprennent de leurs expériences, raisonnent, communiquent et coopèrent avec d'autres entités, mais en plus, ils réussissent à nous permettre de personnaliser des systèmes avancés de manière efficace. A tout cela s'ajoute le fait que les agents intelligents présentent des qualités de compréhension et de réactivité à l'égard de chacun de nos utilisateurs, tout comme d'analyse, de conseil, de contrôle ou "d'évolutivité". Ces diverses aptitudes, auxquelles nous ajoutons encore la convivialité et surtout la performance, seront indispensables dans l'élaboration et la conception de notre nouvelle interface intelligente.

Nous remarquons toutefois la petite différence entre les caractéristiques intelligentes de [Lieberman, 1995] et de [Maglio & Barrett, 1997]. De cette seconde référence transparaît, certes quelque peu par excès de tentatives d'améliorations visant l'interface, un système moins porté vers les propres intérêts et habitudes de l'utilisateur, de même qu'assurément plus insistant et intrusif vis-à-vis de celui-ci. C'est d'ailleurs aussi un des rares points faibles de [Pynadath, Tambe & al., 2000], dont l'avantage émane de ses capacités d'autonomie et de prise d'initiatives envers la gestion des données. Quant à [Maulsby & Witten, 1997], nous noterons un manque flagrant d'application et d'évaluation au niveau du réseau *Internet*, en opposition à une réussite pourtant intéressante de la modularité et d'un traitement plus évolué des besoins individuels.

De son côté, la qualité principale apportée par [Nangle, Cunningham & Evans, 1998], et qui supprime de loin les autres références abordées, demeure sa constante recherche d'amélioration au niveau de la convivialité, de la simplicité réellement efficace, de l'humanisation et de la clarté dégagée par son système multi-agents. Ces travaux se caractérisent également par la mise en évidence d'agents intelligents capables de négocier et de collaborer entre-eux.

1.2.4.6. Application des interfaces intelligentes en entreprise

Véronique De Keyser, de la Faculté de Psychologie et des Sciences de l'Éducation de Liège en Belgique, introduit ainsi le présent sujet : « Peut-on capturer réellement le savoir de l'ergonome et lui substituer, en tout ou en partie, une aide intelligente ?... » [Kolski, 1997]

« En effet, pour les opérateurs dans les entreprises, chaque situation est unique et ils sont eux-mêmes irremplaçables. Sans doute, mais jusqu'à quel point ? Quelle est la part de routine, et la marge au-delà de laquelle l'informatisation, ou les techniques de l'intelligence artificielle sont impuissantes, inefficaces ? Essayons, vérifions. Jouons. Le domaine évolue vite, sous la pression des technologies nouvelles. »

En confirmant l'importance actuelle de la haute technologie, des enjeux financiers et de la maîtrise du temps, [Créance, 2001] corrobore ces interrogations par une forte insistance sur les besoins que l'entreprise, "machine à rationaliser", a de concilier au quotidien concepts et mise en œuvre. De plus, la mise en place de réseaux dans les entreprises fait apparaître de nouveaux comportements de groupe qu'il est primordial de pouvoir analyser et modéliser afin de mieux les comprendre et de pouvoir surtout les consolider [Adam, Kolski & Vergison, 1998]. Les bouleversements structurels qui ébranlent le monde industriel d'aujourd'hui mettent en lumière le besoin d'appréhender correctement les mécanismes de fonctionnement de l'entreprise et, en particulier, des processus techniques et administratifs qui sous-tendent ses activités.

Dans ce contexte, les facteurs humains jouent un rôle prépondérant. De même en est-il pour la création en commun et le partage d'informations, de connaissances et de documents. L'émergence de nouvelles technologies et d'outils informatiques puissants, simples dans leur conception et dans leur application contraste souvent avec le manque de méthodes de mise en œuvre appropriées. Par ailleurs, si des solutions sont trouvées pour résoudre des problèmes locaux de nature le plus souvent technique, il n'en existe pas pour résoudre des problèmes organisationnels complexes où les facteurs humains sont critiques.

C'est pourquoi, il demeure indispensable d'avancer l'insertion de nouvelles interfaces intelligentes, avant tout dédiées à l'amélioration du travail et du bien-être des utilisateurs, comme une méthode ou un outil d'analyse en vue de la spécification de systèmes coopératifs. Simple tant dans sa conception que dans sa mise en œuvre, une interface intelligente appliquée à l'entreprise, et plus généralement au monde industriel, doit permettre aux différents intervenants dans un processus, acteurs et décideurs, d'abord d'en acquérir la perception et la maîtrise, pour ensuite également en améliorer, voire même en optimiser, le fonctionnement.

LEAR (Learner's Living Repository) en est un exemple [Oppermann & Thomas, 1996]. Support d'utilisateurs pour l'exploitation et l'échange, dans un groupe de travail, d'informations apprises et consultées, ces travaux décrivent, par "mémos" (pense-bêtes) ou copies d'écrans, des problèmes ou solutions trouvées lors de l'utilisation d'un outil déterminé. L'accomplissement de tâches spécifiques est alors associé à des commentaires et des stockages dans une "démothèque" personnelle. Basé sur un procédé itératif d'apprentissage tourné vers l'obtention d'une solution, ce système s'applique à réutiliser toutes les facettes de la compétence disponible :

- "mémos" individuels (attachement d'une description électronique à des éléments du système pour expliquer des caractéristiques utilisées pour des besoins individuels).
- enregistrement d'interactions (répétition et exploration de son propre historique d'interaction, par exemple dans le cas d'une situation d'erreur).
- définition de démonstrations pertinentes (sélection d'un épisode intéressant et conservation dans la "démothèque" personnelle pour des situations futures similaires).
- annotation de démonstrations (adjonction de commentaires et avertissement concernant les actions de chacun).
- recherche de démonstrations (sélection de différentes méthodes d'accès pour rechercher l'épisode pertinent de la "démothèque").
- sélection de vues (sélection de différents points de vues pour exploiter un épisode pertinent de la "démothèque" afin de supporter un nouvel apprentissage d'une solution et le transfert de cette dernière sur la tâche courante).
- échange de questions et de réponses (consultation distante entre l'utilisateur et un consultant pour échanger questions et réponses indépendantes des contraintes temporelles et spatiales).
- échange de solutions (échange de tâches et de compétences envers l'outil entre utilisateurs d'un environnement de travail coopératif par la fourniture et la requête de solutions typiques concernant l'espace de travail).

En définitive, il ressort ici déjà quatre aspects bien distincts au niveau de l'apprentissage destiné à augmenter la compétence d'un utilisateur envers un outil logiciel :

- omniprésence (l'apprentissage doit être supporté dans chaque situation de travail, pas seulement dans des phases ou des environnements d'apprentissage particuliers) ;

- combinaison d'exploration et d'instruction (les utilisateurs apprennent en s'essayant à des situations et en consultant des facilités d'aide technique et humaine) ;
- phénomène itératif (évolution étape par étape en utilisant la connaissance initiale pour une compréhension à venir) ;
- activité individuelle et sociale (les utilisateurs apprennent seuls mais apprécient aussi le support et l'échange de connaissances des interactions sociales).

Autre ressource appréciée au sein des entreprises et sociétés industrielles de fabrication, la réalisation d'interfaces d'aide à la décision en temps réel, pour la supervision notamment de procédés continus complexes [Lambert, Riera & al., 1998]. Dans ce cas, il s'agit de seconder l'opérateur humain au niveau de la surveillance, la détection et la localisation de défauts éventuels, tout en le laissant maître des actions à appliquer. Les procédés continus automatisés sont effectivement un excellent exemple de systèmes où l'interaction homme-machine prend un caractère primordial ; la qualité des interfaces permet de rendre leur supervision plus efficace et surtout plus sûre. Le procédé physique est complexe, à la fois par le nombre important de variables qu'il y a lieu de surveiller et par les multiples relations qui existent entre elles. De plus, les réactions sont difficiles à interpréter du fait du caractère en boucle du système de commande.

Cependant, un tel système présente les nombreuses données, acquises par exemple en ligne, en termes plus synthétiques et symboliques qu'une simple liste numérique de valeurs ou de dépassements de seuils, de façon à ce que l'opérateur maintienne aisément sa vigilance et se focalise sur les vrais problèmes lorsqu'ils apparaissent et non sur des problèmes secondaires. Ainsi, même si le système d'aide à la décision lui propose un diagnostic, entre-autres grâce à une représentation causale de l'installation visualisée par un graphe orienté facilitant sa décision, l'homme reste maître de déterminer la séquence d'actions à appliquer sur le procédé.

Un protocole expérimental complet a été mis en œuvre pour évaluer les performances des utilisateurs assistés par ce type d'interfaces intelligentes. La prise en compte d'un simulateur d'entraînement dédié aux opérateurs sur procédé de retraitement de combustibles nucléaires s'est avérée profitable, notamment du fait que ce système a pu fournir un jeu de données représentant le fonctionnement de référence et différents jeux de données ensuite correspondant à un fonctionnement avec un ou plusieurs défauts. Les qualités ergonomiques des vues proposées ont été jugées suffisamment claires pour que le bénéfice apporté par ces outils de supervision soient évalués dans de bonnes conditions. Certes, l'outil étudié n'était pas d'une grande complexité (une cinquantaine de variables à surveiller) et les pannes simulées n'étaient pas représentatives de toutes les possibilités de pannes. Mais les opérateurs étaient en situation suffisamment réaliste lors des tests pour entraîner, en particulier, des situations où stress et fatigue pouvaient se sentir.

Davantage orienté vers la bureautique, la célèbre société *Microsoft* a récemment dévoilé sa "Secrétaire Intuitive" [Vladyslav, 2001-2]. Logiciel permettant de hiérarchiser la délivrance des *e-mails*, ce système, couplé à une caméra et un microphone, détermine l'activité du destinataire, choisit s'il doit le prévenir ou non, et sur quel terminal, en fonction de l'importance qu'il attribue au message. A la différence des "Assistants Personnels de Communication" [Nangle, Cunningham & Evans, 1998] qui se chargent d'effectuer un filtrage intelligent des appels télématiques d'un utilisateur en lui apprenant à mieux les gérer, l'idée du système abordé est d'aller plus avant en cherchant à déterminer dans quel environnement se trouve l'utilisateur et ce qu'il est en train de faire, afin de lui envoyer les messages cette fois-ci triés en fonction de leur urgence. Imaginons un "pauvre" employé de bureau submergé par les courriers électroniques : messages d'une cousine éloignée, messages de groupes de discussion, annonces commerciales "forwardées" par un collègue scrupuleux, notes de son responsable, alertes prévenant d'un virus, etc. Autant d'informations qu'il faut hiérarchiser et qu'il faut lire rapidement, si ce n'est immédiatement.

Eric Horvitz, un des responsables de *Microsoft Research*, explique que « ce nouveau système est justement là pour comprendre comment un utilisateur perçoit l'urgence. Le système consulte ensuite le corps des *e-mails*, l'objet, la structure de l'en-tête, et prend même en compte les relations entre l'envoyeur et le récipiendaire, pour finalement assigner un niveau d'urgence compris entre 1 et 100 aux messages. Le logiciel permet ainsi de déterminer des profils différents (lieu de travail, domicile, en dehors du bureau) en s'appuyant sur un outil de hiérarchisation des priorités. De plus, le système sait que l'utilisateur est de face, sa tête est là, il sait que l'humain est là. Si ce dernier avait une fenêtre *Outlook* d'ouverte, le système le verrait dans *Outlook*, et s'il va dans *PowerPoint*, il est vu dans *PowerPoint*. Le système détermine par conséquent aussi si l'utilisateur regarde l'écran ou non, et peut alors réellement savoir ce qu'il fait. » Ceci explicité, le but étant d'adapter la délivrance des messages électroniques en fonction de la situation de travail caractérisant l'utilisateur, nous pouvons tout-de-même nous interroger ici sur le niveau et l'acceptation par cet utilisateur de l'indiscrétion inhérente à une telle interface intelligente...

Enfin, à l'image de nos évaluations réalisées au sujet des terminaux de communication, les interfaces intelligentes envahissent aussi très rapidement les sites du réseau *Internet*. Dans ce contexte, lorsque la distance ne compte plus, que l'offre n'est plus facteur de différenciation, que reste-t-il pour fidéliser le client ? Comment créer la proximité sur le "Net", une relation de proximité étant construite pas à pas dans la vie réelle ?... Cette dépendance, avant tout propre au commerce électronique entre un fournisseur et un client, possède heureusement un équivalent de taille sur le *Web*, à savoir la personnalisation [Dhénin & Lichter, 2000]. Abondance de l'offre et boutiques à un clic de souris les unes des autres : voilà à quoi ressemble l'environnement concurrentiel sur le *Web*. Pas très réjouissant pour les "cyber-marchands". Dans le commerce classique, non électronique, un client choisit une boutique plutôt qu'une autre en raison de sa situation géographique, ou de la relation qu'il entretient avec le commerçant. Sur le *Web*, comment apporter cette différence, sinon en tentant de réinventer cet interlocuteur qui connaît si bien son client, l'oriente vers des produits qu'il apprécie et lui donne le conseil attendu au bon moment ? La personnalisation désigne ainsi l'aptitude d'un site à servir ses visiteurs au plus près de leurs besoins en leur proposant un contenu individualisé.

Deux manières d'y parvenir sont alors possibles : interroger les clients sur leurs besoins (personnalisation explicite), ou bien dresser leurs profils en examinant, en temps réel ou non, leurs opérations sur le site pour ensuite leur suggérer des choix auxquels ils n'ont pas forcément pensé (la personnalisation est alors dite implicite). A cela, il faut ajouter qu'un client demeure très volatile. Parce qu'il se caractérise par une abondance de l'offre et, ainsi, par un commerce sans cesse en renouvellement, le *Web* crée plus qu'ailleurs un "consommateur qui n'est pas dupe". Ce client-là apprend alors vite, très vite ; il devient de plus en plus difficile de lui prendre de son temps pour lui communiquer un message de "marketing". Sauf s'il y trouve un intérêt...

Technique de personnalisation basée sur l'étude comportementale de l'internaute, le filtrage collaboratif offre alors une analyse prédictible à partir du cycle d'achat de clients présentant des profils voisins. [Net Perceptions, 2001] nous propose à ce niveau des outils basés sur des statistiques véritablement précises. Formulaire d'achat, questionnaire pour une inscription à une lettre électronique, demande de documentation, jeux-concours sont, tout d'abord, autant d'occasions de recueillir des informations sur les visiteurs d'un site *Web* [Dhénin & Lichter, 2000]. Ces données sont nécessaires pour déterminer les profils d'achat des internautes et représentent une mine d'or pour les entreprises qui peuvent ainsi mener, sur le *Web*, des actions de "marketing one-to-one", segmenter le marché pour adapter un produit à toutes les catégories d'utilisateurs [Créance, 2001], identifier les cibles et adapter les offres en conséquence. Cette personnalisation-là s'effectue avec la participation active et la complicité de l'internaute.

Une approche complémentaire consiste à étudier le comportement d'achat à l'insu du visiteur, mais en respectant son anonymat et en enregistrant son cheminement à travers chaque clic de souris. Cette seconde technique, connue sous le nom de "*clic-stream tracking*", est mise en œuvre au travers des outils de filtrage collaboratif. [Net Perceptions, 2001] s'appuie alors sur "l'expérience" des utilisateurs possédant un profil similaire pour recommander un nouveau service ou produit. A l'aide de serveurs de contenu qui se chargent d'associer des mots-clés à des objets d'une page *Web* dans le but de personnaliser un site en fonction du profil de l'utilisateur, l'entreprise dispose de statistiques précises pour formuler de nouvelles recommandations. Les logiciels de [Net Perceptions, 2001] se fondent sur leur propre base de données (appelée aussi base de recommandations) et sur un moteur de filtrage. La base de données est alimentée en temps réel par le serveur *Web* de l'entreprise ou par des routines, ou bien encore directement à partir du panier d'achat du consommateur. Toutes les informations concernant les achats de l'internaute sont ainsi enregistrées. Quand a-t-il acheté ? Combien de fois ? Quels produits ?...

En parallèle, le moteur de filtrage collaboratif traite les informations et stocke le résultat dans la base de recommandations. Ces traitements s'appuient sur deux fonctions principales, l'analyse et la recommandation qui, toutes deux, s'effectuent en temps réel. [Net Perceptions, 2001] adjoint même la possibilité d'extraire des informations, dans le but de mener une analyse plus poussée pour établir des règles qui s'appuient sur les préférences du client. Cependant, nous pouvons signaler que de telles techniques de filtrage collaboratif se heurtent de nos jours à une importante problématique de montée en charge. Les sites très fréquentés enregistrent désormais des millions de clics, qui sont l'expression de centaines de milliers de visites, elles-mêmes génératrices de plusieurs milliers de paniers d'achat en préparation [Dhénin & Lichter, 2000]. Les logiciels doivent donc être capables de traiter autant de ces profils. Pour les éditeurs d'outils, le défi consiste alors autant à affiner leurs moteurs qu'à améliorer leur aptitude à tenir la charge.

En définitive, le **Tableau 1.2.4.6** sur la page à suivre nous permet de résumer cette partie dédiée à la mise en valeur des modèles d'interfaces intelligentes dans le monde de l'entreprise, en laissant transparaître que ces procédés sont promis à un bel avenir dans une industrie plus que jamais pragmatique. Avant tout destiné à asseoir les notions contenues dans le sujet de la Thèse, ce chapitre initial nous aura ainsi permis d'intégrer l'ensemble du contexte inhérent à nos recherches, de même qu'à valider nos premières définitions et caractérisations ayant rapport aux interfaces intelligentes appliquées aux terminaux de communication de toute dernière génération.

Nous relevons alors que la relative faiblesse de [Oppermann & Thomas, 1996] par rapport aux autres références résulte d'une absence presque totale d'automatisation et d'auto-adaptation vis-à-vis des utilisateurs. Seules, une classification évoluée des nombreuses informations extraites du travail coopératif de ces derniers et "l'adaptabilité" de la caractéristique ergonomique générale nommée "flexibilité" en constituent de très remarquables exceptions. De son côté, si [Lambert, Riera & al., 1998] se démarque des précédentes investigations de par sa capacité à gérer des situations de réactivité et de sécurité extrêmes, ces travaux demeurent limités par l'ignorance d'un modèle dédié aux opérateurs et qui serait pourtant à ce niveau bien profitable.

La quête d'une auto-adaptation confortable et de modèles de l'utilisateur efficaces correspond par contre, dans leurs registres et contextes bien entendu respectifs, aux avantages majeurs émanant des deux derniers logiciels mis en évidence. Cependant, nous avons aussi pu observer que [Vladyslav, 2001-2] apparaît intrusif envers les utilisateurs, au point que la "secrétaire intuitive" ne laisse que peu de liberté à son interlocuteur en s'immisçant de manière exagérée dans ses activités privées. Quant à [Net Perceptions, 2001], les théories et expérimentations que concrétise cette référence sont visiblement plus discrètes et respectueuses des utilisateurs, tout en demeurant promues à un avenir intéressant, notamment en ce qui concerne la richesse du niveau d'appréhension et la pertinence des recommandations.

Tableau 1.2.4.6 : *Caractérisation hiérarchique des références d'interfaces intelligentes remarquables traitant de leurs applications respectives en entreprise*

Référence abordée	Valeur obtenue pour HII	Classe de l'interface	Principaux qualificatifs de l'intelligence
[Oppermann & Thomas, 1996]	10	Intelligence moyenne	échange dans groupe de travail - "démothèque" personnelle - réutilisation de la compétence - omniprésence - combinaison exploration et instruction - procédé itératif d'apprentissage - historique d'interaction - situation d'erreur - annotation - avertissement - méthodes d'accès - consultation distante
[Lambert, Riera & al., 1998]	12	Intelligence moyenne	aide à la décision en temps réel - supervision de procédés continus complexes - détection et localisation de défauts - visualisation par un graphe orienté facilitant le diagnostic - acquisition de données en ligne - termes symboliques - fonctionnement de référence - situation de stress et de fatigue
[Vladyslav, 2001-2]	13	Intelligence forte	"secrétaire intuitive" - hiérarchisation de la délivrance des e-mails - activité du destinataire - choix de prévention - terminal - importance des messages - environnement - triage - rapidité - compréhension - assignation d'un niveau d'urgence - profils - priorités - situation de l'utilisateur
[Net Perceptions, 2001]	15	Intelligence forte	personnalisation - profils explicites et implicites - suggestion de choix - étude comportementale - filtrage collaboratif - "clic-stream tracking" - serveurs de contenu individualisé - statistiques - segmentation de marché - anonymat - recommandations - "Customer Relationship Management"

1.3. Conclusion et orientation de notre projet de recherche

1.3.1. Récapitulatif et discussion des caractéristiques recensées

Beaucoup de références sont disponibles aujourd'hui, à l'image de toutes celles que nous avons répertoriées dans la bibliographie associée à ce mémoire. Cependant, dans un permanent souci de synthétisation et un besoin légitime de précision, nous n'en avons retenues et exposées que les plus significatives, celles estimées majeures pour la compréhension de nos recherches.

Voici donc un récapitulatif, en insistant très fortement sur les diverses caractéristiques issues des publications abordées vis-à-vis de nos définitions et de nos hiérarchies d'interfaces intelligentes (*Tableau 1.3.1* à suivre). Ainsi, en rappelant le rôle primordial joué par les valeurs obtenues pour notre degré **HII** et permettant en conséquence d'établir une hiérarchie de l'ensemble des interfaces intelligentes étudiées, s'en suivra l'établissement de nouveaux et divers commentaires généraux au sujet des nombreuses informations recueillies. Au fil des références et de l'ensemble des années qui se sont succédées pour composer nos présents états de l'art, nous relevons par conséquent à ce niveau que le besoin de programmation dédiée à l'utilisateur final est, de nos jours, de plus en plus apparent.

Tableau 1.3.1 : *Récapitulatif de nos caractérisations hiérarchiques appliquées aux références d'interfaces intelligentes mises en avant par nos états de l'art*

Catégorie thématique de l'interface	Référence étudiée	Valeur obtenue pour HII	Classe de l'interface
<i>Aides automatiques et tutoriels informatiques</i>	[Benyon, 1993]	8	Intelligence faible
	[Harrington, Banks & Santos, 1996]	11	Intelligence moyenne
	[Gavrilova & Voinov, 1997]	14	Intelligence forte
<i>Techniques de base en exploitation des connaissances</i>	[Garbay & Maitre, 1997]	9	Intelligence moyenne
	[Rhee, Kim & al., 1997]	9	Intelligence moyenne
	[Kohonen, Kaski & Lappalainen, 1997]	12	Intelligence moyenne
<i>Enseignement Assisté par Ordinateur</i>	[Murray, Schell & Willis, 1997]	8	Intelligence faible
	[Brusilovsky & Schwarz, 1997]	8	Intelligence faible
	[Murphy & McTear, 1997]	12	Intelligence moyenne
<i>Agents logiciels intelligents</i>	[Lieberman, 1995]	13	Intelligence forte
	[Maglio & Barrett, 1997]	11	Intelligence moyenne
	[Maulsby & Witten, 1997]	12	Intelligence moyenne
	[Nangle, Cunningham & Evans, 1998]	14	Intelligence forte
	[Pynadath, Tambe & al., 2000]	15	Intelligence forte
<i>Applications en entreprise</i>	[Oppermann & Thomas, 1996]	10	Intelligence moyenne
	[Lambert, Riera & al., 1998]	12	Intelligence moyenne
	[Vladyslav, 2001-2]	13	Intelligence forte
	[Net Perceptions, 2001]	15	Intelligence forte

Beaucoup d'utilisateurs souhaitent personnaliser leurs applications. Si les professionnels trouvent eux-mêmes de savantes tâches répétitives qu'ils déléguaient autrefois aux employés et secrétaires, nombreux sont les gens qui subissent aujourd'hui une tension répétée et qui ont par conséquent besoin de réduire l'usage du clavier et de la souris [Maulsby & Witten, 1997]. C'est pourquoi, les développeurs actuels répondent avec un torrent de nouvelles caractéristiques, toutes plus originales et attrayantes que les unes que les autres.

Face à la quantité des travaux et autres courants, théoriques et pratiques, nous avons cherché à parcourir, découvrir et élaborer un panel et une catégorisation aussi efficaces que possible des domaines abordés par nos recherches. En ce but, nous avons opté pour l'extraction, la compréhension et le développement des interfaces intelligentes les plus représentatives ces dernières années, non sans avoir effectué des études complètes de multiples autres travaux ayant rapport avec ce sujet, comme en témoigne notre bibliographie détaillée avant les annexes.

Un premier enseignement déjà corrobore nos présentes investigations : les capacités et qualités d'intelligence des interactions homme-machine étudiées sont en très nette et constante amélioration avec la succession des années. Les idées, les impulsions théoriques tout comme les divers choix pragmatiques effectués dans le contexte des interfaces intelligentes s'avèrent majoritairement judicieux, accentués qu'ils sont encore par l'évolution émérite de chacune des cinq catégories que nous avons réussi à extraire, au travers notamment du **Tableau 1.3.1**.

Nos divers “échantillons” remarquables d’interfaces intelligentes demeurent également catalysés par l’évolution brillante des techniques et des modèles d’Intelligence Artificielle. Ces derniers ne présentent cependant aucune caractéristique d’intelligence véritablement très forte. En effet, même si les applications exemplaires du type de [Pynadath, Tambe & al., 2000] comme de celui de [Net Perceptions, 2001] n’en sont plus éloignées et n’attendent qu’un développement plus prononcé du réseau *Internet* actuel, il se dégage de nos états de l’art un constat de perfectibilité : il y a, aujourd’hui, vraiment matière à progression dans ce domaine. Du point de vue contraire, aucune intelligence très faible n’est non plus à déplorer... Nous pouvons d’ailleurs relever que, dans le domaine de l’Intelligence Artificielle qui est le nôtre, c’est par définition l’apprentissage par cœur qui correspond au niveau le plus bas de l’intelligence. Or, tous les travaux que nous avons étudiés se situent, sans exception, à un niveau d’intelligence déjà bien supérieur à celui-ci.

Il apparaît intéressant de saisir que le calcul du “Degré d’Intelligence” **HII** d’une simple interface intelligente (et de ses indispensables caractéristiques de base) qui présente un unique apprentissage par cœur aboutit à une valeur sensiblement limitée à 1 sur un total défini à 20 par nos investigations précédentes. Correspondant donc à une intelligence très faible, cette catégorie d’interfaces représente, pour nos travaux d’auto-adaptation à l’utilisateur, le minimum de départ, autrement dit l’intérêt minimal requis pour une communication homme-machine dite intelligente ou “avancée”. Un parcours d’informations uniquement séquentiel associé à un traitement, sans aucune appropriation notable, des besoins individuels d’apprentissage pour un utilisateur donné équivaut par conséquent, sur notre échelle des interfaces intelligentes, à un degré d’intelligence égal à l’unité. De cette manière, l’assise de notre système de définitions devient attrayante : une interface présentant un niveau nul pour **HII** est correctement estimée comme étant en effet dénuée de toute intelligence propre. Une autre évaluée à un seul et unique degré appartient à une structure mettant en avant la plus faible des intelligences existantes.

En conséquence, nous pouvons tout d’abord remarquer que les deux catégories que nous avons retenues en premier lieu se distinguent assez nettement l’une de l’autre. Les références tournées vers les aides automatiques et les tutoriels informatiques présentent d’avantageuses caractéristiques interactives, liées à une communication bilatérale agréable entre l’homme et sa machine. Celles plutôt dédiées aux différentes techniques de base de l’Intelligence Artificielle choisissent de mettre tous leurs atouts dans l’amélioration de leurs qualités purement théoriques, entièrement immergées dans un contexte de recherche fondamentale. Pour en avoir fait nous aussi l’expérience, il reste difficile, à l’avènement d’un nouveau concept d’interface intelligente, d’allier technique et convivialité. Dès lors que l’on s’attache à élaborer d’innovants modèles, l’interface perd très souvent en “humanité” puisque le côté technique entraîne un délaissement des préoccupations d’image concernant l’application étudiée, de sa couche de plus haut niveau, et parfois même de la prise en compte “sympathique” de l’utilisateur. Dans le cas des travaux issus de ces deux premiers domaines, il en est ressorti de même au niveau du traitement sous-jacent des éventuelles données erronées émanant des manipulations successives des utilisateurs, voire également des conséquences en matière de sécurité des informations.

De leur côté, les principes de l’Enseignement Assisté par Ordinateur présentent aussi quelques déficiences puisque cette troisième classe se distingue par deux inconvénients majeurs [Bruillard, 1997]. En effet, ce domaine est caractérisé par une interaction et un imposant support de données dont les mises en œuvre demeurent très lourdes. La préparation d’un cours d’EAO requiert un temps considérable en recherches, élaborations et conceptions informatiques. Des multiples et diverses questions aux réponses attendues des apprenants, des critères de déclenchement aux nouveaux points de continuation pour chacun de ces branchements définis, tout ceci doit être soigneusement spécifié par l’auteur du cours. Et cela ne s’arrête pas là puisqu’il ne faut pas omettre non plus que le résultat obtenu est peu flexible, donc difficilement

modifiable et réutilisable à terme dans le cadre d'une nouvelle problématique : changer un ensemble de questions ou des éléments du cours conduit toujours à un gros travail de réécriture du programme considéré.

Mais l'EAO, à l'image d'ailleurs de ses nombreuses variantes comme par exemple l'EIAH (Environnement Informatique pour l'Apprentissage Humain) ou encore l'EIAO (Environnement Interactif d'Apprentissage avec Ordinateur), n'est pas moins privé d'avenir prometteur pour autant. Ses défenseurs ont su s'orienter au bon moment vers la quatrième catégorie d'interfaces intelligentes que nous avons pu mettre en avant, à savoir les agents logiciels en général. En effet, la solution qui semble s'imposer aujourd'hui en EAO fait appel à des personnages virtuels, désignés par le terme "agents" et véritables auxiliaires pédagogiques de plus en plus évolués qui s'attellent à la rude tâche d'aider les élèves à apprendre [Alberganti, 1999]. Nombre de chercheurs en Intelligence Artificielle, en imagerie et en animation, en multimédia aussi, tentent de concocter toute une panoplie de nouvelles créatures virtuelles, que ce soient des professeurs, des compagnons ou encore des tuteurs... Destinés à assister les enseignants, voire déjà à les remplacer dans certaines tâches, ces nouveaux outils logiciels doivent relever, outre l'adaptation à l'ordinateur des diverses disciplines enseignées, un défi majeur : reconstituer entre l'ordinateur et l'élève une relation s'approchant le plus près possible du rapport humain.

Ces créatures peuvent jouer de multiples rôles. Elles endossent la peau du professeur, celle de l'élève ou se transforment en agent de renseignement. Les chercheurs exploitent à ce niveau toutes les variations possibles d'une unique situation, à savoir l'élève face à la machine, plus proche de la leçon particulière que du cours collectif dispensé dans une classe. Poussant cet avantage encore plus loin, ils explorent des stratégies pédagogiques inapplicables dans le cadre de l'enseignement classique. Grâce à l'Intelligence Artificielle, les agents virtuels analysent de plus en plus finement les réactions des élèves, scrutant le moindre signe d'incompréhension ou d'ennui. Insensibles à la fatigue et à la lassitude, ils varient leurs techniques d'enseignement jusqu'à identifier, pour chaque élève, celle qui donne les meilleurs résultats. Mais ces agents ne s'arrêtent pas là puisqu'ils permettent en outre d'introduire un paramètre dit "de comportement". Suivant l'option choisie, l'agent peut en effet prendre son temps, ou répondre rapidement à l'étudiant pour témoigner à ce dernier d'une grande confiance en soi. L'analyse des interactions entre comportement et connaissances conduit à la définition de "critères de performance".

En conséquence, nous aboutissons au développement et à la migration de nos premières catégories d'interfaces intelligentes recensées vers les innovants agents logiciels intelligents et les systèmes multi-agents qui leur sont associés. Nous découvrons que ce sont ces agents qui présentent de nos jours le meilleur intérêt en matière d'interfaces intelligentes, ce en constituant notamment la base de travail la plus prometteuse pour le domaine étudié. Ces procédés, issus des plus récentes recherches en Intelligence Artificielle comme en attestent nos références recueillies à ce sujet, permettent d'identifier et de collecter automatiquement l'information dépendant d'un utilisateur, puis d'aider à construire un modèle d'utilisateur prédicateur [Motoda, Washio & al., 1997]. Cependant, le nombre et la complexité de telles caractéristiques, certes développées dans le but de personnaliser au maximum nos applications, peuvent néanmoins entraîner la création illicite de beaucoup de soucis d'utilisabilité [Maulsby & Witten, 1997]. Du fait de l'enrichissement fonctionnel et graduel des interfaces, l'utilisateur se trouve de plus en plus souvent confronté à une complexification de leur présentation et de leur mode d'interaction [Wolff, 1999]. Pire, cette sophistication tend à réduire leur intérêt, conduisant les interfaces à s'écarter de leur objectif original, c'est-à-dire celui se résumant à simplifier les interactions entre l'homme et la machine !

Nous avons montré qu'une catégorie de travaux a commencé à maîtriser ces nouvelles difficultés afin d'adapter les agents intelligents à d'innovantes applications quotidiennes comme industrielles, notamment en ce qui concerne un des domaines informatiques les plus attractifs du

moment : le réseau *Internet*. Ainsi, personnaliser pour fidéliser, faire coïncider le contenu d'un site *Web* avec le profil de l'internaute courant constitue aujourd'hui le principal enjeu du déjà inévitable "commerce électronique". Ce marché de la gestion de contenu explose : le *Gartner Group* estime à 5,7 milliards de dollars le budget mondial pour 2002 accordé aux dépenses concernant ces technologies de personnalisation [Dhénin & Lichter, 2000]. Face à la prolifération des sites *Web*, une entreprise offrant la vue d'un site personnalisé en fonction du profil de celui qui le consulte a toutes les chances de capter l'attention du visiteur. Mieux encore, en établissant une interaction avec le consommateur, c'est-à-dire en lui proposant des produits ou des services associés à ses habitudes d'achat, l'entreprise offre un service à forte valeur ajoutée, susceptible de le fidéliser. Nous avons étudié deux approches standard en matière de personnalisation. Rappelons que la première, à l'initiative de l'internaute, effectue une requête classique à partir d'un moteur de recherche ou d'un portail. Les "alertes" permettent de recevoir des informations en fonction des centres d'intérêts et les "mailing-lists" assurent l'envoi d'une lettre thématique régulière. La seconde approche consiste à utiliser les technologies de personnalisation sur le site *Web* lui-même. Parmi ces techniques, les "cookies" sont les plus connus : le profil de l'utilisateur est stocké dans un fichier sur son disque dur, ce qui permet, lors d'une connexion, de le reconnaître. Mais les "cookies" ne sont pas suffisants pour établir un profil pertinent.

En effet, les outils de personnalisation de contenu font désormais leur apparition. Maillons essentiels dans la chaîne du *CRM* (*Customer Relationship Management* ou "gestion de la relation client"), ils permettent de comprendre l'utilisateur en vue de maximiser son panier d'achat. Cependant, s'il est vrai que les solutions de personnalisation s'adressent avant tout aux responsables "marketing" et aux forces commerciales qui ont besoin de solutions simples et efficaces à utiliser, la personnalisation doit être en conséquence utilisée de manière appropriée suivant les différents stades de la relation avec le client. Une utilisation "intelligente" améliore ainsi le service envers chaque client et utilisateur potentiel.

Il apparaît maintenant clairement que les agents intelligents rendent le contexte général du travail informatique plus productif et plus agréable [Maulsby & Witten, 1997]. Ils demeurent également porteurs d'une nouvelle technologie très avantageuse pour tous les utilisateurs. En effet, si les interfaces personnifiées aident constamment les utilisateurs à engager une tâche précise, les interfaces basées sur les agents demeurent relativement flexibles pour supporter la diversité des préférences et la nature de diverses tâches à traiter [Koda & Maes, 1996]. L'utilisation d'agents intelligents entraîne ainsi la possibilité réelle de construire des modèles d'activités humaines intelligentes dans de larges domaines, spécialement à grande échelle et pour des problèmes toujours plus complexes [Yunfei, Renshou & al., 1997]. Nous constatons également que l'émergence de nouvelles technologies d'interfaces comme de mécanismes d'interaction homme/machine "intelligents" modifie notablement les modes de consommation des services de communication, ce en développant l'usage de services existants et favorisant l'apparition de nouveaux usages (technologies vocales, univers virtuels, touches et gants à retour d'efforts, sécurité et domotique, adaptation au contexte d'usage grâce aux services de transcription et de traduction automatique, ...) [Sorin & Colaitis, 1999].

En conclusion, si nous sommes conscients que nos états de l'art comme nos définitions concernant les concepts inhérents à notre Hiérarchie des Interfaces Intelligentes seront sûrement encore appelés à être complétés par le futur, ces nouvelles spécifications nous ont déjà permis de découvrir, comprendre et évaluer ce vaste domaine des interfaces intelligentes appliquées aux terminaux de communication. Et, si nous étions dans la situation très compliquée de ne devoir retenir ici qu'une seule image de toutes nos investigations, nous garderions en mémoire cette indiscutable vérité actuelle : « Les utilisateurs jugent de plus en plus la qualité d'une application informatique au travers de son interface homme-machine » [Crampes, 1997].

1.3.2. Dégagement de nos axes stratégiques de travail

Nos diverses investigations nous amènent finalement à nous orienter vers une maquette des différentes approches jugées attractives pour concrétiser nos futurs travaux d'implémentation au sujet du modèle d'interface intelligente recherché (**Figure 1.3.2**). Si cette réflexion s'appuie bien entendu sur les travaux et définitions que nous avons mis en œuvre, elle sera encore très largement étoffée, en partie grâce aux prochaines spécifications de nouveaux concepts d'assistantes intelligents. Nous aborderons alors un nombre important de propositions entraînant une personnalisation efficace des terminaux de communication, ainsi qu'un modèle d'utilisateur qui devra prendre en compte la majeure partie des caractéristiques individuelles et "intelligentes" de l'utilisateur.

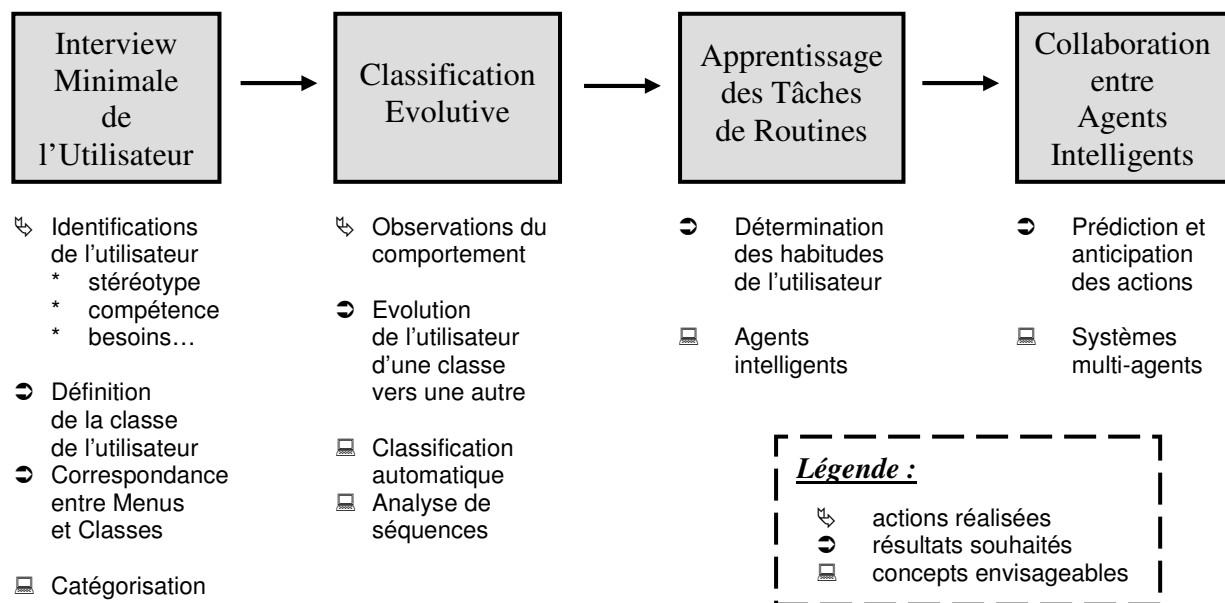


Figure 1.3.2 : *Mode expérimental et premiers concepts opératoires envisageables pour la mise en œuvre de notre modèle d'interface intelligente*

S'il va s'agir de spécifier, discuter et valider la puissance du raisonnement, l'efficacité, la faisabilité ainsi que la réalité de mise en œuvre d'un tel système, nous tenterons constamment de réduire au maximum les contraintes pouvant être imposées à chacun de nos utilisateurs potentiels. Nous verrons, au cours du **Chapitre 3** notamment, qu'il devient absolument nécessaire de limiter, voire de complètement annihiler, la première phase introduisant nos concepts exposés précédemment. En effet, et bien que souhaitée minimale, l'étude de cette "interview directe de l'utilisateur" nous montrera qu'elle demeure intrusive, gênante et parfois déséquilibrante pour l'homme. En conséquence, nous avons cherché, et réussi, à la remplacer par un innovant modèle auto-adaptable totalement implicite, que nous avons entièrement intégré au second modèle de la "classification évolutive".

Par suite, nous développerons et approfondirons également les deux dernières étapes du schéma de fonctionnement, principalement au sein des **Chapitres 4 et 5**. Ainsi, nous aboutirons à l'application d'une récente théorie d'apprentissage automatique permettant avantagement le traitement en temps réel des diverses habitudes répétitives de chacun des utilisateurs ainsi que

leurs exploitations respectives et auto-adaptables. En étudiant alors, entre-autres, l'intéressant concept du "séquençement d'actions", nous insisterons encore sur la conséquente base de travail apportée par les catégories étudiées jusqu'ici en matière de conception d'interfaces intelligentes. Nous verrons effectivement que toutes les techniques d'assistances automatiques, d'Intelligence Artificielle, d'enseignement informatique automatisé, d'agents logiciels intelligents, et ce sans oublier la personnalisation de l'utilisateur et la traduction de ses caractéristiques en entreprise, nous permettrons de définir, d'élaborer et de construire, en plusieurs prototypes interactifs, un modèle d'interface que nous avons cherché à établir aussi intelligente que possible, de même que pleinement dédiée au monde des télécommunications.

Pour résumer, nous allons fournir un accès à la fois aisé, convivial, adapté, automatisé et aussi peu intrusif que possible à des services de communication, de diffusion et d'information, rendus intelligents. Tout utilisateur doit pouvoir alors en être bénéficiaire, qu'il soit débutant ou expérimenté, calme ou débordé par ses propres activités, attentif ou non, opérant dans un besoin précis ou simplement intuitif, organisé ou désordonné, etc. A l'image de [Sorin & Colaïtis, 1999], nous devons offrir des interactions "utilisateurs-services" efficaces pour, par exemple, accéder rapidement à une information recherchée et communiquer avec d'autres personnes (ou leurs avatars) avec le meilleur sentiment de confort et de "télé-présence". Nous devons organiser et faciliter le travail en proposant à l'homme d'avoir accès de manière "intelligente" aux ressources bureautiques disponibles. Dans ce domaine, les attentes des utilisateurs se définissent, de façon très exigeante, par une demande de fluidité certaine des accès, des contraintes opérationnelles minimales, un confort d'utilisation et d'assistance, une personnalisation aussi, une sécurisation et enfin une confidentialité.

Afin d'œuvrer à la réalisation de ces nombreux et ambitieux intérêts, nous retiendrons que, si toute communication avec les machines, toute commande avec celles-ci s'effectue par l'intermédiaire d'un ensemble d'instructions, tout ce qui fait la souplesse du langage ordinaire réside dans l'implicite et l'imprécis. Or, avec les machines, rien de cela n'est malheureusement autorisé : « seule la lettre du texte est considérée, jamais l'esprit... » [Ganascia, 1993]. La conquête des machines passe donc par la conception d'un langage commun à l'homme et à la machine, un langage qui soit suffisamment léger pour convenir à la machine ; un langage bref, incisif, au moyen duquel nos ordres seraient transmis rapidement, sans ambages, sans ambiguïtés, sans que l'on s'embarrasse de formules compliquées... On le voit, les questions qui se posent aux spécialistes des machines ne sont plus uniquement des questions de physique ou d'électronique. Indépendamment de leur réalité matérielle, les machines modernes ont aussi une réalité abstraite, d'ordre logique, mathématique et linguistique, une réalité qu'on peut qualifier "d'immatérielle".

Le rôle joué par une interface est donc central, voir primordial et à jamais indispensable [Bruillard, 1997]. Définissant le cadre réel du travail de l'apprenant, une interface correspond à un milieu filtrant les possibilités d'expression de l'apprenant et pouvant être réactif. Elle doit ainsi répondre à deux contraintes contradictoires : la transparence (s'intégrer dans le monde humain), mais aussi la résistance (ne pas perdre son rôle structurant pour l'utilisateur). C'est pourquoi, toute action que permet ou contraint l'interface constitue une aide précieuse pour l'utilisateur.

Nous nous attacherons à aider l'ordinateur à fournir des représentations pertinentes pour l'apprenant qui a à les interpréter et à les manipuler, sans avoir omis auparavant de permettre une prise en compte de l'utilisateur, non comme un être abstrait idéal, mais comme un des éléments centraux de nos modèles d'interfaces intelligentes.

Chapitre 2 :

Spécifications de Nouveaux Concepts d'Assistants Intelligents

*Le travail de l'ergonome va,
pour la conception d'un logiciel,
de l'analyse de la tâche
à la définition de spécifications fonctionnelles
en passant par la définition des interfaces.*

[Chene, Tijus & Poitrenaud, 1998]

Parmi les différents sujets que nous avons étudiés et suite aux nombreux enseignements issus de nos états de l'art, nous nous devons d'introduire, dès à présent, la notion d'apprentissage par agents intelligents dans une interface-utilisateur dédiée aux terminaux de communication. Cela se conforme à un intéressant travail d'ergonome qui traduit relativement fidèlement une partie de celui qui est le nôtre au cours de nos présentes recherches. Nous abordons ainsi l'identification, l'élaboration, la spécification complète de nouveaux agents intelligents destinés à améliorer et à enrichir les tâches et autres moyens d'épanouissement de nos utilisateurs. Ces derniers demeurant respectivement alloués à une ou plusieurs fonctions précises et spécialisées, nous obtenons en définitive un complet système multi-agents chargé de définir un innovant modèle d'interface intelligente pour terminaux de communication.

Ce procédé, notamment chargé de rendre les opérations courantes moins fastidieuses par leur répétition automatique dès que l'utilisateur le souhaite, s'avère être d'un grand intérêt dans l'interface "homme-machine", c'est-à-dire dans la communication bilatérale entre l'homme et l'appareil. En effet, si le terminal est amené à mettre en confiance, à discuter de la manière la plus conviviale et explicite possible avec l'utilisateur, ce sont ces agents intelligents, et plus précisément les "agents d'interface", qui se doivent de rendre la machine plus humaine et plus sympathique. Nous verrons qu'ils se chargent aussi d'améliorer les modes d'expression entre les deux parties, ce en analysant constamment les opérations et les besoins de l'interlocuteur.

Cette phase donne alors lieu, entre autres avancées, à la conception d'intéressantes structures destinées à assister, aider, faciliter et rendre le travail quotidien de chaque utilisateur plus efficace et donc moins "coûteux", en temps comme en difficultés. Regroupés au sein de nos prototypes que nous avons choisi de nommer « Assistant Individuel de Télécommunication » (AIT), ces modèles auto-adaptables sont dédiés à la mise en œuvre annoncée d'une nouvelle interface intelligente logicielle.

Par extension, nous prenons à ce niveau l'utilisateur réellement en compte, et l'introduisons plus précisément dans le traitement de l'extraction d'informations remarquables. Nous découvrons et définissons alors également la notion inhérente d'auto-adaptation, celle-ci correspondant à chacune des données, pourtant issues de catégories différentes de domaines, ainsi qu'aux intérêts multiples de l'utilisateur concerné. Associé à des systèmes de recherche d'informations balisant des références pertinentes dans un ensemble relativement statique de documents, nous choisissons de plus d'intégrer ce principe d'auto-adaptation à divers systèmes de filtrage d'informations. Ces derniers se voient ainsi caractérisés par l'examen d'un flux dynamique de documents et la conservation à terme de ceux se rapportant uniquement aux mots-clés propres à l'utilisateur [Benaki, Karkaletsis & Spyropoulos, 1997].

Nous remarquons ensuite que l'adaptabilité d'une interface peut porter sur de nombreux aspects qui concernent néanmoins, tous, la gestion du dialogue avec l'utilisateur (mode d'accès aux fonctionnalités, organisation des affichages, guidage des utilisateurs, etc). Cependant, nous comprenons aussi dans ce chapitre que, dans le cadre d'un projet destiné à concevoir un produit manipulable par le "grand public", l'aide fournie aux utilisateurs est un point essentiel dans sa conception [Bernard, 1998]. Mieux encore, la capacité de notre nouveau système auto-adaptable à aider les utilisateurs aux moments adéquats devient source d'une grande valeur dans la fourniture d'un produit, convenant aussi bien aux clients néophytes que plus expérimentés.

Nous nous attelons enfin à préparer l'implémentation proprement dite des modules de notre "Assistant Individuel de Télécommunication", ce en réalisant une spécification complète de nos modèles auto-adaptables, à l'aide de *LOTOS*, langage formel pleinement dédié aux principes de la communication en général et de l'informatique "temps réel" en particulier. A ce propos, nous ne manquons pas de justifier le choix d'un tel outil, non sans avoir auparavant rédigé un état de l'art de tous les langages formels aujourd'hui disponibles et, surtout, attrayants au sein du vaste domaine qu'est celui des télécommunications.

Les premières étapes de la ligne directrice de notre raisonnement passent en conséquence successivement par les notions d'apprentissage automatique grâce aux agents intelligents, ainsi que par l'extraction et la recherche d'informations remarquables. S'en suivent l'aide, la réaction et l'auto-adaptation à toutes les facettes des comportements ainsi recueillis et envisageables pour l'utilisateur. Divers et variés, nous verrons par ailleurs que ces agissements nous ont plongés au sein de multiples domaines, aussi disparates qu'enrichissants : l'ergonomie, l'architecture des systèmes informatiques, le génie logiciel et surtout les spécifications formelles. Ces dernières s'avèrent indispensables à la préparation active d'une future implémentation, rigoureuse et soignée, précise et saine, en un mot... efficace.

Entièrement décrit par le présent chapitre, l'ensemble de ces investigations, au combien complémentaires, nous entraînera ensuite à entreprendre l'apprentissage et l'acquisition de connaissances au travers de l'élaboration d'un nouveau modèle d'utilisateur. Statique et surtout incrémental, celui-ci nous conduira à établir un profil évolutif du comportement d'un utilisateur face à un terminal de communication accédant à *Internet*. Nous pourrions alors développer un autre domaine de l'Intelligence Artificielle, à savoir l'étude des structures et des algorithmes d'apprentissage automatique, qui nous permettra à terme de reconnaître, de traiter et de répondre aux multiples actions de l'utilisateur, et ce de manière de plus en plus complète et personnalisée.

2.1. Le concept « d'Assistant Conversationnel »

2.1.1. Enoncé et définitions

2.1.1.1. Intitulé du procédé

Par ce premier concept intelligent, nous cherchons à améliorer, de manière significative, la convivialité et l'efficacité d'un travail quotidien sur les divers terminaux de communication d'ALCATEL. En nous basant sur chacune des quatre fonctions disponibles sur le *WebTouch*, à savoir la téléphonie avancée, la gestion du courrier électronique, l'accès aux services *Internet* et l'annuaire intégré, nous nous concentrons plus précisément sur l'explication et la spécification de l'application de cet "Assistant Conversationnel" au courrier électronique.

Le concept se base alors sur le principe suivant : un utilisateur quelconque souhaitant joindre un interlocuteur doit pouvoir toujours voir sa requête se réaliser, quelque soit le moyen choisi par le terminal et sans que l'utilisateur ait à choisir lui-même le moyen adéquat.

2.1.1.2. Premières règles de base

Cela s'est traduit par l'entrevue de plusieurs règles de fonctionnement indispensables et adaptées à ce nouveau concept. L'énumération à venir nous en fournit les principales données :

- Lorsque l'utilisateur courant indique au terminal qu'il souhaite joindre une tierce personne à l'aide d'un utilitaire donné ("*e-mail*" par exemple), sa demande doit impérativement aboutir, à savoir que cette tierce personne doit en définitive obligatoirement disposer de l'information que voulait lui faire parvenir l'utilisateur.
[Règle n°1]
- Si, à ce moment-là, "*e-mail*" courant n'est pas connu ou reconnu par le terminal, n'est pas accessible, ne répond pas ou encore renvoie un message d'erreur, le terminal doit pouvoir utiliser, en agissant seul, d'autres moyens à sa disposition (autres adresses "*e-mails*" connues concernant la personne souhaitée, numéros de téléphones fixes, numéros de téléphones mobiles, *fax*, etc).
[Règle n°2]
- Dans le cas où le terminal entre en contact avec un appareil de messagerie vocale ou un répondeur, il interrompt de lui-même la communication et choisit un nouveau moyen de joindre l'interlocuteur souhaité.
[Règle n°3]
- Mais, si cet appareil de messagerie vocale ou ce répondeur reste en définitive le dernier moyen existant pour joindre la tierce personne, le terminal se doit alors de proposer à l'utilisateur de laisser un message contenant l'information qu'il souhaite faire parvenir à son interlocuteur.
[Règle n°4]

2.1.1.3. Adjonction de premières notions d'intelligence

A ce stade, nous attendons de nos modèles un dialogue performant, non plus basé sur un simple contrôle de l'objet de communication, mais construit à partir d'une personnalisation et d'une humanisation "sympathique" du terminal. En effet, si ce dernier est autonome en pratique, il est aussi amené à mettre en confiance et servir l'utilisateur de manière explicite et efficace.

C'est ainsi qu'intervient déjà ce qui restera deux de nos centres d'intérêts principaux, à savoir la reconnaissance et l'apprentissage automatique de toutes les connaissances issues des utilisateurs, depuis les actions et manipulations réalisées par chacun d'entre eux sur les interfaces jusqu'aux ensembles respectifs de leurs données comportementales et personnalisées. En ce sens, et en concordance par trois fois avec les règles élaborées précédemment, les rôles de soutien logistique et de traitement intelligent de l'Assistant Conversationnel peuvent être approfondis :

- Le terminal courant doit pouvoir connaître à tout moment, et donc apprendre en temps réel ou avoir déjà appris au préalable, les différents moyens qui existent (ou qui lui sont disponibles) pour pouvoir joindre l'interlocuteur souhaité.

[Extension de la Règle n°2]

- Dans le cas où le terminal rencontre un échec dans sa tentative d'appel de l'interlocuteur, entre en contact avec un appareil de messagerie vocale ou encore un répondeur, il doit pouvoir choisir un nouveau moyen capable de joindre l'interlocuteur souhaité, mais surtout savoir opter pour l'utilitaire qui semble le plus pertinent et le plus adéquat parmi les restants.

[Extension de la Règle n°3]

- En proposant à l'utilisateur de laisser un message d'informations sur la messagerie vocale ou le répondeur de l'interlocuteur, le terminal gagne à connaître déjà les "petites habitudes" de son utilisateur, et ce en matière de mise en page, d'entête et de signature, de références, de style, d'expressions et de mots-clés, etc. L'utilisateur peut de cette manière, grâce aux abréviations et autres raccourcis, transmettre son message à son interlocuteur sans prendre trop de temps.

[Extension de la Règle n°4]

- Nous pouvons également envisager d'implémenter dans cet outil la notion de filtrage, mais il s'agit alors de filtrage de moyens d'information. En effet, le terminal peut proposer à l'utilisateur d'éviter, temporairement ou non, de fonctionner avec certains des moyens pourtant bel et bien existants et utilisables. Mieux : une personne susceptible d'être jointe, mais ne désirant pas l'être à travers certains moyens (par le téléphone si elle est en réunion par exemple), peut demander à filtrer ses différentes façons d'être appelée.

Cependant, dans ce cas, il est à noter que l'Assistant Conversationnel ne reste plus entièrement autonome : l'interface devient effectivement semi-automatique puisqu'une information émane de l'utilisateur, et nous perdons par conséquent le côté "sans décision" ("decisionless") concernant ce dernier.

[Règle n°5]

Il reste à énoncer l'indispensable interface "homme-information" qui, si elle autorise la navigation du terminal intelligent et donc de l'utilisateur (sur le réseau *Internet* notamment), elle permet aussi à cet utilisateur d'obtenir la représentation et le tri des données recueillies, de même que le travail et la confirmation de l'information au cours du temps (mémorisation, rappel des

dernières opérations, ...). Cela s'avère primordial en matière de convivialité et mise en confiance de l'utilisateur de la part du terminal ainsi "humanisé", si ce n'est que pour l'assurer que son interlocuteur a bien été joint (dans le cas où il ne lui a pas parlé directement) ou a bien reçu son message, complet et en temps réel.

Le choix de baptiser notre concept par la combinaison des deux termes "Assistant" et "Conversation" résulte, au sein notamment des équipes du *Département Recherche* d'ALCATEL, d'importantes discussions de terminologie, qu'elles aient été technologiques, industrielles, conceptuelles ou même encore commerciales. Dans notre quête "d'humaniser" au possible nos différentes machines, la "Conversation" apparaît la plus directe et représentative d'une entente ou relation privilégiée, voire presque inter-personnelle, entre l'homme et la machine. Par contre, concernant le nominatif d'une idée d'assistance, nous avons tout d'abord envisagé les termes de "Secrétaire" et "Standard", que ceux-ci soient considérés comme "Electroniques", "Intelligents", "Automatiques", "Autonomes" ou "Personnalisés"... Cependant, si nous avons alors perçu un côté très mécanique et relativement impersonnel pour le second de ces deux mots principaux, le titre de "Secrétaire" nous est tout-de-même apparu comme traduisant un caractère presque "trop humain", et donc *a fortiori*, dans l'esprit des utilisateurs, sujet justement à nombre d'erreurs... "humaines" ! La qualité de dévouement insatiable émanant naturellement du terme "d'Assistant" nous a alors définitivement fait pencher en faveur de l'expression entérinée dans ces pages.

Cette étape correspondant à l'exploration et à la mise en forme de notions d'intelligence destinées à notre concept d'assistant personnalisé, nos éléments de réflexion se partagent en deux groupes complémentaires. Nous distinguons, d'un côté, le domaine de l'extraction et de la reconnaissance des connaissances (moyens de communication accédés, choix d'un nouveau moyen adéquat, situations de filtrage, ...) et, de l'autre, l'apprentissage automatique des données découvertes (moyens de communication disponibles, "petites habitudes" des utilisateurs, ...).

2.1.2. Premières propositions adaptées d'agents intelligents

2.1.2.1. Corrélations avec les règles de fonctionnement établies

En poursuivant nos travaux de spécification des procédés d'apprentissage automatique imputables à notre concept d'assistant intelligent, nous pouvons nous atteler à l'élaboration et à la mise en avant de plusieurs fonctionnalités, ou agents d'interface, chargées d'effectuer les tâches de notre "Assistant Conversationnel". En nous appuyant sur les règles détaillées jusqu'ici, voici donc quelques exemples de ces nouveaux agents intelligents :

- Agent de reconnaissance de l'utilisateur par l'Assistant Conversationnel (mot de passe, données biométriques, ...) et ouverture de son bureau personnalisé associé.
- Agent de détermination de tous les moyens existants (et disponibles si l'on tient compte de la [Règle n°5]) permettant de joindre la personne souhaitée, en temps réel ou à fréquence(s) paramétrable(s).
- Agent de choix du moyen le plus pertinent parmi les moyens non encore utilisés.
- Agent d'extinction d'un ou plusieurs des moyens d'information alloués à l'utilisateur, ce à la demande de celui-ci.

- Agent d'interruption automatique de la communication dans le cas d'une messagerie vocale ou d'un répondeur.
- Agent d'élaboration de la (ou des) proposition(s) de message(s) à déposer dans la messagerie électronique, sur le *fax*, dans la messagerie vocale ou encore sur le répondeur de l'interlocuteur...
- Agent d'information de l'utilisateur sur le déroulement des opérations, que ce soit en temps réel ou une fois l'interlocuteur joint (moyens existants, différents moyens appelés, connexions en cours, résultats de tous les appels, interruptions éventuelles, nouvelles connexions détectées, tri des données recueillies, rappel des dernières opérations, etc).

Nous recensons, à l'issue de la présentation de nos premiers agents logiciels pourvus d'intelligence, trois catégories bien distinctes quant aux différents mécanismes à percevoir et à appliquer au niveau de nos modèles auto-adaptables. Le **Tableau 2.1.2.1** à suivre nous en fournit la liste détaillée. Ainsi, il insiste sur les manières selon lesquelles nous envisageons d'exploiter et de traiter les multiples informations particulières relevant des préférences de tous les utilisateurs, premières habitudes de manipulation d'un terminal intelligent de communication entérinées par le fondement de notre Assistant Conversationnel.

Par ailleurs, nous pouvons aussi constater que l'ensemble de ces concepts intelligents continue de faire appel, à l'image de nos règles de fonctionnement précédemment développées, non seulement à la reconnaissance d'informations remarquables, mais également au tri et à la classification de telles données.

Tableau 2.1.2.1 : *Récapitulatif des premiers mécanismes intelligents issus des règles de fonctionnement du concept de l'Assistant Conversationnel*

Catégories d'application sur le terminal	Mécanismes définissant nos premiers agents intelligents
Personnalisation du bureau interactif	Reconnaissance des personnes appelées le plus fréquemment
	Reconnaissance du type des moyens utilisés le plus fréquemment
	Reconnaissance du contenu de messages envoyés le plus fréquemment
	Reconnaissance des situations les plus fréquemment configurées par l'utilisateur en cas de filtrage de moyens d'information (réunion, entretien, absence, travail)
Disponibilité des moyens de communication	Reconnaissance de tous les moyens utilisables pour joindre une personne par traitement antérieur ou en temps réel
	Reconnaissance du (ou des) moyen(s) le(s) plus adéquat(s) à une situation
Tri des opérations	Classification du déroulement des investigations effectuées par le terminal

2.1.2.2. Vers un nouveau concept d'assistant auto-adaptable

Nous pouvons maintenant étendre nos idées à un assistant entièrement dévoué à chacun de nos utilisateurs, concept plus général et plus complet aussi. Incluant les différents principes de notre Assistant Conversationnel, ce nouveau modèle se doit en effet d'élargir nos notions d'aide, de personnalisation des outils appelés, de substitution et d'allègement des actions courantes. Nous pouvons alors imaginer que ce concept va notamment gérer le processus de navigation de l'utilisateur sur le réseau *Internet* en lui proposant de le personnaliser et de privilégier ses centres d'intérêts préférés. En outre, les machines cibles au sein d'*ALCATEL* regroupent non seulement tous les *WebTouch* et terminaux de communication actuellement à l'étude ou en développement, mais aussi les applications de messagerie électronique telles que l'*ALCATEL Unified Messaging*.

L'apprentissage, à travers des obligations plus ou moins répétitives et les divers centres d'intérêts de l'utilisateur, tient une place indispensable dans ce concept, de même que le travail par systèmes multi-agents et le développement de systèmes associés dits "intelligents".

2.2. « L'Assistant Individuel de Télécommunication »

2.2.1. Introduction et caractéristiques générales

En détaillant les scénarios et ensembles de fonctionnalités envisagés pour la création, la spécification et l'implémentation d'un modèle d'Interface Intelligente destinée en premier lieu aux terminaux de communication d'*ALCATEL*, nous apportons dans cette partie une définition complète de notre nouveau concept d'assistant dédiée à l'utilisateur et intitulé plus précisément « Assistant Individuel de Télécommunication » (ou AIT). En introduisant ainsi dans une interface-utilisateur les notions d'intelligence, d'ergonomie et d'apprentissages automatiques à l'aide d'agents logiciels et de séquencements d'actions, nous abordons deux domaines majeurs : l'assistance personnalisée de l'utilisateur pour communiquer avec tout interlocuteur, et la classification automatique de toutes les informations manipulées.

Il apparaît alors comme acquis qu'une interface se doit d'être attirante et intéressante, à la fois pour les novices (apprentissage et patience, aides et soutiens multiples, ...) que pour les habitués (rapidité d'exécution, raccourcis et compléments de travail, ...). Nous nous devons donc d'apporter, au sein d'*ALCATEL*, des innovations supplémentaires dans les importants domaines du "grand public" et de l'ouverture des terminaux de communication vers le monde entier, à travers la téléphonie avancée ou encore l'avènement du réseau *Internet*.

Des agents intelligents d'interface ou d'apprentissage, incorporés dans un ou davantage de systèmes multi-agents, ainsi que l'élaboration de plusieurs modèles d'utilisateur prenant en compte un maximum des caractéristiques individuelles et "intelligentes" de chaque utilisateur, vont nous permettre de rendre notre outil informatique aussi automatique et avancé que possible. A tout ceci, nous appliquerons ensuite nos souhaits d'aide et d'assistance continue à l'utilisateur, de personnalisation et d'humanisation des différentes fonctionnalités disponibles, de substitution et d'allègement enfin des actions effectuées. Notre but principal se résume donc à la tentative de simplification de la vie de l'utilisateur devant la machine téléphonique et informatique.

En spécifiant alors la notion "d'Auto-Adaptation" telle que nous souhaitons l'insérer dans notre Interface Intelligente, nous travaillerons selon les trois caractéristiques essentielles de l'Intelligence Artificielle que sont la compréhension, l'apprentissage et la résolution de problèmes. Graphes d'actions à l'appui, nous aborderons ainsi en détail les différents scénarios et fonctionnalités envisagés pour le développement de nos modèles auto-adaptables.

2.2.2. La notion "d'Auto-Adaptation"

2.2.2.1. Définition précise du concept

L'amélioration d'une "Interface Homme-Machine" (IHM) passe aujourd'hui par l'intégration d'une intelligence la plus adaptative, voire auto-adaptative, possible [Benyon, 1993]. Il apparaît donc obligatoire d'établir notre propre définition d'une "Interface Auto-Adaptable", caractérisation qui se doit d'être complète et détaillée, mais aussi concise, pratique et orientée autant que possible vers les utilisateurs "grand public" des terminaux de communication actuels.

Le terme "Auto-Adaptation" se compose du substantif "Adaptation", c'est-à-dire de la capacité du système à faire face aux changements en relation avec son utilisateur [Mataric, 1994], et du préfixe "Auto" que nous pouvons traduire selon plusieurs sens distincts à venir. En effet, si nous pensons à "l'automatique", rappelons que ce terme qualifie tout dispositif exécutant de lui-même certaines opérations définies à l'avance. [Whitehead, 1998] présente "l'automatisation" qui correspond à l'exécution de fonctions sélectionnées indépendamment du contrôle immédiat de l'utilisateur, bien que sous l'entière conduite des requêtes précédentes issues de ce dernier. Cela évoqué, [Lieberman, 1997] note encore que "l'autonomie" correspond de son côté à un agissement simultané et sans intervention de l'utilisateur, que ce dernier soit occupé ou non.

Mais attention cependant à ne pas interpréter finalement ce même préfixe "auto" par la notion "d'autoritarisme" qui serait alors synonyme de restrictions et d'insistances [Akoulchina & Ganascia, 1997] de la part de la machine, empêchant tout contrôle et libre choix pour l'utilisateur dans la définition manuelle des paramètres du système [Schlimmer & Hermens, 1993].

A ce niveau, nous pouvons alors définir notre nouvelle "Interface Auto-Adaptable" en faisant appel aux différentes caractéristiques énoncées jusqu'à présent, ce en spécifiant la notion "d'Auto-Adaptation" telle que nous l'entendrons au cours de nos travaux de recherche à suivre.

Entité, faculté, qualité ou encore important concept, "l'Auto-Adaptation" se dévoile à nous comme étant pleinement basée sur l'apprentissage automatique, ainsi que fournissant la capacité d'agir en fonction des comportements variables et particuliers de chacun des utilisateurs appréhendés. Fortement autonome et surtout implicite aussi souvent que possible, notre Interface Intelligente se voit chargée de prendre en considération l'ensemble des divers intérêts, habitudes et préférences des utilisateurs, tous étudiés en temps réel à partir de leurs manipulations. Conduisant de plus à une construction permanente et évolutive d'un profil de l'utilisateur, aussi nommé modèle individuel, ce principe d'interface auto-adaptable est destiné à une prédiction ou même à une anticipation active de ce que l'utilisateur va commander et vouloir réaliser. Il va se traduire par l'établissement et la présentation à l'utilisateur d'une ou plusieurs suggestions de séquences d'actions. Aboutissant à des initiatives de manipulations artificielles mais tout-à-fait appropriées et toujours souhaitées par l'utilisateur courant, ces interventions de notre Interface Intelligente se doivent ainsi d'être spécifiques, personnalisées et aptes à moduler leurs actions auto-adaptables en fonction des attitudes, comportements et besoins successifs de l'utilisateur.

En conséquence, nous parvenons à saisir l'intérêt que représente une telle implantation auto-adaptable dans les terminaux de communication "grand public" d'ALCATEL, puisqu'il s'agit de "concerner" l'utilisateur de manière pertinente en cherchant à faire venir la machine vers l'être humain [Blandet & Bohrer, 1998]. Il est toujours intéressant, voire de nos jours indispensable, que l'utilisateur sente la présence d'une aide implicite, d'un apprentissage automatique (et constant) provenant de la machine avec laquelle il tente de travailler couramment.

Cependant, nous devons aussi éviter l'autre extrême, à savoir l'embrigadement total de notre utilisateur. Nous nous obligeons à analyser et à spécifier une interface intelligente définie "auto-adaptable", et non "auto-adaptative" ou "auto-adaptée". Il nous semble en effet primordial que notre interfaçage intelligent "propose" son soutien automatique et logistique à l'utilisateur, lui "permette" d'être assisté en toute discrétion et toute humilité, et non oblige au contraire celui-ci à constamment subir l'outil informatique et travailler sans détour possible. En d'autres termes, il nous apparaît tout-à-fait concevable d'adapter les différentes fonctionnalités de notre interface intelligente à l'utilisateur, de manière automatique certes, mais sans pour autant interdire à ce dernier de s'intéresser au terminal de communication en dehors de notre assistance intelligente, de manière "manuelle" ou "normale" donc, et ce dès qu'il en ressent le besoin ou l'assurance.

Ainsi, par notre interface "auto-adaptable", nous remplissons les critères des concepts de non-restriction et de non-insistance vis-à-vis de nos utilisateurs [Akoulchina & Ganascia, 1997], à savoir que ceux-ci peuvent toujours ignorer la suggestion qui leur est proposée par le système intelligent de communication, et préférer exécuter une action différente, voire contradictoire.

2.2.2.2. Proposition d'un schéma architectural auto-adaptable

Dans le but essentiel de rester entièrement compatibles avec un maximum de terminaux de communication d'ALCATEL, nous nous sommes attachés à tendre vers un complet système en couches architecturales successives, portables et indépendantes les unes des autres. En respectant ainsi les interfaces, applications et modules logiciels déjà existants, très peu de modifications au niveau des multiples codes de programmation ont été à envisager au sein des différents cas de prototypes. Cela est issu de notre choix de travailler avec une architecture logicielle en couches. Insérée entre l'interface déjà implémentée et les applications de haut niveau correspondant aux diverses fonctionnalités présentes dans les terminaux, notre couche d'Intelligence Artificielle, baptisée "AI Layer" selon la traduction en langue anglaise de l'intitulé, apporte à son échelle toutes ses informations et ses compétences à chacun de nos terminaux, sans jamais en altérer les données initiales respectives (**Figure 2.2.2.2** à venir sur la page suivante).

Ceci avancé, chaque problème de cette Interface Intelligente est ensuite résolu de façon automatique et autonome, que ce soit au niveau du Modèle d'Utilisateur que nous découvrirons dans le prochain chapitre, ou en ce qui concerne la Base de Connaissances Factuelles (ensemble des informations recueillies, analysées et traitées) que nous développerons au **Chapitre 4**. De plus, les problèmes principaux que doit résoudre notre "Couche IA" naissante apparaissent au nombre de trois. La compréhension initiale d'une situation, d'un fait ou d'une commande quelconque se voit en effet suivie par un apprentissage automatique des actions, des décisions et des réactions courantes, répétées ou pertinentes émanant de l'utilisateur. Finalement, le dialogue interactif est souhaité à terme aussi implicite que possible, de même que la résolution concernant les différents "problèmes" constatés lors de la manipulation de l'interface d'un tel terminal de communication.

C'est pourquoi, nous allons décrire, selon ces trois points de vue, chacune de nos idées, à savoir chacun des scénarios correspondant aux différentes fonctionnalités envisagées, ce dans le but d'insuffler une certaine "intelligence" au comportement de l'interface des terminaux actuels.

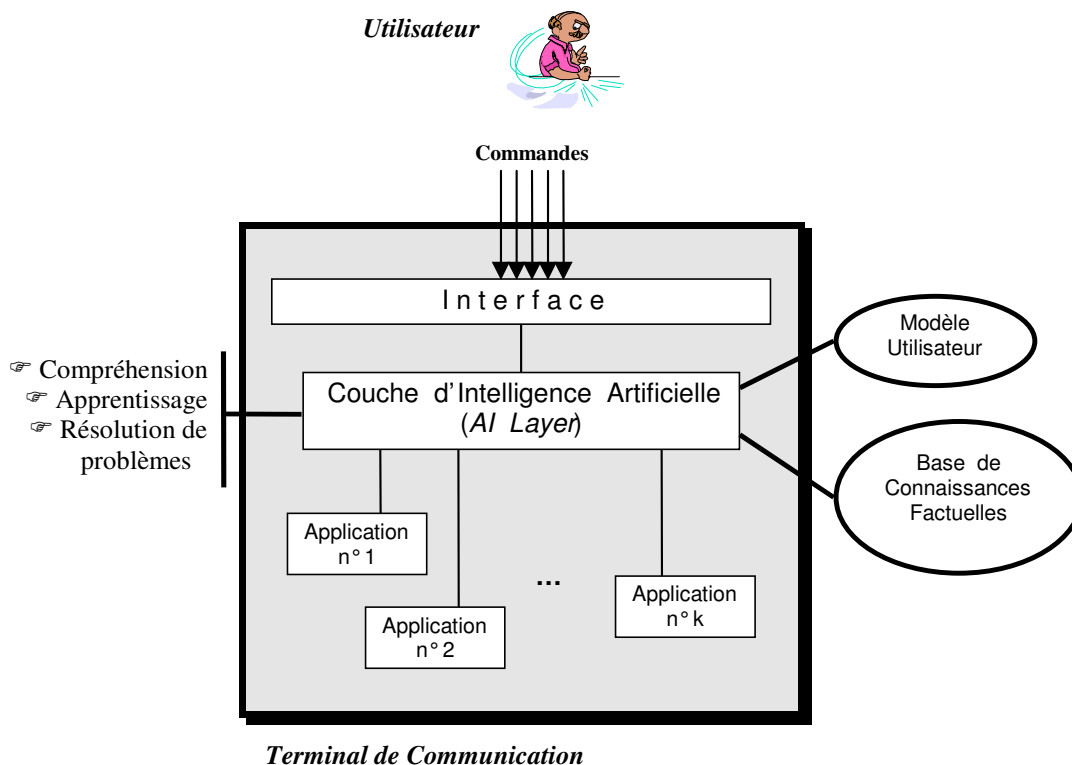


Figure 2.2.2.2 : Schéma architectural de l'Interface Auto-Adaptable

2.3. Définitions structurées des divers scénarios envisagés

2.3.1. « L'Assistant Conversationnel Auto-Adaptable »

Ce nouveau concept "d'Assistant Conversationnel Auto-Adaptable", dont nous avons spécifié le principe de base en 2.1.1.1, peut permettre d'améliorer l'efficacité et la convivialité de chacune des quatre fonctions disponibles sur une majorité des terminaux de communication d'ALCATEL. Il s'agit là plus précisément de la téléphonie avancée, de la gestion du courrier électronique, de l'accès aux services *Internet* et, enfin, de l'annuaire intégré.

Voici donc à présent les descriptions et études complètes des différents scénarios que nous envisageons de concrétiser, en insistant systématiquement sur les avantages procurés à l'utilisateur par chacune des fonctionnalités proposées. Nous débutons tout-de-suite la page suivante par le descriptif d'une première fonctionnalité traitant d'une arrivée à destination obligatoire pour chaque information issue des terminaux de communication (**Scénario 2.3.1.1**).

Scénario 2.3.1.1 : *Fonctionnalité générale traitant de l'aboutissement obligatoire d'une information*

Description :

Lorsque l'utilisateur indique au terminal qu'il souhaite joindre une personne, sa demande doit impérativement aboutir, à savoir que cette tierce personne doit en définitive obligatoirement disposer de l'information que voulait lui faire parvenir l'utilisateur, et ce dans un délai aussi court que possible (recherche d'une efficacité maximale à proposer à l'utilisateur du terminal).

Compréhension : souhait d'envoyer de l'information à une personne.

Résolution : faire obligatoirement parvenir l'information à l'interlocuteur.

Mais nous notons alors d'emblée plusieurs cas particuliers dans ce premier scénario, à savoir les trois différents exposés à présent (**Scénarios 2.3.1.2 à 2.3.1.4**).

Scénario 2.3.1.2 : *Première fonctionnalité particulière traitant de l'aboutissement obligatoire d'une information par une utilisation auto-adaptable des moyens de communication*

Description :

Si l'outil de messagerie de l'interlocuteur (que le terminal a choisi de joindre) n'est pas reconnu, n'est pas accessible, ne répond pas ou encore renvoie un message d'erreur, le terminal doit pouvoir utiliser, en agissant seul, d'autres moyens mis à sa disposition (numéros connus de téléphones fixes ou mobiles attribués à la personne demandée, *e-mail*, *fax*, visio-conférence, etc). Le terminal doit par conséquent pouvoir connaître à tout moment, et donc apprendre en temps réel ou avoir déjà appris au préalable, les différents moyens qui existent (ou qui sont disponibles) pour pouvoir joindre l'interlocuteur : il n'y a en effet rien de pire pour l'utilisateur que de devoir répéter à l'interface de la machine ce qu'il souhaite réaliser.

Compréhension : l'outil appelé n'est pas disponible.

Apprentissage : connaître à l'avance (ou découvrir en temps réel) et permettre l'évolution du contenu de l'ensemble des moyens existants capables de joindre l'interlocuteur.

Résolution : utilisation automatique des moyens appris.

Scénario 2.3.1.3 : *Seconde fonctionnalité particulière traitant de l'aboutissement obligatoire d'une information par une interruption automatique face à un appareil inadéquat*

Description :

Dans le cas où le terminal entre en contact avec un appareil de messagerie vocale (ou un répondeur), il interrompt de lui-même la communication et choisit un nouveau moyen de joindre l'interlocuteur souhaité. Le terminal doit par conséquent savoir opter pour l'outil qui semble le plus pertinent et le plus adéquat parmi ceux restants à sa disposition, et ce déjà en matière de corrélation entre l'information et son support (une image ou une animation par exemple ne peut être fournie vocalement par téléphone ou répondeur, mais par contre plus ou moins bien par *fax*, voire nettement mieux par *e-mail* ou vidéo-conférence...).

Compréhension : entrée en contact avec un appareil de messagerie vocale.

Apprentissage : connaître en permanence l'ensemble des différentes adéquations informations-supports.

Résolution : interruption automatique de la communication, choix automatique du(des) moyen(s) le(s) plus adéquat(s).

Scénario 2.3.1.4 : *Troisième fonctionnalité particulière traitant de l'aboutissement obligatoire d'une information par un choix auto-adaptable de l'outil courant le plus adéquat*

Description :

S'il ne reste en définitive plus que des moyens d'enregistrements vocaux pour fournir l'information courante à la tierce personne, le terminal se doit de proposer à l'utilisateur de laisser un message contenant l'information qu'il souhaite faire parvenir à son interlocuteur, soit sur la messagerie vocale, soit sur le répondeur, mais dans tous les cas sur l'outil le plus adéquat parmi l'ensemble des outils d'enregistrements vocaux existants.

Compréhension : seule subsiste la possibilité de fournir l'information souhaitée par les moyens d'enregistrements vocaux.

Apprentissage : connaître l'ensemble des outils d'enregistrements vocaux disponibles.

Résolution : proposer à l'utilisateur de déposer un message sur l'outil vocal le plus pertinent parmi ceux existants.

Tout ceci étant déjà clairement établi, abordons à présent une seconde mais toute aussi importante fonctionnalité à propos des différentes caractéristiques détaillant notre nouvel "Assistant Conversationnel Auto-Adaptable" (**Scénario 2.3.1.5**).

Scénario 2.3.1.5 : *Fonctionnalité générale traitant de la mise en forme des habitudes de l'utilisateur*

Description :

Lorsque l'utilisateur souhaite non seulement écrire un *e-mail* ou un *fax*, mais aussi laisser un message sur la messagerie vocale ou le répondeur d'un interlocuteur, le terminal gagne à connaître déjà les "petites habitudes" de son utilisateur, et ce en matière de mise en forme(s), d'entête et de signature, de références, de style, d'expressions et de mots-clés, etc. La machine se doit aussi de tenir compte bien évidemment des différents interlocuteurs à joindre (l'utilisateur ne s'adressera par exemple pas de la même manière à son(sa) conjoint(e) qu'à ses enfants, ses parents, ses amis, ses collègues de travail ou encore des vendeurs).

L'utilisateur peut alors en effet, grâce à toutes ces informations déjà établies, aux différentes abréviations et autres raccourcis, voire même à une déduction automatique en temps réel de mots d'après leurs premières lettres tapées, transmettre le message (en partie déjà élaboré, construit et vérifié automatiquement) à son interlocuteur sans prendre trop de son temps et sans omettre d'y inclure des informations importantes (date et lieu d'une réunion, signature complète, etc).

Compréhension : souhait d'écrire un *e-mail*, d'envoyer un *fax*, de laisser un message dans la boîte vocale ou sur le répondeur d'un interlocuteur.

Apprentissage : connaître et faire évoluer l'ensemble de toutes les habitudes de l'utilisateur en matière de présentation de ses messages.

Résolution : utilisation rapide et efficace des informations apprises, des abréviations et autres raccourcis, déduction automatique et en temps réel de mots-clés d'après leurs premières lettres tapées, adjonction d'informations importantes manquantes.

Cependant, nous pouvons, à ce niveau également, pousser encore nos investigations plus loin en caractérisant les trois nouveaux procédés supplémentaires définis sur les pages à suivre maintenant (**Scénarios 2.3.1.6 à 2.3.1.8**).

Scénario 2.3.1.6 : *Première fonctionnalité particulière traitant de la mise en forme des habitudes d'utilisateur au niveau de la lecture des messages arrivés*

Description :

A l'ouverture des fenêtres de réception des différents messages (boîte aux lettres des *e-mails*, récapitulatif des *fax*, liste des messages sur le répondeur, etc), notre système d'assistant automatique à la communication se doit de proposer une prise de connaissance des nouveaux messages arrivés selon une certaine hiérarchie représentant les préférences et habitudes de lecture de l'utilisateur. En effet, dans une liste de documents à consulter, l'utilisateur se dirige toujours d'abord vers ceux présentant pour lui le plus d'intérêts : une hiérarchie des messages peut donc être établie par un apprentissage évolutif, non seulement à partir d'une observation continue des manipulations antérieures de l'utilisateur, mais également grâce à de constantes confrontations et réévaluations comparatives entre les nombreuses informations détectées.

Nous pouvons alors par exemple concevoir que chaque nouveau message se voit automatiquement attribué, dès son arrivée dans une boîte de messagerie, une valeur de priorité résultant de la fréquence habituelle de lecture par l'utilisateur d'un tel message (analysé et reconnu grâce, notamment, aux mots-clés, connus ou non, le composant). A noter aussi que, de cette manière, nous résolvons aisément les nombreuses interactions constamment évolutives qui existent entre les différents interlocuteurs, sujets abordés ou encore dates d'émission (respect de la chronologie des messages) caractérisant l'ensemble des documents perçus par l'utilisateur.

Compréhension : souhait de prise de connaissance des nouveaux messages, de manière globale (émanant de tous les moyens de communication disponibles) ou ciblée (obtenus d'un ou plusieurs moyens précis).

Apprentissage : connaître en permanence la hiérarchie complète et évolutive de consultation des messages, c'est-à-dire des nombreuses informations détectées dans les différents messages consultés par l'utilisateur.

Résolution : lecture des données fournies par le système chargé de la hiérarchisation des informations remarquables, analyse du contenu du nouveau message, calcul et attribution d'une valeur de priorité au message arrivant, établissement d'une liste de nouveaux documents arrivés en fonction de la hiérarchie courante des informations-clés.

Scénario 2.3.1.7 : *Seconde fonctionnalité particulière traitant de la mise en forme des habitudes au niveau de l'élaboration d'une réponse auto-adaptable*

Description :

Mise en place d'un système d'aide à la décision concernant l'élaboration éventuelle d'une réponse auto-adaptable : développement d'un procédé d'apprentissage chargé de déterminer, au fur et à mesure de la manipulation de l'outil, quels sont les différents messages parmi ceux reçus pour lesquels il devient astucieux de préparer automatiquement une réponse individualisée, c'est-à-dire si tel ou tel message demande ou "mérite" une réponse de la part de l'utilisateur.

Il nous vient alors inévitablement à l'esprit, afin de déterminer précisément et efficacement ce "mérite", certains des points essentiels déjà énoncés plus haut, comme le moyen de communication utilisé, l'identité de l'interlocuteur, l'importance attribuée au message reçu. En outre, il ne faut pas non plus omettre la possibilité de faire appel à un "historique" automatique des actions de l'utilisateur, scénario que nous définissons un peu plus tard dans ce mémoire et qui permet entre-autres d'analyser avantageusement le déroulement chronologique des travaux réalisés par l'utilisateur, et par conséquent ses réactions en fonction des messages entrants.

Compréhension : réception et lecture d'un nouveau message.

Apprentissage : connaître de manière évolutive l'ensemble des caractéristiques des différents messages auxquels l'utilisateur a tendance à répondre.

Résolution : déterminer, en récupérant les informations du système d'apprentissage, si un message entraîne une réponse automatique (immédiate ou non).

Scénario 2.3.1.8 : *Troisième fonctionnalité particulière traitant de la mise en forme des habitudes au niveau de la détermination d'un moyen préférentiel*

Description :

Le système peut alors encore ajouter un nouvel aspect décisionnel à son apprentissage des habitudes de l'utilisateur en déterminant également par quel(s) moyen(s) cette précédente réponse auto-adaptable est préférable d'être envoyée, en s'appuyant sur une hiérarchie pré-établie et constamment évolutive des outils de communication les plus souvent manipulés par l'utilisateur (mais aussi certainement sur l'adéquation entre informations et supports correspondants, relation définie dans le second cas particulier du scénario général précédent).

Compréhension : décision d'une réponse auto-adaptable à envoyer.

Apprentissage : connaître l'ensemble des habitudes et des préférences de l'individu en matière d'utilisation des outils télématiques mis à sa disposition.

Résolution : choix automatique du moyen préféré, parmi les plus adéquats en ce qui concerne la correspondance entre information et supports.

Nous poursuivons ainsi par toutes les explications relatives aux diverses fonctionnalités que nous envisageons pour l'implémentation de nos systèmes intelligents dédiés à l'utilisateur (*Scénarios 2.3.1.9 à 2.3.1.11*).

Scénario 2.3.1.9 : *Fonctionnalité traitant du filtrage des moyens d'information*

Description :

Le terminal peut proposer à l'utilisateur le filtrage de ses propres moyens d'information, c'est-à-dire d'éviter de fonctionner avec certains de ses moyens, pourtant bel et bien existants et utilisables. Par extension, une personne susceptible d'être jointe, mais ne désirant pas l'être au travers de certains moyens (par le téléphone de bureau si elle est en réunion, par l'*e-mail* si sa réponse doit être directe ou "immédiate", par le *fax* si elle est absente, par les téléphones encore s'il y a du bruit environnant, etc) peut ainsi demander au terminal de se charger de filtrer ses façons d'être appelée. Nous nous devons en effet de ne jamais négliger la situation, le contexte, les multiples circonstances qui catalisent, parfois même simultanément, une communication.

Nous pouvons par exemple imaginer un tableau de commandes facile à gérer puisque présentant une liste d'expressions telles que "réunion", "conversation directe", "absence", "bruits ambiants", etc. La liste informe alors l'utilisateur sur le filtrage en cours et sur ceux disponibles à ce moment-là, leurs durées étant également automatiquement adaptables aux besoins courants de l'utilisateur en matière de catégories distinctes d'outils de communication. Sans oublier non plus la possibilité de basculer de situation en situation, rapidement par quelques "cochages" et "décochages" (manière explicite), mais aussi par observation continue des différentes actions effectuées par l'utilisateur afin de pouvoir régler ensuite un filtrage automatique (et donc implicite) des différents moyens disponibles en fonction des besoins réels de l'utilisateur.

Compréhension : intention explicite de changer le contexte de filtrage des divers moyens d'informations par "cochage" ou "décochage" de la part de l'utilisateur dans le tableau de commandes.

Apprentissage : établir et maintenir en permanence un tableau évolutif des commandes, c'est-à-dire l'ensemble des configurations de filtrage explicitement souhaitées par l'utilisateur, mais aussi implicitement observées à partir du travail effectué par celui-ci.

Résolution : basculement explicite d'une situation de filtrage à une autre, basculement implicite du contexte de filtrage après observation des différentes actions réalisées par l'utilisateur et déduction automatique de la nécessité ou de l'avantage d'un tel changement.

Scénario 2.3.1.10 : *Fonctionnalité traitant de l'information auto-adaptable de l'utilisateur*

Description :

Le terminal doit pouvoir ensuite fournir automatiquement une information adaptable et exhaustive à l'utilisateur sur le déroulement des différentes opérations de communication, que ce soit en temps réel (information continue, explication de ce que le système fait) ou une fois l'interlocuteur joint (information de récapitulation de la communication, ce que le système a fait), et ce sur les outils existants et disponibles, le contexte courant de filtrage des informations, les différents moyens appelés, les tentatives de connexion en cours, les résultats de tous les appels, les interruptions éventuelles, les nouvelles connexions et appels détectés, le récapitulatif et tri des données recueillies, le rappel des dernières opérations souhaitées et effectuées, etc.

En outre, afin d'améliorer encore l'automatisation d'un tel apport d'informations, que nous nommons par ailleurs "historique" ou chronologie des différentes actions de l'utilisateur, nous pensons définir également (et éventuellement proposer à l'utilisateur de préciser de manière interactive sa valeur souhaitée) un seuil d'information destiné à permettre une information totalement implicite de l'utilisateur, après un certain temps sans demande explicite de sa part ou après un nombre relativement important d'actions réalisées ou d'interlocuteurs appelés.

Compréhension : souhait ou besoin (établi par le seuil d'information) de connaissances exhaustives (continues ou récapitulatives) pour l'utilisateur.

Apprentissage : détecter, appréhender et conserver en permanence l'ensemble des actions, explicites et/ou implicites, effectuées par l'utilisateur et/ou globalement par l'interface intelligente.

Résolution : définition du seuil d'information et/ou proposition de détermination interactive de sa valeur.
affichage de l'ensemble des différentes informations demandées à la demande ou à la suite d'un dépassement du seuil d'information défini.

Scénario 2.3.1.11 : *Fonctionnalité traitant des statistiques issues du travail de l'utilisateur*

Description :

Nous proposons enfin la gestion d'un service, également automatisé et par ailleurs très intéressant pour spécifier le modèle d'utilisateur que nous envisageons, présentant un recueil d'informations apprises en continu au cours de la manipulation des différents utilitaires de nos systèmes. Nous définissons alors ces informations courantes comme des statistiques de travail, à savoir par exemple les personnes appelées ou en communication le plus fréquemment, les types de moyens utilisés le plus fréquemment, les contenus de messages envoyés le plus fréquemment, les situations les plus fréquemment configurées par l'utilisateur au niveau du filtrage de ses moyens d'information, etc.

Il reste à noter que, par extension, nous associons également à ce service l'ensemble des statistiques retenues au fur et à mesure de l'utilisation des outils de navigation sur le réseau *Internet* proposés par le terminal (nombre total d'accès au réseau, dates et périodes horaires correspondantes, durées moyenne et totale de navigation sur le réseau, durée moyenne de visite d'une page, nombre de pages respectives visitées par centre d'intérêts, etc).

Compréhension : souhait, accord de la part de l'utilisateur d'établir des statistiques complètes sur l'ensemble de son travail courant.
arrivée d'une nouvelle information à assimiler par l'interface.

Apprentissage : appréhender et traiter en permanence l'ensemble des statistiques recueillies et représentant la globalité du travail de l'utilisateur.

Résolution : affichage de l'ensemble des différentes statistiques demandées et établies, en temps réel ou encore à la demande de l'utilisateur.

2.3.2. La « Classification Auto-Adaptable » des informations

La vision que nous mettons en avant au sujet de notre concept de l'Assistant Individuel de Télécommunication ne se limite pas seulement à un nouvel Assistant Conversationnel orienté vers l'Auto-Adaptation.

En effet, en plus de l'apprentissage continu et automatique, de la caractérisation individuelle de l'utilisateur courant ainsi que de l'assistance logistique et évolutive de ce dernier, nous pensons qu'il est de nos jours inévitable d'essayer encore de fortement lier, à toute volonté de communiquer, l'information correspondante que l'on souhaite fournir à son interlocuteur. Nous remarquons alors tout de suite que nos différents scénarios définis précédemment nous ont même déjà emmenés plus loin que cela. Ils nous proposent effectivement, grâce notamment au concept des adéquations entre les informations et leurs éventuels supports respectifs, de gérer l'action de communication directement en fonction de son contenu, c'est-à-dire de l'ensemble des informations à communiquer à une tierce personne.

C'est pourquoi, nous allons nous attacher à décrire à présent le deuxième volet de notre outil informatique, partie qui correspond à la gestion "intelligente" de toutes les informations recueillies et manipulées par l'utilisateur au travers de l'interface auto-adaptable des terminaux de communication d'ALCATEL.

En insistant cependant encore lourdement sur le fait que nous ne personnalisons ni ne travaillons en aucune manière directement sur le *Web*, mais plutôt au niveau de sa "couche de niveau supérieur" qui traite de la navigation et de la recherche des informations, nous proposons d'adjoindre aux déjà nombreuses fonctionnalités les deux récents et importants scénarios à venir (**Scénarios 2.3.2.1 à 2.3.2.2**).

Scénario 2.3.2.1 : *Fonctionnalité globale traitant d'une mise à jour auto-adaptable des informations recueillies*

Description :

Une mise à jour, bien évidemment auto-adaptable et opérant en temps réel, apparaît déjà à ce niveau tout-à-fait primordiale pour la légitimité et la pertinence à tout instant de l'ensemble des informations régulièrement recueillies par les composants de notre nouvelle Interface Intelligente. En spécifiant puis en implémentant une complète concordance de travail et de temporisation entre les différents outils gérés par le terminal, cela se doit par exemple d'entraîner la réactualisation automatique de l'agenda de l'utilisateur en fonction des messages qu'il a reçus mais aussi envoyés, la prise en compte des "cahiers" de réservations des différentes salles de réunion dans le cas de l'élaboration d'un message destiné à organiser une conférence, un séminaire ou encore un entretien...

Compréhension : arrivée d'une nouvelle information à assimiler par l'interface.

Apprentissage : connaître en permanence l'ensemble des applications proposées par le terminal de communication utilisé, ainsi que toutes leurs caractéristiques respectives de gestion des informations.

Résolution : mise à jour évolutive des différentes fonctionnalités disponibles.

Scénario 2.3.2.2 : *Ultime fonctionnalité globale traitant de tris auto-adaptables des diverses informations recueillies*

Description :

Tri en temps réel, classification automatique des toutes les informations découvertes et recueillies par l'utilisateur, non seulement à travers la multitude de messages et données obtenus de ses diverses communications gérées par l'Assistant Conversationnel Auto-Adaptable, mais également lors de ses investigations effectuées grâce à l'accès au réseau *Internet* du terminal. En second lieu, toutes ces données doivent ensuite évidemment pouvoir être retrouvées, rapidement, exhaustivement, que ce soit par catégories, par mots-clés, par moyens ou outils de communication, par titres, par interlocuteurs ou encore auteurs de messages ou de pages *Web*...

Nous nous devons cependant remarquer, et par suite de ne pas oublier, qu'il nous faudra alors certainement définir un concept bien précis de barrières, d'isolants ou encore de limitations de domaines d'intérêts concernant cette classification automatique, et ce pour éviter un mélange dénué de sens des différents sujets principaux à gérer.

A ne pas négliger non plus la nécessité d'une éventuelle fonctionnalité de réorganisation, implicite ou demandée (voire même amorcée) par l'utilisateur. Ceci concerne toutes les informations classifiées, afin d'établir une nouvelle classification courante, rafraîchie, plus "simple" et plus aisée de compréhension, et donc d'utilisation.

Nous pensons ainsi définir (et éventuellement autoriser l'utilisateur à préciser de manière interactive sa valeur appropriée) un seuil de réorganisation souhaité, cet utilitaire étant destiné à permettre une clarification, une amélioration totalement implicite de la classification globale des informations proposée à l'utilisateur. Ces avantages suivront par exemple le dépassement d'un nombre trop important de niveaux imbriqués et hiérarchiques composant cette classification auto-adaptable ou encore la détection d'une trop grande fréquence d'apparitions disparates d'un mot-clé, c'est-à-dire à de nombreux endroits différents au sein de l'arborescence de la classification.

Compréhension : recueil d'une nouvelle information intéressant l'utilisateur et devant donc être stockée par l'Interface Intelligente Auto-Adaptable, souhait de rechercher une information précise classifiée, souhait ou besoin (établi par le seuil de réorganisation) d'élaborer une nouvelle arborescence de classification auto-adaptable à l'utilisateur.

Apprentissage : classification, auto-adaptable et en temps réel, des nombreuses et parfois très différentes données recueillies par l'utilisateur, au travers de l'Assistant Conversationnel ou grâce au réseau *Internet*, réorganisation, amélioration, raffinement de toute l'architecture du procédé de classification automatique des données, connaissance permanente des différentes fréquences d'apparition de chaque mot-clé spécifiant les noeuds de l'arborescence de la classification établie.

Résolution : détermination de la disponibilité du moyen de stockage courant, définition d'un concept de limitations de domaines d'intérêts, recherche rapide et exhaustive d'une information précise, stockée et classifiée selon certaines caractéristiques, et ce en vue d'une réutilisation immédiate ou non, de même que temporaire ou non, définition du seuil de réorganisation souhaité et/ou proposition de détermination interactive de sa valeur, amorçage interactif d'une réorganisation de la classification courante par l'utilisateur, détection du nombre de niveaux hiérarchiques composant la classification automatique courante.

2.3.3. Enseignements et perspectives de travail

Deux concepts essentiels se dégagent des différentes caractérisations de fonctionnalités établies : le soutien logistique et l'apprentissage, tous deux automatiques, évolutifs et tendant à devenir aussi adaptables que possible envers l'utilisateur. L'interface "homme-machine", c'est-à-dire la communication bilatérale entre l'homme et l'appareil, s'en voit améliorée, notamment en ce qui concerne l'assistance automatisée, donc implicite ou "cachée" à l'utilisateur. Le second aspect de nos recherches, à savoir celui de l'interface "homme-information", s'avère tout aussi indispensable à l'établissement de notre Interface Intelligente Auto-Adaptable, si ce n'est que de par sa gestion automatique et individualisée. Sa capacité de classification évolutive des multiples informations recueillies par l'utilisateur, données inséparables du travail quotidien de ce dernier, est également à relever à ce niveau.

Les agents intelligents, que nous avons scrupuleusement définis en **1.2.4.5**, ainsi que les systèmes multi-agents associés, jouent un rôle primordial quant à l'implémentation de telles fonctionnalités nouvelles. Ils sont les garants d'une mise en œuvre efficace de l'interface pluri-disciplinaire souhaitée et de la répétition automatique de l'ensemble des opérations courantes et souvent fastidieuses dès que l'utilisateur en montre le besoin. Les séquencements et encore les graphes d'actions semblent par ailleurs être aussi adaptables à la gestion du flux des quantités d'informations exhaustives, apprises et fournies ensuite à l'utilisateur.

Subsiste l'élaboration d'un indispensable modèle d'utilisateur destiné à établir un profil évolutif du comportement d'un utilisateur quelconque, ce dans le but de prendre en compte ses caractéristiques individuelles et "intelligentes" face aux fonctionnalités mises à disposition par le terminal et face, également, à leurs propres évolutions respectives.

Nous entrevoyons alors deux problèmes majeurs concernant la personnalisation de toute interface : la distinction de diverses classes d'utilisateurs et l'observation des actions de chacun en vue d'une tentative d'automatisation. L'apprentissage et l'approfondissement de la notion de modèle d'utilisateur passe, non seulement par l'établissement du savoir, des connaissances, de ce que sait ou connaît la personne qui utilise notre interface, mais également par la gestion de l'évolution de ce savoir [Gavrilova & Voinov, 1997]. Nous pouvons même ajouter que, dans cette optique, l'acquisition mais aussi la déperdition de l'information sont à considérer, puisque moins une action est effectuée régulièrement, plus il faut aider, assister, la rappeler et éventuellement l'expliquer à l'utilisateur lorsque celui-ci souhaite à nouveau la réaliser.

A tout cela s'ajoute encore la conception d'un tout aussi primordial modèle du domaine chargé de mettre à disposition de notre système intelligent auto-adaptable le profil, également en constante évolution, de la situation, du contexte, des circonstances d'actions de l'utilisateur : réunion, entretien, indisposition temporaire, déjà en communication, absence, bruit environnant, et ainsi de suite...

2.3.4. Schématisation récapitulative des différents concepts définis

Dans le cadre général de description et de récapitulation plus schématisées que nous allons apporter maintenant, les principaux modèles pratiques demeurent les agents intelligents, comme nous l'avons développé auparavant. A l'aide de leurs nombreuses qualités, ils vont nous permettre de spécifier de manière claire et très rigoureuse l'ensemble des idées et scénarios que nous avons mis en évidence jusqu'ici.

En effet, si nous estimons déjà très intéressant pour nos présents travaux de recherche de pouvoir faire appel à leurs nombreux avantages conceptuels (mobilité, autonomie, interaction, coopération et adaptation [Carolan, Collins & al., 1997]), nous possédons également l'assurance de grandes pré-dispositions quant à une riche mise en œuvre. Crédibilité, réactivité, sociabilité, modélisation et communication [Carrez, 1998] doteront ainsi nos modules logiciels de qualités très appréciables, dans le but aussi d'habituer progressivement l'utilisateur à l'automatisation de son interface de communication de même qu'à la technologie somme toute relativement particulière des agents d'interface [Lashkari, Metral & Maes, 1994].

En conséquence, nous proposons ici une ébauche générique et hiérarchique à l'aide des cinq graphes orientés à suivre (**Figure 2.3.4.2**). Nous mettons avant tout en évidence les diverses actions exécutables par notre nouvel outil informatique, intelligent et auto-adaptable. Cependant, pour cela, il apparaît encore indispensable de prendre au préalable connaissance des différents formalismes et autres symboles manipulés au travers de ces graphes (**Figure 2.3.4.1** ci-dessous).

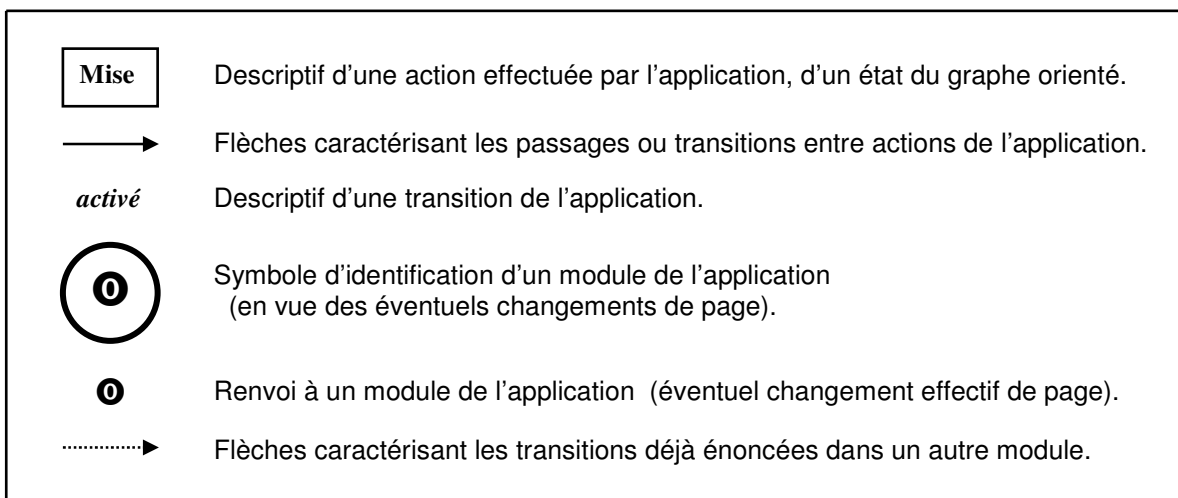


Figure 2.3.4.1 : *Synthèses graphiques de la description fonctionnelle de nos concepts auto-adaptables*

Ceci spécifié, nous abordons donc à présent, sur les cinq prochaines pages, l'ensemble conséquent des différents graphes orientés introduits ci-dessus. Rappelons également par ailleurs que ces schémas successifs sont mis en œuvre afin de décrire, de manière fonctionnelle mais aussi relationnelle et récapitulative, tous nos nouveaux concepts auto-adaptables. Nous sommes alors à même de pouvoir définir plus globalement, selon la manière informelle suivante, les diverses fonctionnalités de notre nouvelle application logicielle décrite :

- Page 78 : Lancement de l'application et initialisation de ses fonctionnalités.
- Page 79 : Fourniture d'une information à un interlocuteur.
- Page 80 : Consultation des nouveaux messages arrivés.
- Page 81 : Traitement en temps réel de toute action effectuée.
- Page 82 : Traitement en temps réel des nouvelles informations obtenues.

Si de tels schémas fonctionnels nous apportent effectivement la description complète et relationnelle de tous nos concepts auto-adaptables mis en avant, nous nous devons également d'en tirer d'utiles enseignements dans le but de nos implémentations. C'est ainsi que nous nous attarderons sur la spécification formelle et rigoureuse des divers systèmes intelligents obtenus.

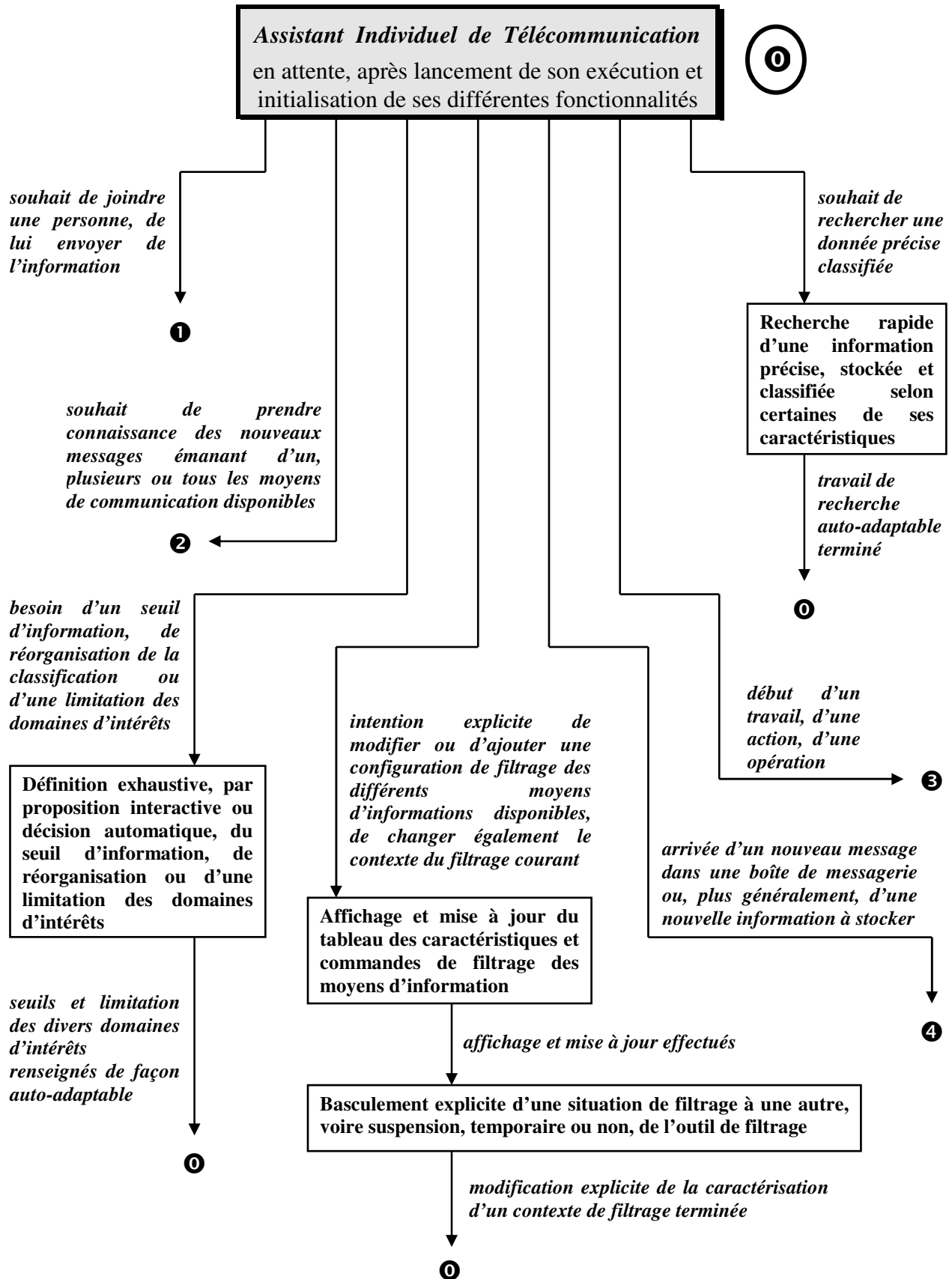


Figure 2.3.4.2 : Description fonctionnelle de nos concepts auto-adaptables

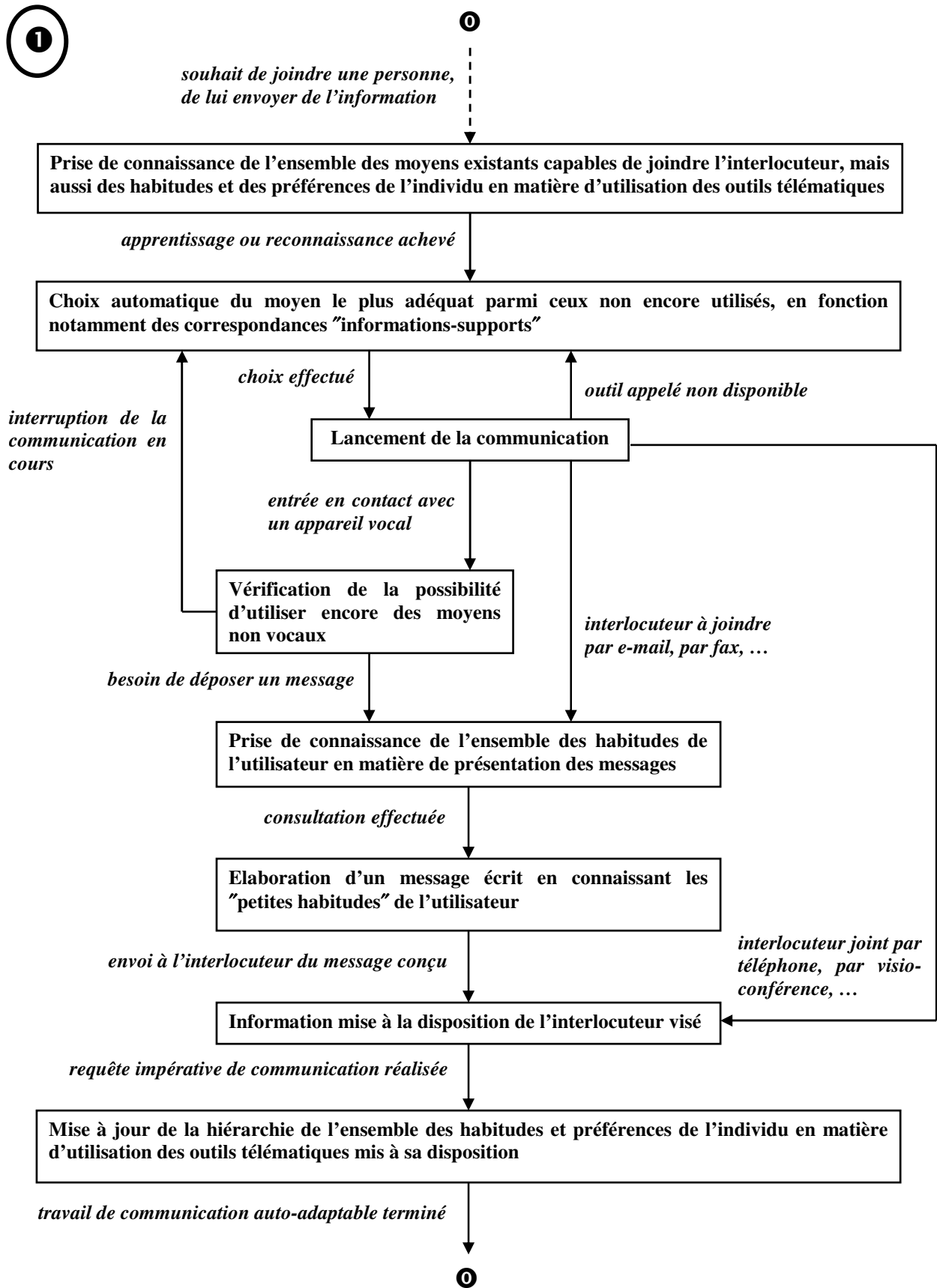


Figure 2.3.4.2 : Description fonctionnelle de nos concepts auto-adaptables

2

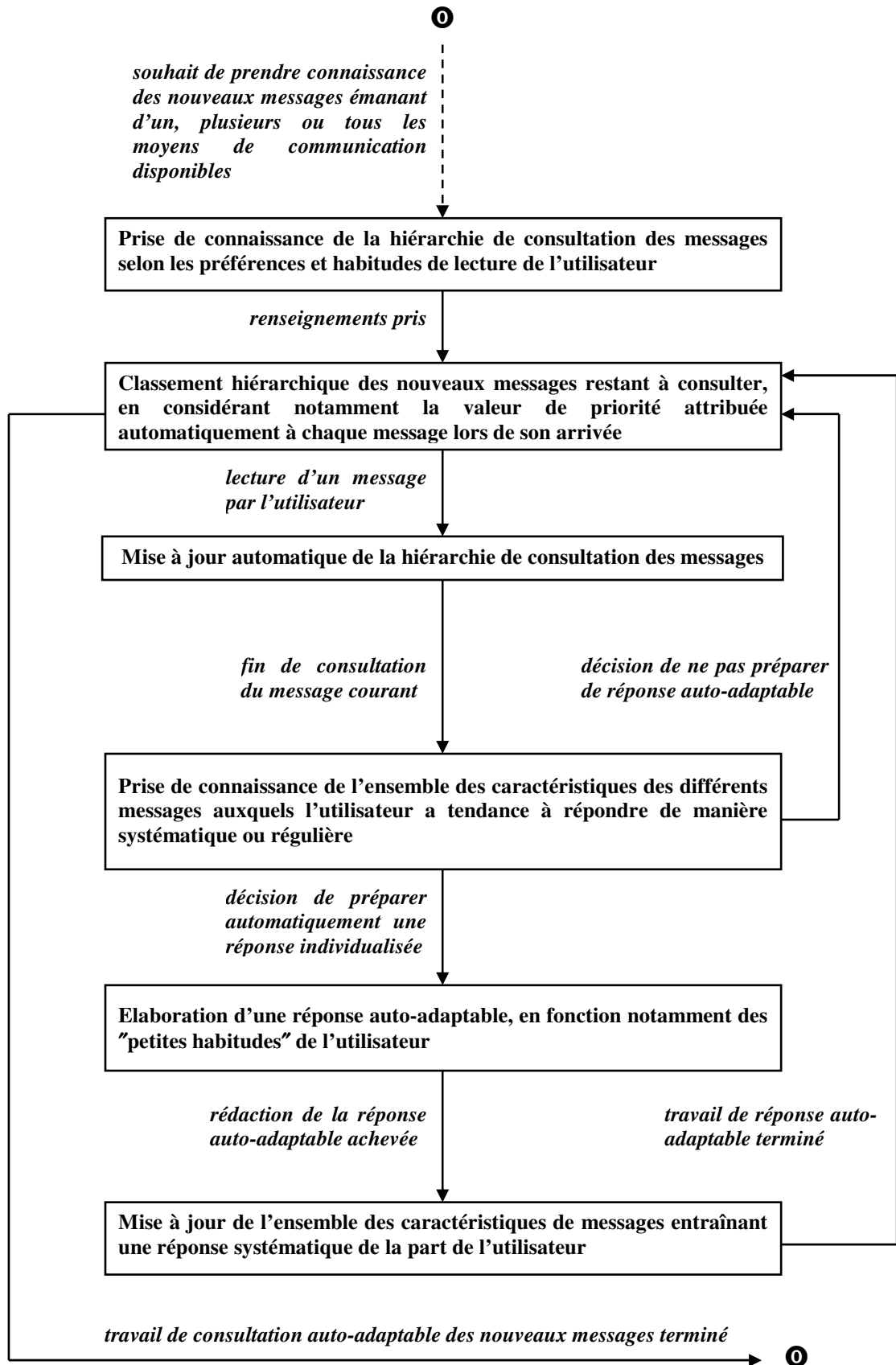


Figure 2.3.4.2 : Description fonctionnelle de nos concepts auto-adaptables

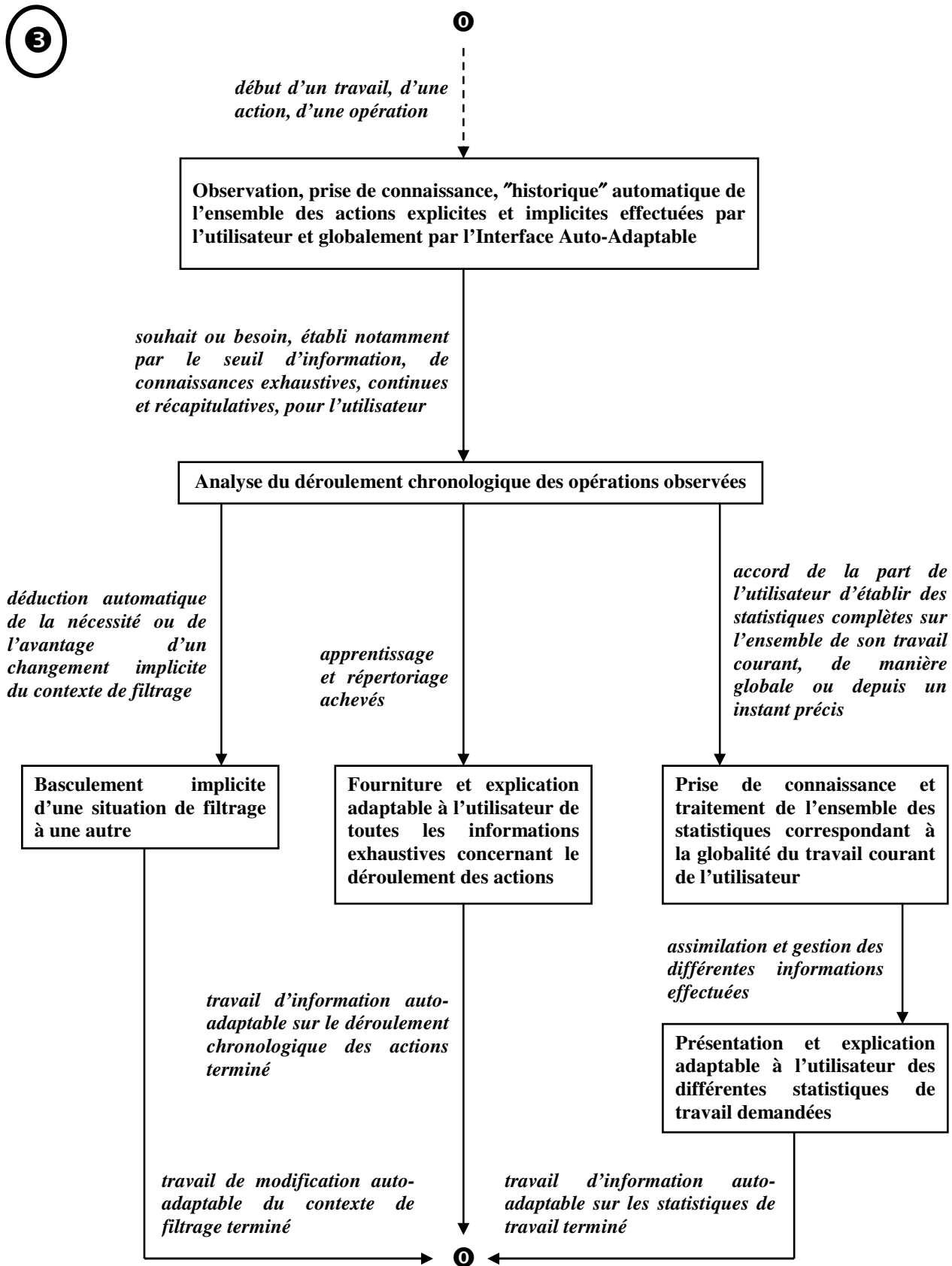


Figure 2.3.4.2 : Description fonctionnelle de nos concepts auto-adaptables

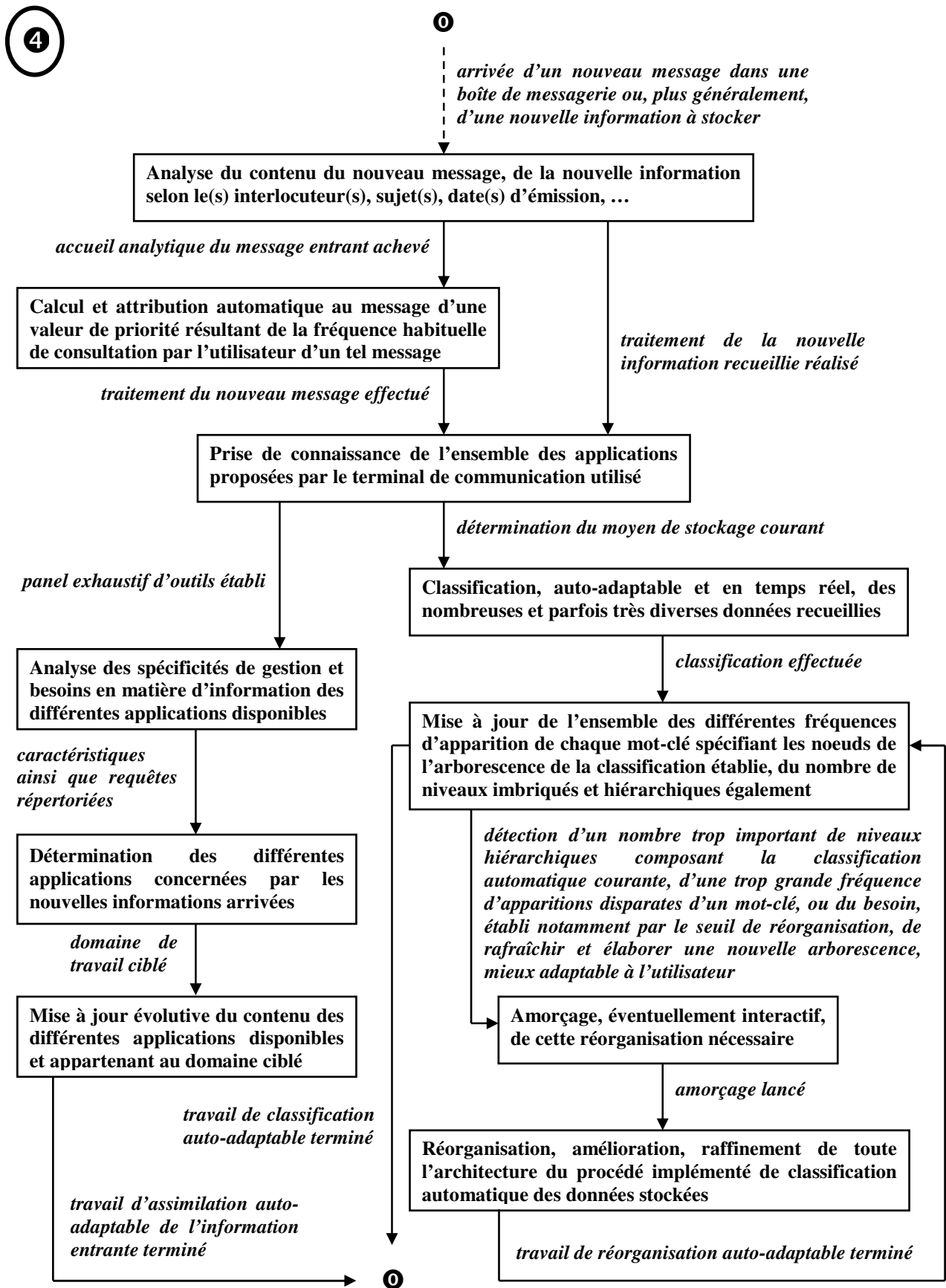


Figure 2.3.4.2 : Description fonctionnelle de nos concepts auto-adaptables

2.3.5. Vers une indispensable spécification formelle de nos modèles

En nous référant aux *Définitions 1.2.2.2 et 1.2.2.3* qui nous apportent une définition qualitative du domaine des interfaces intelligentes, nous pouvons tout d'abord répertorier ici, au sein-même de nos concepts auto-adaptables, les différentes propriétés relevant de l'intelligence telle que nous l'avons précisément déterminée au cours de nos travaux. Cette qualité de l'intelligence pouvant être associée de manière graduelle à chacun des systèmes d'interfaçage avancés, l'ensemble des nombreux critères de caractérisation que nous avons mis en place demeurent plus que jamais catalyseurs d'une conception et d'un développement efficaces de tous nos précédents scénarios. Par suite, nous constatons que seuls trois de nos vingt critères listés n'apparaissent pas encore à ce niveau de nos recherches. Plus particulièrement, il s'agit :

- de l'intégrité et de l'accessibilité universelle de l'information,
- de la surveillance et de la sécurité des données ensuite, notamment en ce qui concerne la protection des conséquences d'erreurs,
- de la tolérance caractérisant les diverses variations et imperfections de commandes émanant de l'utilisateur.

Ces ultimes critères appellent en conséquence certains travaux complémentaires. Dans ce but, notre Assistant Individuel de Télécommunication, bien que déjà composé de l'Assistant Conversationnel Auto-Adaptable et du dernier procédé de Classification Auto-Adaptable, attend beaucoup de l'étude et de l'adjonction de nos prochains modèles, que ce soit le Modèle Auto-Adaptable de l'Utilisateur ou ceux de nos mécanismes d'apprentissage automatique. Néanmoins, ces systèmes se trouveront fortement guidés par nos catégories d'intelligence élaborées jusqu'ici.

Dans un second temps, chaque scénario, chaque module, chaque utilitaire nouvellement décrit peut également être assimilé et entièrement défini suivant un concept de "prothèse". Ce dispositif de remplacement ou de consolidation, ce module additionnel ou encore cette "greffe d'intelligence", devient alors capable de suppléer, de se substituer à une capacité complète de l'utilisateur, ou même seulement à une partie d'une, et de réduire ainsi la charge cognitive de ce dernier. Si nous tentons au maximum de rendre auto-adaptables nos modules de "substitution intelligente" traitant des actions de l'utilisateur, il en résulte un travail plus clair et plus pertinent. Les manipulations deviennent en effet inévitablement plus poussées, grâce notamment à un approfondissement maximal dû à un suivi automatisé des diverses possibilités existant au niveau du système appréhendé. Cela conduit donc l'utilisateur, en définitive, à un travail nettement plus efficace que celui qu'il aurait effectué seul, c'est-à-dire sans assistance auto-adaptable.

En conclusion, et avant d'explorer les techniques de conception et d'implémentation applicables à nos différents modèles auto-adaptables, nous ne pouvons poursuivre nos recherches sans nous attacher maintenant à spécifier et à entièrement formaliser les nombreuses actions gérées par notre Assistant Individuel de Télécommunication. Par ailleurs, nous nous devons aussi, vu notamment le contexte bien spécifique de la Thèse, d'établir ces formalisations selon des principes et des modèles de spécifications issus du propre domaine des télécommunications. Ainsi, une importante partie de nos investigations va traduire à présent l'application des capacités du domaine de la spécification formelle, et plus particulièrement de "LOTOS" après comparaison des caractéristiques qualitatives de plusieurs méthodes envisagées, à la validation de nos multiples actions auto-adaptables : sont-elles bien toutes théoriquement "correctes" ainsi que réellement "implémentables" en pratique ?...

Nous vérifierons alors l'absence de conflits, d'effets concurrentiels entre les différents agents intelligents envisagés, ce qui reviendra à garantir une bonne collaboration entre tous nos présents agents, et donc une suffisance satisfaisante de leur ensemble pour gérer correctement les nombreux scénarios mis en place dans notre interface de communication "grand public".

2.4. Spécifications formelles des modèles

2.4.1. Introduction

2.4.1.1. Définitions de base

Pour bien débiter cette partie dédiée à la spécification de nos concepts auto-adaptables, nous nous devons tout d'abord d'introduire les différentes catégories qui caractérisent la notion d'action. Les actions s'articulent en effet autour de trois axes distincts, à savoir le vertical concernant la décomposition entre une action générique et ses actions élémentaires, l'horizontal ou "séquençement d'actions" faisant état des liens chronologiques entre actions (cf. par ailleurs 4.1.2 pour bien davantage d'explications à ce propos), et l'axe paradigmatique enfin pour les actions amenées à se substituer [Lippold & Pomian, 1995].

Le comportement d'un utilisateur face à un système étant toujours motivé par un but, le concepteur d'une interface homme-machine avancée se doit de commencer par analyser les tâches que les utilisateurs réalisent ou souhaitent exécuter, afin d'identifier ensuite la structure de base des buts avoués et les procédures nécessaires à la réalisation de ceux-ci [Whitehead, 1998]. Cependant, les procédures doivent être aussi naturelles, simples, abordables, compréhensibles et efficaces que possible, sans oublier d'analyser au préalable la vue que possède l'utilisateur sur les actions basiques et autres objets interactifs.

2.4.1.2. Justification de l'approche préalable orientée scénarios

Les scénarios correspondent très bien à l'ensemble des nécessités demandées par un tel développeur. En effet, ils se définissent par des étapes intermédiaires entre une analyse et un prototype, des interactions hiérarchiques typiques entre l'utilisateur et le système, ainsi que des descriptions concrètes et compréhensibles de la performance des tâches [Whitehead, 1998].

En outre, l'obtention d'une spécification à partir d'exigences informelles peut s'avérer une tâche fastidieuse et parsemée d'embûches si une approche méthodique ou rigoureuse n'est pas employée [Amyot, Logrippo & Buhr, 1997]. Les approches basées sur les scénarios permettent alors aussi de se concentrer sur les principaux aspects fonctionnels du système à spécifier. Mais, une bonne intégration dans un processus de développement itératif et incrémental, de même que la facilitation d'ajouts ou de modifications de fonctionnalités et de documentation en sont également des caractéristiques toujours appréciables et appréciées.

Enfin, n'oublions pas non plus que cette approche s'utilise largement dans l'industrie puisqu'elle permet notamment l'expression de séquences d'activités à partir d'événements déclencheurs [Amyot, Bordeleau & al., 1995]. Nous userons d'ailleurs souvent de ce dernier atout au cours des prochains chapitres de ce mémoire.

2.4.1.3. Justification du besoin inhérent de formalisation

Si une méthodologie efficace de conception logicielle se doit en premier lieu de présenter un langage expressif et flexible, elle se caractérise également toujours par une grande puissance de ses méthodes d'analyse et de validation [Amyot, Bordeleau & al., 1995]. Ces méthodes d'analyse formelle, basées sur des fondations théoriques solides, se chargent alors de vérifier que le système possède bien les propriétés désirées. Les méthodes formelles en ingénierie logicielle vont même aujourd'hui jusqu'à proposer de fournir un fondement purement mathématique aux différents traitements de conception, transformation et validation logicielles, bien qu'un certain degré de non-formalisme soit essentiel dans les étapes préalables du processus de développement [Amyot, 94].

Dans l'industrie en particulier, les concepteurs utilisent régulièrement une importante variété de notations pour saisir les besoins et les solutions candidates. Ces notations, bien que partiellement informelles, sont très utiles en tant qu'outils pensants. Cependant, on ne peut pas passer de l'informel au formel par des moyens formels ; on ne peut aller automatiquement de ces diagrammes et esquisses informels à une spécification formelle complète. Il est en outre presque impossible de saisir correctement des besoins en utilisant, directement, des méthodes formelles. Des décisions de conception doivent être prises, et plusieurs étapes intermédiaires sont souvent demandées.

Les méthodes formelles doivent ainsi éviter de proposer un remplacement de la totalité du processus de conception, mais leur intégration dans le processus de développement peut conduire à des solutions avec moins d'erreurs, et dans une plus courte période temporelle. En ce qui concerne notre propre domaine, à savoir les interfaces homme-machine, les spécifications formelles fournissent des modèles complets, non-ambigus et cohérents [Palanque & Bastide, 1995] : preuves de propriétés sur la conception, elles valident le fonctionnement de l'application créée avant sa mise en œuvre.

En conséquence, la nécessité de telles spécifications formelles étant démontrées, nous aborderons la question suivante, avant d'établir en un second temps la spécification proprement dite de tous nos concepts auto-adaptables. De quelles manières, en effet, et selon quels procédés les plus adaptés à nos besoins pouvons-nous définir et spécifier formellement, de manière précise et efficace, chacun de nos divers scénarios auto-adaptables, voire plus spécifiquement chacune des nombreuses caractéristiques, actions et réactions envisageables pour l'utilisateur ?...

Nous allons ainsi étudier en détail, et surtout aussi comparer, des outils de spécification formelle nommés respectivement *Estelle*, *LOTOS*, *PRS*, *SDL* et *UML*. Quant aux très nombreuses autres possibilités de formalismes existant à l'heure actuelle, l'étude bibliographique que nous avons associée à ces travaux a démontré, comme précisé en introduction, que les cinq outils présentés ci-dessus apparaissent de loin comme étant les plus impliqués dans le domaine des télécommunications qui nous intéresse.

2.4.2. Description des diverses caractéristiques qualitatives

2.4.2.1. *Estelle*

Standard international de description formelle pour spécification de systèmes distribués, ce premier langage correspond plus précisément à une technique à base de machines étendues à états finis, communicantes et concurrentes.

La spécification d'après *Estelle* peut être utilisée comme entrée pour les techniques de génération de tests de conformité, ces derniers atteignant alors une haute couverture de défauts [Amer, Sethi & al., 1994]. Langage expressif, bien défini et bien structuré, capable de spécifier de manière complète, consistante, concise et non-ambigüe, *Estelle* possède une sémantique formelle, mathématique et indépendante de l'implémentation. De plus, cet outil demeure une des deux *Formal Description Techniques (FDT)* de l'*International Standard Organisation (ISO)* pour la spécification de protocoles de communication informatique. Les besoins principaux pour des implémentations de protocoles, à savoir l'exactitude mais aussi l'efficacité, sont très bien comblés puisque les *Formal Description Techniques* sont conseillées pour assurer la conformité avec les besoins en exactitude. Avec leur syntaxe et sémantique formelles, elles proposent une description précise de systèmes et supportent la vérification d'aspects fonctionnels et non-fonctionnels [Thees & Gotzhein, 1998]. Enfin, il est encore intéressant de noter qu'*Estelle* opère la description du comportement par des clauses de transition (*from, to, provided, when, priority, delay, any, etc.*). Il s'agit ainsi d'une méthode de spécification abordable, visuelle et rapide.

2.4.2.2. *LOTOS*

Abréviation signifiant en anglais *Language Of Temporal Ordering Specification*, *LOTOS* est un standard international, mais défini comme un langage algébrique de spécification basé sur l'analyse formelle, le classement temporel et la validation du comportement observé.

Plus précisément, la Technique de Description Formelle *LOTOS* se caractérise par la définition des relations temporelles relevées tout au long des interactions constituant le comportement extérieurement observable d'un système [Amyot, 1994]. Langage spécialement développé pour la description formelle des interfaces, des services et des protocoles, *LOTOS* vérifie constamment la bonne saisie des informations pertinentes vis-à-vis des besoins. Demeurant applicable aux systèmes distribués et concurrents en général, ce langage se charge aussi de la découverte et de la compréhension d'éventuels problèmes de concurrence, de non-déterminisme, de collision et d'ordonnancement des événements. Puissance de modélisation utilisée pour les systèmes temps-réel et la conception d'interfaces dédiées aux aspects comportementaux [Palanque & Bastide, 1995], cette technique est, en outre, supportée par de nombreux outils en constante amélioration et particulièrement utiles pour la puissance d'analyse et de validation. Nous pouvons citer, pour exemples, *ELUDO* et *LOLA* (exécution pas-à-pas, tests d'analyse et de conception), *LMC* (vérification des propriétés), *GOAL* (contrôle de l'accessibilité des actions), mais aussi et encore *SMILE*, *CAESAR*, *TOPO*, *CADP*, etc.

Finalement, depuis que *LOTOS* est un standard *ISO*, son domaine d'action ne cesse de s'étendre : partie matérielle (*hardware*) et téléphonie, mais également systèmes d'exploitation, embarqués comme temps-réel.

2.4.2.3. PRS

Traduit de son côté par l'expression *Procedural Reasoning System*, PRS est un langage de contrôle et de supervision adapté aux robots autonomes afin de représenter et d'exécuter des procédures et des scénarios dans des environnements dynamiques [Ingrand, Chatila & al., 96]. Ainsi davantage dédié à la cybernétique proprement dite qu'aux divers domaines de l'interface homme-machine, sa capacité à manipuler en temps réel des demandes rigoureuses et des tâches multiples amène toutefois PRS à construire et agir sur des scénarios conséquents, de même qu'à poursuivre des buts en étant constamment sensible au changement des modèles d'événements.

2.4.2.4. SDL

SDL, qui se voit développé en *Specification and Description Language*, devient, lui aussi, un standard international largement utilisé dans l'industrie des télécommunications et est défini plus spécifiquement comme un langage de description des systèmes de communication. SDL s'attache à décrire un système en termes de processus interactifs, appartenant chacun à des machines étendues à états et échangeant des messages [Mellor, Tockey & al., 1998]. Plusieurs bénéfices significatifs en sont alors extraits, et notamment celui de la vérification précoce. Celle-ci correspond à l'utilisation de spécifications d'actions précises afin de toujours fournir une simulation basée sur une spécification et des preuves formelles d'exactitude, tôt dans le cycle de vie du logiciel. Les problèmes détectés tôt peuvent en effet être résolus avec beaucoup moins de travail supplémentaire que les autres, voire dirigés vers une réduction significative des coûts.

2.4.2.5. UML

Langage de modélisation "objet" né de la fusion des trois méthodes "objet" les plus diffusées dans le monde (*OMT*, *Booch et OOSE*), UML (*Unified Modeling Language*) correspond à une notation en passe de devenir un standard mondial en matière de notation et de communication autour de l'objet [Divoux, 2001].

Langage standard de l'industrie, à la fois riche et puissant, UML permet entre-autres la conceptualisation et la visualisation d'un problème, la spécification d'un système logiciel complet et la construction aussi d'une implémentation ou d'une documentation. En simplifiant le traitement complexe d'une conception logicielle par établissement d'un plan détaillé de construction [Rational, 1997], ce langage couvre en effet un large panel de résultats, depuis des cas d'utilisation et des scénarios jusqu'au comportement d'états et à la déclaration d'opérations [Mellor, Tockey & al., 1998].

D'ores-et-déjà supportée par de nombreux ateliers de génie logiciel, cette notation UML est composée de quatre diagrammes de structure et cinq diagrammes de comportement, vues partielles d'un modèle trop complexe pour être représenté en totalité. Nous retiendrons alors le diagramme de classe (premier diagramme de structure) qui propose avant tout le modèle central utilisé pour la génération du code. De son côté, le diagramme des cas d'utilisation, quant à lui premier diagramme de comportement, s'occupe de recenser les fonctionnalités du système étudié

du point de vue de l'utilisateur, c'est-à-dire de regrouper toutes les actions que doit réaliser ce système pour aboutir à un résultat observable par l'utilisateur. Notons que le groupe d'actions ainsi défini rejoint directement notre notion primordiale de "scénario" explicitée auparavant.

2.4.3. Dégagement de la méthode de formalisation adéquate

Afin de déterminer à présent quel langage de formalisation nous allons choisir pour l'établissement des spécifications d'actions inhérentes à nos fonctionnalités auto-adaptables, nous nous devons de lister ci-dessous les avantages que nous souhaitons voir apparaître au sein des méthodes que nous privilégierons. Ainsi, la méthode de formalisation attendue au niveau de nos systèmes intelligents se doit d'être, à la fois :

- théoriquement complète, correcte et robuste,
- relativement récente, normalisée et promise à un avenir évolutif,
- dédiée aux télécommunications et autres protocoles de communication,
- orientée vers l'interfaçage homme-machine et les agents intelligents communicants,
- tournée vers les aspects comportementaux des systèmes traités,
- capable de gérer le concept du "temps réel",
- efficace dans la traduction exhaustive de scénarios en formalismes,
- associée à des outils ou environnements de développement efficaces.

Au vu des informations que nous avons rassemblées, nous optons pour le langage de spécification *LOTOS* (*Language Of Temporal Ordering Specification*). Face à ses concurrentes, cette méthode possède en effet la gestion la plus exhaustive et la plus précise de l'ensemble des "qualités" que se doit d'être capable de traiter la spécification formelle destinée à travailler sur nos scénarios d'assistance auto-adaptable de l'utilisateur. Nous en résumons d'ailleurs tout-de-suite les multiples avantages et caractéristiques avancées :

- standard international de description et d'analyse formelles,
- langage algébrique de spécification,
- interfaçage orienté services et protocoles,
- classement temporel et ordonnancement des événements,
- applicable aux systèmes distribués, concurrents et temps-réel,
- observation du comportement,
- vérification de la pertinence des informations et validation,
- nombreux outils pratiques et évolutifs.

Bien évidemment, nous nous devons aussi d'explicitier les raisons qui nous amènent à privilégier le langage *LOTOS* par rapport aux quatre autres techniques de spécification formelle. Voici en conséquence, à venir, les différents points faibles qui nous sont apparus décisifs au sujet des méthodes non choisies :

- *Estelle* est ancienne, plus vraiment évolutive, non orientée vers l'interfaçage et ne présente plus d'outils disponibles depuis des années.
- *PRS* et *UML* ne sont pas dédiées aux télécommunications, ni davantage orientées vers l'interfaçage par ailleurs.
- *SDL*, enfin, ignore les aspects comportementaux indispensables à nos travaux.

2.4.4. Spécification formelle effective de nos actions auto-adaptables

Par définition, une spécification formelle se charge de produire le modèle d'un système à un niveau d'abstraction donné. Cette modélisation peut ensuite être testée, validée, et même utilisée, pour une analyse de performance notamment [Saloña, Vives & Gómez, 1993]. De son côté, une spécification *LOTOS* se caractérise plus précisément par la description dynamique d'un système par ordonnancement "d'événements" ou "instances de communication". Autrement dit, une telle formalisation cherche à définir tous les ordonnancements d'événements possibles qu'un observateur extérieur peut percevoir.

[Amyot, 1994] nous fournit alors un détail relativement complet des principaux concepts de spécification selon le langage *LOTOS*. En effet, si la description des systèmes s'effectue toujours sous *LOTOS* en terme de "processus", ce dernier y est vu comme une boîte noire, en interaction avec son "environnement" via une "entrée" visible appelée "porte" (les actions internes de tout processus *LOTOS* sont donc invisibles de l'environnement). L'élément de base de toute spécification est "l'action" qui représente une synchronisation entre des processus, entre un processus et son environnement, ou encore les deux (les actions en *LOTOS* sont également atomiques : arrivée instantanée, sans consommer de temps). Aussi, nous devons constamment garder à l'esprit que chaque "expression comportementale" est construite en combinant les actions avec des opérateurs, mais aussi éventuellement des instantiations d'autres processus.

Nous aboutissons ainsi à la formalisation complète, en *LOTOS* donc, de l'ensemble de nos scénarios auto-adaptables. En nous référant à la **Figure 2.3.4.2**, l'**Annexe 2** nous fournit alors tous les détails de cette spécification. Nous constatons que cette dernière s'avère très justement, comme nous l'avons démontré auparavant, indispensable à tout travail de qualité en matière d'implémentation. Par conséquent, les multiples développements de cette spécification formelle nous a permis de mettre en place une appréciable rigueur concernant la conception de nos modèles auto-adaptables. Ceci établi, ils nous ont aussi et surtout fourni les capacités d'entériner nos différentes fonctionnalités de manière théorique, rationnelle et exhaustive.

2.5. Conclusion : enseignements et perspectives

Le langage formel *LOTOS* nous est apparu comme étant le plus évolutif et le plus adapté à la description, à la simulation et surtout à la validation de fonctionnalités auto-adaptables issues de la télécommunication et de l'interfaçage homme-machine. Les très nombreuses idées et, *a fortiori*, les multiples actions des divers scénarios auto-adaptables caractérisant notre Assistant

Individuel de Télécommunication s'avèrent "bien pensées", bien agencées les unes par rapport aux autres et très complètes. Elles deviennent donc performantes, efficaces et implémentables.

En ce sens, une importante réflexion et une progression forte de notre part ont permis d'échaffauder des bases solides et indispensables aux implémentations à venir au sujet de notre nouvel outil intelligent. Les enseignements pour la suite de nos recherches ont effectivement été riches et puissants, que ce soit au niveau des types et structures de données à mettre en œuvre, en ce qui concernent les besoins en processus et autres procédures à concevoir, l'intégration des périphériques caractérisant les terminaux actuels, ou encore les paramètres d'interaction entre les différentes actions et phases auto-adaptables.

D'autres points déterminants subsistent néanmoins dans la découverte et une traduction concrète de nos nombreuses idées en termes d'agents intelligents et de scénarios descriptifs ciblés, une nouvelle définition de l'auto-adaptation telle que nous souhaitons l'implémenter dans nos terminaux, le schéma architectural et relationnel de nos fonctionnalités auto-adaptables.

Afin d'insister aussi sur l'indissociable contexte d'application qui domine ces constats, nous devons relever à ce niveau que l'avènement de nos recherches s'est de plus révélé jusqu'ici par l'approfondissement de la conception et la réalisation d'interfaces physiques, "matérielles" ou encore multi-médias. Ces travaux correspondent en effet à une définition technologique dite "des sens" [Sorin & Colaïtis, 1999]. Par suite, c'est une technologie plus orientée vers "le sens" en particulier que nous avons réussi à illustrer, ce grâce à l'élaboration et à la définition, au cours du second et présent chapitre, de processus "intelligents" d'interaction, d'innovation et de créativité. Cela s'est opéré au travers d'études de perception, du traitement précis de contenus, d'ingénierie avancée de la connaissance aussi et de procédé de raisonnement enfin.

Par conséquent, si nous avons, en permanence, associé toute cette technologie aux utilisateurs "grand public", l'efficacité et l'intelligence de nos systèmes doivent également être améliorées par des capacités de réactions adéquates à des situations réelles et parfois même impossibles à décrire complètement lors de la conception de ces systèmes. Cela passe alors par la théorie et la mise en pratique de notre nouveau panel de procédés d'apprentissage automatique et de séquençement d'actions que nous aborderons en détails lors des deux derniers chapitres.

C'est pourquoi, il s'agit pour nous de chercher à promouvoir, à valider et donc surtout à concrétiser au mieux les scénarios auto-adaptables et actions successives mis en avant jusqu'ici. Cependant, nous nous rendons bien compte que l'auto-adaptation que nous souhaitons, et principalement sa propriété automatique, discrète et non-insistante vis-à-vis de l'utilisateur, ne peuvent être implémentées que grâce à un modèle d'utilisateur implicite, robuste et efficace.

Dans ce but, nous allons maintenant poursuivre nos travaux par la prise en considération du principal intéressé, du facteur primordial de nos recherches : l'homme, l'utilisateur lui-même. En effet, nous nous devons de définir et surtout exploiter un nouveau modèle dédié aux propres actions auto-adaptables de l'utilisateur dans le traitement des fonctionnalités caractérisant nos concepts d'assistants intelligents. Nous découvrirons ainsi, entre-autres, les très intéressantes informations et données que pourront nous apporter une analyse et des observations résultant des manipulations successives de l'utilisateur, de même qu'une judicieuse prédiction couplée à une anticipation auto-adaptable d'actions en fonction de la classe comportementale de celui-ci.

Chapitre 3 :

Modèle Auto-Adaptable de l'Utilisateur

*L'interface doit être adaptable et évoluer,
pour tenir compte de la multiplicité des utilisateurs
et du fait que l'apprentissage est un processus dynamique.*

Eric Bruillard

[Bruillard, 1997]

En prolongeant cette remarquable pensée aux nombreuses ramifications, Eric Bruillard insiste sur cette nécessité d'évolution permanente de la part des interfaces en y associant une qualification au sujet de "l'auto-adaptation". Cette dernière approche pour lui de la capacité d'un système à découvrir automatiquement, par observation à partir et au travers d'une interface de communication, tous les choix particuliers d'un utilisateur quelconque [Bruillard, 1997].

Bien qu'à ceux-ci nous adjoindrons nos propres concepts de prédiction et d'anticipation, la transition avec les deux premiers chapitres se profile naturellement puisqu'à la problématique essentiellement technologique, nous devons également associer maintenant celle du domaine des "Sciences Humaines". Il s'agit de prendre en considération les informations, données et autres caractéristiques issues des diverses actions réalisées successivement par l'utilisateur, ce toujours en complète concordance avec ses états courants de comportement.

Dans ce but, il faut intégrer une adaptation dynamique de l'interface aux comportements de nos utilisateurs, ainsi qu'à leurs environnements respectifs. En effet, indissociable des apports d'usage, d'acceptabilité et de convivialité des services d'assistants rendus à l'utilisateur, les "Sciences Humaines" sont aujourd'hui perçues en priorité par la construction de représentations du comportement des utilisateurs pris en compte et l'appropriation des modalités de présentation et/ou de tâche que nous appelons encore, de nos jours, "ergonomie" [Sorin & Colaitis, 1999].

C'est pourquoi, en plus des méthodes de représentation des connaissances internes à l'interface et ses mécanismes de raisonnement, et afin aussi d'identifier au mieux les besoins de l'utilisateur en matière de télécommunication, nous allons apprendre les opérations routinières de celui-ci, sans oublier la prédiction ou l'anticipation active de ce que l'utilisateur va commander et vouloir réaliser. Dans ce cadre, nous serons amenés à établir et à étudier son profil évolutif du comportement et, par extension, différents algorithmes d'extraction des connaissances à partir des données obtenues alors précisément sur le comportement de l'utilisateur.

Tout ceci exposé, nous nous sommes en outre plus particulièrement intéressés à trois aspects principaux, bien complémentaires. Ceux-ci sont, successivement, la représentation de connaissances factuelles au niveau de l'utilisation globale d'un terminal de communication, la construction proprement dite d'un modèle d'utilisateur, ainsi que la modélisation de mécanismes de raisonnement à partir des diverses données recueillies du comportement de chaque utilisateur.

Par ailleurs, si nos agents intelligents sont à terme incorporés dans un ou davantage de systèmes multi-agents, nos procédés d'apprentissage automatique, quant à eux, doivent aussi pouvoir s'avérer adéquats, tels que les séquençements d'actions. L'élaboration, au cours de ce troisième chapitre, de modèles d'utilisateur prenant en compte un maximum des caractéristiques individuelles et "intelligentes" de chaque utilisateur va alors nous permettre de rendre notre outil informatique aussi auto-adaptable et complet que possible. Nous aboutissons ainsi à la mise en œuvre de nos notions initiales d'aide et d'assistance continue à l'utilisateur. De plus, ajoutés à de nouvelles personnalisations et individualisations des différentes fonctionnalités de l'interfaçage de chaque terminal appréhendé, nous obtenons enfin des substitutions et allègements permanents des multiples actions effectuées de manière courante.

D'un autre point de vue, si la terminologie d'interface intelligente présente *a priori* un concept de travail provenant avant tout du domaine de l'Intelligence Artificielle, nous souhaitons également nous atteler aux mécanismes plus psychologiques et comportementaux, à travers des modèles d'utilisateur essentiels à un fonctionnement crédible de notre interface auto-adaptable.

En conséquence, si nos recherches se résument à la tentative constante de simplification automatique de la vie de l'utilisateur et de la réduction aussi de l'effort cognitif fourni par celui-ci devant un terminal de communication, nous devons impérativement nous poser la question à suivre. De quelles manières et selon quels procédés pouvons-nous exhaustivement définir chacun de nos utilisateurs, chacune de leurs caractéristiques, actions et réactions ? Ceci sans perdre également de vue qu'elles devront s'avérer "auto-analysables" et apprises automatiquement, afin d'être par la suite activement prévisibles et anticipées.

C'est donc dans cette optique que nous souhaitons maintenant définir un entier concept de modélisation auto-adaptable de l'utilisateur, destiné à compléter notre intégration de la notion d'intelligence dans une interface évolutive pour les terminaux de communication "grand public". Après avoir alors exploré différentes techniques de conception et d'implémentation des modèles d'utilisateurs déjà existants, nous aborderons plusieurs domaines, et agents significatifs, de la représentation mentale et contextuelle de l'utilisateur d'aujourd'hui. Puis, nous prolongerons nos travaux de recherche par une définition qualitative d'un modèle auto-adaptable de l'utilisateur, suivie de l'étude d'une méthode appropriée d'analyse des comportements et de la conception d'un prototype "d'Interview Comportementale".

3.1. Théorie des procédés actuels dédiés à l'utilisateur

3.1.1. Introduction

Contrairement à la détection ou la reconnaissance des personnes par des méthodes dites biométriques (visages, empreintes digitales, signature vocale, géométrie de main, iris, dynamique de la signature, etc) et donc grâce aux caractéristiques physiques de chaque utilisateur, le modèle auto-adaptable d'utilisateur s'attache plutôt à étudier le côté "mental" et "comportemental" de ce dernier. De plus, ce système s'avère plus complexe à concevoir puisque notamment plus changeant et modulant au cours du temps que les qualités purement physiques de nos utilisateurs.

Nos travaux nécessitent par conséquent un prototypage, une évaluation et une précision considérables avant de capturer tous les aspects de l'utilisateur avec suffisamment d'efficacité

[Benyon & Murray, 93-1]. En outre, certains critères d'acceptation par les utilisateurs sont à prendre en considération, comme le fonctionnement "sans heurts" du matériel ou l'absence de sources d'inconfort dues par exemple à d'éventuelles difficultés d'utilisation. C'est pourquoi, il nous apparaît indispensable, en premier lieu, d'aborder et d'étudier plus avant différentes méthodes d'élaboration et d'implémentation d'un tel modèle, de même que plusieurs procédés possibles de représentation du raisonnement et des réactions d'un utilisateur face à la machine.

3.1.2. Définition d'un modèle de connaissances de l'utilisateur

Le modèle des connaissances d'un utilisateur peut être défini par une structure de données caractérisant, pour l'ensemble du système considéré et mis en place, l'état de toutes les connaissances de cet utilisateur. Cette mise en oeuvre est aujourd'hui indispensable pour réaliser des agents et autres tuteurs artificiels ou "intelligents" suffisamment adaptatifs [Bruillard, 1997].

Il s'agit d'une définition à différencier du cas de l'Enseignement Assisté par Ordinateur correspondant plus particulièrement à l'ensemble des tactiques de parcours prédéterminées, puis associées aux réponses correctes ou erronées de l'élève. En effet, ce principe permet de mesurer l'écart qui se situe entre les connaissances (supposées) de l'apprenant et les "connaissances-cibles" (enjeu de l'apprentissage) telles qu'elles sont représentées dans le système utilisé. Nous notons en outre que nos modèles de connaissances doivent se construire de manière dynamique et incrémentale, en s'appuyant fortement sur le comportement observable de l'utilisateur.

De ce nouvel objectif apparaissent alors trois problèmes, bien reliés entre eux puisque nous constatons en effet ci-dessous que la solution adoptée pour l'un conditionne de manière très importante la façon de traiter les deux autres. Ces nouvelles difficultés sont :

- le choix d'une représentation efficace pour enregistrer les données sur l'utilisateur (constitution de son modèle de connaissances attitré),
- la mise en place ensuite de mécanismes pour initialiser et mettre à jour ce modèle tout au long de l'interaction avec le système,
- l'implantation finalement d'un processus d'interprétation de ce modèle afin d'orienter les procédures de décision liées à nos futures stratégies d'auto-adaptation.

Pour réussir à apporter un maximum de solutions adéquates à ces trois obstacles, nous allons nous focaliser sur le sujet de la description du comportement de chacun de nos utilisateurs potentiels dans la réalisation de ses propres tâches. Puis, nous chercherons à établir, à partir de ces comportements analysés, des états cognitifs correspondants, complets, fidèles et exploitables pour l'ensemble de nos modèles auto-adaptables.

3.1.3. Techniques de conception des modèles d'utilisateurs

Liés aux états de l'art développés dans le contexte des interfaces dites "intelligentes", les modèles de l'utilisateur demeurent, sans exception, des systèmes ayant été mis en oeuvre très récemment. En effet, si la caractéristique essentielle d'un produit destiné au "grand public" est

de convenir à une grande diversité de personnes (clients plus ou moins âgés, habitués ou pas du domaine, utilisateurs ou non des systèmes automatiques actuels de vente, etc), elle se retrouve aussi dans des contextes d'utilisation très variés (heure de pointe *versus* période calme, voyage fréquent ou exceptionnel, acheter un titre de transport ou seulement s'informer, etc). Les outils de modélisation du comportement des usagers sont longtemps restés, pour d'évidentes raisons de capacités matérielles insuffisantes, techniquement impossibles à concevoir [Bernard, 1998]. C'est pourquoi, dans le cas notamment de systèmes d'aide adaptatifs et d'assistance automatique, de nombreuses approches dédiées à l'utilisateur reviennent en force et sont envisagées par le biais d'études détaillées du comportement de chaque type de client dans chaque contexte d'utilisation.

Fonder un modèle de l'utilisateur sous-jacent à un système auto-adaptable relève alors d'un placement de tous les utilisateurs potentiels sur un axe allant d'une absence complète de connaissance à une pratique experte de celle-ci. Si une telle approche tient compte de l'utilisateur et de toutes ses interactions avec le système considéré, le niveau de connaissance mis en jeu dépend aussi essentiellement de la fréquence de ces utilisations. Il est donc possible de repérer expérimentalement l'évolution, dans le temps comme dans la durée, de ce facteur. Cette manière de procéder n'est en outre pas exclusive et peut toujours être complétée par d'autres dimensions issues d'études empiriques supplémentaires ou futures.

L'assistant auto-adaptable, tel que nous le percevons à présent, s'appuie ainsi sur un modèle de l'utilisateur composé, par définition, des trois assertions à suivre [Bernard, 1998] :

- les utilisateurs sont censés avoir d'autant plus besoin d'aide qu'ils sont peu expérimentés,
- les utilisateurs sont censés prendre d'autant plus aisément l'initiative dans le dialogue interactif qu'ils sont plus expérimentés,
- les utilisateurs sont censés agir d'autant plus rapidement qu'ils sont plus expérimentés.

Par ailleurs, l'efficacité de l'architecture du système de modélisation d'un utilisateur est également une contrainte primordiale à respecter [Ardissono & Torasso, 2000]. Nous ne pouvons raisonnablement allouer, pour la révision dynamique d'un tel modèle de l'utilisateur, qu'un temps minime. Dans le cas contraire, la perception explicite d'un tel système serait extrêmement gênante au niveau de l'utilisateur.

D'un autre point de vue, nous devons penser à subdiviser le processus en deux étapes distinctes, la surveillance de l'utilisateur et la traduction de ses comportements, ne serait-ce que dans le but d'alléger et d'accélérer les temps de calcul en machine. Cela dit, nous reviendrons encore largement sur ce point essentiel du fractionnement des tâches à réaliser par l'ordinateur en matière de traitement des séquences d'actions effectuées par nos utilisateurs. De nouvelles distributions du travail intrinsèque de nos logiciels seront ainsi, entre-autres, destinées à améliorer l'efficacité et les capacités de nos terminaux et applications intelligentes de communication.

Enfin, il ne nous reste plus qu'à constater l'importante progression, théorique comme technique, qui s'est opérée depuis quelques années dans le cadre des modèles de l'utilisateur. Dans le secteur des "systèmes-experts" par exemple, considérés comme les précurseurs ou les premières structures indépendantes ayant fait preuve d'une certaine "intelligence artificielle", seulement quatre types d'utilisateurs, discrets, stricts, séquentiels et surtout non évolutifs, pouvaient effectivement être pris en considération [Harrington, Banks & Santos, 1996] : l'utilisateur de l'application, l'expert attaché à l'application, l'ingénieur chargé de la connaissance et, en définitive, l'indispensable informaticien (cf. 1.2.4.2 pour davantage de précisions).

Maintenant, nous ne pouvons plus imaginer autres modèles-utilisateurs que ceux traitant les actions de ces derniers de manière dynamique, incrémentale, ... auto-adaptable en somme.

3.1.4. Capacités des agents de représentation contextuelle de l'utilisateur

En nous appuyant sur les états de l'art introduits, nous pouvons à présent traiter de la véritable nature, du rôle exact et significatif de tout "générateur de profils d'utilisateurs" qui s'emploie à correspondre plus précisément à un système d'observation et de représentation de l'interaction courante et passée de l'utilisateur avec une application donnée. La prise en compte de toutes les actions et réactions directes de l'utilisateur, dans le but de mettre à jour son état et de mieux répondre ensuite à ses intérêts, nous permettent d'assister à ce niveau à l'avènement d'une intéressante prédilection pour diverses et récentes méthodes d'Intelligence Artificielle. Les apprentissages distribués et les mécanismes de représentation sous la forme de systèmes multi-agents évolutifs génèrent et maintiennent des caractérisations d'utilisateurs, ou des agents de découverte et de filtrage de l'information [Moukas, 1997].

Le comportement de chaque utilisateur est alors surveillé discrètement, implicitement, grâce à la mise en œuvre d'une structure efficace destinée au traitement dynamique des modèles d'utilisateurs, à l'image des architectures que nous avons étudiées précédemment. Les actions exécutées par l'utilisateur et ainsi mémorisées se doivent ensuite d'être récapitulées (notamment dans des bases de données avancées), analysées et utilisées pour mettre à jour le modèle de chaque utilisateur, modèles qui vont eux-mêmes induire des prises de décisions auto-adaptables. Nous nous concentrons par conséquent sur le secteur des caractéristiques de l'utilisateur qui décrit tous ses besoins et définit ses multiples intérêts, comme son style d'interaction ou encore son niveau d'expertise.

De telles caractéristiques de l'utilisateur incluent obligatoirement la "réceptivité" de celui-ci en décrivant sa disposition face à l'acquisition et à la maîtrise des grandes quantités de données qui lui sont constamment proposées [Ardissano & Torasso, 2000]. Il est d'ailleurs à relever que certains utilisateurs deviennent confus s'ils sont exposés à beaucoup d'informations, et il faut donc être préparé à ce que l'interaction de la machine avec ces personnes demande à être améliorée en contraignant notamment le contenu de l'interface mise en place. De l'autre côté, des utilisateurs très réceptifs peuvent au contraire apprécier des écrans contenant toutes sortes d'informations en grand nombre. Cependant, la réceptivité de l'utilisateur peut aussi faiblir au cours de son travail, principalement en raison de facteurs de fatigue... Par conséquent, même un utilisateur très ouvert, après une longue interaction, peut apprécier et vouloir bénéficier d'interfaces synthétiques.

En second lieu, les intérêts extraits des manipulations successives de l'utilisateur vont être amenés à décrire ses différentes attitudes face à toute information technique, esthétique ou fonctionnelle issue des caractéristiques principales de notre interface. Ce type d'informations peut servir de base pour classifier les préférences de l'utilisateur. En effet, des données comme la taille et la couleur d'icônes sont susceptibles d'être classifiées en caractéristiques esthétiques, alors que les "équipements" offerts par l'interface (par exemple, des barres d'outils proposant diverses commandes pratiques) peuvent être reliés aux aspects fonctionnels.

Finalement, un ensemble de caractéristiques d'expertise peut aussi servir à représenter la connaissance de l'utilisateur à disposer des principales bases, classes et fonctionnalités mises en avant par notre interface. Plusieurs caractéristiques d'expertise différentes peuvent même être maintenues pour chaque notion ou catégorie principale d'une entité automatique spécifique. Une première conclusion à cette partie nous ayant permis et amené à nous immiscer dans ce monde nouveau de la modélisation des utilisateurs, [Savidis, Akoumianakis & Stephanidis, 1997] nous rappelle que les capacités propres aux interfaces intelligentes se destinant à un utilisateur "grand public" entraînent, non seulement, une auto-adaptation aux besoins individuels de cet utilisateur,

mais aussi et surtout une individualisation et une personnalisation automatiques à l'exécution. Ces dernières propriétés se laissent alors conduire par des paramètres orientés "utilisateur", tels que des compétences, des besoins ou encore des préférences. Mais ne perdons pas de vue ici que nous avons aussi déjà découvert auparavant que les interfaces intelligentes sont particulièrement appropriées à la construction d'interactions pouvant être façonnées, de manière autonome et automatique, à l'intention des utilisateurs individuels.

C'est ainsi, sur les bases d'un modèle architectural compréhensif envers chacun de nos utilisateurs, de même que pleinement dédié à une interaction auto-adaptable avec ceux-ci, que nous allons poursuivre nos recherches par l'identification des principaux besoins fonctionnels et communicatifs de chaque personne.

3.1.5. Récapitulatif des principaux enseignements établis

La modélisation de l'utilisateur, c'est-à-dire une réelle spécification de l'utilisateur ou la considération de l'utilisateur comme une entité instable que le système essaie de maîtriser, paraît vraiment essentielle pour parvenir à un dialogue auto-adaptable [Stémart, 1997]. Il faudra parfois aller jusqu'à tenir compte d'une évolution éventuelle de l'interface toute entière en fonction du niveau de compétence de l'utilisateur courant. En outre, on s'accorde aujourd'hui à accepter que le développement de systèmes adaptatifs nécessite, dans le but d'obtenir des données pour le modèle de l'utilisateur, à la fois une interrogation explicite des usagers et la gestion d'inférences implicites émanant de l'interaction [Benyon & Murray, 1993-2].

Il faut cependant, en accord avec notre concept d'auto-adaptation, tirer un avantage maximum de l'information que l'utilisateur fournit librement, c'est-à-dire sans avoir recours à une interaction séparée, de même que du temps de réflexion de cet utilisateur, en notant par ailleurs que l'attention de l'utilisateur peut être irrégulière dans le temps [Lieberman, 1997]. Les méthodes "traditionnelles" de génération des profils d'utilisateurs, à l'image des interviews d'utilisateurs ou des stéréotypes (modèles de groupes d'utilisateurs), restent caractérisées par un coût très important en temps. Le refus parfois des utilisateurs à identifier convenablement les catégories de leurs intérêts entraîne un coût très élevé de la maintenance des différents systèmes [Moukas, 1997]. Par conséquent, nous devons opérer l'acquisition de notre modèle auto-adaptable d'utilisateur à partir d'interviews seulement occasionnelles de l'utilisateur, en leur préférant un système plus élaboré de règles (description, souvent simple et sans détours, des entrées dans le modèle de l'utilisateur pouvant notamment être la conséquence d'actions observées à propos de l'utilisateur) et des procédures d'inférence à base d'hypothèses implicites [Pohl, 1996].

Sans perdre également de vue que nous pouvons même pousser notre étude jusqu'à une mesure de la productivité (ou efficacité) de l'utilisateur à l'intérieur d'une interface concrète, mesure pouvant être notamment basée sur l'émulation d'une situation de stress graduellement renforcée par une réaction d'apprentissage négative [Gavrilova & Voinov, 1997] !...

De tels modèles ou représentations de l'utilisateur s'avèrent ainsi intéressants pour une interface intelligente de qualité. Nous allons alors maintenant chercher à déterminer comment nous pouvons nous appuyer sur les divers procédés présentés jusqu'ici, et dans quelles directions ceux-ci peuvent nous emporter vers de nouvelles améliorations auto-adaptables.

3.2. Définition "intelligente" d'un nouveau "modèle d'utilisateur"

3.2.1. Description des caractéristiques qualitatives

Aussi intelligente que devienne l'interface, il y aura toujours des situations où l'utilisateur comprendra mieux les problèmes que l'interface elle-même, et sera par conséquent la meilleure personne pour effectuer des adaptations à l'interface [Lennard & Parkes, 1997]. De plus, le développement de systèmes adaptatifs nécessitent la prise en compte de l'importante variété des utilisateurs [Benyon & Murray, 1993-2] susceptibles de faire appel à notre interface intelligente auto-adaptable.

En rejoignant la problématique globale de notre sujet de travail, ces citations résument toute la difficulté pour nous de définir précisément un modèle auto-adaptable d'utilisateur. Nous devons palier au maximum à ce "fossé" entre l'utilisateur et sa machine, afin d'apporter des innovations supplémentaires au niveau des modélisations d'utilisateur existantes. Pour cela, nous nous plaçons au cœur des importants domaines du "grand public" et de l'ouverture des terminaux de communication vers le monde, notamment à travers la téléphonie avancée ou l'avènement du réseau *Internet*. [Lennard & Parkes, 1997] insiste encore en nous proposant de nous demander quels utilisateurs sont préparés à posséder un ordinateur analysant tous leurs mouvements, jusqu'à leur efficacité à utiliser les différentes applications disponibles. La réflexion sur cette remarque nous apporte une nouvelle confirmation de l'intéressante direction dans laquelle nous nous engageons. L'auto-adaptation que nous souhaitons implémenter dans notre interface intelligente se doit ainsi d'être principalement discrète et non-insistante vis-à-vis de l'utilisateur.

Toutefois, il apparaît comme acquis que notre système informatique doit en permanence pouvoir être employé par tous les utilisateurs et à différents niveaux d'expertise [Rhee, Kim & al., 1997] puisque la diversité (de conception notamment) reste une des caractéristiques essentielles en interaction homme-machine [Benyon, 1998]. Il n'est donc pas moins nécessaire d'amorcer notre étude en précisant qu'un "modèle d'utilisateur" peut se définir comme un prérequis à toute interaction intelligente et personnalisée entre les ordinateurs et leurs utilisateurs [Pohl, 1996].

Nous pouvons alors décrire l'ensemble des caractéristiques ou énumérer les nombreuses "qualités" qui sont susceptibles de nous aider à distinguer, et par là-même à spécifier, chacun de nos utilisateurs lors de tout travail avec notre interface auto-adaptable :

- connaissances, niveau social et culturel, états d'esprit [Benyon, 1998]
- travail (vitesse, efficacité, précision, spécificité) [Benyon, 1998]
- environnement (hautement encadré, isolé, confortable, hostile) [Benyon, 1998]
- expériences et fréquence d'utilisation des ordinateurs [Benyon & Murray, 1993-2]
- domaine d'action et niveau de détail souhaités [Benyon & Murray, 1993-2]
- expérience du *Web*, connaissances, formation [Brusilovsky & Schwarz, 1997]
- touristes, citadins, employés, personnes âgées, aveugles ou handicapées (variété d'utilisateurs aux besoins différents) [Fink, Kobsa & Schreck, 1997]
- préférences, buts, humeurs et attitudes [Garrido, Brena & Sycara, 1996]
- facteurs démographiques (âge, sexe, première langue, lieu de naissance, particularités sociales et culturelles) [Gavrilova & Voinov, 1997]
- facteurs professionnels (position, expérience professionnelle, compétences en informatique, éducation) [Gavrilova & Voinov, 1997]

- facteurs physiologiques (réaction, maniabilité, attention) [Gavrilova & Voinov, 1997] :
 - * perception (globale, périphérique et spatiale),
 - * mémoire (opérative, iconique, à long terme),
 - * moteurs dynamiques (temps et orientation de réponse, performance senso-moteur),
 - * neurophysiologie (assymétrie fonctionnelle cérébrale),
 - * fonctions cognitives (estimation de vitesse et de distance, tolérance réactive au stress à long-terme, intelligence numérique, poursuite),
 - * attention (environnement monotone, focalisation, vigilance).
- facteurs psychologiques [Gavrilova & Voinov, 1997] :
 - * couche communicative (perception, compréhension, clarté, disponibilité, habileté),
 - * couche cognitive (style personnel cognitif, mentalité logique et déductive).
- intérêts et habitudes de l'utilisateur [Moukas, 1997]
- forces, faiblesses, préférences et niveau de compétences [Murphy & McTear, 1997]
- buts, plans, croyances, préférences de l'utilisateur [Pohl, 1996]
- croyances, intentions, objectifs et connaissances [Quaresma & Lopes, 1997]
- capacités individuelles, exigences et préférences [Stephanidis, 1997]
- motivation, émotions et humeurs de l'utilisateur [Vassileva, 1997]

3.2.2. Dégagement de la définition initiale de notre modèle

En conséquence à nos recueils d'informations construits jusqu'ici, nous conservons et travaillerons plus avant avec des données que nous avons surtout cherché à rendre synthétiques, orientées vers la pratique et les implémentations futures. Notons également qu'en matière de télécommunication, la partie appelée doit immédiatement pouvoir obtenir des informations sur l'identité et les intentions de l'appelant, et ne pas négliger les caractéristiques des précédentes communications associées à cet appelant [Nangle, Cunningham & Evans, 1998]. Nous aboutissons à la **Définition 3.2.2** ci-dessous faisant suite à de maintes réflexions dans le but principal de créer un "Modèle Auto-Adaptable de l'Utilisateur" à la fois ergonomique, efficace et exhaustif.

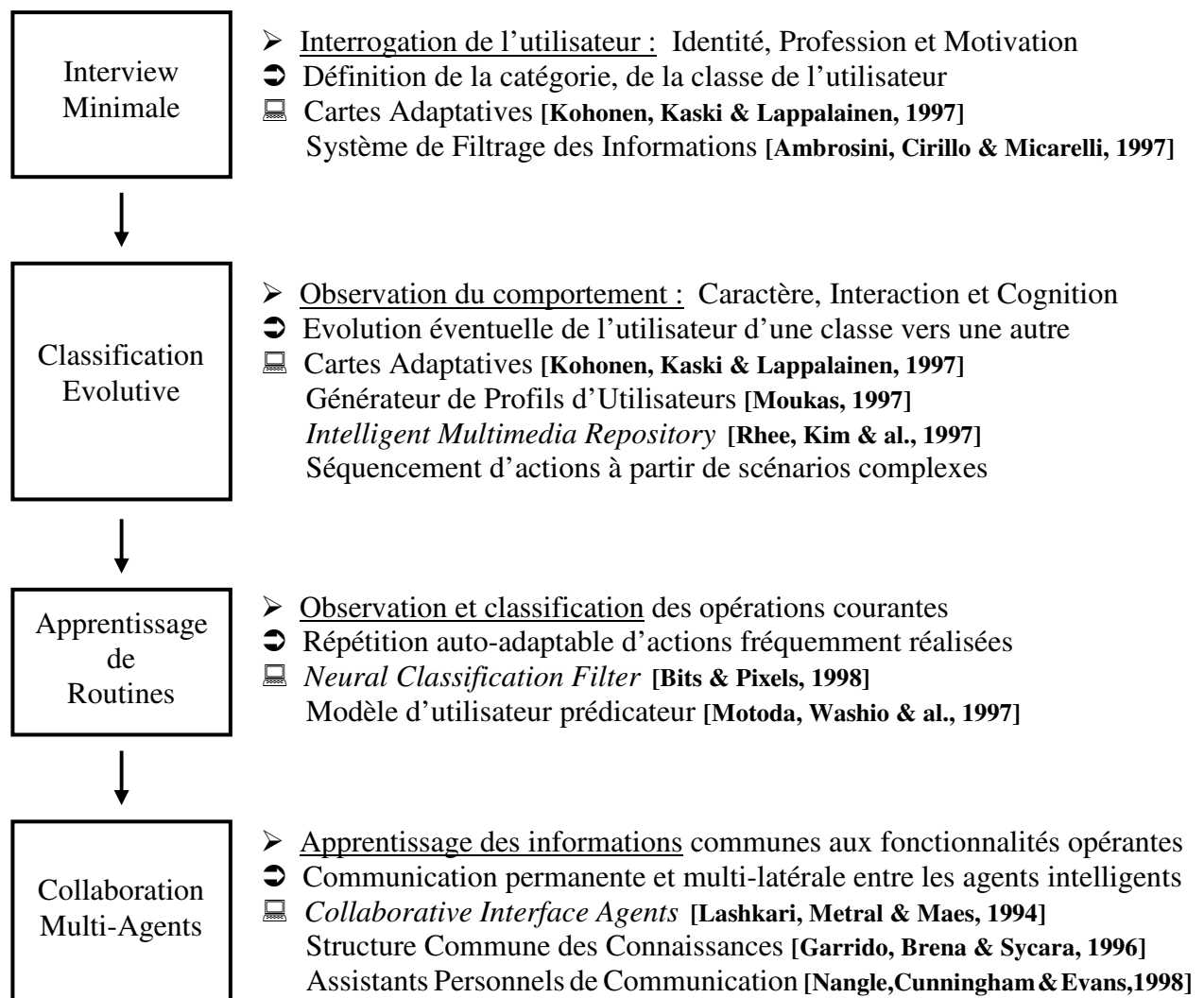
Définition 3.2.2 : *Définition initiale de notre "Modèle Auto-Adaptable de l'Utilisateur" chargée d'établir les caractéristiques remarquables d'un utilisateur en communication avec une machine*

- *Identité* (nom, prénom, surnom, sexe, âge, handicaps, origines, langue principale, éducation, croyances)
- *Caractère* (forces, faiblesses, attitudes, humeurs, réactivité au stress, habitudes, préférences)
- *Profession* (fonction, position, habilitations, cadre de travail, formation, connaissances, expérience, compétence en informatique, fréquence d'utilisation des ordinateurs et d'Internet)
- *Motivation* (objectifs, intentions, besoins, exigences, intérêts, domaine d'action, contexte, niveau de détail, précision)
- *Interaction* (perception, compréhension, disponibilité, clarté du dialogue)
- *Cognition* (style mental de réflexion, compétence logique, capacité déductive, attention, vitesse de travail, habileté, efficacité)

3.2.3. Mise en oeuvre envisagée

3.2.3.1. Schéma global de fonctionnement

A ce niveau, toutes nos précédentes investigations nous permettent d'aboutir à une première proposition de personnalisation et d'humanisation des terminaux de communication "grand public" d'ALCATEL. La **Figure 3.2.3.1** présente alors le schéma global de fonctionnement du modèle d'utilisateur tel que nous le percevons.



Légende :

- actions à effectuer
- ➡ résultats finaux souhaités
- 📄 implémentations de référence

Figure 3.2.3.1 : Schéma global de fonctionnement de notre modèle d'utilisateur

Concept comportemental, notre Modèle de l'Utilisateur est ainsi destiné à établir un profil évolutif des agissements d'un utilisateur donné. Il demeure conforme aussi à l'ensemble de nos souhaits "d'Auto-Adaptation", grâce notamment à la prise en compte des caractéristiques individuelles et "intelligentes" de chaque utilisateur ainsi qu'aux propres évolutions respectives de chacun d'eux. A nous de spécifier maintenant la puissance de raisonnement, la faisabilité, la rapidité et la facilité de mise en œuvre d'un tel système. Nous relèverons en conséquence que ce dernier possède d'indéniables avantages quant à l'ensemble de ces quatre qualités.

3.2.3.2. Analyse du système mis en avant et nouvelle définition qualitative

Grâce à un tel modèle auto-adaptable de l'utilisateur, nous parvenons à mieux cerner les différentes façons d'agir de l'utilisateur, non seulement en matière de télécommunication, mais aussi en matière de communication en général. De plus, nous appréhendons aussi plus justement les réactions prévisibles ou non d'un utilisateur "grand public" face à une machine informatisée.

Nous entrevoyons alors deux problèmes majeurs concernant la personnalisation de toute interface : la distinction de différentes classes d'utilisateurs et de l'observation des actions de chacun en vue d'une tentative d'automatisation. L'apprentissage et l'approfondissement de la notion de modèle d'utilisateur passe en effet aujourd'hui, non seulement par l'établissement du savoir, des connaissances, de ce que sait ou connaît la personne qui utilise notre interface, mais également par la gestion de l'évolution de ce savoir [Gavrilova & Voinov, 1997]. Le schéma de fonctionnement de notre modèle auto-adaptable insiste justement sur la distinction qu'il existe entre les "qualités" définies par interrogation de l'utilisateur et celles spécifiées par observation de ses différentes actions et réactions. Nous obtenons alors une seconde définition ou plutôt une mise-à-jour plus évoluée et fonctionnelle de l'énumération exhaustive précédemment détaillée, indispensable au répertoire de chacun de nos utilisateurs respectifs (*Définition 3.2.3.2* à venir).

D'un autre côté, nous remarquons encore qu'au modèle de l'utilisateur s'ajoute parfois la conception d'un procédé tout aussi primordial nommé "modèle du domaine" [Benyon, 1996]. Ce dernier est chargé de mettre à disposition de notre système auto-adaptable le profil, également en constante évolution, de la situation, du contexte et des circonstances d'actions de l'utilisateur. De plus, ce modèle du domaine est entièrement pris en compte par notre modèle auto-adaptable, ce à travers l'application partielle de deux des caractéristiques de notre précédente définition, à savoir la profession (cadre de travail) et la motivation (domaine d'action, contexte).

L'interview préalable et l'obtention "en direct" d'informations concernant l'utilisateur courant se doivent alors d'être aussi succinctes et implicites ou automatiques que possible (c'est du reste pour cela que nous l'avons spécifiée auparavant comme étant "minimale"). Il nous faut effectivement éviter de demander trop de perspicacité et d'efforts à l'utilisateur par des requêtes trop fréquentes, longues et fastidieuses. En conséquence, notre outil auto-adaptable se doit de rester toujours très discret, en évitant de submerger son utilisateur : c'est l'image que nous nous faisons par ailleurs d'un tel système intelligent, à savoir celle d'un assistant personnalisé regardant en permanence par-dessus l'épaule de son utilisateur [Akoulchina & Ganascia, 1997].

Nous rejoignons alors ici le cœur-même de notre interface auto-adaptable par la prise en compte et la gestion complète du dialogue entre l'utilisateur et son interface homme-machine individualisée.

Définition 3.2.3.2 : *Nouvelle définition plus fonctionnelle des différentes qualités remarquables d'un utilisateur en interaction avec une machine*

Interview directe de l'utilisateur :

- *Identité* (nom, prénom, surnom, sexe, âge, handicaps, origines, langue principale, éducation, croyances)
- *Profession* (fonction, position, habilitations, cadre de travail, formation, connaissances, expérience, compétence en informatique, fréquence d'utilisation des ordinateurs et d'Internet)
- *Motivation* (objectifs, intentions, besoins, exigences, intérêts, domaine d'action, contexte, niveau de détail, précision)

Observation évolutive du comportement :

- *Caractère* (forces, faiblesses, attitudes, humeurs, réactivité au stress, habitudes, préférences)
- *Interaction* (perception, compréhension, disponibilité, clarté du dialogue)
- *Cognition* (style mental de réflexion, attention, compétence logique, capacité déductive, vitesse de travail, habileté, efficacité)

3.3. Méthode d'analyse des comportements au sein d'ALCATEL

3.3.1. Définition du concept

A la recherche d'une modélisation capable d'entraîner une implémentation judicieuse des caractéristiques d'utilisateurs susceptibles de manipuler les terminaux de communication "grand public" d'ALCATEL, nous avons découvert que la société faisait déjà appel, depuis quelques années, à une méthode d'analyse des comportements [Julhiet, 1993]. De plus, il apparaît à présent que, concernant les 4 concepts à dominante comportementale qui composent les 6 "qualités" fonctionnelles nous permettant de spécifier et d'étudier nos utilisateurs (à savoir précisément le "Caractère", "l'Interaction", mais aussi la "Motivation" et la "Cognition"), cette méthode d'ALCATEL s'avère très intéressante et complète.

En effet, cette caractérisation du profil de chacune des personnes de l'entreprise permet de "classer" ces dernières en fonction de leur propre comportement dans les différentes et quotidiennes relations inter-personnelles. Cette méthode d'analyse du comportement individuel, répertoriée sous le terme de *Persona* du nom du séminaire pendant lequel elle est enseignée, s'attache à distinguer 2 critères pour "classifier" l'ensemble du personnel d'une entreprise :

- la personne étudiée est-elle introvertie ou extravertie, réservée ou expansive ?
- la personne étudiée est-elle dominante ou dominée, ascendante ou consentante ?

Pour chacun de ces critères, une échelle a été établie pour éviter que nous ne travaillions qu'avec 2 valeurs distinctes. La combinaison des 2 critères définis ci-dessus fournit 4 "profils comportementaux" (intitulés plus précisément *Dominant* ou *Ascendant*, *Dominé* ou *Consentant*, *Extraverti* ou *Expansif*, et *Introverti* ou *Réservé*). Ces comportements sont également attribuables à des catégories ou "styles sociaux" d'utilisateurs bien distincts (eux-mêmes respectivement nommés *Promouvant*, *Facilitant*, *Contrôlant* et *Analysant*) et dont les différentes interactions sont décrites en détail par le schéma à venir (**Figure 3.3.1**). Il convient de noter alors tout de suite que les tendances *Dominant* et *Ascendant* correspondent au fait "de dire" ou "d'affirmer", comme les adjectifs *Dominé* et *Consentant* se traduisent par les actions "de demander" et "de suggérer". Quant aux différents termes associés à l'axe vertical du schéma, ils se chargent de définir concrètement une certaine "émotivité" de l'utilisateur étudié.

En conclusion, la méthode affirme, visiblement avec raison, qu'il n'y a pas de bon ou de mauvais profil, mais qu'il est utile pour optimiser les relations inter-personnelles de connaître son profil comportemental ainsi que celui de ses différents interlocuteurs.

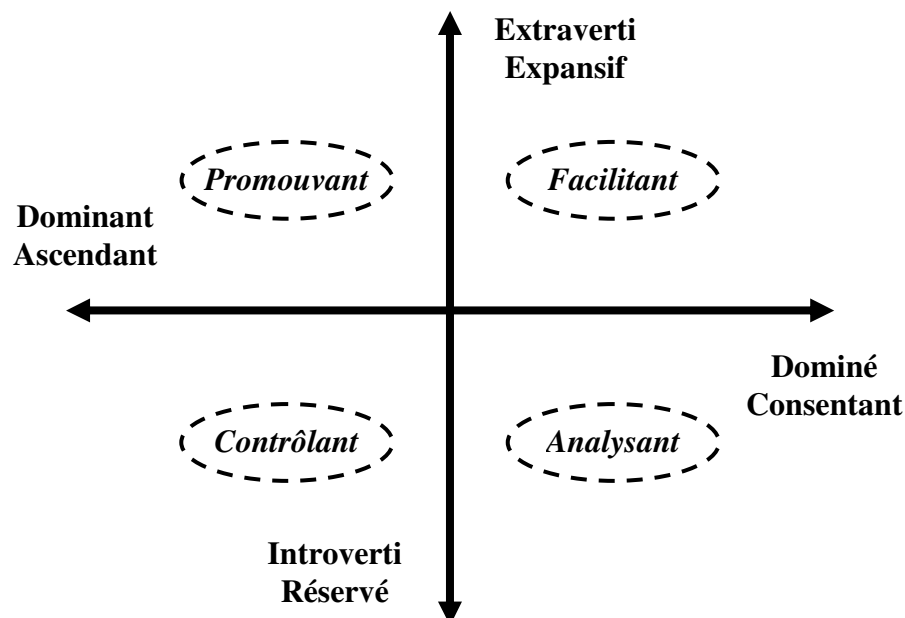


Figure 3.3.1 : Schéma général des interactions entre comportements individuels et catégories d'utilisateurs proposées par la méthode "Persona"

3.3.2. Adaptation à l'interfaçage intelligent "grand public"

Nous constatons, en découvrant *Persona* de manière plus approfondie, que les principes et autres concepts d'étude avancés par cette méthode d'analyse des comportements individuels s'avèrent, certes, complètes et instructives, mais beaucoup trop touffues, voire de compréhension difficile et trop délicate, pour les utilisateurs "grand public" que nous visons.

Par conséquent, à nous de tenter maintenant, devant l'intérêt indéniable de ce procédé de spécification des motivations et du caractère, ainsi que des capacités de communication et de raisonnement de nos utilisateurs, d'adapter *Persona* à nos besoins concernant l'interface auto-

adaptable envisagée pour nos terminaux. Nous nous proposons de conserver et, par suite, de faire appel aux différentes distinctions et caractéristiques à venir. Notons à ce sujet que, si celles-ci correspondent à d'importantes synthèses émanant de la méthode d'analyse initialement établie, elles sont toujours entièrement capables de cerner, ou de très précisément spécifier, l'ensemble des "qualités" humaines régissant notre présent Modèle Auto-Adaptable de l'Utilisateur.

En ce sens, nous retenons dans un premier temps les caractéristiques pouvant qualifier nos utilisateurs dans un contexte déjà particulier et complexe, à savoir celui de leurs interactions avec l'interface homme-machine d'un terminal de communication d'ALCATEL (**Figure 3.3.2.1**).

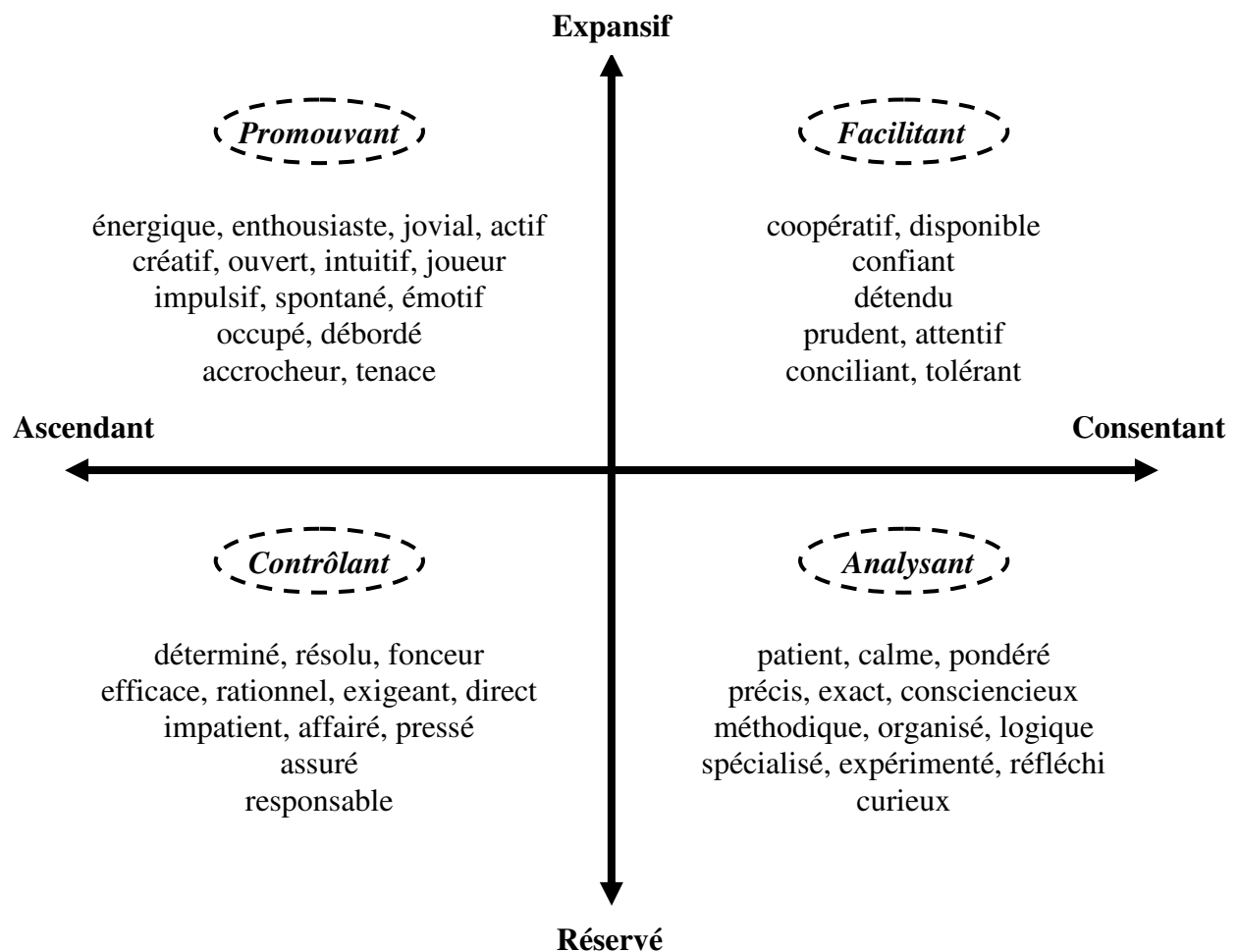


Figure 3.3.2.1 : *Caractéristiques générales de la méthode "Persona" adaptée dans un premier temps à l'interface homme-machine*

En second lieu, si nous conservons comme but l'adaptation de notre nouvelle méthode au fait que les utilisateurs visés seront qualifiés de "grand public", la présente analyse de leurs comportements respectifs doit être généralisée, ce au niveau notamment des caractéristiques des différents "styles sociaux". De plus, il nous apparaît aussi quelque peu subjectif, dénué de considérations pratiques, voire même dangereux au point de pouvoir entraîner une catégorisation des utilisateurs loin d'être réellement efficace, le fait que l'axe vertical du schéma général de cette méthode *Persona* traite de "l'émotivité". C'est en effet une notion encore trop difficilement perceptible de nos jours quant aux réactions de l'utilisateur devant la machine : durée de travail

ou rapidité de réponse à une interrogation posée par la machine ne saurait correctement caractériser nos différents utilisateurs. Beaucoup trop de facteurs peuvent en être la cause, tous plus compréhensibles et légitimes les uns que les autres.

C'est pourquoi, nous nous sommes attachés à modifier la prise en compte de cette émotivité, trop subjective dans une interface homme-machine, par celle plus concrète de l'expérience mise en avant par chaque utilisateur lors de ses manipulations respectives. Parmi les attributs qui peuvent être inclus dans un modèle d'utilisateur, le niveau de la connaissance, ou l'expertise dans un domaine précis, est effectivement d'importance fondamentale : il permet notamment l'utilisation d'un niveau adéquat de détail et d'un choix approprié du vocabulaire dans le dialogue avec l'utilisateur [Desmarais, Maluf & Liu, 1995]. Si l'aide fournie à l'utilisateur doit s'adapter à sa connaissance, elle correspond par exemple au texte standard de l'aide en ligne, ou à quelque chose de plus avancé tels que leçons et cours particuliers ou caractères indicateurs.

Cette "expérience", réellement palpable et matérialisable pour une machine, entre alors d'autant mieux dans nos processus d'interfaçage intelligent qu'elle se traduit et ressort très bien de l'étude des répétitions et autres habitudes de l'utilisateur au niveau de ses actions et séquences d'actions. Et ce sont justement ces dernières qui nous intéresseront avant tout lors des prochains chapitres. Ainsi, voici à venir (**Figure 3.3.2.2**) un ultime schéma de notre méthode d'analyse des comportements des utilisateurs de terminaux de communication dits "grand public", ce suite à l'insertion d'un "axe de la connaissance", plus utile et fiable pour notre modèle de l'utilisateur.

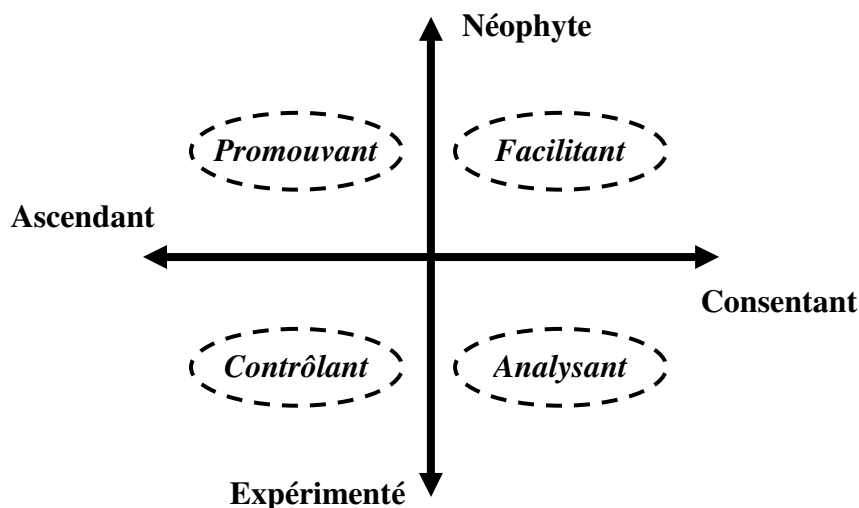


Figure 3.3.2.2 : *Caractéristiques principales de la méthode "Persona" avec la mise en œuvre de notre nouvel "axe de la connaissance"*

De toutes nos investigations se dégagent finalement des informations et caractéristiques tout-à-fait implémentables dans notre interface auto-adaptable puisqu'entièrement orientées, non seulement vers l'interaction et l'interfaçage homme-machine, mais aussi surtout vers tout travail à l'aide d'une machine automatisée, quelles que soient ses fonctionnalités et autres contraintes. Et, si les différents critères de comportement, au nombre de 48 et composant les schémas établis, s'avèrent être les qualificatifs les plus adaptés à la spécification du profil comportemental d'un utilisateur actuel de machine informatisée, ils n'en demeurent pas moins des termes relativement positifs, pour la plupart gratifiants et surtout tous dénués de critiques éventuelles vis-à-vis de l'utilisateur. Nous les avons en effet tous délibérément avancés dans le but de toujours conférer à notre interface auto-adaptable une image ou un "caractère" optimiste, agréable et convivial.

3.3.3. Conception d'un prototype « d'Interview Comportementale »

3.3.3.1. Introduction

Nous devons par suite nous interroger sur la nécessité et les moyens à mettre en œuvre afin que nos utilisateurs arrivent à débiter le mieux possible sur notre Interface Intelligente. Autrement présenté, quels peuvent être les fonctionnalités à avancer pour que l'utilisateur puisse être bien accueilli par la machine et ainsi sereinement s'atteler au travail qui est le sien ?...

Nombre de systèmes spécialisés proposent un procédé relatif à une sorte d'interview individuelle destiné à chacun des utilisateurs débutant un travail sur une application donnée. Ce questionnaire en ligne, puisque c'est bien de cela dont il s'agit en majorité, est chargé de fournir à la machine, le plus rapidement possible, des informations initiales au sujet des caractéristiques personnelles de l'utilisateur. Cela permet au modèle de cataloguer ses utilisateurs, afin d'éviter de proposer des aides déplacées ou inintéressantes à un utilisateur, ce dernier pouvant dans ces cas critiques se forger très rapidement une mauvaise opinion des méthodes ou des capacités d'adaptation de la machine à son égard. Dans cette optique, nous pouvons alors spécifier un tel module pouvant faire partie intégrante de notre interface intelligente des terminaux d'ALCATEL. Présentation initiale ou description individuelle préalable concernant chacun de nos utilisateurs, cela correspond en définitive à une intéressante prise de contact entre l'homme et la machine, entrée en matière donc indispensable à beaucoup de modèles d'utilisateur actuels.

Cependant, ces spécifications dédiées et propres à notre nouvelle Interface Intelligente resteront essentiellement dans le domaine du théorique. En effet, rappelons ici que nous désirons mettre en place un modèle d'interface aussi peu intrusif ou invasif pour l'utilisateur que ce soit possible. Or, nous allons constater que ce système d'interview, obligatoire pour la majorité des systèmes de modélisation d'utilisateur actuels, correspond aussi à une requête contraignante, voire difficile, pour l'utilisateur. De plus, cette dernière ne peut apparaître que dès le début du lancement de l'application en question, avant même un travail effectif de la part des utilisateurs.

Par conséquent, en tentant de remédier à ce problème, nous nous attacherons avant tout à conserver l'efficacité de notre nouveau Modèle Auto-Adaptable de l'Utilisateur. Nous verrons alors, dès le début du prochain chapitre, que nous nous sommes efforcés à initier les actions de l'utilisateur avec convivialité, mais tout en évitant de le mettre d'emblée sur la défensive face à la machine avec une telle interview... Nos modèles restent ainsi tous fidèles à nos objectifs initiaux, caractérisés par une discrétion maximale de l'interfaçage vis-à-vis de nos utilisateurs.

3.3.3.2. Première approche exhaustive

Voici donc selon quels schémas nous avons envisagé d'interviewer l'utilisateur sur ses caractéristiques comportementales lors de sa première approche du terminal de communication, en précisant que celle-ci reste unique pour chacun des utilisateurs, de même que toujours très succincte, agréable et conviviale. Demande est faite à l'utilisateur de simplement se situer parmi les divers critères comportementaux affichés (*Figure 3.3.3.2* à suivre). Ces données vont permettre à notre interface auto-adaptable d'établir la distinction entre les quatre classes de notre modèle d'utilisateur et de répertorier rapidement l'utilisateur dans l'une d'entre-elles. Les conséquences

au niveau du fonctionnement des nombreux scénarios et multiples fonctionnalités déjà proposés par nos procédés d'auto-adaptation en sont ensuite entièrement déduits.

<input type="checkbox"/> accrocheur	<input type="checkbox"/> débordé	<input type="checkbox"/> fonceur	<input type="checkbox"/> pondéré
<input type="checkbox"/> actif	<input type="checkbox"/> détendu	<input type="checkbox"/> impatient	<input type="checkbox"/> précis
<input type="checkbox"/> affairé	<input type="checkbox"/> déterminé	<input type="checkbox"/> impulsif	<input type="checkbox"/> pressé
<input type="checkbox"/> assuré	<input type="checkbox"/> direct	<input type="checkbox"/> intuitif	<input type="checkbox"/> prudent
<input type="checkbox"/> attentif	<input type="checkbox"/> disponible	<input type="checkbox"/> joueur	<input type="checkbox"/> rationnel
<input type="checkbox"/> calme	<input type="checkbox"/> efficace	<input type="checkbox"/> jovial	<input type="checkbox"/> réfléchi
<input type="checkbox"/> conciliant	<input type="checkbox"/> émotif	<input type="checkbox"/> logique	<input type="checkbox"/> résolu
<input type="checkbox"/> confiant	<input type="checkbox"/> énergique	<input type="checkbox"/> méthodique	<input type="checkbox"/> responsable
<input type="checkbox"/> consciencieux	<input type="checkbox"/> enthousiaste	<input type="checkbox"/> occupé	<input type="checkbox"/> spécialisé
<input type="checkbox"/> coopératif	<input type="checkbox"/> exact	<input type="checkbox"/> organisé	<input type="checkbox"/> spontané
<input type="checkbox"/> créatif	<input type="checkbox"/> exigeant	<input type="checkbox"/> ouvert	<input type="checkbox"/> tenace
<input type="checkbox"/> curieux	<input type="checkbox"/> expérimenté	<input type="checkbox"/> patient	<input type="checkbox"/> tolérant

Figure 3.3.3.2 : *Liste des qualités issues de la méthode "Persona" et composant l'approche initiale de notre prototype "d'Interview Comportementale"*

A l'utilisateur alors d'uniquement cocher à l'écran les cases qui correspondent le mieux à son comportement, et à notre interface de compter ensuite et d'établir enfin le profil ou style social dominant de cet individu. Notons que nous avons choisi de présenter les différents critères par ordre alphabétique, ce pour n'influencer en rien le choix de l'utilisateur en matière de comportements apparentés, voisins ou encore liés. Celui-ci pourrait, en effet, être tenté de rester trop complet et orienté quant à la définition préliminaire de son caractère comportemental.

Il subsiste toutefois une difficulté : le nombre des critères comportementaux attribués à chacun des styles sociaux n'est pas égal. En effet, nous décrivons le comportement promouvant par 15 adjectifs, celui d'analysant par 13 termes, le style contrôlant par 12 alors que le facilitant ne demande que 8 critères pour être entièrement spécifié. Nous nous proposons donc d'obtenir la dominante comportementale attribuable initialement à l'utilisateur, non pas par simple et inégale comparaison du nombre des différents critères choisis et cochés dans chaque style social, mais plutôt par confrontation de ces nombres respectifs divisés chacun par le nombre total de critères comportementaux chargés de décrire leurs styles sociaux correspondants. De cette manière, si nous définissons successivement par *P*, *F*, *C* et *A* le nombre de critères cochés par l'utilisateur dans les domaines des styles *Promouvant*, *Facilitant*, *Contrôlant* et *Analysant*, notre interface auto-adaptable se charge de comparer les valeurs suivantes : $P/15$, $F/8$, $C/12$ et $A/13$. Les valeurs ainsi obtenues apparaissent en outre tout-à-fait traduisibles en pourcentages, voire même en un rapide graphique, ces procédés étant toujours très représentatifs pour l'utilisateur.

Notons enfin qu'en cas "d'égalité" entre plusieurs de ces 4 nouvelles valeurs obtenues, nous pouvons envisager de placer l'utilisateur à la "frontière" des styles coïncidants. Mais, nous pouvons aussi demander à l'utilisateur de revoir ou de préciser son analyse, ou le laisser débiter ses manipulations en lui attribuant aléatoirement un style donné parmi ceux en concurrence. Cela s'avérerait sans grande conséquence : notre modèle d'utilisateur demeure évolutif et réorienterait de manière auto-adaptable cet utilisateur vers un des 3 autres styles comportementaux en cas d'inadéquation de son style initial, et ce au vu des actions et réactions futures de l'utilisateur.

3.3.3.3. Vers la recherche d'une approche moins intrusive

Une seconde version de notre "Interview Comportementale" nous paraît présenter aussi un intérêt pour l'utilisateur "grand public", ce concept étant plus court, donc plus abordable, voire de compréhension et de manipulation plus rapides et moins astreignantes (**Figure 3.3.3.3**). Nous constatons alors qu'un important avantage de cette nouvelle méthode réside dans le fait que les 4 styles sociaux sont décrits et analysés selon uniquement 5 catégories chacun. En effet, puisqu'elle s'effectue alors sur une échelle finie de 20 critères répartis équitablement entre les 4 tendances de comportements étudiées, la comparaison des cases cochées par l'utilisateur est ainsi rendue instantanée pour l'interface, mais également justifiée et validée par la méthode *Persona*.

<input type="checkbox"/> accrocheur, tenace	<input type="checkbox"/> créatif, intuitif, joueur, ouvert
<input type="checkbox"/> actif, énergique, enthousiaste, jovial	<input type="checkbox"/> curieux
<input type="checkbox"/> affairé, impatient, pressé	<input type="checkbox"/> débordé, occupé
<input type="checkbox"/> assuré	<input type="checkbox"/> détendu
<input type="checkbox"/> attentif, prudent	<input type="checkbox"/> déterminé, fonceur, résolu
<input type="checkbox"/> calme, patient, pondéré	<input type="checkbox"/> direct, efficace, exigeant, rationnel
<input type="checkbox"/> conciliant, tolérant	<input type="checkbox"/> émotif, impulsif, spontané
<input type="checkbox"/> confiant	<input type="checkbox"/> expérimenté, réfléchi, spécialisé
<input type="checkbox"/> consciencieux, exact, précis	<input type="checkbox"/> logique, méthodique, organisé
<input type="checkbox"/> coopératif, disponible	<input type="checkbox"/> responsable

Figure 3.3.3.3 : Liste contractée des qualités entraînant une seconde approche moins intrusive de notre "Interview Comportementale"

Le calcul de fractions mis en place tout-à-l'heure devient par conséquent quelque peu trivial, puisqu'il suffit maintenant de comptabiliser et de comparer le nombre de lignes cochées pour chaque style de comportement, avec la même démarche qu'auparavant en cas d'égalité.

En conclusion, si les trois premières caractéristiques purement interrogatives de notre modèle qualitatif et fonctionnel de l'utilisateur ne présentaient *a priori* que peu de difficultés (interview directe globalement suffisante pour "l'Identité", la "Profession" ainsi que la partie purement technique de la "Motivation", sous forme de formulaire par exemple), les quatre styles sociaux spécifiés peuvent très bien s'apparenter de manière évolutive à nos classes d'utilisateurs recherchés, que ce soit à propos de la "Motivation" de chaque utilisateur, de son "Caractère", de "l'Interaction" avec la machine ou même de ses propres caractéristiques de "Cognition"...

Tout ceci entériné, il demeure néanmoins intéressant de noter encore ici que la méthode *Persona* nous apprend, d'un nouveau point de vue, qu'il existe quatre autres éléments à considérer si l'on souhaite construire et instaurer un véritable climat de confiance au sein d'une interaction : l'ouverture, l'acceptation, l'authenticité et la fiabilité. Ainsi, afin de mettre à terme chacun de nos utilisateurs en confiance face à un terminal de communication "grand public", nous devons toujours nous assurer que notre interface intelligente reçoit des informations, puis fournit, apporte, voire explique celles-ci à l'utilisateur (notion d'ouverture), met l'utilisateur à l'aise et respecte ses motivations (acceptation), communique librement avec l'utilisateur sans rien lui cacher (authenticité), pour tenir finalement ses engagements en fonctionnant sans heurts (fiabilité).

3.4. Conclusion

Si ces analyses achèvent l'élaboration de notre Modèle Auto-Adaptable de l'Utilisateur, nous avons constaté que le rôle de l'homme ne cesse d'évoluer. En effet, nous ne le considérons plus comme un simple utilisateur manipulant une application, mais comme une personne possédant des connaissances qui lui sont propres et qu'elle utilise pour interagir [Stémart, 1997]. Le système se doit donc d'intégrer des mécanismes pour tenter de capturer et de comprendre ces connaissances, ce pour, en définitive, mieux maîtriser l'utilisateur. Ce dernier est alors impliqué dans le système dès la spécification initiale de nos modèles.

Par extension, l'Intelligence Artificielle ne saurait en rester à des considérations techniques relatives aux seuls ordinateurs : c'est l'homme qui est au centre de nos investigations actuelles et non la machine, et ce n'est pas l'homme comme modèle qui importe, mais l'homme comme destinataire. En fait, l'homme n'a créé les machines que pour éviter de commettre des erreurs, et pour aller plus vite aussi [Ganascia, 1993]. Quel que soit le critère retenu, l'homme est présent ; c'est lui qui anime les machines, c'est lui qui donne sens à leurs activités, c'est lui qui délègue ses pouvoirs. Ainsi, une machine ne peut prétendre à "l'intelligence" que si elle dispose d'un savoir analogue au savoir humain, donc de connaissances nombreuses et surtout variées.

Au travers de nos pages, est apparu le développement de plusieurs formes d'interaction dynamique comme le dialogue et la résolution de problèmes. Outil permettant de contrôler et d'augmenter les capacités de l'homme, l'ordinateur devient également un guide, un partenaire de moins en moins impersonnel. Il observe minutieusement les différents cheminements de son utilisateur, structure ses compétences et ses performances dans le but de le conseiller de manière aussi efficace que possible. Si le vecteur fort de la recherche sur les machines reste aujourd'hui d'adapter les interfaces aux besoins et aux capacités des hommes plutôt que de façonner le comportement humain en fonction des exigences de la technologie informatique, nous découvrons ici une très intéressante et nouvelle ubiquité de nos ordinateurs. En effet, ceux-ci deviennent réellement capables, non seulement de la gestion de l'environnement de travail et de la fourniture de documents, mais aussi de l'enseignement, d'une aide à la réalisation d'une tâche ainsi que de l'apport de points de vue dédiés à l'utilisateur.

La progression de telles capacités intelligentes est également loin d'être achevée, de même qu'elle ne cesse de nous surprendre, puisque nous venons notamment de découvrir qu'il est *a priori* envisageable de faire appel à une souris informatique d'un tout nouveau type et donnant à l'ordinateur la faculté de déterminer l'état émotionnel de son utilisateur [Sors, 2000]. Pour rester bref, cette souris, d'apparence ordinaire si ce n'est son revêtement en cuivre, est en effet capable d'évaluer la température du corps, les pulsations cardiaques et l'activité somatique de l'utilisateur. La machine peut ainsi détecter le stress ou la somnolence de ce dernier, et réagir en conséquence, par une stimulation visuelle et auditive entre-autres.

Mais nous devons tout-de-même grandement veiller à ne pas empiéter sur la vie privée de nos utilisateurs, notamment celle d'un client potentiel ou encore d'un salarié... Nous avons ainsi constamment cherché à faire face à une trop forte ingérence dans le travail quotidien de l'utilisateur, intrusion pouvant malheureusement être associée à trop de mauvaises intentions. A terme, celles-ci pourraient entraîner de bien fâcheuses conséquences, ne serait-ce que pour le bel attrait que traduisent aujourd'hui les machines pour chacun de nous.

S'il nous a ainsi été donné d'introduire beaucoup d'idées afin de préparer au mieux les implémentations et l'importante mise en œuvre de nos différentes méthodes d'auto-adaptation, nous nous devons maintenant de mettre en pratique les procédés selon lesquels celles-ci peuvent régir, voire diriger et commander, les nombreuses actions des scénarios intelligents caractérisant notre Assistant Individuel de Télécommunication.

En particulier, nous pouvons imaginer qu'au sein de l'Assistant Conversationnel Auto-Adaptable, le choix prioritaire des moyens de communication puisse s'effectuer en fonction de la classe attribuée à l'utilisateur (plutôt le téléphone pour un comportement contrôlant, et davantage l'*e-mail* pour un analysant ; ...). Dans le même ordre de raisonnement, la Classification Auto-Adaptable et sa réorganisation associée pourraient déterminer automatiquement leurs mots-clés en fonction des différentes catégories (plutôt les objectifs pour le style contrôlant, les concepts pour le promouvant, les méthodes pour l'analysant, ...).

Pour ce qui est enfin de la classification évolutive des utilisateurs et du passage éventuel d'un utilisateur d'une classe vers une autre, notre Interface Auto-Adaptable se charge alors aussi d'analyser les différentes actions de l'utilisateur pour en déduire des enseignements adéquats. Si un "Analysant" utilise par exemple de plus en plus l'outil téléphonique, son comportement se rapprochera de la frontière de son homologue "contrôlant" et l'utilisateur basculera peut-être dans ce nouveau style, après le dépassement d'un seuil pré-établi que nous programmerons de telle sorte qu'il soit paramétrable et modifiable selon les catégories des comportements humains face à nos machines.

Par conséquent, nous pouvons relever que notre Modèle Auto-Adaptable de l'Utilisateur pourra intervenir à pas moins de trois niveaux successifs dans le fonctionnement de l'interfaçage intelligent que nous implémenterons.

En effet, si nous omettons sans souci la détermination interactive de la classe initiale de l'utilisateur, et ce pour des raisons essentielles d'efficacité et d'indépendance indispensable au travail de celui-ci, notre modèle dédié entièrement à chacun de nos utilisateurs s'attachera tout d'abord à fournir une analyse complète ainsi que des enseignements riches résultant des diverses actions de l'utilisateur. Il deviendra par suite rapidement un maillon primordial du module de prédiction et d'anticipation des actions à traiter de l'utilisateur courant, en fonction de la classe comportementale de ce dernier avant tout...

Chapitre 4 :

Mécanismes d'Apprentissage Automatique

*Pouvoir extraire et rentabiliser
des formes significatives et la connaissance stratégique
de grands stocks d'information tenus électroniquement
est de plus en plus un défi
face aux mondes des affaires et de la science.*

[Michalski, Bratko & Kubat, 1998]

En réponse à ce constat bien réaliste, [Michalski, Bratko & Kubat, 1998] enchaîne aussitôt en soulignant qu'à notre époque, il est indispensable de chercher à développer des méthodes avangardistes capables de découvrir toutes sortes de formes remarquables et de construire un maximum de règles générales à partir de l'information contenue dans les diverses bases de données que l'homme exploite quotidiennement.

Ainsi, nous allons concrétiser chacune de nos modélisations auto-adaptables spécifiées.

Avec l'objectif audacieux d'établir des méthodes informatiques implémentant diverses formes d'apprentissage, en particulier des mécanismes capables d'induire des connaissances à partir d'exemples ou de vastes groupes de données, le champ de l'apprentissage automatique est aujourd'hui avant tout consacré à la mise en œuvre de processus d'apprentissage fondamentaux. Une partie significative de la recherche en apprentissage automatique est dédiée à des méthodes de développement pour déterminer des descriptions générales de concepts à partir d'exemples.

Dans cette optique, et parce que le développement de logiciels est aussi devenu un des principaux goulets d'étranglement de la technologie de l'informatique, l'idée d'introduire de la connaissance dans des ordinateurs semble de nos jours particulièrement attrayante. Une telle forme d'induction des connaissances est en effet souhaitable dans des problèmes qui manquent de solutions algorithmiques, peuvent également être mal définis, ou seulement décrits de manière informelle tel que cela apparaît justement être le cas pour nos propres modèles intelligents.

En outre, si nous avons démontré auparavant qu'il est à l'heure actuelle quasi impossible d'ignorer les diverses caractéristiques propres à chacun des utilisateur lors de l'intégration d'une interface homme-machine innovante, nous découvrons à ce niveau qu'il se dresse naturellement un second obstacle de taille pour l'élaboration et la conception d'une interface dite "intelligente". Si l'utilisation d'un environnement interactif s'accompagne incontestablement de la répétition de nombreuses actions [Ruvini, 2000], il s'agit alors de mettre en œuvre des procédés pour rendre l'interface, et *a fortiori* plus généralement la machine, auto-adaptable et donc intelligente.

Ceci exposé, nous verrons que les enjeux d'une telle problématique technologique deviennent très importants. Les systèmes abordés doivent effectivement disposer d'une certaine compréhension de tout le domaine considéré, c'est-à-dire d'un formalisme pour représenter les connaissances de ce domaine et de mécanismes pour les manipuler. De plus, les machines se doivent d'être capables, afin d'être toujours adaptables à part entière, de savoir résoudre tous les problèmes qui peuvent se poser [Bruillard, 1997]. Enfin, il ne faut pas non plus perdre de vue que la modélisation des mécanismes de compréhension, de raisonnement ou encore de génération de réponses sont tout aussi vitaux pour nos modèles d'interfaces intelligentes que le développement de méthodes et d'outils chargés ensuite du traitement des connaissances par acquisition, gestion, validation également, et même exploitation [Sorin & Colaitis, 1999].

C'est pourquoi, nous nous proposons maintenant de présenter en premier lieu différents mécanismes capables de jouer ce rôle primordial au sein de nos modèles auto-adaptables. Pour cela, nous commencerons par analyser plusieurs applications susceptibles d'être rendues auto-adaptables, extraites bien sûr de différents logiciels actuels d'ALCATEL. Puis, nous expliciterons l'intégration de nos apprentissages automatiques en fonction du profil de chaque utilisateur par de nouvelles représentations, plusieurs formalismes adaptés et la gestion de toutes les données disponibles, mais également par le détail des raisonnements et mécanismes d'inférence adéquats. En définitive, nous avons réussi à aller encore plus avant en cherchant à prédire le comportement de l'utilisateur afin de pouvoir anticiper, au travers d'une modélisation rigoureuse des fonctions "Tell Me" et "Do It" [Lieberman, 1997], une majorité de ses actions et de ses séquences d'actions.

4.1. Dégagement des concepts opératoires

4.1.1. Mode expérimental

Nous proposons de commencer cette étude par le détail descriptif des différentes étapes telles que nous les envisageons pour la prise en compte, l'analyse et l'insertion de notre procédé d'auto-adaptation au sein de nos interfaces "homme-machine" et "intelligentes" (**Figure 4.1.1**).

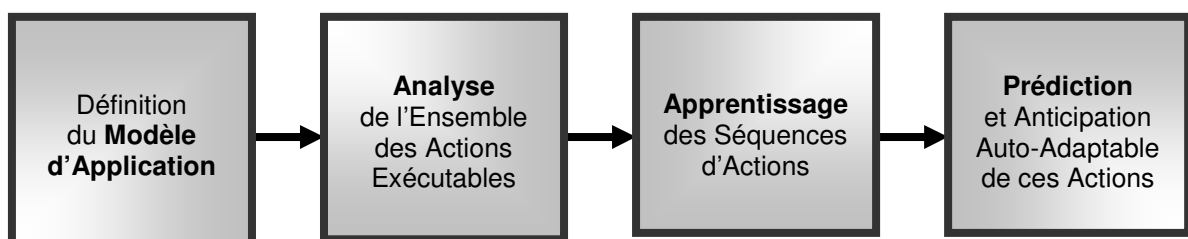


Figure 4.1.1 : Mode opératoire de mise en œuvre de notre concept d'auto-adaptation

Si nous avons pris connaissance de nos modèles d'application au cours du **Chapitre 1**, nous les avons ensuite tous caractérisés de manière très précise. Nous détaillons cette nouvelle phase de nos travaux au sein de la partie répertoriée **A1.3** dans l'**Annexe 1** dédiée à l'étude des terminaux de communication. En outre, chaque modèle d'application de nos fonctionnalités de

messagerie électronique, ainsi que tous les concepts auto-adaptables que nous souhaitons leur associer, sont exploités et mis en valeur grâce à une succession de trois phases expérimentales.

Dans un premier temps, l'analyse de l'ensemble des actions exécutables, et ce en temps réel, nous entraîne vers la découverte et le recensement de toutes les informations relatives à l'exécution du modèle d'application. Suivie de l'extraction et du stockage des diverses données redondantes, cette étape initie l'apprentissage des séquences d'actions remarquables. Cela nous permet d'aboutir à la configuration automatique d'un outil interactif et surtout progressif, chargé de l'exécution auto-adaptable des manipulations courantes de l'utilisateur. Nous entendons plus précisément, à ce niveau, la prédiction et l'anticipation, toutes deux auto-adaptables, opérant à terme sur toutes les séquences d'actions fréquemment réitérées par chacun de nos utilisateurs.

Ainsi, en fonction du profil de chaque utilisateur, nous proposons d'aborder en détail la conception de nos représentations et formalismes adaptés à ces modèles d'application. Puis, nous verrons qu'une gestion rigoureuse et efficace de toutes les données rendues disponibles par nos modèles d'application nous permettra de programmer un ensemble conséquent de raisonnements et de mécanismes d'inférence chargés de rendre notre interface intelligente.

4.1.2. Introduction de la théorie des séquencements d'actions

Principalement relayée ces dernières années par les travaux de Craig Nevill-Manning, la notion des séquencements, que nous dédions aux diverses actions successives réalisées par un utilisateur donné, permet essentiellement de développer toutes sortes de techniques révélatrices d'une structure intéressante, à base de séquences, dans un ensemble réel.

Robustes ainsi par production d'un résultat dans tout domaine envisageable, mais aussi efficaces puisque capables d'analyser une quantité de données très importante (secteur de prédilection de l'interaction homme-machine qui plus est), ces méthodes se caractérisent en outre par leur généricité et leur utilité aujourd'hui indéniables. En effet, applicables à une large variété de séquences et libres de trouver des structures surprenantes, voire non-anticipées par le concepteur, ces techniques brillent également par la mise en valeur de productions d'explications compréhensibles et visuelles, de même que par la constante possibilité d'identifier maintes unités morphologiques et autres expressions significatives.

Tout ceci avancé, c'est aussi au travers de leurs capacités à générer des mécanismes d'inférence performants, par reconnaissance de répétitions ou de similarités notamment, que ces procédés ont gagné leur notoriété. Nous proposons d'ailleurs, grâce à la **Figure 4.1.2** à suivre, de découvrir quelques exemples de grammaires récursives faisant état de ces avantages, en précisant encore que c'est avant tout dans cette direction des répétitions d'actions identiques que nous nous orienterons dans le cadre de nos propres travaux de recherche. Nous développerons aussi alors différentes définitions au sujet des actions et des séquences d'actions que nous serons amenés à manipuler et à traiter.

Par suite, nos présentes lignes ne peuvent également occulter que le premier chapitre de ce mémoire, et tout particulièrement les études successives menées au travers de nos états de l'art de la partie **1.2.4**, nous ont amenés à prendre connaissance, dans le cadre de la réalisation des interfaces intelligentes, d'encore bien d'autres qualités propres aux séquencements d'actions.

	Séquences	Grammaires		Séquences	Grammaires
a	$S \rightarrow \text{abcdbc}$	$S \rightarrow \text{aAdA}$ $A \rightarrow \text{bc}$	b	$S \rightarrow \text{abcdbcabcdbc}$	$S \rightarrow \text{AA}$ $A \rightarrow \text{aBdB}$ $B \rightarrow \text{bc}$
c	$S \rightarrow \text{abcdbcabcdbc}$	$S \rightarrow \text{AA}$ $A \rightarrow \text{abcdbc}$	d	$S \rightarrow \text{aabaaab}$	$S \rightarrow \text{AaA}$ $A \rightarrow \text{aab}$
		$S \rightarrow \text{CC}$ $A \rightarrow \text{bc}$ $B \rightarrow \text{aA}$ $C \rightarrow \text{BdA}$			$S \rightarrow \text{AbAab}$ $A \rightarrow \text{aa}$

Figure 4.1.2 : Exemples de séquences et de grammaires les reproduisant [Nevill-Manning & Witten, 1997] :

- (a) séquence avec une seule répétition ;
- (b) séquence avec une répétition imbriquée ;
- (c) / (d) deux grammaires différentes respectivement proposées pour une même séquence

En effet, nous pouvons alors déjà nous remémorer les diverses qualités, et notamment ergonomiques, caractérisant les interfaces chargées d'assister l'opérateur humain au niveau de la supervision, de la détection et de la localisation de défauts éventuels émanant des complexes procédés automatisés de l'industrie [Lambert, Riera & al., 1998]. Présentant de très nombreuses données en termes plus synthétiques et symboliques qu'une simple liste numérique de valeurs ou de dépassements de seuils, un tel système d'aide à la décision en temps réel propose à l'homme un diagnostic lui permettant de déterminer la meilleure séquence d'actions à appliquer.

D'un autre côté, nous avons également pu grandement apprécier les différents avantages des "Cartes Adaptatives de Kohonen" entraînant une exploration toujours ajustable d'un réseau de formes, de documents ou donc d'actions remarquables [Kohonen, Kaski & Lappalainen, 1997]. De plus, ce système met encore en évidence une préparation à l'extraction de toutes ces dernières caractéristiques en permettant le décodage de chaque classification correspondante. En découle ainsi un apprentissage adaptatif des diverses classes relevant pleinement des séquences d'actions.

Dans un second temps, et dans le but de compléter plus avant notre présente étude sur les diverses qualités distinguant les séquencements d'actions, [Ardissono & Torasso, 2000] insiste sur les attraits d'une formation aisée, et toujours en temps réel, d'historiques locaux répertoriant toutes les séquences continues d'actions propres à un utilisateur, sans introduire de quelconques changements dans l'attention de celui-ci. [Bruillard, 1997] nous rappelle alors également que ce dernier procédé ne peut, ensuite, se passer d'une autre méthode de séquencement, associée elle, au sein de l'historique, à la reconnaissance de la structure de toute séquence observée, et ce afin de discerner bien entendu le but courant poursuivi par nos utilisateurs. Il se peut d'ailleurs qu'il y ait, à ce niveau, plusieurs buts, voire même aucun.

Nous aboutissons en conséquence aux activités typiques rencontrées dans la conception et l'implémentation de toute interface intelligente, ainsi que dans le cadre d'une modélisation d'utilisateurs [Ye, 1997]. Il s'agit en effet ainsi plus précisément de reconnaissance de formes pour l'identification de séquences d'actions répétées par l'utilisateur, de classification et d'association de formes par catégories d'actions suivant les caractéristiques de l'utilisateur, ce pour révéler entre-autres les relations (de groupage par exemple) décrivant les actions ou séquences d'actions réalisées par l'utilisateur. En outre, nous constatons que le stockage et la recherche de formes tiennent aussi ici une place indispensable pour nos mécanismes de séquencements d'actions

puisque'ils permettent la mémorisation courante et le rappel de formes de l'utilisateur manipulant des répliques partielles ou bruitées. De même, la prédiction de formes [Ruvini, 2000], que nous étudierons très en détail en 4.5.3, nous ouvre les portes de l'automatisation des manipulations et de chaque séquence d'actions émanant de l'utilisateur.

A ce propos, nous cautionnons simultanément une autre qualité essentielle des concepts de séquencements d'actions, aptitude que nous ne manquerons pas de grandement développer dans nos différentes implémentations à venir. En effet, les prédictions de tâches après exécutions répétées de séquences d'actions font apparaître des conditions suffisamment riches pour garantir une méthode correcte et automatique de construction de modules appropriés et de règles individuelles de contrôle pour chacune des manipulations d'utilisateurs [Baral & Son, 1998]. Nous verrons que cette approche peut être ensuite très bien étendue aux théories et modules spécialisés dans les actions intuitives, les anticipations auto-adaptables d'actions ainsi que dans le cas d'un grand nombre de simplifications de séquences d'actions.

Afin de spécifier à présent globalement la théorie abordée, et dans un souci de concision, nous pouvons retenir que le séquencement d'actions se définit en premier lieu par la découverte de régularités significatives dans une séquence de symboles ou d'événements discrets.

« Découvrir automatiquement et rendre explicites des structures issues de séquences » [Nevill-Manning, 1996], puis intégrer et traiter les informations apprises des nouvelles séquences ainsi extraites au sein de tous nos modèles, pour entraîner enfin tous les résultats obtenus vers une prédiction en temps réel et une anticipation auto-adaptable des actions effectuées de manière préférentielle par chacun des utilisateurs pris en charge, résume alors bien la ligne directrice principale des étapes que nous nous sommes fixés tout au long de cette Thèse. Plutôt que d'examiner la signification de symboles individuels dans une séquence, il apparaît plus efficace, et "intelligent", de détecter une structure suivant le fait que certaines combinaisons de symboles se reproduisent. Dans le même ordre d'idée, et pour des répétitions exactes, les séquences mettent aussi souvent en évidence une structure de branchement ou de boucle, parfois à grande échelle, qui peut être inférée, mais aussi prédite et anticipée.

A noter aussi ici la faisabilité de la détection de ces structures basées sur des séquences. De tels systèmes peuvent en effet être détectés, efficacement et incrémentalement, sans référence à la signification des divers symboles individuels. Une fois détectés, ils peuvent être utilisés pour expliquer et compresser les séquences respectives et initiales. Cet apprentissage d'une structure à partir d'une seule séquence, ainsi que l'évaluation de cette structure en termes de plausibilité et de densité, peuvent alors aboutir, à l'image de nos travaux dévoilés plus loin, aux intéressantes constructions d'automates et modélisations inhérentes de séquences particulières ou répétitives.

En poursuivant nos recherches, nous remarquons qu'il s'avère enfin également possible de construire incrémentalement une complète représentation hiérarchique de chaque séquence originale obtenue. En effet, nous pouvons relever à ce niveau que beaucoup de séquences de symboles discrets présentent aujourd'hui une structure hiérarchique naturelle [Nevill-Manning & Witten, 1997]. Dans le cas précis des enregistrements du comportement d'une interface homme-machine et des tâches exécutées par un utilisateur, [Nevill-Manning & Witten, 1998] nous apprend cependant que cela demeure encore très difficile d'extraire une véritable structure remarquable, et hiérarchique ou non, du travail courant de chaque utilisateur.

En conclusion, si les idées mises en avant à ce niveau apparaissent déjà bien efficaces pour identifier des répétitions précises, et selon un temps linéaire avec la longueur de la séquence courante, il demeure nécessaire d'également relever ici chacune des possibilités de réalisations robustes, génériques et ergonomiques, ainsi que toutes les capacités d'inférences automatiques et auto-adaptables, applicables à ces structures significatives extraites des séquencements d'actions.

En outre, plusieurs aspects s'avérant indispensables dans le cadre d'une modélisation d'interface intelligente n'ont, à ce jour, pas encore été traités :

- les fréquences numériques supérieures à la "réplique" de base définissant une répétition,
- la similitude métrique entre les symboles individuels qui composent une même séquence,
- toutes les structures imbriquées ou récursives pouvant caractériser chaque séquence dédiée à l'interfaçage homme-machine.

Ainsi, l'ensemble des très nombreuses capacités que nous avons pu mettre en avant au sujet des séquencements d'actions, à commencer par la production d'explications ergonomiques, l'extraction et la prédiction aussi de séquences d'actions significatives, nous convainc que ces mécanismes pourront grandement nous soutenir au cours des travaux à venir relevant notamment de l'apprentissage automatique. A l'image finalement du propre séquencement du temps réel en "étapes successives et homogènes" [Yunfei, Renshou & al., 1997], nous allons donc nous attacher, par suite, à valoriser et à appliquer cette synthèse de méthodes à nos procédés auto-adaptables.

4.2. Extraction des actions de l'utilisateur

4.2.1. Représentation des connaissances correspondant aux séquences

Concernant cette seconde phase de notre mode opératoire, nous précisons d'emblée que cette étape d'observation se doit d'être "brute" et exhaustive. En effet, nous devons récupérer le maximum d'informations ou types d'informations disponibles (données émanant de l'utilisateur, de la machine, du réseau considéré, ...) pour pouvoir construire ensuite, en toute connaissance de cause et de façon efficace, un ou plusieurs procédés d'apprentissage, adaptables non seulement à l'utilisateur, mais aussi éventuellement à l'interface ou à la machine elle-même. Il nous semble également utile de bien prendre en compte le phénomène de la distinction de perception qu'il peut exister entre les différentes séquences d'actions effectuées par l'utilisateur et les opérations propres à la machine qui en sont obligatoirement déduites.

Du point de vue matériel, nous avons examiné la toute dernière version de test de ce logiciel du *WebTouch*, non sans avoir auparavant été tenus d'installer entièrement, sur *Solaris 1.2*, le *SUN Java Development Kit* (version 1.1.7A) associé à cette émulation. Nous n'omettons pas non plus l'installation et notre étude de l'utilisation du logiciel *SNiFF+* (version 3.1), outil de gestion du code-source de tout produit informatique travaillé en partage au sein de chaque équipe de recherche d'*ALCATEL*.

Ces manipulations initiales nous ont permis de rapatrier toutes les données de ce logiciel nécessaires à nos travaux. Et cela comprend l'ensemble du code-source exécutant l'application "*e-mail*" existante afin de permettre le lancement de nos extractions et observations souhaitées (à l'aide notamment si besoin de "drapeaux" de passage d'étapes ou autres "rapporteurs" d'actions effectuées), au sujet des nombreuses séquences manipulées, et par suite répétées, de l'utilisateur.

Mais commençons par nous pencher sur l'observation et la représentation des séquences d'actions engendrées par l'utilisateur. Si nous nous basons une nouvelle fois sur les opérateurs de *LOTOS* étudiés antérieurement, nous commenterons nos travaux en détail après la présentation de

la **Figure 4.2.1.1** (ci-contre) et de la **Figure 4.2.1.2** (sur la page suivante). Ces schémas techniques apportent alors une base complète à nos recherches au travers d'un graphe de représentation des diverses actions exécutable sur une application de courrier électronique émanant d'ALCATEL.


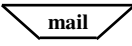
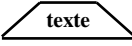

	Descriptif d'une action réalisable par l'activation d'un bouton de l'interface.
	Descriptif d'une action caractérisée par la désignation spatiale d'un "mail".
	Descriptif d'une action spécifiant l'entrée au clavier d'une donnée textuelle.
	Flèches décrivant les passages ou transitions entre actions de l'application.
;	Opérateur LOTOS du séquençement (séparateur d'actions successives).
□	Opérateur LOTOS du choix (séparateur d'alternatives, disjonction d'actions).

Figure 4.2.1.1 : Synthèses graphiques de la représentation des actions exécutable

4.2.2. Commentaires et analyse des informations extraites

Au sujet de l'arbre hiérarchique que nous avons construit à la **Figure 4.2.1.2**, plusieurs avantages, mais aussi quelques contraintes non négligeables, méritent à ce niveau d'être précisés. Tout d'abord, il apparaît que l'automate correspond *a priori* à nos attentes et besoins en matière de vision des possibilités d'interaction homme-machine correspondant à l'application "e-mail" étudiée. Il s'avère en effet complet quant à la description des différentes actions disponibles et réalisables par chaque utilisateur : il prend en considération toutes les activations possibles au niveau des multiples boutons de l'interface et intègre également les autres apports d'informations interactives ou de données textuelles émanant de l'utilisateur (désignation spatiale d'un "mail" par la souris, renseignement des adresses des correspondants d'un "mail", écriture exhaustive du sujet et du corps d'un message, adjonction éventuelle de fichiers attachés, ...).

Nous avons donc cerné, sur un automate hiérarchique, l'ensemble complet des diverses interactions homme-machine que peut rencontrer l'utilisateur en appelant l'application "e-mail" d'ALCATEL. Pas moins de 23 actions distinctes sont alors répertoriées et agencées par cet outil suivant les relations qui caractérisent chacune d'entre-elles vis-à-vis de toutes les autres actions. Cependant, à ces 23 actions, se rajoutent en fait encore 2 autres, d'autres boutons pour être précis que l'on peut qualifier d'actions implicites puisqu'il s'agit du bouton "retour" et du bouton "page d'accueil". La fonctionnalité "retour" permet un renvoi à l'action parente de l'action courante, voire bien entendu plusieurs renvois successifs dans le cas de plusieurs activations de ce bouton. Quant à l'accès à tout moment à la "page d'accueil" principale du logiciel, il faut noter que cela signifie inévitablement l'arrêt immédiat de la session courante de l'application "e-mail".

L'application "e-mail", encore intitulée "courrier", ainsi que, par extension, l'automate hiérarchique, proposent en définitive 25 actions différentes à l'utilisateur. Il est alors également essentiel de relever que ces 25 actions se composent plus précisément de 3 catégories distinctes : les 23 activations de boutons, dont les 2 boutons des actions constamment utilisables que sont le "retour" et l'accès à la "page d'accueil" principale de l'application, la désignation spatiale de tout "mail" à l'aide d'un simple "clic" de la souris, et enfin l'écriture ou l'entrée au clavier de chaque donnée textuelle, cette dernière étant par exemple amenée à être intégrée à un "mail" à envoyer.

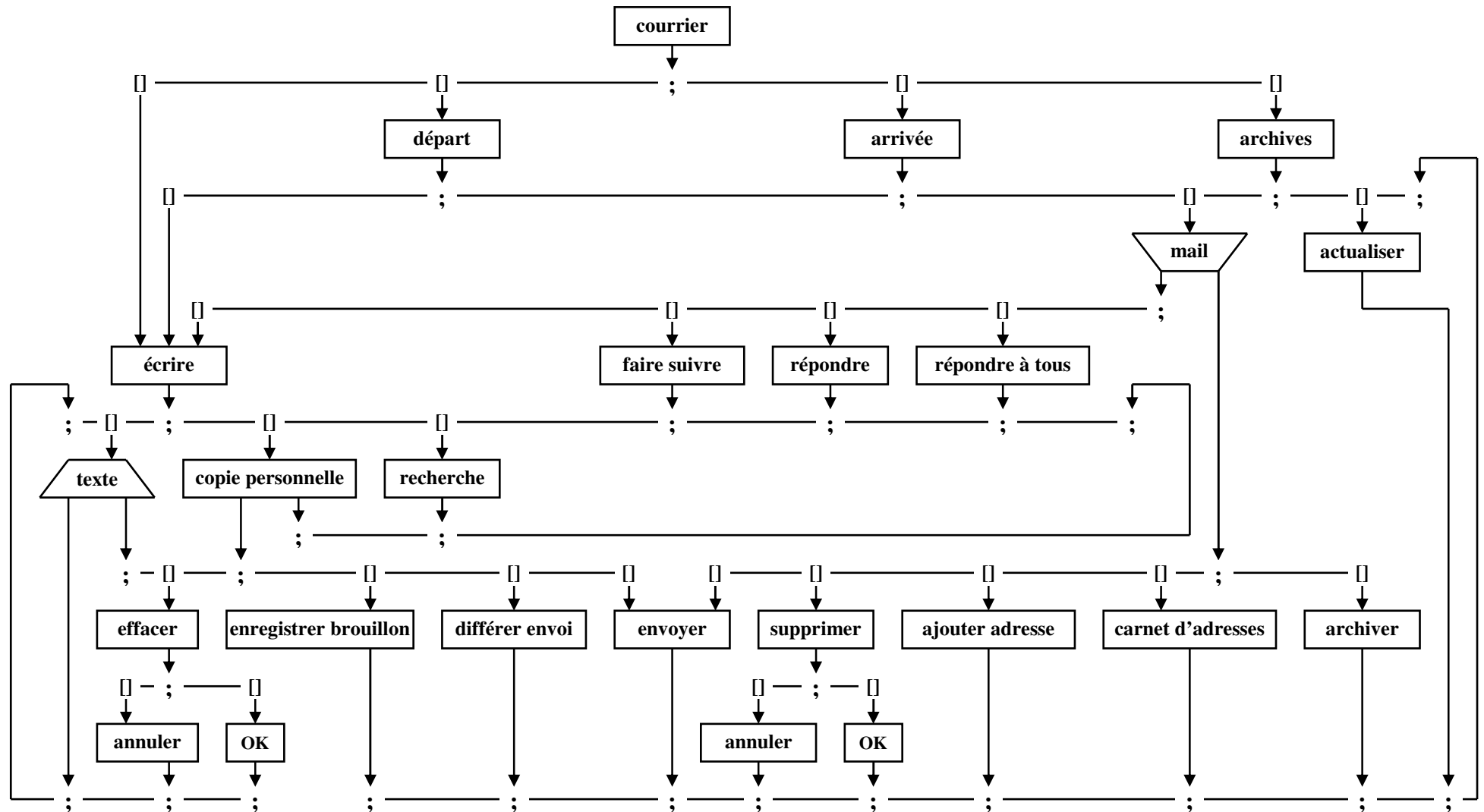


Figure 4.2.1.2 : Automate hiérarchique de toutes les manipulations réalisables par chaque utilisateur sur l'application "e-mail"

4.2.3. Représentations canoniques des actions et séquences d'actions

La "représentation canonique" d'une action correspond à une structure théorique capable de définir toute manipulation quelconque effectuée par l'utilisateur au niveau de l'interface de l'application "e-mail". Ainsi, en fonction des paramètres apparus (cf. **Figure 4.2.1.2** précédente) et leurs multiples possibilités d'instantiation, nous obtenons le modèle suivant (**Définition 4.2.3.1**). Nous appliquerons cette représentation à quelques exemples "concrets" de séquences d'actions relativement simples, mais cependant couramment rencontrées au cours des maintes exécutions de l'application "e-mail" (**Tableau 4.2.3** sur la page suivante).

Par extension, nous aboutissons également à l'énoncé d'une nouvelle définition issue de nos travaux de recherche, à savoir cette fois-ci la définition canonique d'une séquence d'actions (**Définition 4.2.3.2**).

Définition 4.2.3.1 : Représentation canonique d'une action

Action \equiv *symbole ouvrant* **descriptif de l'action** *symbole fermant*

où

- les deux symboles de la représentation se chargent d'emblée de distinguer le type de la manipulation effectuée par l'utilisateur, selon qu'il s'agisse :
 - * de l'activation d'un bouton de l'interface (traits verticaux : | ... |),
 - * de la désignation spatiale d'un "mail" (traits inclinés vers le bas : \ ... /),
 - * ou encore de l'entrée au clavier d'une donnée textuelle (traits orientés vers le haut : / ... \).
- le descriptif de l'action, au centre de l'expression, fournit ensuite :
 - * l'intitulé du bouton si l'on est dans le premier cas,
 - * l'identifiant exact du "mail" en cas de désignation spatiale de celui-ci (à savoir ici le nom du fichier sous lequel le "mail" a été sauvegardé),
 - * la chaîne entière de caractères entrée au clavier par l'utilisateur.

Définition 4.2.3.2 : Représentation canonique d'une séquence d'actions

Séquence d'actions \equiv *action 1 ; action 2 ; ... ; action n*

où

- chaque **action** se doit d'être représentée selon la **Définition 4.2.3.1** ($n \in \mathbf{N}$).
- ; \leftrightarrow opérateur *LOTOS* du séquençement (séparateur d'actions successives).
- **action 1** \leftrightarrow action de tête (début, ouverture de la séquence d'actions).
- **action n** \leftrightarrow action de queue (fin, fermeture de la séquence d'actions).
- **action 2 ; ... ; action n-1** \leftrightarrow succession des actions internes (parcours de la séquence d'actions).

Tableau 4.2.3 : Exemples de séquences d'actions fréquentes

Intention de l'utilisateur	Séquence(s) d'actions pouvant répondre à l'intention courante de l'utilisateur
Ouverture d'une session "e-mail"	courrier
Envoyer un nouveau "mail"	courrier ; écrire ; recherche ; /jjk@dpt-info.u-strasbg.fr\ ; /jc@ms.alcatel.fr\ ; recherche ; /cge@sxbsf.alcatel.fr\ /Rapport\ ; /Bonjour à tous.\ ; copie personnelle ; envoyer
	courrier ; archives ; écrire ; /P'tit mail de confirmation :\ ; /Mon Séminaire aura bien lieu demain 21 Juillet.\ ; envoyer
	courrier ; arrivéel ; actualiser ; écrire ; recherche ; /pl\ ; /dds\ ; enregistrer brouillon ; \ml251106/ ; envoyer
Différer l'envoi d'un "mail"	courrier ; écrire ; /all\ ; /Question : to bug or not to bug ?\ ; /Bonne Année !!!!\ ; copie personnelle ; différer envoi
Envoyer un "mail" en attente	courrier ; départ ; \bg030101/ ; envoyer
Envoyer tous les "mails" en attente	courrier ; départ ; actualiser
Lecture d'un "mail" arrivé	courrier ; arrivéel ; \bg030102/
Répondre à un "mail" reçu	courrier ; arrivéel ; actualiser ; \repedem16/ ; répondre ; /Re-bonjour !\ ; /Message bien reçu...\ ; envoyer
Archiver un "mail"	courrier ; arrivéel ; \bg030102/ ; archiver
Effacer ou supprimer un "mail"	courrier ; écrire ; /jc@sxbsf.alcatel.fr\ ; effacer ; OK
	courrier ; arrivéel ; \repedem16/ ; supprimer ; OK
Renvoi à l'action parente	retour
Arrêt d'une session "e-mail"	page d'accueil

Ceci avancé, nous commençons par ailleurs à déterminer et assimiler ces exécutions au travers des notions d'actions ou séquences d'actions estampillées "fréquentes". Ce terme signifie en effet explicitement que les actions ou séquences les plus remarquables demeurent celles dont le nombre des itérations courantes apparaît important, de même qu'évolue de façon suffisamment sensible pour intéresser, mais aussi justifier, une application de nos systèmes auto-adaptables.

Par simple déduction de nos premières définitions, nous pouvons alors encore proposer à suivre deux nouvelles spécifications, peut-être même un peu plus explicites dans le contexte de "cheminement" lié à notre automate hiérarchique (*Définition 4.2.3.3* et *Définition 4.2.3.4*).

Définition 4.2.3.3 : *Seconde représentation des séquences d'actions*

Séquence d'actions \equiv

action de tête ; succession d'actions internes ; action de queue

où

- chaque *action* se doit d'être représentée selon la **Définition 4.2.3.1** ($n \in \mathbf{N}$).
- ; \leftrightarrow opérateur *LOTOS* du séquençement (séparateur d'actions successives).

Définition 4.2.3.4 : *Troisième représentation des séquences d'actions*

Séquence d'actions \equiv *début ; parcours ; fin*

où

- ; \leftrightarrow opérateur *LOTOS* du séquençement (séparateur d'actions successives).
- chaque *action* se doit d'être représentée selon la **Définition 4.2.3.1** ($n \in \mathbf{N}$).

Précisons enfin, pour clore l'établissement de ces différentes définitions indispensables à la bonne compréhension des structures de données que nous avons choisi de manipuler, qu'une liste des actions, ouvrant ou fermant chacune de nos séquences d'actions, sera encore apportée en rapport avec la gestion des informations d'utilisateurs, notamment au niveau du **Tableau 4.4.1**. Nous pouvons également relever ici le lien direct et primordial qui apparaît entre nos scénarios d'assistantats intelligents et tous les mécanismes d'apprentissage automatique : les fondements théoriques nécessaires aux développements de nos concepts auto-adaptables sont à présent posés.

4.3. Traitement temporel et gestion des données

4.3.1. Introduction

Après les détails de l'analyse et la compréhension de notre automate de représentation des séquences d'actions réalisables par l'utilisateur sur les interfaces des modèles d'application, nous poursuivons nos recherches afin d'obtenir un maximum d'efficacité concernant les données extraites, ce en intégrant à nos formalismes la nécessaire notion de temps. Cette dernière devient en effet primordiale puisque nous souhaitons élaborer par la suite un mécanisme d'apprentissage automatique à partir des informations émanant de l'exécution des séquences d'actions. Les informations temporelles apparaissent alors réellement indispensables, notamment du fait que tout apprentissage demande obligatoirement une certaine chronologie en ce qui concernent les objets qu'il est amené à traiter.

En conséquence, nous nous proposons d'adjoindre la considération du temps, de la date, du "moment d'exécution", à la représentation de chacune des actions réalisées par l'utilisateur.

4.3.2. Définition de la base de données des actions manipulées

Nous conservons le descriptif complet de chaque action rencontrée par chacun de nos multiples utilisateurs, et ce lors de leurs différentes et respectives utilisations (**Tableau 4.3.2**). A noter alors que notre base de données ainsi constituée est prévue, non seulement pour traiter les actions de l'application "e-mail" étudiée par nos présents travaux, mais également pour prendre en compte toutes celles des autres applications proposées par chaque logiciel courant manipulé.

Tableau 4.3.2 : *Extrait de la base de données des actions*

Numéro d'identification	Login de l'utilisateur	Intitulé de l'application	Exécution		Descriptif
			Date	Heure	
8898	osa	courrier	12/10/1999	08 : 06 : 16	courrier
15578	jjk	courrier	16/11/1999	08 : 14 : 58	écrire
15654	jc	courrier	16/11/1999	08 : 21 : 26	actualiser
15722	jjk	courrier	16/11/1999	08 : 37 : 33	/Bonjour Olivier !\
17416	jjk	courrier	16/11/1999	09 : 40 : 04	envoyer
99254	cge	courrier	25/11/1999	14 : 32 : 07	\ng241110/
127916	osa	courrier	26/11/1999	13 : 19 : 48	courrier
127917	osa	courrier	26/11/1999	13 : 19 : 53	arrivé
127998	osa	courrier	26/11/1999	13 : 41 : 09	\dr261102/

Le numéro d'identification, unique pour chacune des actions effectuées, nous est apparu comme étant un moyen supplémentaire, mais relativement simple et pratique, permettant une distinction rapide et efficace des actions recensées. Ce procédé, correspondant par définition à la clé primaire de la base de données, nous évite ainsi la lourde et constante manipulation simultanée de la date et de l'heure d'exécution des différentes actions, avec en outre les différents intitulés d'applications correspondant à celles-ci (il nous faut effectivement considérer le cas où plusieurs applications sont lancées en même temps).

Ces dates et heures d'exécution respectives pourraient aussi avantageusement participer au permanent recueil de renseignements que nous souhaitons obtenir pour notre Modèle Auto-Adaptable de l'Utilisateur sur les humeurs de l'individu, ses habitudes, sa réactivité au stress, sa vitesse de travail, son attention, son habileté, son efficacité, sa compréhension enfin ou encore sa disponibilité du moment.

Les trois derniers champs du **Tableau 4.3.2**, par exemple, traduisent alors bien, pour un même utilisateur travaillant avec l'application "e-mail", une séquence complète de trois actions permettant la lecture d'un "mail" arrivé (se reporter pour plus de précisions au **Tableau 4.2.3**). Quant aux actions numérotées **15578**, **15722** et **17416**, elles correspondent ensemble à l'écriture et à l'envoi d'un nouveau message : bien que non consécutives dans la numérotation permettant de les identifier, elles le sont pourtant pour un seul et même utilisateur caractérisé (en effet, dans ce cas, plusieurs utilisateurs travaillent simultanément).

A noter par conséquent la nécessité déjà de respecter plusieurs contraintes quant à la considération et à l'analyse précise des séquencements d'actions répertoriées au travers de notre base de données. En effet, pour une même séquence, c'est-à-dire pour déterminer l'ensemble des actions qui appartiennent à un même séquencement, tous les identifiants d'actions doivent être croissants, voire strictement consécutifs dans le cas où une seule application est exécutée pour un unique utilisateur présent. Bien sûr, ces contraintes, et d'autres, seront très précisément détaillées au cours de nos recherches à venir sur l'apprentissage automatique des séquences d'actions.

Par contre, il n'est pas nécessaire que les actions d'une même séquence présentent une même date, ni des heures strictement chronologiques. En effet, un passage, en plein travail d'un utilisateur, du cap de minuit reste possible, de même que l'arrêt du travail d'un utilisateur sur une application (en soirée par exemple), et une reprise ultérieure par cet utilisateur du cours logique de son travail (le lendemain, si l'on se réfère au même ordre d'idées).

En toute dernière partie de l'**Annexe 1**, le **Tableau A1.4.1** nous propose alors finalement une liste plus importante d'exemples tels que nous serons amenés à en manipuler au cours de nos prochaines expérimentations chargées de nous aider à caractériser un nouvel apprentissage automatique des différents séquencements d'actions. Aussi, nous poursuivons nos investigations par une analyse précise ainsi que l'élaboration de nouveaux mécanismes indispensables à une bonne compréhension, et donc à un travail ultérieur efficace à partir de cette base d'actions.

4.3.3. Traduction de la base de données pour l'extraction des répétitions

4.3.3.1. Récapitulatif par utilisateur

Il s'agit maintenant, afin de manipuler et de traiter les nombreuses actions et séquences d'actions ainsi mémorisées, d'expliquer les procédés selon lesquels nous pouvons extraire, à partir de notre base de données, les répétitions des actions listées ainsi que les redondances des différentes séquences pouvant en être déduites. Voici, en premier lieu, une sorte de contraction du tableau présenté précédemment et correspondant plus précisément à une vue restreinte des séquences émanant de chaque utilisateur (**Tableau 4.3.3.1** à suivre). Nous découvrons alors aussi une mise en évidence des répétitions pouvant être constatées lors de l'utilisation des applications.

4.3.3.2. Décompositions de séquences d'actions

En effet, il nous faut remarquer, au sujet de la constitution de ce **Tableau 4.3.3.1**, que nous nous sommes tenus à répertorier, dans un souci de clarté et de compréhension vis-à-vis du grand **Tableau A1.4.1**, une seule séquence d'actions par utilisateur et par date d'exécution aussi. Cependant, nous pouvons à ce niveau envisager de caractériser les diverses séquences selon une catégorisation plus fine. Ceci n'entraîne *a priori* aucune difficulté supplémentaire, si ce n'est la mise en place d'un système capable de détecter et de "morceler" les longues séquences d'actions.

Tableau 4.3.3.1 : *Récapitulatif des séquences présentées par le Tableau A1.4.1, et vues pour chacun des utilisateurs listés aux différentes dates considérées*

Login de l'utilisateur	Date d'exécution	Séquence(s) d'actions effectuées par l'utilisateur
cge	23/11/1999	courrier ; écrire ; /engi\ ; /Réunion... à 14h !!\ ; envoyer
jc	12/10/1999	courrier ; écrire ; /all\ ; /Réunion à 9h30...\ ; /En salle intelligente.\ ; /Jean\ ; envoyer
	23/11/1999	courrier ; arrivée ; actualiser ; \rp231131/
	26/11/1999	répondre à tous ; /Re: Réunion PND\ ; copie personnelle ; /En salle intelligente.\ ; /Jean\ ; envoyer ; actualiser
jjk	12/10/1999	courrier ; arrivée ; actualiser ; \ml251101/
	23/11/1999	courrier ; départ ; écrire ; /sandel\ ; /Bonjour Olivier !\ ; /Séminaire demain...\ ; copie personnelle ; envoyer
osa	12/10/1999	courrier ; arrivée ; actualiser ; \rp231131/ ; répondre à tous ; /Re: Brainstorming\ ; /Bonjour à tous !\ ; /Message bien reçu :\ ; /à quelle heure ?!!\ ; /M'enfin !\ ; copie personnelle ; envoyer ; actualiser
	23/11/1999	courrier ; arrivée ; actualiser ; \repedem16/ ; répondre ; /Message bien reçu :\ ; /c'est OK pour moi.\ ; envoyer ; \rep23111/ ; archiver ; \repedem16/ ; supprimer ; OK
	24/11/1999	courrier ; arrivée ; actualiser ; \um241112/ ; ajouter adresse ; écrire ; copie personnelle ; envoyer
	26/11/1999	courrier ; arrivée ; actualiser ; écrire ; retour ; \dr261102/ ; faire suivre ; recherche ; /pl, dds\ ; /Poursuite du travail\ ; envoyer ; page d'accueil

Nous pouvons par suite choisir, selon les besoins du moment voire même de l'utilisateur considéré, de "découper" les séquences répertoriées suivant qu'elles s'attachent au traitement d'un "mail", qu'elles distinguent les séquences dites de fonctionnement de l'application (pouvant être constituées de l'entrée dans l'application, de la consultation de la boîte des "mails", de l'envoi ou la suppression de messages, etc) de celles plutôt dédiées au traitement des informations propres à l'utilisateur (lecture d'un "mail", écriture d'un message, mise-à-jour du carnet des adresses, etc).

Nous aboutissons à l'exemple du tableau proposé sur la page à suivre et qui met en pratique nos présentes idées sur deux cas concrets déjà découverts auparavant (**Tableau 4.3.3.2**).

4.3.3.3. Extraction et comptabilisation des répétitions

Nous sommes à présent capables d'analyser et de traduire les actions de nos utilisateurs.

Tableau 4.3.3.2 : Exemples de décomposition fine pour deux des séquences d'actions présentées dans le **Tableau A1.4.1**

Login de l'utilisateur	Date d'exécution	Séquence(s) d'actions effectuées par l'utilisateur
osa	23/11/1999	courrier ; arrivée ; actualiser
		\repdem16/ ; répondre ; /Message bien reçu : \ ; /c'est OK pour moi.\ ; envoyer
		\rep23111/ ; archiver
		\repdem16/ ; supprimer ; OK
	26/11/1999	courrier ; arrivée ; actualiser
		écrire ; retour
		\ dr261102 / ; faire suivre ; recherche ; /pl, dds\ ; /Poursuite du travail\ ; envoyer
		page d'accueil

Toujours à partir du **Tableau A1.4.1** et donc du **Tableau 4.3.3.1**, nous pouvons extraire ici les premières répétitions d'actions et de séquences d'actions, ce afin d'associer finalement à la majorité d'entre-elles un nouveau mécanisme d'apprentissage automatique dans le but d'assister, voire aussi surtout de faciliter, le travail quotidien de chaque utilisateur (**Tableau 4.3.3.3**).

Tableau 4.3.3.3 : Extraction et comptabilisation des différentes répétitions d'actions et de séquences d'actions issues des exemples du **Tableau 4.3.3.1**

Login de l'utilisateur	Séquence(s) d'actions répétées par l'utilisateur	Nombre de répétitions
jc	/En salle intelligente.\ ; /Jean\ ; envoyer	2
osa	courrier ; arrivée ; actualiser	4
	/Message bien reçu : \	2
	copie personnelle ; envoyer	2

Il faut cependant posséder, ce par définition-même, plusieurs séquences d'actions pour un même utilisateur pour pouvoir découvrir et répertorier une première répétition de séquences. Cette contrainte évidente remplie, nous avons associé notre présente extraction de connaissances à une comptabilisation rigoureuse du nombre de répétitions des diverses séquences découvertes.

Cette comptabilisation fournit, en temps réel, la fréquence de répétition caractérisant chacune de nos séquences redondantes. En considérant alors par exemple la première séquence de l'utilisateur connecté sous l'intitulé *osa*, nous possédons immédiatement l'information selon laquelle cette séquence a déjà été répétée 4 fois : nous pouvons donc sérieusement envisager l'automatisation de cette manipulation, d'autant plus que nous constatons en second lieu que l'utilisateur effectue cette séquence à chaque fois qu'il lance l'application "courrier" étudiée.

D'un autre point de vue, il se dégage du tableau mis à l'instant en avant deux types bien distincts de répétitions, à savoir celles d'une ou plusieurs chaînes de caractères strictement identiques et celles plus générales d'au moins deux actions successives. A noter au passage que la répétition d'une seule et unique action non textuelle, c'est-à-dire caractérisée par l'activation d'un bouton de l'interface ou par la désignation spatiale d'un "mail" existant, ne présente pour nos travaux aucun intérêt : il est en effet, dans ces cas-là, dénué de tout sens réel de chercher à automatiser les exécutions successives d'une même action, puisque celle-ci s'avère ainsi privée de tout contexte ou de but de réalisation compréhensible.

Il faut d'ailleurs aussi inévitablement associer notre procédé de recherche des répétitions d'actions à un système chargé d'analyser les nombreux "mails" présents ou entrants dans la boîte de messagerie de chacun des utilisateurs. L'actuelle simple sauvegarde reste insuffisante quant à l'extraction des multiples redondances d'informations pouvant émaner du contenu de chacun des "mails". L'unique nom du fichier, sous lequel tout "mail" est aujourd'hui sauvegardé, ne peut en effet en aucun cas nous permettre d'obtenir des renseignements quant aux nécessaires données correspondant aux interlocuteurs du "mail", à l'intitulé également du sujet de chaque message, ainsi qu'aux nombreuses habitudes d'écriture de l'expéditeur... Si nous possédons en outre toutes les informations entrées au clavier par les divers utilisateurs et qui nous renseignent directement sur les habitudes et préférences d'écriture de ces derniers, c'est le contraire en ce qui concerne la lecture des "mails", que ce soient ceux déjà écrits ou encore ceux réceptionnés.

4.4. Mise en oeuvre de nouvelles définitions

4.4.1. Dégagement d'actions remarquables

Dans le cadre du principe de décomposition des séquences que nous avons déjà abordé, un autre système se doit également d'être mis en place. Nous proposons en effet le mécanisme à suivre qui consiste à rechercher, repérer et marquer certaines actions primordiales, manipulations que l'on peut qualifier "d'actions-clés" et qui correspondent alors au début ou à la fin de chacune de nos séquences d'actions traitées (pour davantage de précisions, se référer à nos différentes définitions déjà établies en 4.2.3).

Nous pouvons ainsi, dans le cadre de nos derniers exemples, avancer les "actions-clés" de la page suivante (**Tableau 4.4.1**).

De plus, nous découvrons la possibilité d'instaurer, pour chacune de nos actions traitées, une notion supplémentaire que nous pouvons qualifier de relation ou facteur de "dépendance", voire également au contraire "d'indépendance". Ces nouvelles propriétés font ainsi l'objet des définitions successives jalonnant la prochaine partie.

Tableau 4.4.1 : *Liste des actions pouvant faire office "d'actions-clés"*

Situation au sein des séquences d'actions décomposées	Descriptif des "actions – clés"					
Début	courrier	départ	arrivée	archives	mail/	écrire
Fin	actualiser	différer envoi	envoyer	archiver	page d'accueil	

4.4.2. Actions dépendantes ou indépendantes

Cette nouvelle propriété de dépendance peut se définir, non seulement d'un point de vue théorique au travers de notre automate hiérarchique de la **Figure 4.2.1.2**, mais aussi, en temps réel et pour chaque utilisateur, à partir des répétitions de séquences recensées par le **Tableau 4.3.3.3**.

Pour le premier cas, nous ne pouvons que considérer, conformément à l'essence-même du graphe, que chaque action dépend de sa ou ses "actions-mères" respectives. Nous découvrons alors que la racine, à savoir plus précisément l'action d'activation du bouton "courrier", ne peut être qu'une action indépendante puisqu'elle ne présente, par définition, aucune "action-mère". Concernant la mise en pratique du second cas, nous proposons de traduire une nouvelle fois le **Tableau 4.3.3.3**, mais pour ne laisser apparaître que les actions dépendantes, c'est-à-dire associées et répétées les unes avec les autres (**Définition 4.4.2.1** à venir). Nous pouvons même énumérer les séquences *a priori* indépendantes, c'est-à-dire n'apparaissant cette fois-ci que de manière isolée ou non subordonnée à une précédente action au sein du tableau des répétitions de séquences (**Définition 4.4.2.2**). A noter qu'il nous faut aussi considérer que toutes les actions de l'utilisateur sont, initialement, sans propriété de dépendance ou indépendance : cela signifie que, tant qu'une première séquence d'actions répétée n'a pas été cataloguée pour cet utilisateur, aucune action réalisée par lui ne peut être dépendante d'une autre, et *a fortiori* encore moins indépendante.

En conséquence, les pages à suivre mettent aussi en évidence deux tableaux faisant état des différentes actions dépendantes et indépendantes que nous pouvons recenser, ce tout d'abord au contact des répétitions des manipulations de nos utilisateurs pris en exemples, puis de manière plus générale (**Tableau 4.2.2.1** et **Tableau 4.2.2.2**).

Définition 4.4.2.1 : *Relation de dépendance entre actions*

Une action **b** est dépendante d'une action **a** si et seulement si **b** suit **a** dans une séquence d'actions.

Définition 4.4.2.2 : *Propriété d'indépendance d'une action*

Une action peut être définie indépendante si et seulement si elle n'est pas dépendante d'une autre.

Tableau 4.4.2.1 : *Liste des interactions de dépendance pouvant être extraites des répétitions de séquences d'actions découvertes par le **Tableau 4.3.3.3***

Login de l'utilisateur	Action courante	Actions avec lesquelles l'action courante présente une relation de dépendance	
jc	/En salle intelligente.\	(action indépendante)	
	/Jean\	/En salle intelligente.\	
	envoyer	/Jean\	
osa	courrier	(action indépendante)	
	arrivée	courrier	
	actualiser	arrivée	
	/Message bien reçu :\	(action indépendante)	
	copie personnelle	(action indépendante)	
	envoyer	copie personnelle	

Nous pouvons alors déjà remarquer qu'en pratique, une action indépendante correspond toujours à une action de tête ou d'ouverture d'une séquence d'actions, voire parfois à une action de queue en cas d'action isolée (cf. **Définitions 4.2.3.2** et **4.2.3.3**). La corrélation avec la définition d'action de tête se comprend par ailleurs aisément puisqu'en première place, il est théoriquement impossible d'être précédé d'une autre action.

Cependant, les exemples présentés nous font aussi découvrir une notion supplémentaire que nous choisissons de nommer "dépendance transitive". En effet, nous pouvons aborder cette nouvelle relation en relevant qu'une action c peut être, par transitivité, dépendante d'une action a si c est dépendante d'une action b et que b est elle-même dépendante de l'action c . Par extension, l'action b peut également correspondre à plusieurs actions, c'est-à-dire par définition à une séquence d'actions intermédiaires. Ainsi, dans le **Tableau 4.2.2.1** et pour le cas notamment de l'utilisateur **jc**, l'action `|envoyer|` est par exemple dépendante transitive de l'action de tête de la séquence concernée, à savoir l'action `/En salle intelligente.\`. Concernant notre présente notion de dépendance entre actions, nous proposons alors une troisième spécification (**Définition 4.4.2.3**).

Définition 4.4.2.3 : *Relation de dépendance transitive entre actions*

Une action c est dépendante transitive d'une action a si et seulement si c est dépendante d'une action intermédiaire b et b est aussi dépendante de a .

Si, dans nos divers cas exposés, nous avons traité les relations de dépendance qu'il peut exister entre des actions, nous avons, par extension, aussi la possibilité de spécifier des séquences (d'actions) dépendantes, des séquences indépendantes et des séquences dépendantes transitives. Les trois définitions que nous venons d'établir peuvent en effet s'adapter à la définition générale d'une séquence d'actions (**Définition 4.2.3.2**) : il suffit de remplacer respectivement, dans nos trois définitions, les termes "action" par l'expression "séquence d'actions".

En exemples et pour conclure, nous pouvons alors affirmer, toujours en référence au **Tableau 4.3.3.3**, que la séquence d'actions `/Jean\ ; |envoyer|` est dépendante de celle décrite par `/En salle intelligente.\`, de même que les séquences d'actions respectivement représentées par `|courrier| ; |arrivée| ; |actualiser|` et `/Message bien reçu :\` sont, pour l'utilisateur **osa** et au niveau de

ses manipulations mises en avant par nos recherches, indépendantes. Nous proposons en outre, au travers du **Tableau A1.4.2** clôturant l'**Annexe 1**, un glossaire complet regroupant les différents types et propriétés par lesquels nous avons caractérisé nos actions et séquences manipulées.

Tableau 4.4.2.2 : *Liste des différentes relations de dépendance pouvant être extraites de manière abstraite de l'automate hiérarchique de la Figure 4.2.1.2*

Descriptif de l'action courante	Descriptif des différentes actions avec lesquelles l'action courante peut présenter une relation de dépendance					
courrier	<i>(action indépendante)</i>					
départ	courrier					
arrivée	courrier					
archives	courrier					
\mail/	départ	arrivée	archives	enregistrer brouillon	différer envoi	
	envoyer	annuler	OK	ajouter adresse	carnet d'adresses	
	archiver	actualiser				
actualiser	départ	arrivée	archives	enregistrer brouillon	différer envoi	
	envoyer	annuler	OK	ajouter adresse	carnet d'adresses	
	archiver	actualiser				
écrire	courrier	départ	arrivée	archives	enregistrer brouillon	
	différer envoi		envoyer	annuler	OK	ajouter adresse
	carnet d'adresses		archiver	actualiser	\mail/	
faire suivre	\mail/					
répondre	\mail/					
répondre à tous	\mail/					
/texte\	/texte\	annuler	OK	écrire	faire suivre	répondre
	répondre à tous	copie personnelle		recherche		
copie personnelle	/texte\	annuler	OK	écrire	faire suivre	répondre
	répondre à tous	copie personnelle		recherche		
recherche	/texte\	annuler	OK	écrire	faire suivre	répondre
	répondre à tous	copie personnelle		recherche		
effacer	/texte\	copie personnelle				
enregistrer brouillon	/texte\	copie personnelle				
différer envoi	/texte\	copie personnelle				
envoyer	/texte\	copie personnelle		\mail/		
supprimer	\mail/					
ajouter adresse	\mail/					
carnet d'adresses	\mail/					
archiver	\mail/					
annuler	effacer	supprimer				
OK	effacer	supprimer				

4.4.3. Hiérarchie de représentation des séquences d'actions effectuées

Conformément aux travaux décrits jusqu'alors, nous aboutissons à la spécification d'une nouvelle et complète hiérarchie concernant tous les séquençements d'actions mis en œuvre.

En effet, nous pouvons rappeler à ce niveau que nous nous sommes tenus, tout d'abord, à répertorier une seule séquence d'actions par utilisateur et par date d'exécution (**Tableau 4.3.3.1**). Nous avons ensuite mis en place un système capable de détecter et de "découper" les longues séquences d'actions extraites précédemment en séquences d'actions plus courtes, mais avant tout plus significatives vis-à-vis des besoins courants de l'utilisateur considéré (**Tableau 4.3.3.2**). Sans oublier enfin la couche de base de notre hiérarchie, sur laquelle d'ailleurs nous nous appuyons depuis le début de nos recherches, à savoir la quantité colossale d'informations que représentent toutes les actions exécutées de manière successive et indépendante par les différents utilisateurs pris en compte par notre système intelligent et auto-adaptable (**Tableau A1.4.1**).

Par conséquent, en précisant encore que nous choisissons de nommer par "ensemble de séquences d'actions" chacune des grandes séquences obtenues initialement du **Tableau 4.3.3.1**, une autre notion, que nous avons cette fois-ci étiquetée "transformation", apparaît également ici comme la décomposition des séquences en des entités, séquences ou actions, plus petites. Voici donc un graphe orienté complet chargé d'entériner de manière théorique notre actuelle hiérarchie du concept des séquençement d'actions, et ce avec $n \in \mathbf{N}$ (**Figure 4.4.3**).

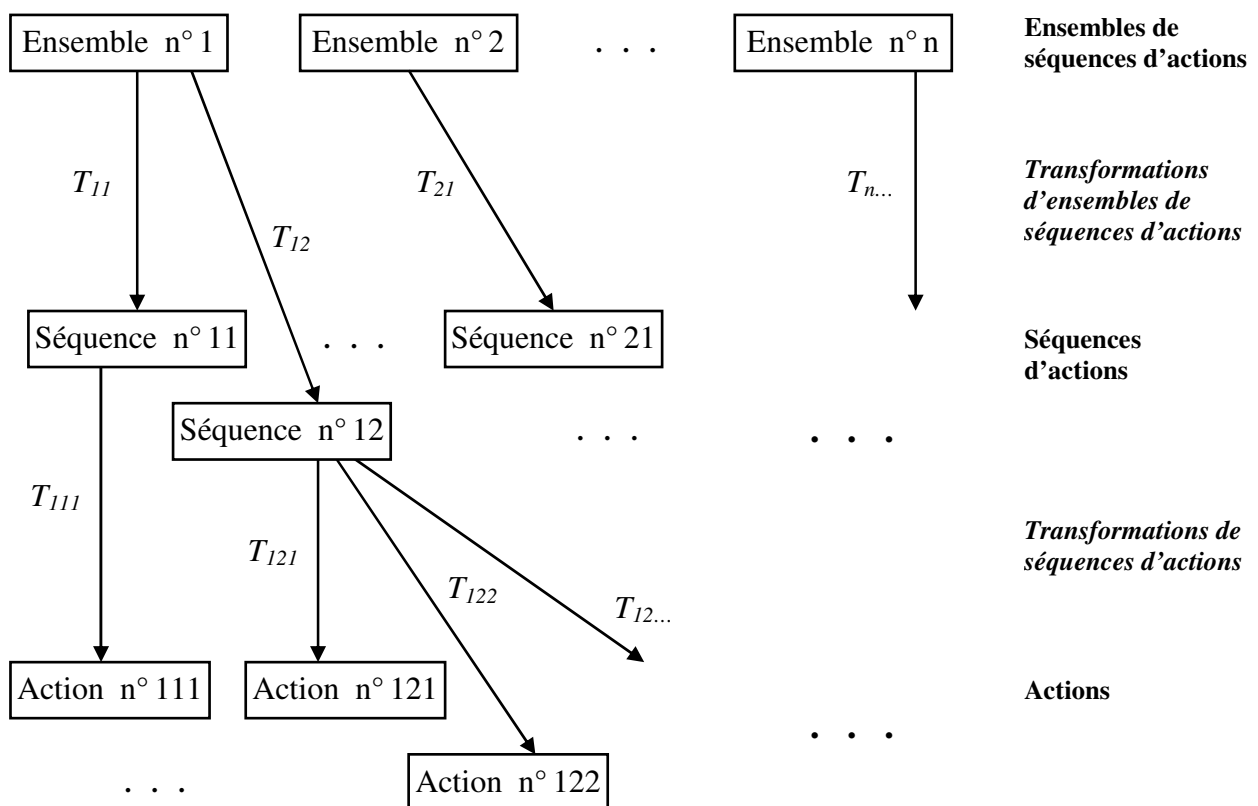


Figure 4.4.3 : Structure hiérarchique représentative de nos séquences d'actions

4.4.4. Dégagement de l'algorithme d'extraction des répétitions d'actions

Voyons maintenant par suite comment notre système élaboré par nos travaux de recherche peut aboutir aux résultats présentés auparavant par le **Tableau 4.3.3.3**. Pour cela, précisons encore qu'une séquence d'actions est considérée comme étant "répétée" si deux ou davantage d'actions (définies alors comme étant les différentes composantes de la séquence d'actions courante) se succèdent, pour une même application et un même utilisateur, suivant un schéma d'exécution strictement identique (**Définition 4.4.4**).

Définition 4.4.4 : *Algorithme d'extraction des répétitions d'actions et de séquences d'actions à partir d'une base de données*

Parcours de la base des actions :

Variables : descriptif d'action **a**, donnée textuelle **dt**, séquence d'actions **sa**

Pour chaque application

Pour chaque utilisateur

Pour chaque descriptif d'action **a**

Si **a** correspond à la désignation spatiale d'un nouveau "mail"

Alors extraire les informations contenues dans le "mail"

Sinon

Si **a** correspond à l'entrée au clavier d'une donnée textuelle **dt**

Alors

Si **dt** est rencontrée pour la seconde fois dans la base des actions

Alors répertorier **dt** comme donnée textuelle redondante
initialiser à "2" la comptabilisation de cette répétition

Sinon

Si **dt** est déjà répertoriée comme donnée textuelle redondante

Alors incrémenter d'une unité son nombre de répétitions

Sinon

Si **a** correspond à l'activation d'un bouton de l'interface
amorçant une séquence d'actions **sa**

Alors

Si **sa** est activée pour la seconde fois dans la base des actions

Alors répertorier **sa** comme séquence d'actions redondante
initialiser à "2" la comptabilisation de cette répétition

Sinon

Si **sa** est déjà répertoriée comme séquence redondante

Alors incrémenter d'une unité son nombre de répétitions

4.4.5. Généralisation du séquençement d'actions obtenu

Finalement, nous nous devons encore à ce niveau de généraliser notre nouvelle méthode d'extraction des différentes séquences d'actions provenant de nos utilisateurs. Cela apparaît bien entendu nécessaire afin d'aboutir à l'établissement et à la constitution recherchée de profils ou de groupes d'utilisateurs présentant plusieurs caractéristiques similaires.

Dans ce but, nous pouvons effectivement envisager de généraliser tous les descriptifs de "mails", respectivement demandés par les utilisateurs au cours de leur travail, ce en remplaçant l'identifiant exact du fichier de sauvegarde de chacun de ces "mails" (cf. **Définition 4.2.3.1**) par l'action générique précisant uniquement qu'il s'agit là de la désignation spatiale d'un "mail". Puis, en procédant de la même manière, nous étendons à terme également toute entrée au clavier d'une chaîne complète de caractères à son action générique correspondante relatant simplement l'action d'entrée au clavier d'une donnée textuelle par l'utilisateur.

Voici alors ci-dessous (**Tableau 4.4.5**) l'application de ces approches de généralisation au **Tableau 4.3.3.3**, tableau qui nous a auparavant déjà présenté l'extraction et la comptabilisation de l'ensemble des répétitions d'actions émanant de nos différents utilisateurs.

Tableau 4.4.5 : *Extrait de généralisation de nos répétitions de séquences d'actions*

Login de l'utilisateur	Séquence(s) d'actions répétées par l'utilisateur	Nombre de répétitions
jc	/texte\ ; envoyer	2
osa	courrier ; arrivéel ; actualiser	4
	courrier ; arrivéel ; actualiser ; \mail/	3
	copie personnelle ; envoyer	2
	/texte\ ; /texte\ ; envoyer	2

Ainsi, nous constatons explicitement qu'il est à présent possible de constituer des profils communs ou des groupes d'utilisateurs tels que nous le souhaitons dès le début de nos travaux. Chaque texte et chaque "mail" ne restent à ce niveau plus du tout individuels ou nominatifs : des séquences d'actions répétées apparaissent en effet identiques pour plusieurs de nos utilisateurs. D'où la possibilité tant recherchée de pouvoir spécifier des groupes d'utilisateurs présentant des caractéristiques de manipulation communes sur les différentes applications proposées.

A noter cependant une restriction de taille, à savoir que, si nous souhaitons généraliser les différentes séquences d'actions extraites du travail des utilisateurs, il ne faut en contre-partie surtout pas omettre que nous perdons alors toute information concernant le contenu précis des textes entrés au clavier par chacun d'eux, de même que les données pouvant être extraites des différents identifiants de "mail". Retenons par conséquent qu'en choisissant de travailler avec le processus de généralisation des séquences d'actions, nous décidons également de ne plus nous intéresser aux informations exclusivement individuelles et propres à chacun de nos utilisateurs.

4.5. Elaboration détaillée des mécanismes d'apprentissage

4.5.1. Spécifications précises

4.5.1.1. Introduction

Après avoir construit notre premier "Modèle d'Application" par l'analyse complète des données manipulées par la fonctionnalité "e-mail" du *Webtouch* d'ALCATEL (voir 4.1.1), nous avons décrit une des toutes nouvelles applications appelée AUM (*ALCATEL Unified Messaging*). Nous y reviendrons d'ailleurs, encore plus en détail, au cours du prochain chapitre focalisé sur nos travaux d'implémentation et de validation de nos modèles auto-adaptables. Et pour tout cela, nous n'avons souvent pas hésité à pousser nos investigations jusqu'à régulièrement avoir recours au procédé du "Reverse Engineering". En outre, nous avons toujours été amenés à suivre, à nous adapter et quelque fois même à anticiper les multiples changements, nombreuses contraintes et autres lois du marché des télécommunications de pointe qu'est celui du groupe ALCATEL.

En conséquence, nous disposons à ce niveau de deux importants supports, modélisés et pouvant accueillir de bien diverses possibilités de raisonnement et mécanismes d'inférence que nous envisageons de considérer pour aborder notre apprentissage automatique des différentes actions réalisées par nos utilisateurs. Nous proposons donc d'expliquer à présent suivant quelles manières nous comptons nous appuyer sur l'ensemble de nos précédents travaux de recherche pour réaliser un prototypage approfondi de nos mécanismes d'apprentissage automatique. De plus, grâce aux formalismes fonctionnels et hiérarchiques, aux traitements temporels et à la base de gestion des données, tous successivement spécifiés jusqu'ici, nous pouvons affirmer que toute action, effectuée par chacun des utilisateurs pris en compte par notre système auto-adaptable lors d'une quelconque utilisation des applications d'ALCATEL, est à présent intégralement représentée.

Nous allons ainsi nous préoccuper de notre véritable centre d'intérêt initialement avoué, à savoir les mécanismes d'apprentissage automatique appliqués aux actions effectuées, à leurs répétitions respectives, voire à des séquences plus évoluées en émanant. Nous envisageons alors différentes possibilités de progression, de raisonnement comme de systèmes dits d'inférence.

Dans un premier temps, nous allons aborder, de manière algorithmique, l'apprentissage symbolique mais aussi statistique des séquences d'actions réalisées de manière répétitive par chaque utilisateur sur nos "Modèles d'Application". Nous ne manquerons pas, entre beaucoup d'autres, d'étudier les informations ayant rapport avec les différentes fréquences de répétitions et statistiques d'utilisation de toutes les fonctionnalités mises à disposition par chaque application.

Nous tenterons ensuite d'élaborer un apprentissage, toujours en temps réel, des diverses habitudes de lecture des différents messages de chaque utilisateur, en produisant au préalable une spécification complète d'un innovant tri auto-adaptable concernant les messages arrivés et en attente d'être consultés par l'utilisateur selon ses propres préférences ou priorités de lecture précédemment apprises. Nous envisagerons également un procédé d'apprentissage automatique des données textuelles entrées successivement au clavier par l'utilisateur. Nous chercherons ainsi à définir l'élaboration auto-adaptable d'un message écrit, ou d'une réponse écrite à un message, ceci en connaissant bien au préalable l'ensemble des caractéristiques des différents messages auxquels l'utilisateur a tendance à répondre de manière systématique ainsi que ses "petites habitudes" d'écriture de ses messages (interlocuteurs habituels, listes de diffusion, expressions et sujets courants, formules de politesse, signatures adaptées, ...).

En second lieu, nous expliciterons un mécanisme de simplification et de correction des séquences d'actions réalisées par l'utilisateur pris en compte par notre système d'apprentissage. Ainsi, après une recherche de simplification automatique des nombreux séquençements d'actions effectués, et ce de manière à ce que ce procédé soit adapté aux différentes habitudes de travail de l'utilisateur établies auparavant, notre prototype présentera une complète tentative de correction intelligente d'erreurs émanant de manipulations détectables au sein des séquences d'actions.

Enfin, nous construirons, à l'aide de deux fonctions nommées "Tell Me" et "Do It" [Lieberman, 97], un modèle comportemental complet, et bien entendu toujours auto-adaptable : à l'implémentation d'un modèle de prédiction du comportement de chaque utilisateur suivra alors une anticipation auto-adaptable dédiée aux nombreuses actions issues des utilisateurs respectifs.

4.5.1.2. Répétition de séquences d'actions

En nous référant aux tableaux successifs déclinant notre base de données obtenue de l'observation et de l'extraction de l'ensemble des actions effectuées par nos utilisateurs, il nous faut ici, après avoir isolé toutes les actions correspondant au seul utilisateur courant, parcourir la liste, la vue donc de ces actions particulières afin d'en répertorier les différentes séquences d'actions qui se répètent. Nous aboutissons alors à l'algorithme déjà établi par la **Définition 4.4.4**.

4.5.1.3. Caractéristiques d'écriture du courrier

Dans cet ordre d'idée, il s'agit, et spécifiquement pour les applications de messagerie, d'extraire les données textuelles entrées successivement au clavier par l'utilisateur courant et d'y détecter, en fonction de paramètres également présents textuellement dans le message étudié (interlocuteurs, sujet, signature, ...), des termes, des expressions, des formules ou des phrases que l'utilisateur aura répétés. Ces informations émanent par ailleurs uniquement de l'utilisation d'une seule et unique action dans nos applications, à savoir celle décrite par le formalisme / ... \. Dans ce cas aussi, les nombreux tableaux et explications correspondantes, tous détaillés au sein des parties précédentes, nous ont déjà bien avancé les mécanismes nécessaires.

Ainsi, nous connaissons les multiples "petites habitudes" de chacun de nos utilisateurs pour l'élaboration de leurs messages écrits. Cependant, nous sommes à ce niveau aussi très bien renseignés sur l'ensemble des caractéristiques des différents messages auxquels ils ont tendance à répondre de façon répétitive, ce par l'analyse toujours textuelle expliquée ci-dessus concernant les données particulières et remarquables du travail de tout utilisateur.

4.5.1.4. Tri des messages

Grâce notamment à la description fonctionnelle sous forme hiérarchique des applications de messagerie électronique étudiées, nous disposons de la connaissance des différentes données que ces applications échangent, en entrée comme en sortie, avec leur environnement respectif.

Ainsi, nous avons à disposition les informations qui nous sont nécessaires, uniquement par l'observation d'une seule action de nos applications, représentée ici par le formalisme $\setminus \dots /$. Par suite, nous nous proposons, conformément à l'élaboration de notre "Assistant Individuel de Télécommunication", de calculer et d'attribuer automatiquement, à chaque message entrant dans la boîte aux lettres de l'utilisateur, une valeur de priorité résultant de la fréquence habituelle de consultation de ce message par ce même utilisateur.

C'est par conséquent en nous appuyant sur ces données obtenues des applications que nous allons tenter de trier et d'ordonner, c'est-à-dire d'intégrer un certain classement dans la gestion du fichier des messages de l'utilisateur. Ce fichier des messages, se voulant donc trié, devient le support à considérer dans ces lignes, sans oublier de préciser que nous souhaitons que ce tri soit, à terme, doué d'une intéressante "personnalisation" adaptée au mieux à l'utilisateur.

Différentes possibilités concernant la structure de ce tri s'offrent à nous dans la gamme des procédés de tri interne envisageables pour une fonctionnalité de messagerie, et notamment le principe des "listes triées" et celui des "tables à indexation". Une liste triée est une succession finie et ordonnée de valeurs ou groupes de valeurs. Une table, quant à elle, se voit assimilée à la combinaison d'un ensemble de valeurs accessibles grâce à un autre ensemble d'entrées indexées.

Dans le cas présent, nous souhaitons plus particulièrement travailler avec les tables indexées, ce pour plusieurs raisons. Tout d'abord, la prise en compte d'un nombre important de données devient vite complexe pour le procédé par liste, tant en ce qui concerne la taille (variable ou non) des valeurs ou groupes de valeurs, qu'au niveau de leurs quantités respectives. Une table s'acquiesce au contraire plus aisément de cette tâche, grâce justement à sa propriété permanente d'indexation. L'accès à une information précise ensuite, besoin primordial pour nos présentes investigations, reste très lourd pour les listes triées, car obligatoirement séquentiel. Enfin, les tables indexées restent spécifiquement dédiées à la gestion des fonctions proprement dites, grâce à leur très intéressante relation d'indexation (*clé* \rightarrow *adresse*).

Par conséquent, en approfondissant l'analyse et la définition des tables retenues comme support technique pour notre spécification de tri, nous découvrons qu'une table t est en fait une fonction partielle d'un ensemble R de requêtes (clés, indicatifs, indices, ...) dans un ensemble V de valeurs (**Définition 4.5.1.4**). Ainsi, nous obtenons une complète spécification LOTOS du tri initié jusqu'à présent ; le troisième volet de l'**Annexe 2** nous en fournit l'entier développement.

Définition 4.5.1.4 : *Définition fonctionnelle de notre tri*

$R = \text{Chaîne}$; $V = \text{Ensemble de Données}$

Table_Email :

Chaîne (R) \rightarrow Ensemble de Données (V)

intitulé de la requête (*clé*) \mapsto informations relatives au "mail" correspondant (*valeur*)

Pour conclure ce travail de conception d'un tri dédié aux habitudes de l'utilisateur en matière de messagerie, nous relevons que certains opérateurs *LOTOS*, dits "de garde" (condition préfixée de lancement d'actions) et notamment présents dans les trois processus d'adjonction, de suppression et de modification des "e-mails", se chargent de vérifier et garantir l'indispensable unicité des valeurs manipulées. Plusieurs valeurs différentes ne peuvent en effet correspondre à une même clé ou même indice d'une table de tri des messages émanant d'un même utilisateur. En outre, le processus de modification mis en avant par notre table de gestion des messages ne manipule que l'ensemble des valeurs elles-mêmes. Dans le cas contraire (c'est-à-dire en cas de possibilité de modification aussi des clés), il existe une probabilité réellement non négligeable d'altération du tri des messages au sein de notre table. Sans oublier enfin que, de son côté, la suppression d'un courrier dans la table ne modifie évidemment en rien le tri de cette dernière.

En second lieu, notons également qu'il s'agit d'un tri par insertion, et plus précisément par insertion séquentielle puisque le tri des messages s'effectue dès leur arrivée, dès leur entrée dans la table à indexation. La complexité d'un tel principe de tri se dérive en n^2 , pour un n correspondant au nombre d'éléments (et donc au nombre des messages) à trier dans la table. Ainsi, notre choix se justifie déjà comme étant le plus adapté à une fonctionnalité de gestion d'une table de messages en temps réel (prise en compte et classement de chaque valeur dès son arrivée dans la table). L'efficacité d'un tel procédé s'en trouve grandement améliorée du fait que chaque message traité reste unique : on ne peut avoir à faire face à deux courriers électroniques identiques (même interlocuteur, mêmes dates d'émission et de réception, même sujet, ...).

N'oublions pas cependant que le processus de recherche d'informations mis en évidence par la présente spécification formelle s'effectue à partir des indications contenues dans les mots-clés ou indices permettant de gérer correctement notre table triée par le fait qu'ils décrivent, de manière unique, chacun des messages répertoriés. D'où l'importance de l'analyse du contenu de chaque nouveau message arrivé dans la boîte aux lettres de l'utilisateur, analyse que nous avons déjà abordée au début de ce chapitre et aussi idéalement placée, dans notre "Assitant Individuel de Télécommunication", en toute première position pour le traitement de l'arrivée des messages.

Par extension enfin, et toujours dans la perspective de rechercher des procédés capables d'évoluer de manière pertinente vers des bases de données très conséquentes ou nécessitant un important temps d'accès, nous pourrions encore suggérer ici une autre méthode de tri de nos données. Nous pensons à l'indexation répertoriée sous les termes de "*B-tree*" ou "B-arbre".

Table signifiant précisément "*Balanced-tree*" ou "arbre balancé", la structure d'un tel concept se décline effectivement par une gestion sous forme arborescente, la racine de l'arbre et les nœuds internes étant les blocs contenant l'index. Et si celle-ci présente un arbre d'ordre m :

- chaque nœud contient k clés triées de gauche à droite, avec $m \leq k \leq 2m$, sauf la racine cependant qui contient, elle, un nombre de clés variant entre 1 et $2m$;
- un nœud est soit terminal, soit possède $(k+1)$ fils ;
- le $i^{\text{ème}}$ fils considéré possède des clés comprises entre la $(i-1)^{\text{ème}}$ et la $i^{\text{ème}}$ clé du père.

Toutefois, un tel arbre présente aussi la qualité d'être constamment équilibré, c'est-à-dire que tous les nœuds terminaux sont au même niveau, ainsi que tous les chemins de la racine aux feuilles possèdent la même longueur. En somme, si cette nouvelle méthode envisageable pour le traitement de nos messages propose d'intéressants avantages d'accès et de tri rapide dans des bases de données devenues importantes, elle apparaît aussi comme un index dynamique dont la structure est adaptée au fur et à mesure des besoins grâce à son architecture en arborescence, ainsi qu'à ses capacités de fusion et de fractionnement des divers nœuds de l'arbre mis en œuvre.

4.5.1.5. Simplification et correction automatiques

En prenant une nouvelle fois en référence le **Tableau A1.4.1**, vue expérimentale de notre base des différentes actions effectuées par nos utilisateurs, nous constatons qu'il est relativement délicat de chercher à simplifier automatiquement des séquences d'actions (et donc de les réduire), ainsi que de tenter de détecter de manière intelligente des erreurs d'utilisation répétitives (et donc aussi de chercher à les corriger). Cette tentative d'amélioration de l'efficacité du travail des utilisateurs demande en effet obligatoirement l'élaboration d'une base de donnée exhaustive représentant les diagrammes fonctionnels de chaque application considérée, ainsi qu'une capacité fiable de consultation de la masse "dynamique" d'informations à gérer (représentation des actions proprement dites, mais aussi surtout des relations et autres liens hiérarchiques associés).

Mais cela apparaît tout-à-fait abordable, dès lors que nous nous rappelons que nous possédons déjà, au travers des automates hiérarchiques établis pour chacune de nos applications, un descriptif complet des actions valides et réalisables au cours d'une utilisation de ces diverses applications. Il reste en effet à représenter ces automates de manière à ce qu'ils puissent être implémentés en machine, et consultés ensuite efficacement par notre système intelligent.

C'est pourquoi, nous proposons alors de décrire l'ensemble des manipulations valides et réalisables dans nos applications en recensant, pour chacune des actions courantes possibles, toutes ses "actions-filles" respectives et pouvant donc, par définition, directement lui succéder dans le temps. De cette dernière structure, nous constatons, si nous listons bien les différentes actions disponibles dans chaque application, que c'est surtout par la modélisation des multiples relations et liens hiérarchiques associés que s'enrichit une nouvelle fois notre base de données.

A ce niveau, n'oublions pas non plus que nous avons également déjà spécifié la notion de dépendance d'actions qu'il peut exister au sein de nos automates hiérarchiques respectifs, et par conséquent de nos répétitions de séquences d'actions. Ainsi, l'établissement d'une telle structure devient ici relativement aisé, dès lors que l'on remarque qu'il s'agit en fait d'inverser la relation décrite par les différents tableaux nous ayant auparavant (cf. **Tableau 4.4.2.1** notamment) fourni les interactions de dépendance et les "actions-mères" pouvant être extraites des graphes.

Finalement, en se servant même du travail déjà réalisé en **4.5.1.2** pour accélérer de manière intéressante l'apprentissage des répétitions des diverses séquences d'actions courantes, il faut vérifier si chaque action est bien exécutée suivant le schéma de l'automate correspondant.

En outre, nous pouvons encore améliorer, une nouvelle fois, l'efficacité du travail des utilisateurs en procédant tout de même pour leur indiquer d'éventuelles possibles simplifications automatiques concernant certains de leurs séquencements (boucles, retours en arrière, erreurs de manipulation, ...), et ce en cherchant alors à remplacer ces derniers par des séquences plus petites en nombre d'actions effectuées. Ces séquences réduites, dans le temps comme en taille de mémoire au niveau de la machine, deviennent aussi inévitablement plus efficaces que celles d'origine puisque notre système d'apprentissage garantit des résultats bien entendu strictement identiques à ceux de ces dernières, mais avec des gains de temps et d'espace très appréciables.

Nous détaillerons ainsi plus avant l'ensemble de ces idées lors du chapitre suivant dédié à l'implémentation et à la validation de tous nos prototypes auto-adaptables.

4.5.2. Algorithmique des mécanismes d'inférence

4.5.2.1. Introduction

Nous nous devons maintenant d'apporter toutes les réponses algorithmiques nécessaires à l'implémentation de nos modèles auto-adaptables élaborés jusqu'ici. Ces solutions doivent être, tout d'abord, aussi générales que possible, afin d'être applicables aux différents produits visés au sein d'ALCATEL. En d'autres termes, il faut tirer parti de la somme considérable d'informations que des applications informatiques, concernées par l'interfaçage homme-machine tel que nous l'envisageons, sont amenées à exploiter, ainsi que de toutes les capacités de ces systèmes en matière de mise en rapport des données complexes manipulées par nos modèles auto-adaptables. En approfondissant ainsi, nous constatons que l'apprentissage automatique correspond, pour une machine, à la modification de son comportement en fonction de son histoire [Bruillard, 1997], ou de son historique des actions effectuées dans un passé proche, voire même déjà lointain.

Ensuite, il s'agit également de piloter des dispositifs technologiques par le traitement de représentations symboliques référant à des objets ou encore à des actions du monde réel. Devant la difficulté d'un tel objectif [Ruvini, 2000], une description détaillée de la tâche d'apprentissage automatique, ainsi qu'une précision des contraintes réelles que doivent satisfaire les algorithmes requis pour la mise en œuvre de cette tâche, s'avèrent indispensables. Nous verrons ainsi, pour le confort de manipulation proposé à l'utilisateur, mais aussi pour l'efficacité de son travail, qu'un apprentissage automatique doit être "suffisamment" rapide, peu coûteux en temps de calcul et avec un temps de prédiction éventuel très court. Par ailleurs, si, en plus d'être auto-adaptable, notre interface intelligente se doit d'être "flexible" et "intelligible", c'est-à-dire capable de laisser à l'utilisateur la possibilité de modifier ces précédentes habitudes, l'expressivité et bien sûr la qualité d'interaction, garantes de suggestions intéressantes, ne sont pas non plus à négliger.

Si nous avons déjà tout expliqué dans la partie 4.4 pour ce qui concerne les répétitions d'actions indépendantes telles que les données textuelles entrées au clavier ou les habitudes de lecture de l'utilisateur vis-à-vis de ses messages, nous devons encore spécifier ici les mécanismes propres aux actions identiques qui proviennent des multiples activations de boutons de l'interface. Ce travail se rapporte alors à l'apprentissage des séries d'actions effectuées de manière répétitive au cours des utilisations de chaque application. En outre, la prise en compte des statistiques de manipulation de toutes les actions mises à disposition s'y adjoindra de manière inévitable, ainsi qu'enfin la recherche et la proposition de prédiction, anticipation et simplification automatiques, voire aussi de correction des erreurs, émanant des très nombreux séquençements d'actions appris.

4.5.2.2. Principales orientations issues de l'apprentissage automatique

Conformément aux concepts d'analyses des connaissances que nous avons construits, tout s'oriente dans nos travaux de telle manière à ce que l'entité "action" corresponde au centre névralgique de nos divers modèles auto-adaptables. Ceci établi, et si nous n'oublions donc pas que nous avons choisi d'évoluer dans le domaine du séquençement d'actions (cf. 4.1), nous nous devons aussi nous rappeler que nous sommes toujours limités par de très sévères contraintes de temps, d'espace, de ressources et de robustesse en ce qui concernent les machines d'ALCATEL.

En outre, la mise en œuvre des mécanismes d'inférence que nous abordons à ce niveau se doit également de constamment chercher à prendre en compte et à respecter le traitement de deux des principales qualités de nos modèles avancés. Il s'agit, plus précisément, de la liberté d'évolution future si chère à tout utilisateur et, *a fortiori*, à chaque concepteur de logiciels, mais aussi surtout de la notion de personnalisation qui demeure une des premières caractéristiques intelligentes proposée par notre Assistant.

En premier lieu, nous pensons alors aux nombreuses propriétés décrivant les méthodes et procédés attachés aux réseaux de neurones artificiels. Des études et nouvelles perspectives de travail, indispensables à une intégration efficace et robuste de ces concepts dans nos mécanismes d'apprentissage automatique, se voient ainsi précisées au travers des différents points ci-contre :

- comportement du réseau en cas d'ajout d'une nouvelle catégorie (nommée ou non), c'est-à-dire d'un neurone supplémentaire sur la couche de sortie (validité théorique, gestion des liaisons pondérées, ajustement des poids déjà existants, signification des résultats, etc) ;
- possibilité d'associer la dernière couche cachée (si elle existe) à la notion de "sous-couche" (ou "pré-couche") de sortie, afin d'affiner et de mieux personnaliser la classification des différentes catégories retenues ;
- évaluation de l'efficacité d'un réseau de neurones traitant des séquencements d'actions et fonctionnant entièrement en apprentissage non supervisé ;
- conséquence de l'ajout d'un nouveau neurone d'entrée, à l'image des enseignements issus de l'étude approfondie des caractéristiques et des performances menée sur l'algorithme dit de "Cascade Correlation" [Sandel, 1997] ;
- concrétisation et donc exécution, en temps réel, des algorithmes d'apprentissage associés.

Ces importantes difficultés déjà annoncées, le cadre des réseaux de neurones artificiels présente également un autre concept intéressant, à savoir les réseaux neuronaux récurrents. Au sein de tels systèmes, un avantage coïncide au fait que les liaisons peuvent former des topologies arbitraires : elles ne sont pas toutes unidirectionnelles, et il peut y avoir des cycles. Le cerveau humain se définit d'ailleurs en réseau récurrent [Russell & Norvig, 1995] : parce que l'activation est réintroduite dans les neurones qui sont à son origine, les réseaux récurrents ont des états internes emmagasinés dans les niveaux d'activation des neurones. Les réseaux récurrents peuvent donc implémenter des conceptions d'agents complexes et modéliser des systèmes à états.

La solution apportée par la théorie des réseaux de neurones récurrents consiste en conséquence à boucler tous les neurones entre eux, aussi dans le sens inverse de la propagation [Gurney, 1996]. On comprend alors par suite que l'état des neurones à une étape donnée influence pleinement le calcul de l'étape suivante, et que l'aspect dynamique est ainsi toujours bien pris en compte. En pratique, on retiendra surtout quelques caractéristiques générales [Weinfeld, 2000] :

- capacité de généraliser une fonction continue dans le temps à partir des exemples fournis,
- prédiction de phénomènes sans en connaître les équations ni les lois qui les régissent,
- adaptation à tout phénomène sans apport de connaissance extérieur.

Si la réalisation de réseaux neuronaux à topologie complexe (connexions dans les deux sens, traversant les couches de neurones, entrées/sorties non différentiables, ...) est en théorie possible, le problème reste surtout pour nous d'en trouver un algorithme d'apprentissage efficace. Plusieurs structures récurrentes existent (réseaux de *Hopfield*, machines de *Boltzmann*, réseaux d'*Elman*, réseaux dynamiques, ...) et des algorithmes spécifiques y ont été associés [Russell & Norvig, 1995]. Elles offrent alors de nombreuses possibilités sur le traitement des séquences temporelles, mais leur développement très ciblé et caractéristique demeure encore complexe.

Notre **Chapitre 3** nous a aussi déjà entraîné vers la considération d'une seconde méthode d'apprentissage automatique, à savoir celle des classifications et des catégorisations d'entités. Le terme "classification" recouvre plusieurs significations selon le contexte dans lequel il est utilisé. Le sens qui lui est accordé en analyse de données est plutôt celui de la distribution en classes ou encore catégories : étant donné un ensemble d'objets, il s'agit alors de trouver une répartition en classes de ces objets en regroupant ceux qui possèdent des caractéristiques et un comportement similaires. Par ailleurs, le sens qui lui est plus communément accordé en Intelligence Artificielle est celui du classement et de la découverte de classes d'appartenance. Si la classification automatique est par conséquent la constitution de groupes cohérents à partir d'ensembles de données [Ketterlin, 1995], rappelons à ce niveau qu'il s'agit en outre d'un processus permettant de dériver un ensemble de classes à partir de l'observation d'un ensemble de manifestations de ces classes. C'est un processus inductif, largement exploratoire. Son résultat est une "représentation" de la réalité sous une forme compacte ; et son intérêt est donc fondamental pour l'Intelligence Artificielle. En effet, un système qui ne serait pas capable d'induire un certain nombre de généralisations pertinentes sur la base des faits perçus serait rapidement noyé par une explosion de la quantité d'informations à prendre en compte. La classification permet ainsi de gérer ce flux en assurant la définition de classes décrivant les phénomènes.

De nombreuses méthodes effectuant une classification d'un ensemble d'objets existent. Elles sont toutes de natures très différentes, mais nous pouvons citer ici les trois grandes familles suivantes [Wemmert, 2000]. Ces méthodes proposent, soit uniquement un partitionnement de l'espace des données, soit un regroupement des objets à classer en classes homogènes, soit une hiérarchie de classes, soit une hiérarchie de concepts décrivant chacun une classe d'objets :

- les méthodes de classification statistiques,
- les méthodes de classification neuronales,
- les méthodes de classification symboliques, ou par formation de concepts.

Dans le domaine des méthodes statistiques et probabilistes, nous ne pouvons manquer d'encore souligner, et afin d'initier nos prédictions et anticipations auto-adaptables des séquences d'actions (cf. **4.5.3**), la méthode bien connue sous l'expression des "chaînes de Markov".

En effet, nous pouvons alors couvrir ici le cas particulier où l'agent logiciel surveille et prédit passivement un environnement changeant, plutôt que de directement agir sur lui [Russell & Norvig, 1995]. L'évolution de cet environnement est d'ailleurs modélisée par la distribution de probabilité conditionnelle $P(X_t|X_{t-1}, A_{t-1})$, qui décrit comment l'état X dépend de l'état précédent et de la dernière action A de l'agent. Ce dernier gère ainsi une séquence de X_t valeurs, où chacune d'entre-elles est déterminée uniquement par la précédente au travers de $P(X_t|X_{t-1})$. Cette séquence de X_t valeurs est appelée "modèle d'évolution à états" ou "chaîne de Markov".

Surveillance et prédiction sont donc toutes deux importantes à terme, un agent pouvant très bien utiliser les X_t pour prendre des décisions et réaliser des actions.

D'un point de vue plus éloigné des bases théoriques de l'informatique, nous pouvons également aborder ici le vaste contexte de la robotique et de la cybernétique, sujettes aujourd'hui à des évolutions successives, voire même à des améliorations très intéressantes. Le RIA ("*Robot Institute of America*") définit un robot comme un manipulateur multi-fonctions, programmable et désigné pour déplacer du matériel, des pièces, des outils, ou des dispositifs spécifiques, ce au travers de mouvements variables, programmés pour l'exécution d'une variété de tâches [Russell & Norvig, 1995]. A notre niveau, nous pouvons cependant délimiter cette notion par le fait qu'un robot peut surtout être un agent artificiel et actif dont l'environnement est le monde physique tel que nous, humains, l'appréhendons. Par suite, nous sommes concernés en premier lieu par les robots autonomes, à savoir ceux qui prennent des décisions de leur propre chef, guidés par le

retour de leurs capteurs physiques. La majorité de la conception d'un robot autonome apparaît ainsi identique à celle d'un agent logiciel autonome tel que nous l'avons déjà défini en **1.2.4.5**.

C'est pourquoi, nous parlons avant tout d'entités nommées plus précisément "*softbots*" par la communauté internationale des chercheurs en informatique, et dont l'environnement est constitué des systèmes de fichiers informatiques, des bases de données et des réseaux. De très bons exemples nous ont été fournis par [Lieberman, 1995] et [Lieberman, 1997], ce dans le domaine bien particulier de la recherche d'informations sur le réseau *Internet*. Nous pouvons citer *Letizia*, agent d'interface autonome réalisant des suggestions en temps réel de pages *Web* susceptibles d'intéresser l'utilisateur pendant son déplacement sur *Internet*.

Autres travaux mêlant une nouvelle fois la robotique avec l'apprentissage automatique, [Mataric, 1994] nous présente l'interaction d'agents plongés dans des environnements complexes d'analyse et de synthèse des comportements de groupes intelligents. Descriptions et évaluations empiriques des lois de contrôle et de planification nécessaires aux groupes d'agents pour atteindre des buts et être composés des propriétés appropriées, ces idées sont validées sur plus de vingt robots mobiles utilisant un ensemble de comportements basiques. Ceux-ci comprennent l'errance sûre (minimisation des collisions entre agents et environnement), le suivi (minimisation de l'interférence en structurant le mouvement), l'agrégation (assemblage d'agents), la dispersion et, aussi, le retour à la base (habilitation des agents à poursuivre vers un emplacement particulier).

Mais n'oublions pas non plus de relever l'avènement de la lignée des robots conçus et construits à l'image des animaux domestiques ou de compagnie. "Robokitty" par exemple, et pour ne citer que lui, est un chaton artificiel destiné à tester en grandeur réelle l'interconnexion de modules évolués en une organisation cérébrale adaptée au contrôle de centaines de comportements modélisés [Perret du Cray, 2000]. Le robot apparaît donc pour démontrer les capacités et expérimenter la mise en place d'un cerveau artificiel. Les capteurs embarqués dans le système alimentent en informations les modules neuronaux (vision, reconnaissance sonore, tactile, ...): nous abordons ainsi un autre point tout aussi remarquable de l'apprentissage, à savoir l'ingénierie évolutive ("*Evolutionary Engineering*") et ses algorithmes génétiques.

En effet, l'algorithme génétique est un modèle d'apprentissage automatique qui trouve son comportement dans l'évolution de la nature [Goldberg, 1994]: ce sont les meilleurs individus d'une population, ceux qui sont les plus adaptés, qui arrivent à survivre et qui propagent ainsi leur matériel génétique, représenté par un code et appelé chromosome. Un tel algorithme se charge également d'améliorer la qualité, ou adaptation, de ces individus. Les techniques utilisées sont des mécanismes d'évaluation et de sélection, eux-mêmes suivis d'opérations génétiques nommées "croisement" et "mutation".

Les mécanismes des systèmes de classifieurs évolutifs deviennent alors à ce niveau bien intéressants. Plus que jamais procédés d'apprentissage automatique, ceux-ci apprennent en effet syntaxiquement des règles sous la forme de chaînes, donc appelées à terme des "classifieurs". Les nouvelles caractéristiques ainsi établies guident alors les opérations de chacun des dispositifs évolutifs dans un environnement arbitraire, et ce dans le but de développer des systèmes adaptatifs pour la modélisation du comportement. Les composantes principales et indispensables à la construction de tels modèles cognitifs sont, outre un environnement, des "détecteurs" et des "effecteurs" qui, respectivement, renseignent leur système au sujet de l'actualité des événements et lui permettent ensuite de manipuler son environnement. Nous pouvons encore détailler ici les sept différentes étapes d'un cycle de fonctionnement de tels systèmes de classifieurs :

- introduction de messages environnementaux produits par les "détecteurs",
- collection des règles qui reconnaissent les messages présents dans la liste établie (M),
- élection des règles par un mécanisme d'enchère des règles activables (constitution de M'),

- activation des règles de M' et production d'une nouvelle liste des messages,
- extraction, de cette nouvelle liste, des messages destinés aux "effecteurs" et mise en œuvre de ces derniers,
- rétribution des règles de M' si l'action des "effecteurs" conduit à une amélioration (appelée "récompense"), les règles étant en effet récompensées chaque fois que la différence entre la position voulue et la position courante diminue (elles sont pénalisées dans le cas contraire),
- remplacement de la liste courante des messages par celle qui vient alors d'être produite par le système.

Le champ d'action de telles techniques évolutives présente aujourd'hui un fort intérêt dans le cadre des problèmes d'optimisation. Déjà posés en termes de recherche enveloppant de simples structures linéaires, ceux-ci entraînent de plus l'exploration d'entités plus riches, comme l'automatisation des processus de découverte d'informations dans le vaste domaine des bases de données [De Jong, 1999]. Cependant, les multiples calculs relevant des systèmes de classifieurs évolutifs demeurent très chers. Les contextes attachés à nos besoins apparaissent déterminés et n'entrent en conséquence pas dans le présent domaine du stochastique : les règles, issues des actions de l'utilisateur, ne présentent pas de caractères aléatoires.

Nous pouvons finalement retenir ici quatre différences notoires distinguant l'algorithme génétique des techniques plus conventionnelles, et insister en effet sur les avantages suivants :

- manipulation directe d'un codage de paramètres (consistance et interaction entre les gènes avantageuses) ;
- exploration au moyen d'une population, et non d'un seul point unique ;
- exploration aveugle à partir des seules valeurs de la fonction d'adaptation ou de qualité ;
- optimisation utilisant des opérateurs stochastiques [Porto, Fogel & Fogel, 1995], et non des règles déterministes.

Pour poursuivre, nous devons à présent, tout d'abord justifier, ensuite établir selon quels algorithmes d'apprentissage automatique nous allons pouvoir extraire et traiter chacune des répétitions de séquences d'actions. Pour cela, et en ayant pour objectif de toujours demeurer représentatifs des mécanismes avancés jusqu'alors, nous étudierons deux approches distinctes, abordant successivement les domaines du symbolique (en 4.5.2.4) et du statistique (en 4.5.2.5).

A retenir alors déjà que la différence majeure entre les méthodes statistiques et celles issues de l'apprentissage symbolique fait appel à leurs manières respectives d'appréhender, apprendre et valoriser chacune des informations extraites des bases de connaissances associées. Le but de l'apprentissage automatique, au sens statistique du terme tout d'abord, consiste en effet à considérer, travailler des objets selon leur similarité [Ketterlin, 1995], exprimée par une distance évolutive. L'apprentissage symbolique, quant à lui, a plutôt pour principal objectif de chercher à produire des descriptions correspondant aux informations découvertes, c'est-à-dire de mettre en œuvre les caractéristiques conceptuelles qui permettent de définir ces divers ensembles d'actions.

4.5.2.3. Justification des structures d'implémentation choisies

Nous allons à présent préciser les divers points de repères et autres apports essentiels de l'implémentation des modèles et fonctionnalités auto-adaptables. Pour cela, nous nous aiderons, non seulement de toutes les recherches que nous avons exposées auparavant, mais aussi des outils

logiciels développés depuis le début de ce mémoire et appliqués à nos concepts auto-adaptables. Nous atteindrons ainsi, en définitive, la détermination et le développement complet d'un modèle innovant d'interface intelligente, pleinement dédié à un interfaçage homme-machine.

Dans ce cadre, nous pouvons d'emblée signaler "DataDetective", un outil d'extraction de connaissances et de modélisation informatique chargé de déceler le sens de l'information en général [Sentient Machine Research, 1999]. Basé sur une recherche associative lui permettant de trouver dans un fichier les profils humains les plus ressemblants et non pas que strictement conformes à un signalement, ce produit propose à terme une recherche par profil à partir de critères pondérés, stricts ou même encore flous. En second lieu, [Bits & Pixels, 1998] ainsi que [Net Perceptions, 2001] proposent de filtrer une majorité des messages électroniques entrants, à savoir tous les *e-mails*. Si nous avons déjà minutieusement étudié (en 1.2.4.6) l'interface intelligente mise en exergue par [Net Perceptions, 2001] pour un filtrage collaboratif efficace, le "filtre neuronal de classification" (*neural classification filter*) de [Bits & Pixels, 1998] correspond lui à un produit de filtrage opérant, comme son nom l'indique, à l'aide d'un réseau de neurones, et demeurant en outre capable de fournir un système de filtrage de documents conventionnels.

Nous pouvons également découvrir "Swiftfile", pour *Lotus Notes*, sorte d'assistant aidant l'utilisateur à organiser ses *mails* dans des répertoires par l'utilisation d'un classifieur textuel pour analyser les caractéristiques des messages entrants [Segal & Kephart, 1999]. Et si "Backflip" propose à son tour de classer automatiquement les pages *Web* qu'on lui indique, ou même toutes les pages visitées, ce dans une base de données stockée sur ses propres serveurs [Neuman, 2000], "Nomino" [Plante, Dumas & Plante, 2000] propose lui de construire des bases de connaissances en utilisant le langage naturel comme support d'information. Ce dernier logiciel se charge en effet de structurer toutes les données d'un même utilisateur de telle manière qu'elles soient toujours accessibles par de simples interrogations exprimées en langue naturelle. De programmer un secrétaire personnel qui range et retrouve pour l'utilisateur des mémos de tous genres à la construction "d'auto-réponses" au courrier électronique, cet outil peut alors aussi faire l'analyse, le résumé et le dépouillement terminologique des documents textuels rencontrés.

A l'image déjà du contenu de nos précédents chapitres, nous ne pouvons que confirmer alors que des outils spécialisés tels que nous en découvrons encore dans ces lignes améliorent la réactivité des utilisateurs, que ce soit dans le cas du traitement automatisé des messages entrants ou encore de l'automatisation de réponses à ces derniers. Nous pouvons d'ailleurs citer la société *Akio Solutions* pour laquelle le traitement d'un *mail* coûte jusqu'à 70% moins cher qu'un appel téléphonique [Bordage & Saiz, 2001]. Cet éditeur propose des outils de traitement automatique ou semi-automatique des *e-mails* entrants (*Mail Center*). Véritables répondeurs "intelligents", ces produits haut de gamme analysent le contenu des messages en s'appuyant sur une base de connaissances et une analyse sémantique. Le logiciel de traitement des *e-mails* joue le rôle de poste d'aiguillage entre les différentes personnes chargées de répondre aux messages. Il s'appuie sur un historique de la relation avec le client pour router le message vers la personne la plus apte à répondre. Certains de ces programmes sont aussi capables de préparer une réponse dite "type". Selon le choix retenu par l'entreprise, ils l'envoient directement à l'internaute ou la dirigent vers un collaborateur qui valide la pertinence de la réponse et personnalise le message final avant son envoi au destinataire. Ces logiciels réduisent donc considérablement le temps de traitement des réponses et apportent à l'entreprise une meilleure réactivité vis-à-vis de ses clients.

Sans oublier que c'est aussi une source de productivité pour le personnel de l'entreprise : « A terme, notre solution sera capable de constituer automatiquement une base à partir des messages reçus et d'insérer des offres ciblées dans les réponses renvoyées aux internautes », affirme Hervé Davaine, Directeur de la Communication chez *Akio Solutions*.

A noter enfin ici que la garantie de qualité des *e-mails* sortants demeure également aujourd'hui primordiale et repose sur divers paramètres : organisation des équipes, ergonomie du site ou mise en œuvre d'outils de filtrage, cette fois-ci sortants. En effet, lors de la saisie d'un message électronique, la qualité de la syntaxe, la pertinence de la réponse, le degré de personnalisation sont autant de points qui contribuent à valoriser ou à dégrader l'image de l'entreprise aux yeux du client. Le recours à une plate-forme spécialisée dans le traitement automatisé des *e-mails* n'est donc pas le seul dispositif à mettre en place. Les *e-mails* sortants doivent véhiculer l'image de l'entreprise : professionnalisme, réactivité, service client, etc. Cette communication s'intègre dans une logique plus globale tenant compte des différents canaux de communication utilisés par le client (*e-mail*, téléphone, courrier, etc). Dans le cas spécifique de la messagerie électronique, un *e-mail* sortant peut par exemple reprendre la "baseline" (devise attachée aux activités) de l'entreprise, ou même le slogan publicitaire de la campagne en cours.

En conclusion, nous avons choisi d'orienter notre programmation de telle sorte qu'elle demeure riche et utile pour la communauté informatique propre à l'interfaçage homme-machine. En d'autres termes, face aux différentes directions que prenaient les autres travaux répertoriés précédemment, nous avons conduit notre implémentation en priorité vers la concrétisation des domaines de nos modèles auto-adaptables encore inexplorés à ce jour, à savoir l'extraction en temps réel de données interactives, la prédiction et l'anticipation auto-adaptables d'informations répétées, la modélisation d'utilisateurs "grand public", la simplification de séquences d'actions associée à une recherche de correction "intelligente" des erreurs de manipulation inhérentes.

Méthodes axées par définition sur l'entité "action", les séquencements sont apparus bien adaptés aux bases de données issues de nos modèles d'extraction et d'analyse des connaissances. Ils répondent également, au mieux, à des contraintes incontournables de taille-mémoire, de temps d'accès, de complexité et de robustesse, propres aux machines d'ALCATEL considérées.

4.5.2.4. Approche symbolique

Le succès de la "fouille dans les données" (ou "Data Mining" pour les anglophones) montre l'intérêt théorique et pratique que représente ce domaine d'apprentissage automatique dans le cadre des nouvelles technologies de l'information. En effet, les progrès de la technologie informatique, notamment au sujet du recueil et du traitement de données qui nous intéressent plus particulièrement, font que, dans tous les grands domaines de l'activité humaine actuelle, nous manipulons des informations en quantité souvent gigantesque, comme de toutes sortes (numériques, textuelles, graphiques, etc). De plus, partout dans le monde, se constituent des gisements de connaissances considérables, mais cependant encore difficiles à extraire et à représenter. Résumer et catégoriser l'ensemble de ces données, afin de mieux les appréhender et d'en extraire de nouvelles connaissances plus avantageuses pour nos utilisateurs, constitue alors un nouveau point crucial de nos travaux de Thèse.

Les concepts inhérents à la découverte de nouvelles connaissances sont décrits par des données plus complexes que celles habituellement rencontrées dans le domaine "statistique" que nous étudierons dans la prochaine partie. Les informations que nous allons prendre en compte ici sont dites "symboliques", car elles sont structurées et expriment la variation interne inéluctable des concepts. Dans ce contexte, l'extension des méthodes symboliques, attachées à l'analyse de données exploratoires pour en extraire des connaissances d'interprétation améliorée et plus aisée pour l'homme, devient à notre époque une tâche importante.

En conséquence, l'apprentissage symbolique se charge, par définition, d'élaborer des méthodes permettant d'obtenir des connaissances structurelles ou décisionnelles ("expressions symboliques"), ce à partir d'un ensemble d'instances elles-mêmes peu structurées. Nous retenons en outre que cet apprentissage a pour but principal de construire automatiquement des bases de connaissances, utilisables ensuite par un moteur d'inférence [Ketterlin, 1997]. Les objectifs sont multiples : d'une part, éviter les problèmes liés à l'acquisition des connaissances (interrogation fastidieuse d'un expert, codage rébarbatif de règles, etc), d'autre part, fournir une représentation symbolique de ces connaissances, éventuellement analysable et compréhensible par un humain.

Par suite, nous pouvons également relever ici que la façon basique d'apprendre, de manière artificielle comme pour l'homme d'ailleurs, est d'observer des situations pour lesquelles nous connaissons des réponses respectives adéquates. Un système apprend alors en profitant de cette expérience pour construire une base de connaissances qui lui permettra ensuite de prendre une décision face à une situation dans laquelle la réponse est inconnue. Ce cadre très général constitue en définitive le fondement de l'apprentissage dit "supervisé" qui utilise un ensemble, souvent très gros, d'exemples ou d'objets donnés à la machine avant le premier lancement de l'application considérée. Le système fonctionnant suivant ce principe apprend ainsi en consultant les classes ou les objets qu'on lui a fournis et que, par conséquent, il connaît déjà [Sandel, 1997].

Si notre système et les agents intelligents que nous lui avons associés travaillent déjà en temps réel, ceux-ci se doivent aussi de trouver eux-mêmes les différentes réponses nécessaires à la résolution de tout problème courant. Dit autrement, les informations pouvant caractériser les manipulations de chaque utilisateur, de même que leur nombre, ne sont bien évidemment pas du tout connus à l'avance : nos modules sont donc ainsi perpétuellement en cours de construction puisqu'il leur faut découvrir les différentes séquences d'actions tout en apprenant. Ceci dit, nous reviendrons sur ces techniques issues de notre système multi-agents au cours du **Chapitre 5** traitant plus concrètement des concepts d'implémentation attachés aux modèles auto-adaptables. Nous précisons et analyserons l'intégration, aux présents concepts d'apprentissage symbolique et au Modèle Auto-Adaptable de l'Utilisateur, d'autres outils à l'image des "heuristiques". Nous verrons que de tels procédés désignent des méthodes d'aide à la découverte indispensables à un fonctionnement rapide et puissant de tout logiciel faisant appel à l'apprentissage symbolique. Les heuristiques sont en effet particulièrement utiles en Intelligence Artificielle, car elles permettent de discerner, dans l'ensemble des dérivations possibles, celles qui ont le plus de chances de conduire à un succès de l'opération courante [Ganascia, 1993].

Première technique alors déjà très représentative aujourd'hui de l'efficacité propre à un tel apprentissage symbolique, l'inférence grammaticale (ou encore "*grammar-based technique*") correspond au domaine qui englobe la théorie et les méthodes d'apprentissage de grammaires à partir de corpus de données [Parekh, 1998]. La communauté "apprentissage" accorde par ailleurs de plus en plus d'importance à l'inférence grammaticale, concernée très tôt par les problèmes de reconnaissance de formes. C'est ainsi que l'on assiste à l'émergence de nouvelles méthodes qui s'appliquent à des domaines de plus en plus complexes [Haton & Smaïli, 1996] : reconnaissance de la parole, de caractères, traduction automatique, compréhension, diagnostic médical avancé, etc.

Ce problème de l'inférence grammaticale peut alors être succinctement formulé de la manière suivante : soit un ensemble d'exemples d'un langage donné, trouver la grammaire du langage qui a généré l'ensemble des exemples. Par conséquent, nous pouvons définir l'inférence grammaticale comme un processus d'apprentissage automatique de grammaires à partir d'un ensemble fini d'exemples. Le plus souvent d'ailleurs, l'objectif de telles techniques demeure l'apprentissage des procédures de classification ou de catégorisation d'entités représentatives de l'environnement propre à tout système considéré [Nicolas, 1993].

Toujours dans le cadre des systèmes inductifs issus de l'Intelligence Artificielle, nous pouvons également rechercher des règles associatives intéressantes pour le problème considéré, et faire ainsi appel au concept des ensembles d'items fréquents, plus connus sous les initiales *FIS* ("*Frequent Item Sets*"). En se basant sur un principe qui est d'utiliser les ensembles fréquents de la génération k pour construire ceux de l'itération $k+1$, nous distinguons ici l'algorithme appelé "Apriori". Il est caractérisé par une boucle principale chargée, à chaque passage en son sein, de générer les nouveaux ensembles à traiter et d'incrémenter ensuite un compteur pour chacun de ces candidats. On aboutit à la conservation et à l'intégration des meilleurs candidats, c'est-à-dire ceux dépassant un seuil de présence dans tous les ensembles déterminés [Lachiche, 2001].

Par extension, nous pouvons encore remarquer à ce niveau que plusieurs variantes de ce dernier procédé ont également vu le jour. A retenir notamment l'algorithme intitulé "Apriori partitionné", intéressant avant tout par le fait qu'il débute son activité par une division de la base de données traitée pour ensuite chercher, à l'image des détails déjà apportés plus haut, les ensembles fréquents de chacune des partitions obtenues. Ce principe, judicieux bien évidemment dans le cas de machines partagées et autres systèmes parallèles, ne conserve finalement que les ensembles dits "fréquents globalement", c'est-à-dire ceux fréquents dans au moins une partition.

Pour résumer, l'apprentissage automatique en général, et symbolique en particulier, recouvrent un ensemble de techniques algorithmiques visant à doter les machines de capacités réelles d'apprentissage, que ce soit pour que ces machines tirent parti de leurs expériences afin d'exécuter des tâches routinières, de construire des bases de connaissances complexes, ou encore, pour qu'elles extraient des connaissances à partir de bases de données. Mais la conception des systèmes d'interaction présente également plusieurs approches notables [Garbay & Maitre, 1997] : si les approches concurrentielles analysent et fusionnent ultérieurement les différents résultats obtenus, alors que les approches du type "essai/erreur" évaluent plutôt la qualité du résultat pour formaliser ensuite le couplage "erreur/réparation", celles dites incrémentales ou opportunistes construisent la solution de manière progressive, contrôlée et en temps réel.

C'est dans cette dernière catégorie que s'inscrit l'approche symbolique que nous allons proposer et étudier à présent, puisqu'elle fait appel à un apprentissage évolutif à base de règles, à l'image de la machine "perceptive intelligente" [Garbay & Maitre, 1997] que nous avons découvert dans le premier chapitre. L'apprentissage, appréhendé en termes d'interactions entre systèmes et environnement, acquiert alors de nouvelles règles et méthodes de modélisation, d'interprétation, d'évaluation, de contrôle aussi et même d'adaptation des comportements. En outre, l'intégration de l'utilisateur dans la boucle d'apprentissage, notamment par correction d'erreurs et pré-structuration des problèmes comme nous l'avons explicité au travers de nos précédents tableaux et processus associés, permet la transposition à des cas complexes des connaissances apprises sur des cas simples. Cela est interprété, pour nos présents travaux, par le traitement de séquences d'actions entières à partir des caractéristiques extraites de chaque action effectuée par l'utilisateur.

Ainsi, en s'appuyant sur la définition-même des graphes d'états, et donc en particulier sur ceux que nous avons mis en place pour nos applications, nous découvrons à présent que deux séquences d'actions identiques, qui correspondent toujours à la répétition d'une seule et même séquence initiale, émanent concrètement de deux noeuds identiques eux aussi. Nous pouvons par conséquent concevoir, ce toujours en partant de la base d'actions construite, un algorithme précis s'appuyant sur une approche symbolique pour déterminer des répétitions de séquences d'actions à partir des actions exécutées par nos utilisateurs (**Définition 4.5.2.4** sur la page suivante). Nous constatons alors que, si les algorithmes d'apprentissage automatique permettent la découverte d'importantes "régularités" dans de larges ensembles de données [Mitchell, 1999], cette partie de notre algorithme illustre les règles apprises automatiquement de l'ensemble de nos données, de nos actions extraites auparavant. L'apprentissage symbolique est en effet une méthode qui induit

des règles qui seront elles-mêmes ensuite utilisées pour structurer notre modèle auto-adaptable de l'utilisateur, et plus particulièrement l'agent logiciel intelligent chargé du séquençement des actions (cf. précisément **5.1.3** à ce sujet).

Définition 4.5.2.4 : *Algorithme de détermination des répétitions de séquences d'actions selon l'approche symbolique*

Variables :

$x_{n,n \in \mathbb{N}}$ et $y_{m,m \in \mathbb{N}}$ sont les descriptifs complets de n et m actions respectives ;
 a est l'intitulé d'une application logicielle de communication ;
 u est l'identifiant d'un utilisateur du terminal intelligent ;
 i est un compteur, entier naturel initialisé à la valeur "1" ;
 b est un booléen associé à la terminaison des séquences et initialisé à la valeur "faux" ;

Début

Pour chaque action x_0 répondant à une réalisation sur l'interface d'un utilisateur u

Faire

Si une action y_0 présente une date antérieure à celle de x_0 dans la base des actions,
et x_0 et y_0 relèvent de la même application a ,
et x_0 et y_0 ont été effectuées par le même utilisateur u ,
et x_0 et y_0 correspondent à la même manipulation au niveau de l'interface

Alors x_0 et y_0 sont susceptibles d'amorcer une répétition de séquence d'actions,
et mettre b à "vrai",

et noter x_1 l'action de u dans a succédant à x_0 dans la base des actions,
et noter y_1 l'action de u dans a succédant à y_0 dans la base des actions ;

Tant que b est à "vrai"

Si x_i et y_i correspondent à la même manipulation au niveau de l'interface

Alors la séquence d'actions $\{x_0; x_1; \dots; x_i\}$ répète $\{y_0; y_1; \dots; y_i\}$,
et mettre à jour toutes les données d'apprentissage de cette séquence répertoriée
à présent comme une séquence d'actions redondante dans la base des actions,

et noter x_{i+1} l'action de u dans a succédant à x_i dans la base,

et noter y_{i+1} l'action de u dans a succédant à y_i dans la base,

et incrémenter i d'une unité ;

Sinon mettre b à "faux",

et réinitialiser i à "1" ;

Fin

Le système d'apprentissage présenté vise ainsi à préciser des descriptions de concepts issus d'informations fournis par la base de connaissance. Cette dernière peut inclure des valeurs de variables faisables et des hiérarchies (cf. **4.4.3**), des prédicats, des règles auxiliaires de découverte de données, des préférences d'utilisateurs... Le procédé se construit alors à partir de la taille et de la pertinence de la base de connaissances, en s'appuyant sur les résultats représentatifs et la nature présumée des concepts que l'on cherche à acquérir [Michalski, Bratko & Kubat, 1998].

De telles méthodes orientées vers les connaissances tendent à créer et à manipuler des structures de données symboliques, capables de satisfaire les multiples besoins des systèmes mis en œuvre, et donc de l'utilisateur. Elles peuvent également caractériser nombre d'ensembles de données qualitativement, sélectionner les cas les plus représentatifs (et donc les actions)... Ces techniques sont ainsi particulièrement adaptées au développement de descriptions qui impliquent des variables nominatives (catégoriques) et hiérarchiques au sein des bases de données.

Grâce à des règles de raisonnement présentant des opérateurs symboliques tels que les distinctions ou la concaténation d'entités définies, nous obtenons la liste complète des répétitions de séquences d'actions remarquables dans notre base de données. Il est alors enfin intéressant de noter que nous cherchons constamment à découvrir des séquences d'actions qui, en plus d'être répétitives, soient aussi les plus complètes possibles, et par suite les plus riches en informations.

4.5.2.5. Approche statistique

Si les processus symboliques sont conduits par des règles d'inductions, c'est la prise en compte des fréquences qui va prédominer dans un contexte statistique. En effet, les approches ou modèles statistiques se font fort, notamment au cours d'une extraction de données, de capturer certaines régularités formelles des objets étudiés, en l'occurrence des actions d'utilisateurs, et de leur associer une fréquence ou une probabilité d'apparition [Giguët, 1998]. Ce sont ces régularités, qui, une fois stockées, vont nous servir de connaissances pour identifier, simplifier, prédire puis anticiper les répétitions de séquences d'actions proposées à l'utilisateur. L'identification consiste à calculer, en fonction des régularités obtenues, la probabilité d'une action à appartenir aux séquences remarquables. Deux grands types de ces régularités sont couramment exploités ici : les actions les plus fréquentes et les séquences les plus fréquentes, les séquences étant composées de plusieurs actions. Nous reviendrons plus précisément, au cours de la prochaine partie, sur les trois autres opérations, à savoir la simplification, la prédiction et l'anticipation des séquences.

En nous remémorant encore le rôle primordial joué par nos précédents procédés chargés de l'identification et de l'extraction des actions réalisées par chacun des utilisateurs, nous nous devons également de retenir à ce niveau que les données brutes ont ainsi été mises sous une forme telle qu'on puisse les analyser et les traiter par des algorithmes issus, non seulement des apprentissages symboliques, mais donc aussi des apprentissages statistiques. C'est par ailleurs une des préoccupations de la science statistique que d'extraire d'un ensemble de données une connaissance de nature générale sur les phénomènes que ces données décrivent [Kodratoff, 1994]. Empruntant le formalisme des probabilités, les statisticiens représentent la réalité par un certain nombre de mesures, et leur but est d'énoncer les principes dictant les valeurs de ces quantités.

Nous découvrons en outre, en parlant des modèles statistiques de manière plus générale, des objets allant jusqu'à faire intervenir des variables proches de l'aléatoire et donc presque du hasard [Lauzon, 1999]. En effet, les modèles statistiques liés aux théories de l'apprentissage deviennent à ce niveau intéressants pour plusieurs raisons.

Tout d'abord, les données observées contiennent très souvent du "bruit" : dès lors que nous mesurons ou appréhendons une variable, nous pouvons *a priori* nous attendre à ce que cette mesure ne soit pas exactement la valeur réelle. La différence entre la valeur réelle et la valeur mesurée est alors ce que l'on appelle le bruit. Ce bruit peut dépendre partiellement de la valeur de la variable (une grande valeur aura plus de bruit qu'une petite), mais il est aussi fonction

d'autres variables auxquelles nous n'avons pas forcément accès. Pour cette dernière raison, le bruit est souvent modélisé comme une variable aléatoire, un modèle statistique étant alors utile pour modéliser des variables mesurées puisqu'au moins cette composante du bruit est aléatoire.

Ensuite, la variable extraite peut aussi elle-même dépendre d'autres variables. Par exemple, si nous tentons de prédire l'âge d'une personne étant donnée sa taille, nous savons d'emblée que cette donnée est insuffisante, quoiqu'elle contienne déjà de l'information non négligeable (par exemple, quelqu'un de cinq ans ne peut pas mesurer deux mètres...). Le modèle statistique peut ainsi compenser le manque d'information en donnant une distribution d'âge pour chaque taille. Notons en plus qu'il est souvent impossible d'avoir toutes les informations sur un phénomène et donc de les fournir toutes en entrée à un modèle [Lauzon, 1999].

Enfin, il faut savoir que tout modèle a ses limites. Si nous ne connaissons pas tous les mécanismes d'un phénomène (et c'est presque toujours le cas lorsque nous faisons appel au domaine des algorithmes d'apprentissage), nous pouvons essayer plusieurs modèles dont aucun ne modélisera parfaitement les données, même si les données sont théoriquement déterminées en fonction des entrées. Et là encore, le modèle statistique compense ce manque en ajoutant un phénomène aléatoire plutôt que de générer de manière déterministe une mauvaise réponse.

Néanmoins, si les statistiques dites "classiques" semblent ainsi s'intéresser de plus en plus étroitement aux diverses formes d'apprentissage, deux points caractérisent et différencient les modèles statistiques de la théorie de l'apprentissage. D'une part, ce type de modèles met généralement en avant des hypothèses fortes sur la distribution sous-jacente des données. Pour les tests d'hypothèses par exemple, nous devons en effet considérer tout d'abord une classe de distributions paramétrées et qui a généré les données, puis seulement une distribution de ses paramètres inhérents : nous sommes ainsi amenés à supposer une distribution des données, à estimer ensuite les paramètres de celle-ci, et à tester finalement une hypothèse par rapport à cette distribution [Bengio & Ducharme, 1999]. D'autre part, l'approche statistique classique est aussi généralement asymptotique, du fait que la plupart des conclusions sont basées sur ce qui arrive lorsque le nombre d'exemples croît vers l'infini.

De son côté, la théorie de l'apprentissage s'intéresse plus particulièrement à la prise en compte d'exemples et tente de dissocier les données de leurs modélisations respectives (ce principe ne pose donc pas comme hypothèse que les données manipulées aient été au préalable générées par le modèle). La règle de Bayes est notamment une règle théorique qui représente la meilleure règle ou procédure de classification possible, et ce au sens de la probabilité d'erreur que possède le système de mal classer un exemple donné [Nicolas, 1993]. Cette règle repose sur la connaissance (hypothétique) des probabilités conditionnelles $P(c|x)$, pour x quelconque dans X (l'espace de représentation) et c dans $\{1, 2, \dots, C\}$. La classe qui est désignée par la règle de Bayes, lorsque la situation est x , est la classe i telle que $P(i|x) = \max_c P(c|x)$. Elle constitue une borne indépassable qui représente d'une certaine manière la difficulté intrinsèque du problème considéré. Notons enfin que la règle de Bayes est dépendante de la représentation des situations. Et à citer encore ici, puisque pleinement représentatives aussi de l'apprentissage statistique, les chaînes de Markov que nous avons détaillé lors de l'introduction à la présente étude (cf. 4.5.2.2).

Toutefois, pour en revenir plus spécifiquement à nos propres recherches, nous retenons qu'un modèle statistique peut très bien servir à prévoir et à anticiper toute prochaine observation courante des actions d'un utilisateur. Finalement, de tels modèles statistiques sont aussi souvent associés aux modèles dits "analytiques" [Michel, 2000] et chargés, eux, de la modélisation de motifs suivant leur probabilité (résultats non aléatoires cette fois). A l'image de nos travaux de représentations des connaissances, cela ne fait par conséquent que nous renforcer dans l'idée que, comme les modèles symboliques étudiés au préalable, les modèles statistiques possèdent de puissantes capacités pour prévoir, en globalité, la distribution de données, présentes et futures.

Pour résumer, la méthode statistique choisit de construire les différentes séquences d'actions redondantes suivant qu'une action possède la plus forte chance, la plus importante probabilité de composer à un moment donné la séquence d'action courante. Cela signifie, en d'autres termes, que, lorsque notre système auto-adaptable fonctionne selon cette approche et se charge de proposer à l'utilisateur d'effectuer automatiquement l'action à venir, il choisit cette dernière en fonction de la probabilité élevée de rencontrer cette action dans le présent contexte, c'est-à-dire en fonction des différentes actions qui viennent d'être exécutées et qui caractérisent donc aussi le début de la séquence d'actions en cours.

En conséquence, et pour intégrer cette approche statistique à nos différents systèmes auto-adaptables, nous avons défini et mis en œuvre l'algorithme de détermination des répétitions de séquences d'actions à venir sur la page suivante (*Définition 4.5.2.5*).

En comparant alors cet algorithme à celui déjà établi précédemment (en *4.5.2.4*) selon le principe symbolique, nous découvrons que cette méthode de détermination statistique correspond à une approche de l'extraction et de l'apprentissage des répétitions de séquences d'actions plus directement orientée vers chacune des caractéristiques de manipulation ainsi que vers les comportements successifs des utilisateurs de notre interface de communication.

En effet, nous construisons ici nos différentes séquences en fonction de l'importance respective que celles-ci ont acquise au sein des diverses réalisations de l'utilisateur : seules les actions effectuées en majorité par ce dernier sont par conséquent considérées comme permettant d'améliorer son travail. Par ailleurs, c'est dans ce sens d'une prise en compte dynamique et donc d'ajustements permanents que nous poursuivrons et approfondirons, dès les prochaines pages, nos mécanismes de simplification, de prédiction et d'anticipation que nous avons conçus auto-adaptables pour le traitement de l'ensemble des séquences d'actions apprises à ce niveau.

Cependant, cette dernière méthode statistique présente également l'inconvénient d'être plus coûteuse en nombre d'instructions de même qu'en temps d'exécution. Nous pouvons alors effectivement noter que, si cette approche fait appel à davantage de tests conditionnels, à des structures de données supplémentaires ainsi qu'à des itérations nettement plus conséquentes, elle requiert en outre la mise en place d'hypothèses et de distributions associées à chaque contenu de séquences plutôt que l'utilisation brute de simples descriptifs d'actions déjà établis. Mais il nous faut bien aussi convenir que c'est le prix à payer pour une détermination de séquences d'actions redondantes plus intéressante et significative vis-à-vis de nos nombreux utilisateurs.

En conclusion générale, tous nos précédents modèles et algorithmes sont pleinement dédiés à l'apprentissage automatique et à ses inhérents mécanismes d'inférence que nous nous sommes attachés à judicieusement appliquer à l'ensemble de nos concepts auto-adaptables. Nous insistons également, à ce niveau, sur l'observation que, dans notre système d'implémentation, nous avons continuellement cherché à axer la parité, ainsi que les places respectives réservées aux approches symboliques comme statistiques, en fonction de leurs qualités propres, de manière à toujours exploiter l'ensemble de ces dernières avec un maximum d'efficacité.

Nous verrons alors par la suite qu'il nous a souvent fallu lutter contre la tentation de donner davantage de densité aux ressources catégoriques ou symboliques par rapport aux informations acquises statistiquement ou de manière probabiliste. Cela ressort des simples faits, déjà, que l'application de ces dernières demeure moins rigoureuse en théorie et beaucoup moins maîtrisable en pratique que pour ses concurrentes, dans le cas bien entendu d'un complet système d'interaction entre l'homme et la machine.

Définition 4.5.2.5 : *Algorithme de détermination des répétitions de séquences d'actions selon l'approche statistique*

Variables :

$x_{n,n \in \mathbb{N}}$ et $y_{m,m \in \mathbb{N}}$ sont les descriptifs complets de n et m actions respectives ;
 s^x et s^y sont les descriptifs complets de deux séquences d'actions respectives ;
 a est l'intitulé d'une application logicielle de communication ;
 u est l'identifiant d'un utilisateur du terminal intelligent ;
 i est un compteur, entier naturel initialisé à la valeur "1" ;
 b est un booléen associé à la terminaison des séquences et initialisé à la valeur "faux" ;

Distributions :

F correspond à l'ensemble des descriptifs complets composant notre base d'actions ;
 E englobe tous les descriptifs d'actions relevant d'une même session de travail ;

Hypothèses :

t^s traduit le taux de répétition courant d'une séquence d'actions ;
 t^u est le taux global de prise en considération des répétitions de séquences d'actions correspondant au mieux aux différents comportements successifs de l'utilisateur (et par suite mis à jour dynamiquement en fonction des évolutions de celui-ci) ;

Intervalle de confiance :

d définit la valeur courante des intervalles de confiance (également tous dynamiques) propres à l'apprentissage des répétitions de séquences d'actions ;

Début

Pour chaque action x_0 répondant à une réalisation sur l'interface d'un utilisateur u

Faire

Noter x_1 l'action de u dans a succédant à x_0 dans la base E ;
 Mettre b à la valeur "vrai" ;

Tant que b est à "vrai"

Instancier s^x par $\{x_0; x_1; \dots; x_i\}$;

S'il existe une séquence d'actions $s^y = \{y_0; y_1; \dots; y_i\}$,

et $s^y \in (F \setminus E)$, // restriction chargée d'obtenir la distribution initiale générée

et s^x et s^y relèvent de la même application a ,

et s^x et s^y ont été effectuées par le même utilisateur u ,

et s^x et s^y correspondent aux mêmes manipulations de l'interface

Alors s^x et s^y forment une nouvelle répétition de séquence d'actions,

et modifier t^s le taux de cette répétition dans la distribution E ,

et si $t^s \geq t^u \pm d$

alors intégrer la nouvelle répétition de s^x aux données d'apprentissage,

et incrémenter i d'une unité ;

sinon mettre à jour la valeur de t^s dans la distribution F ,

et instancier y_0 par y_i ,

et réinitialiser i à "1" ;

Sinon si $y_i \in E$

alors mettre b à "faux" ;

Fin

4.5.3. Construction finale d'un modèle comportemental de l'utilisateur

4.5.3.1. Introduction de nos mécanismes de prédiction et d'anticipation

De manière générale, les interfaces dites "prédictives" [Ruvini, 2000] fonctionnent par définition selon deux mouvements. Après avoir en effet mémorisé ou appris des corrélations entre les situations que l'utilisateur rencontre et les actions correspondantes qu'il accomplit, de telles interfaces intelligentes utilisent ensuite ces situations ou ces corrélations pour prédire les actions futures de ce même utilisateur.

Par extension, de nombreuses propriétés apparaissent alors indispensables dans le but de réaliser des prédictions, naturellement suivies de souhaits d'anticipations pertinentes, au sujet des intentions et des différents actes de l'utilisateur courant, ce afin d'assister finalement ce dernier dans ses multiples interactions avec la machine [Stémart, 1997]. Ainsi, et notamment en accord direct avec le Modèle Auto-Adaptable de l'Utilisateur développé au **Chapitre 3**, nos mécanismes de prédiction et d'anticipation des répétitions de séquencements d'actions, émanant de chacun de nos utilisateurs, se devront d'être, à la fois :

- suffisamment complets (il ne s'agit pas de mettre en avant des bribes d'informations, mais de proposer des informations utiles à l'utilisateur),
- compréhensibles (l'utilisateur ne doit pas perdre son temps à essayer de comprendre ce que le système a anticipé, sinon la détection n'a pas d'intérêt puisqu'elle est censée réduire la tâche de l'utilisateur),
- adéquats (le système propose, parmi un ensemble de solutions possibles, celle que l'utilisateur aurait probablement aussi choisie),
- discrets (le système doit faire intervenir le moins possible l'utilisateur, l'intérêt de l'anticipation étant le gain de temps apporté).

Les différents mécanismes d'inférence, d'apprentissage automatique et autres capacités d'induction que nous souhaitons apporter aux applications étant à présent spécifiés, la dernière phase de nos travaux d'intégration d'un modèle d'interface intelligente pour nos terminaux de communication aboutit à la formation d'un modèle comportemental de l'utilisateur, c'est-à-dire à la prédiction de son comportement et à l'anticipation auto-adaptable de ses nombreuses actions.

Pour cela, nous abordons plus précisément la modélisation des deux fonctions nommées "Tell Me" et "Do It" [Lieberman, 1997] que nous avons déclinées selon les principes à suivre.

4.5.3.2. Modélisation dynamique des seuils comportementaux

Des propositions sont faites automatiquement à l'utilisateur, en fonction des actions et séquences d'actions que celui-ci réalise et surtout répète. La machine a en effet pour premier but de chercher à effectuer de manière auto-adaptable les actions répétées. Ces séquences détectées ont toutes, bien entendu, été analysées et apprises de l'extraction des connaissances issues du travail de l'utilisateur, conformément aux tableaux, définitions et algorithmes mis en œuvre.

Cette phase dite du "Tell Me" ("Dis-Moi" en traduisant directement de l'anglais) voit ainsi la machine annoncer à chacun de ses utilisateurs, à tout instant et au travers de son interface devenue "intelligente", quelle séquence d'actions courante elle souhaite tenter de réaliser seule pour aider cet utilisateur et éventuellement accélérer son travail en cours. L'utilisateur répond alors à la machine, grâce à des moyens aisés et surtout très rapides et efficaces (un simple clic de souris la plupart du temps), en lui faisant savoir s'il souhaite effectivement que la machine réalise, à ce moment-là, la séquence des actions qu'elle lui propose d'effectuer.

Puis, en second lieu, et suite à un nombre à présent conséquent mais aussi paramétrable d'acceptations des diverses propositions de la machine par l'utilisateur courant, l'exécution de ces actions répétitives devient réellement "auto-adaptable", non sans un ultime acquiescement de l'utilisateur attendu par la machine. C'est la phase de la fonction "Do It" ("Fais Le") qui entraîne une prise d'initiative, certes toujours modulable, de la part de la machine au sujet de la séquence d'actions en question : dès que la machine détecte la (ou les) première(s) action(s) de cette séquence, elle décide, en temps réel, d'effectuer successivement toutes les autres la composant.

Des valeurs encore à préciser, et que nous avons nommées "seuils comportementaux" au regard de leurs rôles respectifs et séparatifs, commandent alors le déclenchement de ces deux nouvelles phases relevant de notre présent modèle de traitement des comportements d'utilisateur. Le calcul de ces seuils correspond en outre à une caractéristique primordiale des mécanismes avancés à ce niveau. En effet, et c'est ainsi que se justifie de plus le caractère auto-adaptable de nos procédés, nous avons réussi à concevoir et mettre en œuvre ces seuils de manière à ce qu'ils varient, s'adaptent ou encore s'ajustent en fonction de toutes les qualités propres à chacun de nos utilisateurs pris en considération. Déjà entièrement développées au cours du **Chapitre 3**, nous rappelons par ailleurs que ces diverses capacités ont alors donné lieu à l'établissement de notre modèle auto-adaptable des utilisateurs, et par là-même, à une classification et une catégorisation de tous les instants pour chacun d'entre-eux, ce en fonction de leurs compétences respectives. Ainsi, nos derniers seuils de comportements permettent maintenant de moduler, dynamiquement comme de façon indépendante, l'ensemble de nos propositions et prises d'initiatives directement issues des mécanismes d'assistantat intelligent étudiés dans les parties précédentes.

Nous devons aussi insister ici sur les lignes de programmation décrites au cours de la partie **5.2.5**, au travers du **Tableau 5.2.5** ainsi qu'au niveau des **Figures 5.2.5.1, 5.2.5.2 et 5.2.5.3**. Ces références vont en effet nous apporter les ultimes explications nécessaires à la mise en place et à la compréhension des structures composant notre modèle comportemental de l'utilisateur.

En résumé, voici un schéma succinct nous apportant une vision globale et récapitulative des phases et principes de notre Modèle Comportemental de l'Utilisateur mis en œuvre par notre Interface Intelligente pour terminaux de communication (**Définition 4.5.3.2**). S'en suivront alors d'indispensables commentaires sur les importants avantages des procédés que nous avons présentés, ainsi que, bien évidemment, les définitions et descriptions complètes de toutes les caractéristiques et autres fonctionnalités que nous avons réussi à attacher à ce modèle principal.

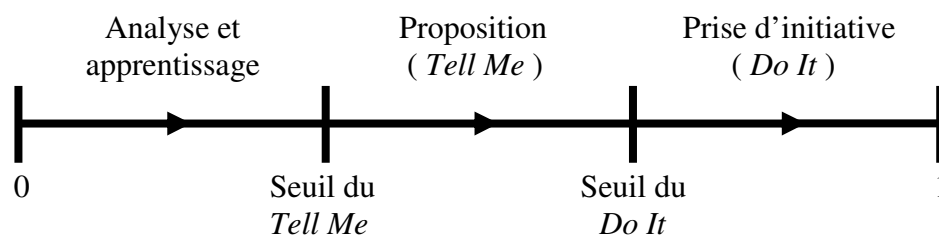


Figure 4.5.3.2 : Modélisation chronologique des fonctions "Tell Me" et "Do It"

Suite à ses apprentissages aux fortes conséquences dynamiques et comportementales, la machine devient ainsi douée d'une toute nouvelle capacité de raisonnement qui lui permet de comprendre, de décider et de réaliser des opérations en temps réel, seule et sans aucune assistance extérieure, mais toujours en complet accord avec les désirs de l'utilisateur concerné et des préférences de celui-ci recueillies au cours des précédentes étapes.

Cependant, il apparaît que nos différents mécanismes d'apprentissage, en plus d'être automatiques, deviennent aussi incrémentaux. En effet, non seulement tous ces apprentissages se réalisent sans aucune difficulté en temps réel au cours du travail de chaque utilisateur, mais en plus ces derniers quittent notre interface intelligente en s'étant enrichis de connaissances supplémentaires, concrètement représentées par un ensemble acquis de séquences d'actions. Utilisées en parfaite adéquation, les fonctions "Tell Me" et "Do It" [Lieberman, 1997] définissent en conséquence un nouvel agent intelligent autonome, dit "d'interface", qui réalise, comme nous le souhaitons, de constantes suggestions et manipulations en temps réel. Ces opérations sont alors toujours susceptibles d'intéresser l'utilisateur pendant son travail : il s'agit donc bien d'une véritable prédiction évoluée, auto-adaptable, proposée à l'utilisateur pour chaque action réalisée.

En outre, cette méthode présente un second avantage tout aussi important puisqu'elle permet aussi de regrouper deux types inséparables d'agents que sont les agents "d'interface" et les agents "autonomes". Si, dans une interface intelligente, les premiers apportent à l'utilisateur l'assistance active, les seconds travaillent sans faire intervenir l'utilisateur, qu'il soit occupé ou non : la conversation, l'interaction avec l'utilisateur continue, "naturelle" et fluide. Et grâce à l'utilisation de seuils précis entre l'apprentissage, la suggestion et la prise d'initiative, nos agents autonomes d'interface sont *a priori* bien outillés pour minimiser au maximum l'effet d'hésitation entre réflexion et action qu'ils peuvent parfois avoir à gérer lors de leurs vies respectives.

Nous pouvons conclure que l'interface présente d'intéressants aspects auto-adaptables, comme le fait d'agir automatiquement non sans avoir auparavant suggéré, comme la manière également de tirer avantage de l'information que l'utilisateur courant lui fournit librement, sans avoir recours à une interaction séparée. C'est à notre avis le point essentiel de nos modèles et prototypes construits lors de cette Thèse, à savoir qu'ils sont tous réellement non intrusifs pour un travail serein mais toujours enrichissant et donc agréable pour chacun de nos utilisateurs.

Ceci entériné, il nous reste à nous pencher plus avant vers la description fonctionnelle et algorithmique de toutes les différentes parties composant notre complet modèle comportemental de l'utilisateur. En effet, nous allons maintenant détailler nos imposants procédés de prédiction et d'anticipation auto-adaptable, de simplifications et corrections avancées ensuite, ce pour aboutir enfin à la description du cœur-même de notre modèle par l'explication des mises en œuvre de la modélisation du comportement et l'ajustement dynamique final de la relation homme/machine.

4.5.3.3. Définition fonctionnelle des mécanismes de prédiction et d'anticipation

En rappelant que le principe d'une méthode de prédiction s'attache, avant toute chose, à déterminer l'entité suivante dans une séquence d'éléments simples dits aussi "nominiaux" [Davison & Hirsh, 1998], nous découvrons diverses caractéristiques intéressantes d'algorithmes qui, appliqués à des séquences d'actions, vont permettre aux interfaces-utilisateur prises en compte de chercher à s'adapter à un modèle d'utilisation cette fois-ci individuel. Dans ce but, nous avons élaboré et développé un module de prédiction et d'anticipation auto-adaptables, déjà présentées

au travers de la **Figure 4.5.3.2** précédente. La **Figure 4.5.3.3** nous en apporte maintenant tous les détails significatifs, de même que les plus profonds fondements et raisonnements.

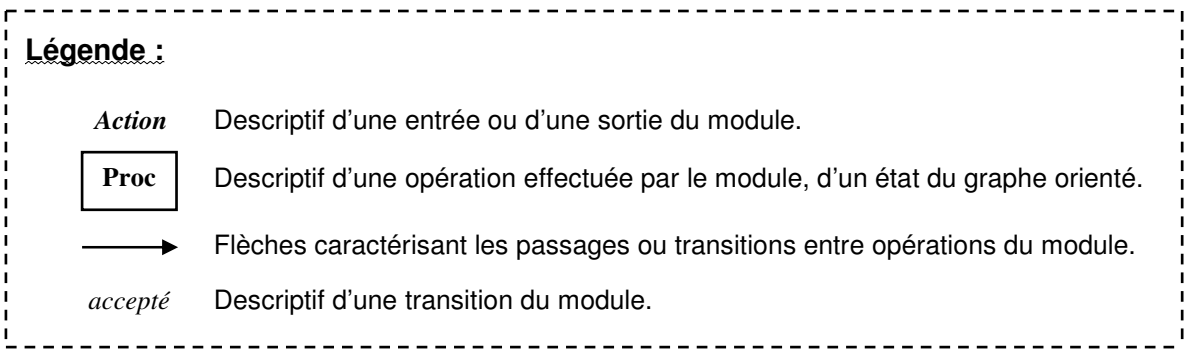
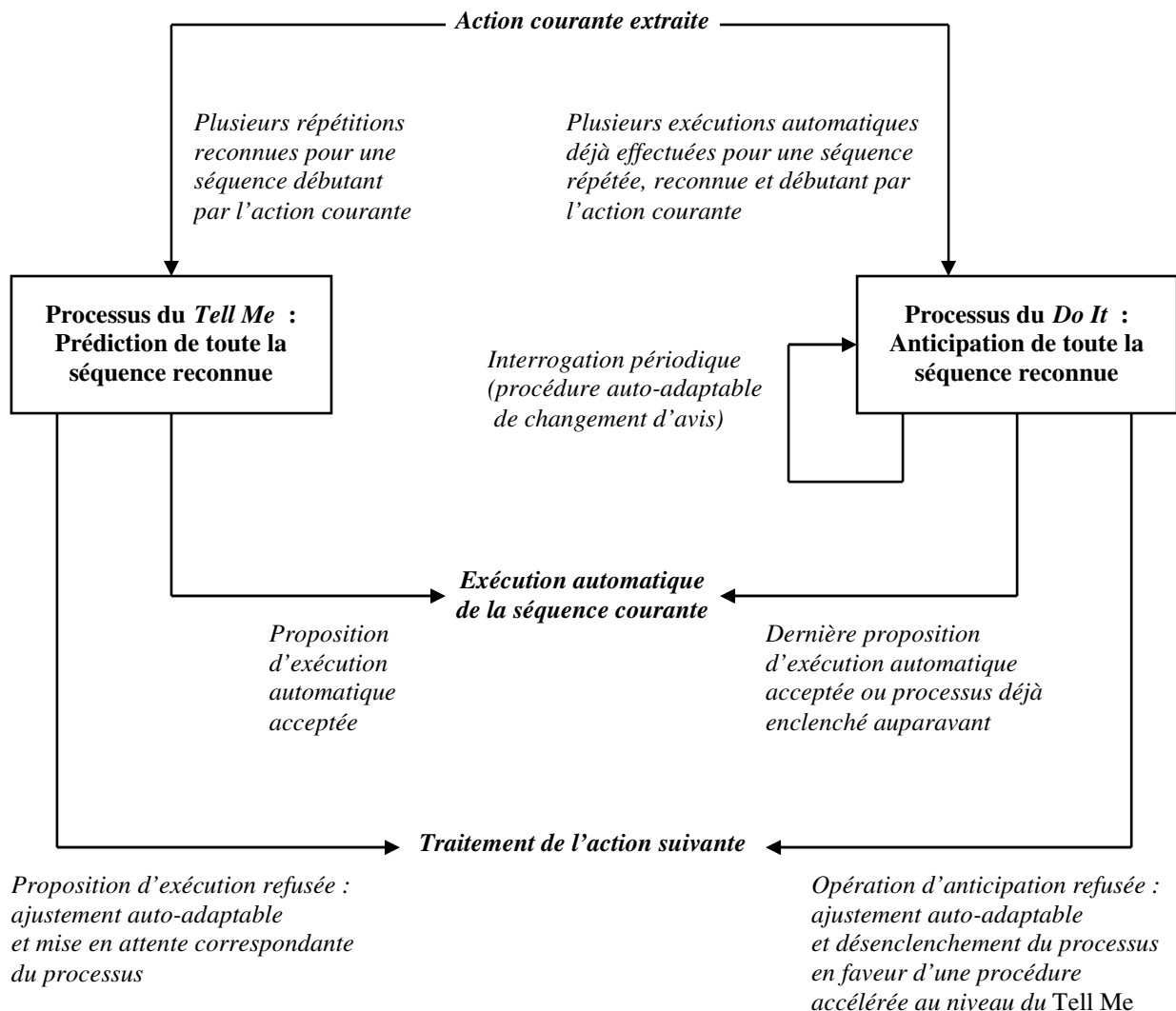


Figure 4.5.3.3 : Schéma fonctionnel du module intelligent de prédiction et d'anticipation de séquences répétées d'actions

En premier commentaire, nous pouvons préciser, en relation directe avec notre concept d'extraction des actions effectuées par nos utilisateurs, que ces nouveaux processus de prédiction et d'anticipation intelligentes ne sont justement en rien des actions proprement réalisées par l'homme, puisque c'est effectivement la "Couche IA" qui les propose et les active. Par conséquent, les propositions successives, attachées à nos deux mécanismes, ne sont pas prises en compte dans la liste uniquement dédiée aux actions réalisées par les utilisateurs eux-mêmes, en notant toutefois que toute action ou séquence exécutée de manière automatique sera bien entendu, au contraire, répertoriée (car cette fois-ci pleinement souhaitée par l'utilisateur).

Ceci indiqué, nous nous devons aussi de relever ici un autre point remarquable, puisque nous avons réussi à implémenter, au sujet du module étudié à ce stade, deux versions différentes mais combinables et même complémentaires comme nous avons pu progressivement le constater. En effet, au vu des nombreuses idées que nous avons pu rassembler dès le début de la Thèse, nous avons permis à notre "Couche IA" d'effectuer des démarrages, déclenchements de propositions et autres automatisations inhérentes, non seulement au niveau de la première, mais aussi de la seconde action composant chaque séquence d'actions reconnue. Pour davantage d'explications à ce propos, nous devons rappeler que le lancement de nos processus de prédiction et d'anticipation se réalise, dans le cas général, dès la détection de la première action d'une séquence susceptible d'être exécutée de manière auto-adaptable. Cependant, une telle opération repoussée au niveau de la reconnaissance de la seconde action composant la séquence s'est, elle aussi, avérée judicieuse. L'exécution par l'utilisateur des deux premières actions d'une séquence est alors requise pour entraîner une éventuelle automatisation. Si ce cas est notamment attrayant pour des sélections de messages où peu de similitudes de séquences existent dès la première des actions, il nous faut en outre éviter de perturber la lecture en temps réel du message concerné.

Dans le même ordre d'idée, nous avons par suite envisagé la mise en œuvre d'une liste répertoriant, pour nos utilisateurs respectifs, les séquences automatisées leur correspondant. Cela se matérialiserait, par exemple, sous la forme d'un menu auto-adaptable où l'utilisateur pourrait cocher et décocher des séquences afin de les rendre automatisées ou non. Mais cela s'est, en définitive, avéré beaucoup trop astreignant pour l'utilisateur. En effet, il s'agit là, ni plus ni moins, d'une manipulation supplémentaire à effectuer par ce dernier, avec prise d'initiative propre à lui-même et donc sans proposition auto-adaptable de la part de la machine.

En conclusion, nous choisissons de conserver ici trois des caractéristiques essentielles et représentatives des recherches réalisées dans le cadre de l'implémentation de notre module de prédiction et d'anticipation auto-adaptables. En effet, les qualités de notre système demeurent, à la fois :

- la capacité régulièrement apportée à l'utilisateur de changer d'avis quand au travail effectué par nos processus de prédiction dédiée, et surtout d'anticipation auto-adaptable pourtant fortement automatisée.
- l'appréhension et le pré-traitement des différents refus et acceptations successives émanant de l'utilisateur en matière de prédiction et d'anticipation des séquences d'actions (cf. d'ailleurs encore à ce sujet les copies d'écrans relevées par les **Figures 5.2.2.4 et 5.2.2.5** pour bien exposer les caractéristiques algorithmiques et la mise en œuvre de ce processus).
- le contrôle permanent mis en place pour éviter d'automatiser un maximum d'erreurs ou d'incohérences effectuées par les divers utilisateurs (cf. **4.5.3.4** sur la page à venir).

Sans oublier que la partie **5.3.3** dédiée à notre étude-utilisateur, mais aussi l'**Annexe 3** et l'**Annexe 4**, se chargeront encore de démontrer, sur un large panel de données réelles, les pleines efficacité, discrétion et adéquation de telles fonctionnalités de nos algorithmes auto-adaptables.

4.5.3.4. Procédé de simplifications auto-adaptables

Facilitant le travail quotidien de chaque individu, néophyte comme expérimenté, notre système intelligent fournit un soutien convivial, rapide, fiable et efficace. Néanmoins, de par ses dernières capacités de simplifications auto-adaptables, il permet également aux divers utilisateurs des terminaux "grand public" et autres logiciels de communication d'ALCATEL de remédier à l'exécution d'une majorité de leurs erreurs. Celles-ci peuvent alors correspondre à des accidents de manipulation comme à des égarements dûs à de mauvaises compréhensions.

Ce nouveau processus intelligent se rapproche des deux principaux types de modèles issus de l'Enseignement Assisté par Ordinateur, à commencer par ceux "d'expertise partielle" ou "de superposition" [Bruillard, 1997]. Structure de données comprenant des connaissances et des compétences élémentaires à acquérir, et ce tout en spécifiant, pour chacune d'elles, le degré de maîtrise de l'apprenant, ce modèle traite, avant toute chose, le fait que l'élève concerné présente des lacunes ou des connaissances encore mal assurées, en quelque sorte des faiblesses, qu'il s'agit d'identifier pour lui permettre de progresser. Si l'objectif d'un tel système tuteur est alors de compléter les connaissances de l'élève pour qu'il acquière les connaissances précisées dans le modèle, la mise à jour du modèle est aussi réalisée par détection des faiblesses de l'apprenant en analysant les actions qu'il effectue afin d'évaluer leurs pertinences respectives. Et ce type de procédé se traduit, dans notre cas, par une extraction et une classification des connaissances focalisées sur les redondances d'informations et l'utilisation d'actions-clé (notamment associées à un retour en arrière) émanant de nos différents utilisateurs. La difficulté de prise en compte d'un tel modèle réside toutefois dans le fait qu'un mauvais choix peut impliquer plusieurs compétences élémentaires : pas facile en effet de déterminer quelle(s) compétence(s) l'apprenant ne maîtrise pas. Et le problème demeure analogue lorsque plusieurs méthodes permettent, en cas de choix multiples et tous corrects, d'aboutir au même résultat.

Nous avons donc opté (cf. les détails de l'**Annexe 4** et les travaux effectués en **4.4.1**) pour relever et simplifier les séquences présentant des faiblesses en nous aidant d'actions que nous avons définies de "remarquables", ce dans le but de différencier les multiples cas probables de manipulations "erronées". Par suite, la conséquence principale, au niveau du traitement et de l'exploitation de ces résultats dans le cadre de notre Modèle Auto-Adaptable de l'Utilisateur, est aussi apparue très intéressante, puisque nous avons pu, de manière standard, créditer ou "blâmer" toutes les compétences de nos utilisateurs en cas de succès ou d'échec, ce qui est revenu pour nous à incrémenter ou décrémenter toutes sortes de compteurs associés à chacune des qualités. Et, nous le verrons au cours des prochaines pages (en **4.5.3.6** surtout), cela s'est avéré d'autant plus efficace qu'il y a eu et aura toujours assez de situations discriminantes pour que les forces et faiblesses, c'est-à-dire les caractéristiques de chaque comportement d'utilisateur, se distinguent.

Nos modèles auto-adaptables se rapprochent également beaucoup des modèles qualifiés de "différentiels" et dont les stratégies sont basées sur la remédiation. De nombreuses erreurs ne sont effectivement pas dues à un comportement erratique des apprenants, mais à l'application correcte de procédures fausses. Nous proposons donc à l'utilisateur, au travers de l'apprentissage automatique et de l'interaction intelligente, de redéfinir, voire de modifier, les séquences d'actions montrant des imperfections de type redondances d'informations ou retours en arrière inutiles.

A noter cependant qu'une difficulté de taille apparaît parfois au niveau de ce modèle théorique puisqu'étiqueter un comportement ne donne pas forcément, ni directement d'ailleurs, les informations nécessaires pour choisir une "thérapeutique" adaptée ou le niveau d'intervention réellement associé à une erreur [Bruillard, 1997] ... Des ajustements pratiques, qu'ils aient été d'ordre comportemental (**Chapitre 3**) ou liés à la pertinence et à l'efficacité d'exploitation de nos présents procédés de simplification auto-adaptable, ont donc été nécessaires à ce niveau.

En conclusion, la comparaison de nos agents auto-adaptables de simplification avec ces deux importantes approches de modélisation s'appuie sur la vision du système "enseignant" : recherche des faiblesses, lacunes et erreurs se mêlent en effet à la détermination des solutions ou actions à réaliser en correction. Une différence notable subsiste cependant avec l'EAO, à savoir que tout n'a pas besoin d'être entièrement prévu à l'avance puisque les détections de lacunes ou d'erreurs comme les décisions d'actions, ou les réalisations de séquences d'actions, peuvent être effectuées (et le sont toutes dans nos modèles) de manière dynamique comme en temps réel.

4.5.3.5. Mise en œuvre de la modélisation du comportement des utilisateurs

Nous avons opté, lors des études du **Chapitre 3**, pour la détermination de la compétence et du comportement de nos utilisateurs selon deux aspects distincts : l'expérience (connaissances) et le caractère d'ouverture ou d'intéressement (facilitation des attentes d'interactions et de requêtes). La modélisation est alors principalement axée sur l'établissement d'un couple "données/comportements" à partir des informations recueillies en temps réel des manipulations successives de l'utilisateur, puis des réactions, attitudes et caractères associés à ces actions par notre modèle auto-adaptable. L'utilisateur devient ainsi un objet modélisé et interagit avec son propre environnement de travail qui va se voir adapté à ses habitudes, ses buts, ses compétences.

Pour cela, nous nous sommes notamment appuyés sur les propriétés très intéressantes de la catégorisation des séquences extraites [Michel, 2000]. Chacun de nos utilisateurs se trouve alors, non seulement caractérisé par le Modèle Auto-Adaptable de l'Utilisateur, mais aussi et surtout directement à l'origine des modifications et des adaptations apportées à l'interface intelligente le concernant. Ces dernières sont issues d'une analyse poussée des classes de séquences d'actions obtenues de toutes nos précédentes opérations et des comportements inhérents attribuables à cet utilisateur. Les **Figures 5.2.5.2 et 5.2.5.3** du prochain chapitre demeurent très explicites à ce sujet, et permettront *a fortiori* d'entériner ces caractéristiques propres à notre modèle comportemental.

D'un autre point de vue, nous aurions pu, comme pour la partie précédente consacrée à nos procédés de simplification auto-adaptable, tenir compte du temps mis par chaque utilisateur pour effectuer ses actions en général, du temps de réaction aussi au niveau des formations de séquences ou fréquences de répétitions de chaque séquence en particulier. Mais trop aléatoire, et surtout lié à beaucoup trop de facteurs aux explications multiples et autres causes variées (fatigue, maladie, dérangement involontaire ou extérieur, travail en multi-tâches, etc), cette idée nous est finalement apparue bien trop irréaliste et fautive, voire pas suffisamment rationnelle ni proprement "scientifique", pour être intégrée à nos concepts et modules implémentés.

4.5.3.6. Ajustement dynamique final de la relation homme/machine

A ce stade de nos investigations, et afin d'informer l'utilisateur de manière aussi peu intrusive pour son travail, mais tout-de-même en temps réel et à chaque fois que cela s'avère judicieux (détection d'une prédiction envisageable, d'une anticipation attendue ou encore d'une intéressante proposition ou automatisation de simplification adaptée), nous avons été amenés à intégrer à nos divers agents intelligents plusieurs formes de dialogue.

Toutes répertoriées au sein de l' **Annexe 4** dédiée à la validation de nos implémentations, ces différentes interfaces de communication mises en place pour nos utilisateurs respectifs ont été orientées, en permanence, vers la construction d'un dialogue aussi efficace et convivial que possible. Emanant de notre système multi-agents, ce dialogue se devait en effet de correspondre au maximum aux attentes des utilisateurs, en tenant compte de la diversité de ceux-ci mise en exergue auparavant par notre processus de modélisation des comportements humains rencontrés. Ainsi, avec l'obtention d'un dialogue interactif toujours agréable, apparaît la possibilité d'éviter de contraindre l'utilisateur à apprendre un langage plutôt "barbare" pour dialoguer avec son application [Stémart, 1997] : nous restons en conséquence suffisamment éloigné des concepts proprement dits du dialogue à utiliser, ce afin d'éviter justement que l'utilisateur ne se sente dépassé par la théorie. Nous demeurons au contraire assez proches de la représentation qu'il se fait naturellement du dialogue, vision donc aisée et explicite pour lui. La qualité du dialogue est évaluée en fonction de la compréhension et de la capacité d'un utilisateur à accomplir sa tâche.

Le rôle joué par l'implémentation en temps réel de tels outils de communication est alors délicat puisque ceux-ci doivent demeurer accessibles pour chaque "type" d'utilisateur, aussi expérimenté que relevant du domaine des personnes néophytes, toujours nombreuses au niveau du "grand public". De plus, toutes les sortes d'explications, que nous avons choisi de porter à la connaissance de nos utilisateurs, se doivent d'être facilitantes, non-ralentissantes et adéquates au travail en cours, la compétitivité correspondant bien entendu à l'un des principaux critères pour une application de qualité.

En outre, ce dialogue régulier entre la machine et l'utilisateur est presque indispensable de nos jours puisque ce dernier a de moins en moins conscience des diverses capacités issues des systèmes qu'il manipule, et n'a donc pas non plus toujours une maîtrise totale des outils mis à sa disposition [Stémart, 1997]. Des explications auto-adaptables s'imposent alors tout autant que des apprentissages destinés à engendrer des propriétés de même équivalence. Œuvrant ensemble pour satisfaire le même but que celui de l'utilisateur concerné, ces outils adaptés à la communication avec l'homme maintiennent la cohérence des dialogues et l'adéquation entre les interactions, l'interprétation que le système en fait ainsi que la traduction des résultats obtenus. Cette capacité à comprendre l'utilisateur à tout moment correspond à un des points essentiels de nos modèles.

Ceci établi, nous découvrons, par extension, que ces procédés de communication auto-adaptable marquent également très bien l'indépendance que nous visons depuis le début de la Thèse entre l'application traitée et notre modèle d'interface intelligente. En effet, ils permettent de fortement insister sur la réalisation de nos actions globales d'Intelligence Artificielle tout en respectant le déroulement des actions propres aux outils de manipulation mis en avant par l'application utilisée. En outre, si la fenêtre de communication courante ne convient pas, elle est modifiée ou changée sans que le reste des actions, globales comme de base, en soient affectées, et, de ce fait, les fondements de l'application considérée ne sont aussi aucunement altérés.

L'utilisateur n'est pas non plus gêné par nos outils auto-adaptables de communication, c'est-à-dire ni préoccupé ni contraint par eux : le choix de les utiliser, quelle que soit sa catégorie déterminée par notre modèle de l'utilisateur, lui est toujours laissé. Et si nous avons décidé de considérer l'utilisateur, de focaliser l'ensemble de nos travaux sur ses propres préoccupations, la communication homme-machine mise en place de manière auto-adaptable est constamment le reflet des capacités, non seulement des utilisateurs, mais également de nos agents intelligents. Cette orientation complète du système en fonction des actes de l'utilisateur courant, de même que les aides pertinentes et implicites apportées à celui-ci, entraînent alors des solutions sérieuses pour répondre au paradigme spécifié par « fournir les bons outils au bon moment » [Stémart, 1997].

Enfin, nous sommes amenés à établir ici un complet "Retour Intelligent d'Informations", inhérent à l'innovante catégorisation maintenant disponible pour chaque utilisateur. En effet,

nous ne pouvions nous résoudre à ne pas exploiter les intéressants résultats offerts par notre modèle auto-adaptable. Un ultime traitement logiciel, une nouvelle fois "intelligent" car destiné à classifier les différentes valeurs obtenues de ce modèle, nous a permis d'aboutir à un concept d'interface encore plus riche, au sein duquel l'utilisateur dirige implicitement la détermination dynamique (en temps réel) des différents paramètres d'interaction caractérisant sa relation courante avec la machine. Sans oublier également que ces dernières investigations ont encore demandé l'établissement de plusieurs protocoles d'ajustement et de production nouvelle des données prises en compte, tels ceux définis par le **Chapitre 5** à suivre et l'**Annexe 3**.

4.6. Conclusion

A l'image de l'introduction de techniques de traitement des connaissances dans le vaste domaine de l'Enseignement Assisté par Ordinateur, nous avons tout d'abord dû relever que les machines actuelles ne doivent plus seulement gérer leurs domaines d'apprentissage et résoudre une multitude de problèmes, mais sont aussi amenées à disposer de données sur leurs propres connaissances afin de pouvoir justifier et expliciter leurs solutions respectives [Bruillard, 1997]. Fondée alors sur un transfert de compétences entre un système intelligent et un humain par l'intermédiaire d'une représentation en machine, cette dernière justification des divers choix effectués, ainsi que la nécessité liée d'ajouter un maximum d'explications de nature causale à l'interaction concernée, se doivent en effet, à notre époque, d'être toujours plus complètes, ce pour demeurer compréhensibles et pleinement adéquats aux différents utilisateurs.

Autre point essentiel mais directement issu de ces principes, la résolution d'un problème apparaît également basée sur deux étapes : trouver une représentation efficace à tous les niveaux d'exécution des actions de l'utilisateur, puis choisir les bonnes transformations ou méthodes à appliquer, notamment en ce qui concernent les modes de résolution (et nombreux enseignements associés) de nos bases de données extraites des actions réalisées en temps réel sur l'interface.

Ce quatrième chapitre apparaît alors très important dans la présente Thèse : s'il justifie déjà scientifiquement nos différents algorithmes mis en œuvre, il assoit aussi l'implémentation de nos concepts auto-adaptables sur des bases bien solides. Pour un exemple particulièrement représentatif, nous avons pu remarquer qu'un agent logiciel et intelligent se voit efficacement qualifié par un apprentissage dit "incrémental", à l'image donc des entités que nous avons choisi d'implémenter. Plutôt que d'appliquer alors un algorithme d'apprentissage à l'ensemble entier des exemples chaque fois qu'une nouvelle prédiction est à effectuer, l'agent logiciel peut, suivant cet algorithme incrémental, juste essayer de mettre à jour ses anciennes hypothèses dès qu'un nouvel exemple arrive [Russell & Norvig, 1995]. De plus, nous verrons que nos agents vont aussi pouvoir grandement bénéficier d'une sorte de "retour", une judicieuse "rétroaction" concernant la qualité des actions qu'ils ont choisies par le passé.

Si l'apprentissage automatique devient ainsi une capacité fondamentale permettant à un système artificiel d'améliorer ses comportements au fil de son expérience [Ketterlin, 1995], nos discussions et mécanismes attachés à nos concepts auto-adaptables sont à présent tous spécifiés ; diverses approches et méthodes applicables ayant été précisément étudiées jusqu'ici. En outre, nous pouvons retenir ici différents indicateurs qui marquent la nécessité d'accentuer les travaux dans ce cadre précis de l'apprentissage automatique : l'augmentation vertigineuse du nombre de bases de données concernant tous les domaines de l'activité humaine (liée au développement du

World Wide Web notamment), les besoins d'autonomie des machines qui doivent pouvoir réagir seules face à des problèmes, et la capacité d'améliorer automatiquement ses performances au cours du temps [Sebban & Venturini, 1999]. Les utilisateurs ont en effet besoin aujourd'hui d'outils effectifs et aisés d'emploi afin de permettre à chacun de déceler l'information pertinente, de réaliser des synthèses et d'interpréter les résultats sans ambiguïté [Sentient Machine Research, 1999].

Nous assistons en conséquence, accentué par la vie trépidante que nous menons tous à l'époque qui est la nôtre, à un besoin émanant de la plupart des utilisateurs et qui est d'éviter une trop grande et fastidieuse séparation entre la formation et la manipulation proprement dite d'un contexte immédiat de travail. Dans tous les domaines informatisés, c'est-à-dire s'appuyant sur ou directement liés à l'informatique comme par exemple les logiciels de messagerie électronique, l'aspect de l'apprentissage doit se confondre dans l'interaction-même avec les autres aspects (documentation, aide, explication, etc) inclus dans l'environnement quotidien de nos actions. L'apprentissage permet en effet à l'être humain, dans la complexité de sa pensée, d'améliorer l'état de ses connaissances et la qualité de ses performances dans la prise de décision ou l'action. La tâche cognitive qu'est l'apprentissage correspond alors à l'ensemble des processus permettant de tirer profit de l'expérience passée pour traiter une nouvelle situation [Haton & Smaïli, 1996].

C'est pourquoi, nous avons veillé à ce que les implémentations de notre complet modèle d'interface intelligente traduisent des fonctionnalités toujours agréables et opportunes pour les utilisateurs, tout en permettant aussi d'assister et de grandement améliorer les capacités de travail de ces derniers. Par extension, nous nous sommes constamment attachés à remédier, parfois en temps réel, aux difficultés pouvant apparaître dans l'usage des diverses activités de nos modèles auto-adaptables, du fait par exemple d'une mauvaise maîtrise ou compréhension par l'utilisateur des concepts proposés. Toutes nos interactions et interventions intelligentes demeurent ainsi liées à de fines descriptions, informations et explications propres à l'état courant de tous nos modèles.

Nous nous orientons à présent vers l'implémentation complète, suivie de toutes ses validations, concernant l'ensemble de nos différents modèles auto-adaptables spécifiés jusqu'ici. Ainsi, nous allons définir nos modèles d'application et leurs interconnexions respectives avec l'homme, formaliser toutes les théories et ensembles de données manipulés en permanence par ces structures, et exposer enfin nos divers systèmes et algorithmes d'apprentissage automatique, de prédiction et d'anticipation auto-adaptables appliqués aux séquencements d'actions. Il nous reste à souligner aussi que nos nouvelles préoccupations, et autres perspectives de recherches, deviennent à ce stade la description en détail de l'implémentation intégrale réalisée à partir de tous nos éléments ainsi que nombre d'analyses de validation inhérentes aux mécanismes auto-adaptables que nous venons de mettre en évidence au travers des quatre précédents chapitres.

Par conséquent, l'essor actuel du gigantesque domaine de l'informatique et de toutes ses applications logicielles associées élargit considérablement le champ des utilisateurs potentiels et susceptibles d'être réellement intéressés par notre modèle d'interface intelligente ainsi développé. Ces usagers n'ont, en outre, ni le temps ni aussi parfois les compétences nécessaires pour l'acquisition de savoir-faire trop complexes. C'est pourquoi, nous découvrirons relativement rapidement que les nombreux raccourcis, alternatives, fonctionnalités nouvelles et assistances auto-adaptables que nous proposons au travers de nos modules implémentés au sein des produits d'ALCATEL, entraînent des démarches et des réactions plutôt pertinentes et très majoritairement appréciées. Et nous verrons que nos modèles auto-adaptables apparaissent finalement comme pleinement remplir leurs fonctions d'assistant intelligent en apportant un soutien à l'ensemble de la population considérée.

Chapitre 5 :

Implémentation et Validation des Prototypes

Parce que le développement de logiciels est devenu un des principaux goulots d'étranglement de la technologie de l'informatique d'aujourd'hui, l'idée d'introduire de la connaissance dans des ordinateurs semble particulièrement attrayante et fait appel au bon sens.

[Michalski, Bratko & Kubat, 1998]

5.1. Contraintes technologiques de l'implémentation

5.1.1. Retour sur les modèles d'application accueillant nos prototypes

Après avoir étudié les fondements théoriques apportant diverses formes d'intelligence à notre nouvelle interface, cet ultime chapitre traite des recherches au sujet de la programmation, de l'implémentation, de la mise en œuvre et de la concrétisation de nos modèles auto-adaptables. Ces travaux ont été suivis de nombreux tests disparates, qui ont tous été menés à bon terme, ainsi que d'une validation très complète y ayant été inévitablement associée au sein d'ALCATEL, dans un contexte de production industrielle inhérent à de tels prototypes.

Mais, rappelons au préalable quelles ont été précisément les ressources matérielles dont nous avons pu disposer pour opérer ainsi sur les différentes applications d'ALCATEL, dédiées avant tout à la messagerie électronique et déjà abordées lors du premier chapitre de ce mémoire. S'il apparaît alors que nos travaux ont été réalisés au sein des deux Départements "Recherche" et "Applications" d'ALCATEL basés à Illkirch-Graffenstaden (Bas-Rhin), en entière collaboration avec le Groupe de Recherche en Intelligence Artificielle du LSIIIT établi, lui, à l'Université Louis Pasteur de Strasbourg, les moyens matériels mis en œuvre se sont principalement composés d'un ordinateur personnel de dernière génération, ainsi que d'une machine SUN Microsystems ayant travaillé sous Solaris (version 1.2) avec les langages "objets" Java (Java Development Kit, version 1.1.7A notamment) et C++ (Microsoft Visual C++ 5.0 plus particulièrement).

Il était alors avancé, dès le début de la Thèse, de tenter d'aboutir par nos recherches à l'implémentation de diverses fonctionnalités spécialisables sur les terminaux de communication bien en vue au sein du Département "Recherche" d'ALCATEL, à savoir les *Internet Screenphone* (ALCATEL 2830, 2840, 2850, puis la gamme *WebTouch* aujourd'hui fabriqué, en partenariat avec THOMSON Multimédia, au sein de la Société ATLINKS) travaillant notamment sous *Personal Java*. Mais les nouveaux choix stratégiques pris, un an et demi après, par la direction d'ALCATEL pour faire face aux très rapides évolutions du contexte planétaire des télécommunications ont entraîné

d'importantes modifications des priorités et structures de travail au sein de toute l'entreprise. Dès lors, nous avons opté, afin de conférer à l'ensemble de nos recherches un maximum d'intérêts et de légitimité, pour une adaptation immédiate de celles-ci aux réalités de notre domaine industriel. Ces importants changements nous ont, entre-autres, finalement amené à découvrir et surtout à implémenter nos différents modèles auto-adaptables au niveau d'une nouvelle application de messagerie électronique et unifiée, appelée *ALCATEL Unified Messaging (AUM)*.

Cette dernière, en correspondance avec l'étude détaillée que nous en avons alors faite (cf. **1.1.3**), nous est même apparu très bien correspondre aux différents besoins, contraintes et objectifs que nous nous étions fixés. En effet, si l'*AUM* regroupe déjà trois médias distincts issus directement de la messagerie (*e-mail*, *voice mail* et *fax*), rappelons brièvement que ce logiciel travaille également de concert avec les fonctions de la base de données et le réseau unique d'administration pour stocker et manipuler l'ensemble des messages en une seule et même interface-utilisateur. Ce système d'*ALCATEL* présente donc de nombreuses et très intéressantes caractéristiques, comme encore une optimisation des différents moyens de communication, une uniformisation du format des messages et une amélioration du flux des informations.

Précédée d'une indispensable formation individuelle sur les matériels logiciels *COM* et *MAPI* (que nous détaillons tous deux à suivre) ainsi que sur les dernières versions de la plateforme de développement *Visual Studio*, cette approche s'est alors traduite par l'implémentation d'un premier prototype d'interface chargé d'extraire et de répertorier les différentes actions ou événements successifs issus du travail d'un utilisateur, ce en respectant une constante mise en adéquation de l'architecture de ce système avec celle déjà actuellement manipulée au sein du *Business Applications Department* d'*ALCATEL*.

Cette phase de mise en pratique d'une partie déjà de nos modèles théoriques a permis ensuite de nouvelles implémentations, beaucoup plus conséquentes, à savoir celle tout d'abord de l'apprentissage automatique des diverses séquences d'actions effectuées de manière répétitive par chaque utilisateur au cours de ses manipulations de l'*ALCATEL Unified Messaging*, puis surtout celles de la construction d'un modèle comportemental complet composé de la prédiction, de l'anticipation et de la simplification, toutes auto-adaptables, des séquences d'actions réalisées. Ainsi, nous avons abouti au principal but que nous avons défini en début de Thèse : la mise en œuvre complète d'un modèle du couple "utilisateur-application", et ce par une prise en compte non-invasive pour l'utilisateur de nos différents modèles caractérisant une Interface Intelligente pour Terminaux de Communication.

5.1.2. Les langages *COM* et *MAPI*

Pleinement liés à la programmation orientée "objet" de l'*ALCATEL Unified Messaging* et plus particulièrement à l'intégration logicielle de ses différents médias de télécommunication, ces deux outils d'implémentation, très complémentaires, constituent également la base-même des multiples langages caractérisant aujourd'hui l'ensemble des applications dédiées à la messagerie électronique en général, et émanant en outre de l'entreprise américaine *Microsoft Corporation*.

Plusieurs raisons de tout premier plan nous ont amenés à opter pour un travail avec les outils de cette dernière société, et donc plus particulièrement avec *COM* et *MAPI*. De plus, si nos présentes explications apparaissent d'emblée très importantes, ce choix a également été réalisé dans un contexte traitant, à la fois, de nos besoins scientifiques et technologiques.

En effet, un des catalyseurs principaux de nos décisions est bien sûr resté la recherche d'un lien, aussi fiable que possible, avec nos propres modèles d'application issus de l'*AUM*. Si des composants de base autres que ceux proposés par *Microsoft* auraient alors pu nous convenir (dont *Linux* et *Unix* notamment), *COM* et *MAPI* sont finalement apparus comme étant les seuls procédés actuels permettant d'opérer et de programmer de manière aussi avancée dans le cadre de la messagerie unifiée que dans le domaine de la téléphonie en général. Par ailleurs, l'initialisation et la mise en œuvre de nos systèmes intelligents distribués n'auraient jamais pu se construire efficacement sans un accès indispensable aux machines, bases de données et sources déjà développées au sein de nos équipes d'*ALCATEL*, en partenariat privilégié avec *Microsoft*.

A relever enfin ici que, pour davantage de détails à ce sujet, nous nous en remettons à l'importante étude que nous avons menée à propos de toutes les intéressantes capacités de ces deux langages, et que nous rapportons dans la partie **A3.1** de l'*Annexe 3*.

Signifiant très précisément "Modèle d'Objet Composant" (*Component Object Model*), *COM* demeure essentiellement destiné à développer des agents ou composants logiciels, en d'autres termes de petits fichiers exécutables qui fournissent des services pour des applications, des systèmes d'exploitation, mais aussi, par héritage et spécification notamment, pour d'autres composants plus conséquents pouvant les englober. Si développer et connecter ensemble des composants *COM* revient alors à développer une application véritablement dynamique et orientée "objet" [Rogerson, 1997], les composants *COM* peuvent aussi être "individuellement" déconnectés et remplacés en temps réel, sans relier ou recompiler l'application considérée.

S'appuyant entièrement sur la technologie *COM*, nous abordons, dans un second temps, le langage *MAPI* qui désigne une "Interface de Programmation pour Application de Messagerie" (*Messaging Application Programming Interface*). Celle-ci se définit par une couche de niveau supérieur à celle de *COM* puisque, au contraire de ce dernier qui fournit un modèle direct et puissant pour implémenter des ensembles sophistiqués d'applications, *MAPI* s'oriente avant tout vers les processus d'interfaçage de la messagerie considérée.

Par suite, les langages *COM* et *MAPI* ont été développés pour fournir une interface standardisée du niveau d'une application et qui laisse des objets ou composants de messagerie dialoguer avec des systèmes de messagerie électronique largement incompatibles. *Microsoft Corporation* aime d'ailleurs à relever que *MAPI* a été écrit en collaboration avec plus de cent vendeurs de logiciels, et représente ainsi un vrai et large consensus industriel des caractéristiques que la nouvelle génération des systèmes de messagerie devrait fournir.

Si *COM* spécifie alors, tout d'abord, le mécanisme pour la création et la destruction de chaque composant de messagerie, *MAPI* définit, de son côté, la forme et surtout la fonction de ces divers objets. A remarquer que *MAPI* fournit également un modèle standard d'abstraction et d'encapsulation des données, en plus de l'utilisation de toutes les qualités de *COM*, par couche superposée. Sans oublier que, si *MAPI* définit précisément un ensemble d'attributs nommés "propriétés" ayant des rôles spécifiques et des types de données prédéfinis, les capacités d'un objet sont analogues aux données privées dans une classe de *C++* : nous pouvons y accéder en appelant des méthodes publiques au niveau de l'interface de propriété de l'objet concerné.

De spécification et de programmation certes parfois très complexes, l'association *COM-MAPI* nous est cependant apparue très efficace dans nos deux domaines que sont les télécommunications et l'informatique. Outil robuste et puissant, *MAPI* brille par son architecture "client/serveur" résolument destinée à palier l'imprévisibilité de l'avenir, de même qu'à apporter aux développeurs une flexibilité et des qualités fort appréciables afin de créer des fonctionnalités de messagerie électronique pleinement orientées vers l'interfaçage homme-machine et une prise en compte personnalisée de chaque utilisateur.

Pour résumer et conclure cette partie directement consacrée aux principaux langages de programmation que nous avons utilisés pour implémenter nos différents travaux de Thèse, nous pouvons retenir que toutes les applications qui suivent l'architecture *COM* sont spécifiées par des interfaces définies. Si *MAPI* spécifie alors chaque nom de telles interfaces, leurs identificateurs d'interface respectifs, leurs définitions de classe (dans un fichier d'entête), la sémantique de chaque méthode, le fournisseur qui l'implémente, la signification des paramètres de la méthode, les valeurs de retour possibles et la "direction" finalement de chaque paramètre, une définition d'interface n'inclut néanmoins pas des données ou un code d'implémentation. C'est ainsi un des concepts les plus attirants et élégants de cette architecture puisque, grâce à cela, un développeur peut concevoir une application pour une architecture donnée et publier sans souci ses interfaces. N'importe quel développeur alors intéressé et voulant utiliser l'interface publiée peut le faire en implémentant les méthodes contenues dans cette dernière. Il n'a donc pas à écrire une seule ligne de code et peut, au lieu de cela, se concentrer au raffinement de la conception et de la spécification.

Enfin, nos interfaces *COM* et *MAPI* sont, typiquement, implémentées en *C++*, à cause, avant toute chose, de l'efficacité incontournable et innée de ce langage : programmation agréable et support pour des caractéristiques orientées objet comme les classes, l'héritage, l'encapsulation de données, l'écriture stricte et la vérification des erreurs à la compilation, qui aident à débusquer des erreurs communes dans la définition et l'utilisation des diverses instances de classe. La programmation orientée objet demeure ainsi, de manière constante et régulière, un gage essentiel de qualité pour l'ensemble de nos implémentations. Tout ceci exposé, nous allons commencer par spécifier en détail les interactions mutuelles à retenir des nouveaux agents intelligents que nous avons implémentés. En ce sens, la prochaine partie nous apporte, entre-autres, un complet schéma fonctionnel, récapitulatif et explicatif de l'ensemble de nos modules informatiques.

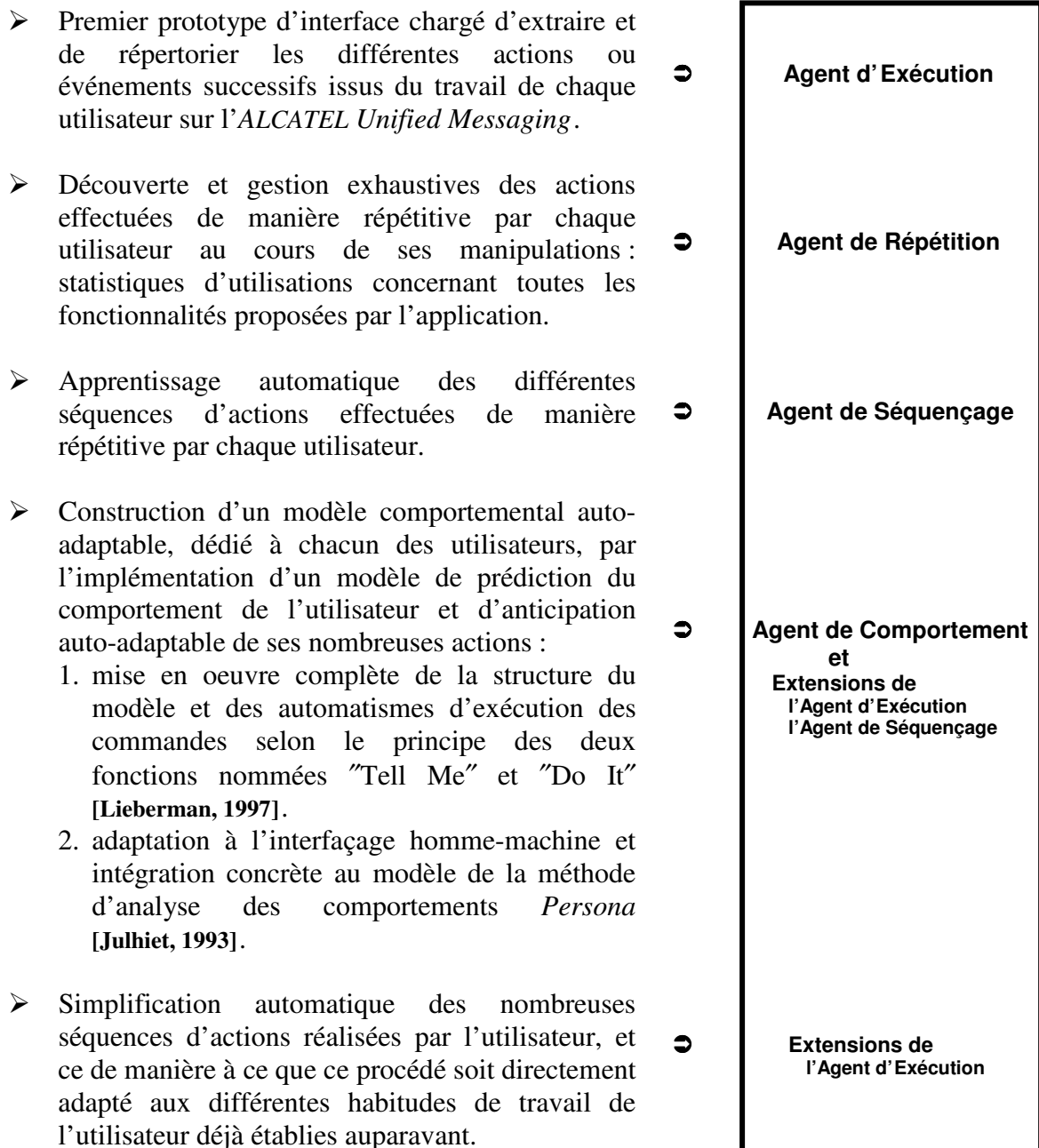
5.1.3. Décomposition de nos modèles en agents intelligents

A l'image du premier schéma à suivre (**Figure 5.1.3.1**) et notamment chargé de nous rappeler l'ensemble des fonctionnalités générales caractérisant chacun de nos agents intelligents, nous avons veillé (cf. aussi **Figure 2.2.2.2**) à spécifier une interface intelligente indépendante des applications sous-jacentes prises en compte. Cela a d'ailleurs permis en retour, à notre système d'interaction, d'être aujourd'hui dédié en priorité à ses utilisateurs. Ceci entériné, il nous reste maintenant à insérer nos composants logiciels dans le système très complexe de l'*AUM*, et ce en apportant, schéma fonctionnel à l'appui, toutes les explications nécessaires à la légitimité et aux choix de fonctionnement caractérisant nos agents intelligents. Ceux-ci assistant les utilisateurs dans leurs tâches informatiques courantes traduisent, finalement, un innovant modèle d'utilisateur généré dynamiquement. Ce dernier sera, de plus, utilisé à terme dans le cadre d'une constante et efficace capacité d'auto-adaptation en temps réel que nous ne manquerons pas de valider.

Alors, si les très nombreuses lignes du "code-source" composant à présent notre modèle d'interface intelligente ont été envisagées et pensées dans un premier temps en *JAVA*, elles ont finalement été réalisées en *Visual C++*, et ce essentiellement pour des raisons d'intégrité avec l'*AUM* et de cohérence avec les pratiques logicielles en vigueur au sein du nouveau *Département Applications* d'*ALCATEL*. Et nos différents agents intelligents se matérialisent plus précisément sous la forme de quatre "Librairies de Liaison Dynamique" (*DLL*, *Dynamic Link Library*). La principale qualité de celles-ci réside dans le fait qu'elles peuvent être rapidement et aisément intégrées à des plate-formes informatiques diverses (*Microsoft Exchange* et *Outlook* notamment,

comme nous l'expliquerons plus loin), de même d'ailleurs que sur toute machine-client propre à un utilisateur en particulier.

Nous proposons par conséquent, sur une nouvelle page, tous les détails correspondant à ces entités logicielles au travers d'un schéma fonctionnel de l'ensemble des opérations attachées à nos agents intelligents et modules associés (**Figure 5.1.3.2** cette fois-ci).



Couche IA

Figure 5.1.3.1 : Schéma récapitulatif des résultats obtenus par implémentation au niveau des caractéristiques respectives de nos agents intelligents

Nous remarquons alors, en découvrant maintenant notre second schéma à venir, que les commandes effectuées par l'utilisateur à partir de l'interface mise en avant, et qui lui est bien sûr entièrement consacrée par l'application courante considérée, se divisent en deux groupes distincts. Le principal, tout d'abord, correspond au recensement de toutes les actions exécutées au niveau de l'application traitée par le modèle intelligent, à savoir les sélections de fichiers, de répertoires et de messages, les commandes par manipulation des menus ou icônes, le travail enfin sur les différentes informations et courriers perçus. De son côté, une seconde flèche, issue de la même interface-utilisateur et entraînant une nouvelle entrée de données dans la "Couche IA", se focalise sur les demandes interactives d'affichage et de renseignement en temps réel des résultats obtenus par nos modèles auto-adaptables, et ce soit par traitement d'opérations au travers du concept des menus, soit aussi grâce aux icônes.

Il est d'ailleurs encore à préciser dans ces lignes que, bien évidemment non spécifié sur notre schéma purement fonctionnel, le lancement de la "Couche IA" donne lieu, en préambule au travail de nos quatre agents intelligents, à une initialisation complète de nos multiples fonctions implémentées de même qu'à toutes les déclarations nécessaires aux événements ou commandes susceptibles d'être, par suite, auto-adaptés. En outre, notre système appelle par les "applications logicielles" tous les programmes informatiques génériques, en considérant également la prise en compte simultanée de différentes versions d'une même application.

Ceci bien précisé, si nous constatons que notre Modèle Auto-Adaptable de l'Utilisateur et son Agent de Comportement associé se composent de trois différents modules informatiques, nous découvrons aussi que le dernier (celui des "Seuils Auto-Adaptés") exploite et entraîne, comme nous l'analyserons par la suite, l'interaction et les conséquences des caractéristiques de l'utilisateur sur ses propres interfaçages. De plus, les indispensables bases de connaissances et leurs successives mémorisations en temps réel sont omniprésentes, tout comme enfin les diverses relations multi-agents (flèches épaisses) ainsi que celles entre modules définis en tant qu'entités individuelles (fléchages fins).

Une qualité importante, voire primordiale pour nos concepts, apparaît de cette manière dès la première flèche épaisse, puisque celle-ci représente les collaborations permanentes entre les domaines de l'applicatif et du comportemental. C'est en effet déjà une preuve que notre "Couche IA" est vraiment dédiée à l'utilisateur et à l'amélioration de ses manipulations.

Nouvelle particularité à remarquer au sein de notre schéma fonctionnel à suivre, les fléchages fins, également introduits ci-dessus, peuvent néanmoins paraître bien surprenants dans la mesure où ceux-ci permettent aux différents modules internes des agents de communiquer directement avec d'autres modules des autres agents.

Nous nous devons alors d'expliquer dans ces lignes que nous avons opté pour une telle représentation spéciale, quelque peu déviée de nos réelles implémentations, dans le but essentiel de favoriser une entière compréhension du fonctionnement de nos diverses entités intelligentes. En effet, nos agents logiciels présentent bien évidemment chacun la structure mise en avant par nos spécifications des premiers chapitres de ce mémoire, et comportent notamment de complètes interfaces indispensables à la communication entre agents. Cependant, il nous est aussi apparu comme étant beaucoup trop lourd et incompréhensible sur un seul schéma, ainsi que surtout très imprécis à terme, de ne pas décrire à ce niveau de quels autres modules intelligents dépend exactement chaque module composant notre "Couche IA".

Nous verrons en conséquence tout-de-suite que toutes nos nouvelles entités intelligentes interagissent non seulement avec chacun des utilisateurs, mais aussi entre-elles et avec nos multiples modèles théoriques, ce dans le but principal de surveiller, d'analyser et d'exploiter au mieux les activités quotidiennes de l'homme face à la machine.

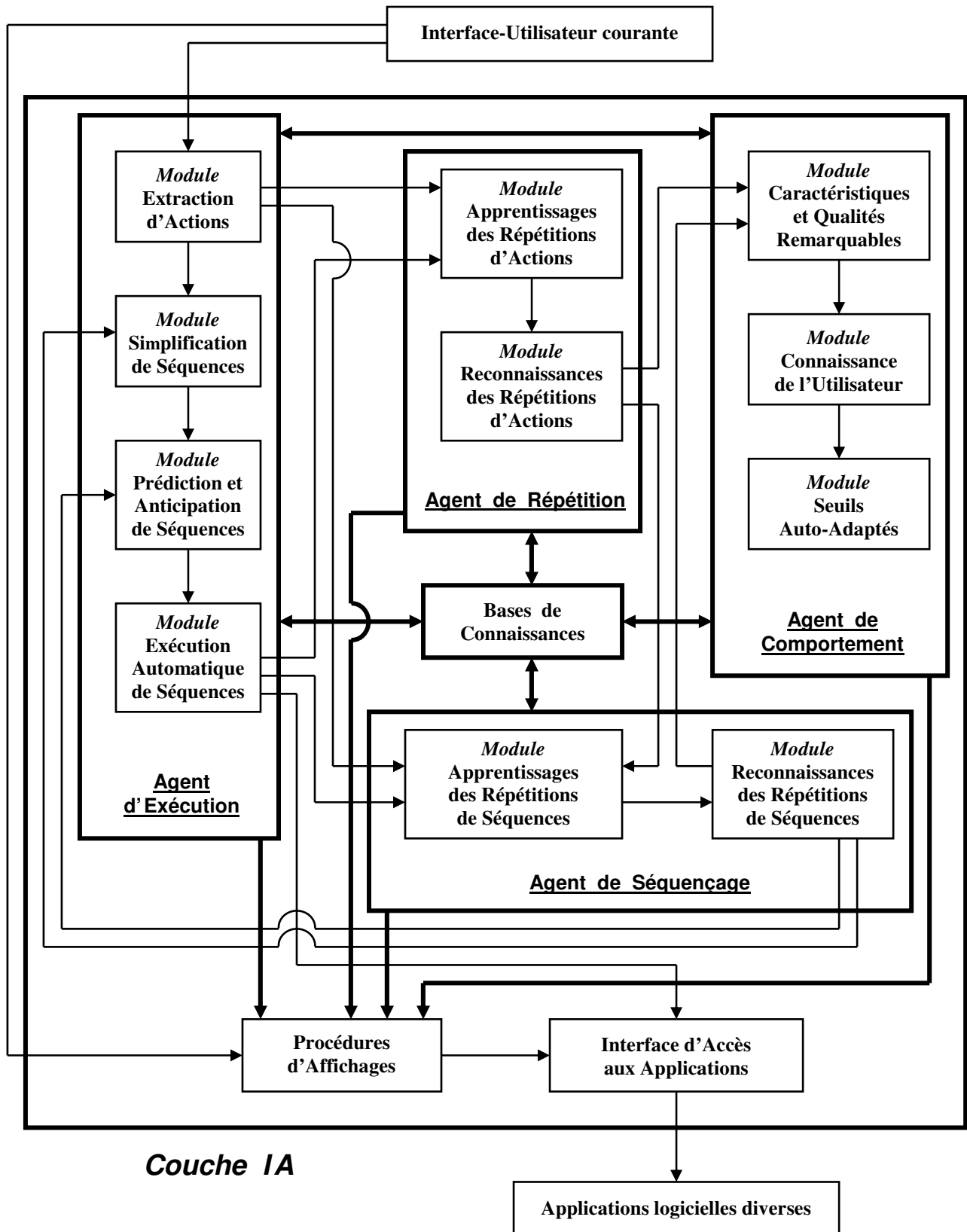


Figure 5.1.3.2 : Schéma fonctionnel de l'ensemble des agents intelligents composant notre "Couche IA"

5.2. Détails de nos procédures d'implémentation

5.2.1. Introduction

En entrant dans le cœur-même des explications inhérentes à l'implémentation de notre modèle d'interface intelligente, nous nous devons de revenir sur les caractéristiques significatives et innovantes de notre système multi-agents ainsi que, par extension, sur les nouvelles propriétés remarquables et programmées au sujet de nos différents agents logiciels, ciblés individuellement. Et toutes ces explications se voient encore grandement complétées de par nos multiples et divers extraits de fichiers de "code-source", lignes mises en avant et commentées au sein de l'**Annexe 3**. Elle enchaîne ainsi son étude des outils *COM* et *MAPI* par la description de tous les mécanismes de démarrage et d'initialisation de nos procédures d'implémentation.

Si nos rapports successivement validés et enregistrés dans la base de données du groupe *ALCATEL* sont bien entendu tous à l'aplomb des écrits de ce chapitre, ce sont surtout les derniers ([Sandel, Korczak & al., 1999-5], [Sandel, Korczak & al., 2000] et [Sandel, Korczak & al., 2001]) qui se chargent d'offrir aux divers développeurs l'indispensable documentation traitant de l'ensemble des aspects de réalisation attachés à notre logiciel. Par ailleurs, il nous faut encore spécifier que, naturellement, les caractéristiques des différents terminaux de communication abordés diffèrent (se référer une nouvelle fois au **Chapitre 1** pour davantage de précisions à ce sujet). Nous avons donc constamment cherché à isoler le code spécifique à chacun de nos modèles auto-adaptables dans des procédures particulières. A l'image de nos travaux précédents, nous verrons alors que la programmation orientée objet nous a grandement soutenus dans cette direction.

5.2.2. Un premier agent intelligent : l'Agent d'Exécution

Construit pour gérer et opérer l'extraction des connaissances propres aux manipulations successives effectuées par l'utilisateur, notre présent "Agent d'Exécution" se voit non seulement attribuer la détermination de l'identifiant décrivant chaque action réalisée, mais aussi chargé de récupérer l'ensemble des données contenues dans tout nouveau message arrivé dans les boîtes aux lettres respectives.

La première copie d'écran rapportée dans cette partie (**Figure 5.2.2.1**) nous permet alors de constater que, dans le cadre de cette phase primordiale d'extraction des actions en temps réel, nous commençons par faire appel à une bien intéressante méthode, nommée "*GetSelectionItem*" et capable de renvoyer, à notre système multi-agents, un descriptif d'item, de *mail*, de répertoire, de commande, d'événement et d'interface, tous sélectionnés. Ensuite, chaque sélection divulgue son propre et unique identifiant (grâce à "*GetCmdId*"), nécessaire à sa définition correspondant également aux clés primaires de nos bases de données, et qui sera aussi indispensable plus tard à une majorité de nos traitements auto-adaptables.

A cette occasion, nous prenons acte, de plus, de la différence essentielle que nous avons rapidement dû apprendre à contrôler entre une session, se caractérisant par la mémorisation de l'ensemble des informations nécessaires au travail courant et "temps réel" des utilisateurs, et un contexte (*Viewer*) qui, contrairement à une session, délivre uniquement un état proposant un support à chaque nouvelle manipulation.

Dans un deuxième temps, et toujours conformément à notre schéma fonctionnel de la **Figure 4.5.3.3**, "l'Agent d'Exécution" s'est attaché à concrétiser nos concepts de prédiction et d'anticipation auto-adaptable. Lié et issu du processus d'extraction des connaissances puisque correspondant à tout éventuel enclenchement d'une nouvelle séquence, le procédé de prédiction, appliqué à chaque action courante, est alors réalisé en partenariat avec "l'Agent de Séquençage", comme nous le détaillerons en **5.2.4**. Quant à l'anticipation, elle est complétée par l'exécution automatique des diverses séquences d'actions et demeure en étroite relation, en ce qui concerne l'interfaçage intelligent proposé à chacun de nos utilisateurs, avec la **Figure A3.3.1.2**.

Cependant, si la **Figure 5.2.2.4** à venir introduit ainsi concrètement l'implémentation de la phase d'anticipation et d'exécution auto-adaptables de toute séquence d'actions, elle-même déterminée par notre système multi-agents et surtout toujours pleinement désirée par l'utilisateur courant, elle s'est encore vu précédée dans nos programmes par notre tout aussi innovant concept de la simplification auto-adaptable (**Figure 5.2.2.2** et **Figure 5.2.2.3**).

```

wsprintf (userBuffer[i], "Action %d (the %s at %s) : User %d selected ",
          i+1, DateBuffer, TimeBuffer, *lpSession);
hr = pmechb->GetSelectedItem (0, &cbeid, &lpeid, &ulType, NULL, 0, NULL, 0);
if (FAILED(hr)) goto error_return;

#ifdef Bibliotheque_des_types_de_selection
#define MAPI_MESSAGE ((ULONG) 0x00000001) // Message
#define MAPI_ADDRBOOK ((ULONG) 0x00000002) // Address Book
#define MAPI_FOLDER ((ULONG) 0x00000003) // Folder
#define MAPI_ABCONT ((ULONG) 0x00000004) // Address Book Container
#define MAPI_STORE ((ULONG) 0x00000005) // Message Store
#define MAPI_MAILUSER ((ULONG) 0x00000006) // Individual Recipient
#define MAPI_ATTACH ((ULONG) 0x00000007) // Attachment
#define MAPI_DISTLIST ((ULONG) 0x00000008) // Distribution List Recipient
#define MAPI_PROFSECT ((ULONG) 0x00000009) // Profile Section
#define MAPI_STATUS ((ULONG) 0x0000000A) // Status Object
#define MAPI_SESSION ((ULONG) 0x0000000B) // Session
#define MAPI_FORMINFO ((ULONG) 0x0000000C) // Form Information
#endif
switch (ulType)
{ case MAPI_MESSAGE : lpSession->OpenEntry (cbeid, lpeid, &IID IMessage,
      MAPI_BEST_ACCESS, &ulType, (LPUNKNOWN FAR *)&lpMessage);
    // Message enveloppe properties
    hr = HrGetOneProp (lpMessage, PR_MESSAGE_CLASS_A, &valeID);
    if (FAILED(hr))
    { hr = S_OK; // we still handled the command
      goto error_return;
    }
    else
    { wsprintf (actBuffer[i], "Message classified %s with\n ",
      valeID->Value.lpszA);
      actInfo = string(actBuffer[i]);
    }
    hr = HrGetOneProp (lpMessage, PR_SUBJECT_A, &valeID);
    if (FAILED(hr))
    { wsprintf (actBuffer[i], "unavailable subject, ");
      actInfo += string(actBuffer[i]);
    }
    else
    { acTmp = string(valeID->Value.lpszA);
      tailTmp = acTmp.size ();
    }
}
    
```

Figure 5.2.2.1 : *Début de notre procédé d'extraction des connaissances (cf. 4.4.4 entre-autres), mis en œuvre, au sein de "l'Agent d'Exécution", en relation directe avec chaque changement de sélection issue de l'interface-utilisateur*

```

// auto-adaptable simplification process for the current sequence of actions
if (seuil >= seuil_predict)
{
    i = 1;
    if (mssgdeux)
        i++;
    while (!testundo && (i <= nbacts))
    {
        if (strActs[i] == "Command \"Undo\" ")
        {
            testundo = true;
            nbactmp = nbacts - 2;
            nivs = i;
            for (j=0 ; j<=(i-2) ; j++)
                strAcTmp[j] = strActs[j];
            for (j=(i-1) ; j<=nbactmp ; j++)
                strAcTmp[j] = strActs[j+2];
        }
        i++;
    }

    if (!testundo)
    {
        i = 0;
        if (mssgdeux)
            i++;
        while (!testidem && (i <= nbacts))
        {
            for (j=(i+1) ; j<=nbacts ; j++)
                if (!testidem && (strActs[j] == strActs[i]))
                {
                    testidem = true;
                    nbactmp = nbacts - (j-i);
                    nivs_i = i;
                    nivs_j = j;
                    for (k=0 ; k<=i ; k++)
                        strAcTmp[k] = strActs[k];
                    for (k=1 ; k<=nbactmp ; k++)
                        strAcTmp[i+k] = strActs[j+k];
                }
            i++;
        }
    }

    if (testundo || testidem)
    {
        if (
            // sequences without any interest of treatment
            (nbactmp <= 0)
            || (
                (nbactmp > 0)
                && ( (strAcTmp[0] == strAcTmp[nbactmp])
                    || (
                        (mssgdeux)
                        && (strAcTmp[1] == strAcTmp[nbactmp])
                    )
                )
            )
        )
        seuil = seuil_wany;
    }
}

```

Figure 5.2.2.2 : *Module de simplification auto-adaptable des séquences d'actions, l'implémentation de la détection d'éventuelles erreurs de manipulation émanant de l'utilisateur (cf. 4.5.1.5 également)*

```

if (testundo)
{ nbacts = nbacts - 2;
  for (j=(nivs-1) ; j<=nbacts ; j++)
  { strActs[j] = strActs[j+2];
    strType[j] = strType[j+2];
    selEidTab[j] = selEidTab[j+2];
  }
}
else // [ (testidem == true) ]
{ nbacts = nbacts - (nivs_j - nivs_i);
  for (k=1 ; k<=nbacts ; k++)
  { strActs[nivs_i+k] = strActs[nivs_j+k];
    strType[nivs_i+k] = strType[nivs_j+k];
    selEidTab[nivs_i+k] = selEidTab[nivs_j+k];
  }
}
}

```

Figure 5.2.2.3 : *Mise en oeuvre de nos corrections auto-adaptables au sujet des erreurs de manipulation détectées à partir du travail de chaque utilisateur*

```

LPOUTLXCALLBACK lpoech = NULL;
hr = pmech->QueryInterface (IID_IOutlookExtCallback, (void **)&lpoech);
if (SUCCEEDED(hr) && lpoech)
{ LPUNKNOWN pUnk = NULL;
  lpoech->GetObject (&pUnk);
  lpoech->Release ();
  if (pUnk)
  { OL::MAPIFolderPtr spFolder;
    hr = pUnk->QueryInterface (__uuidof(OL::MAPIFolder), (void **)&spFolder);
    pUnk->Release ();
    if (SUCCEEDED(hr) && (spFolder != NULL))
    { OL::_ApplicationPtr spAppl = spFolder->GetApplication ();
      OL::_NameSpacePtr spSession = spAppl->GetSession ();
      OL::_FoldersPtr spFolders = spSession->GetFolders ();
      OL::_ExplorerPtr spExplorer = spFolder->GetExplorer ();

      i = begncmpt;
      while (i <= nbacts)
      { if ( (strType[i] != "Menu") && (strType[i] != "Command")
          && (strType[i] != "Context") && (strType[i] != "Toolbar"))
          { // [ origin == 'Selection' ]
            selEid = selEidTab[i];
            OL::MAPIFolderPtr selFolder = FldExtract (spFolders);
            if (selFolder != NULL)
              spExplorer->PutRefCurrentFolder (selFolder);
          }
          else
          { // [ origin == 'Commands' ]
            LRESULT lghr = PostMessage (hWndMem /*HWND_BROADCAST*/,
                                       WM_COMMAND, cmdidMod[i], 0L);

            if (lghr != 1) goto error_return;
            extr_cmd++;
            ActExtract (pmech, cmdidTab[i], 'c');
          }
            ActMemoriz (pmech, string(UserBufferSeq), strActs[i], ".\r\n");
            actAntic = strActs[i];
            actInfoMem = actAntic;
          }
      }
    }
  }
}

```

Figure 5.2.2.4 : *Lancement de la phase d'anticipation et d'exécution auto-adaptables d'une séquence d'actions, déterminée par notre système multi-agents et surtout pleinement désirée par l'utilisateur courant*

Allant jusqu'à permettre un classement en temps réel de tous les messages reçus par nos utilisateurs (cf. **A2.3**), ainsi qu'en complément à des glissements toujours agréables à l'écran, ces précédents procédés de prédiction, anticipation et exécution automatique proposent de traiter également les *mails* au sein-même des commandes automatisées. Ainsi, dans le but d'adjoindre à la **Figure 5.2.2.4** ces précisions, nous poursuivons cette partie par le détail de l'ultime méthode permettant l'implémentation exhaustive de l'algorithme qui confère à notre "Agent d'Exécution" la capacité d'extraire, en temps réel, les caractéristiques de l'item se devant d'être reproduit, de manière automatique, sur l'interface alors traitée (**Figure 5.2.2.5** ci-contre).

```

////////////////////////////////////
// FldExtract ()
//
// Parameters
// trFolders - pointer to the current set of folders to be explored.
//
// Purpose
// This method examines the panel of all the different folders available
// in the treated application, in order to extract the address from that
// which will be executed for the current user in an auto-adaptative way.
//
// Return Value
// crFolder - pointer to the folder which is to be selected automatically.
//
// Comments
// This function is recursive.
//

OL::MAPIFolderPtr FldExtract (OL::_FoldersPtr trFolders)
{
    long lgcr;
    long foldcmpt = trFolders->GetCount ();
    for (lgcr=1 ; lgcr<=foldcmpt ; lgcr++)
    {
        OL::MAPIFolderPtr crFolder = trFolders->Item ((_variant_t)lgcr);
        _bstr_t crFolderName = crFolder->GetName ();
        string strnom = (string)crFolderName;
        _bstr_t crFldEid = crFolder->GetEntryID ();
        if (strnom == selEid)
            return (crFolder);

        if (strnom.compare(0,construct.size(),construct) != 0)
        {
            OL::_FoldersPtr ssFolders = crFolder->GetFolders ();
            OL::MAPIFolderPtr sscrFolder = FldExtract (ssFolders); // recursive call
            if (sscrFolder != NULL)
                return (sscrFolder);
        }
    }
    return (NULL);
}

```

Figure 5.2.2.5 : *Mise en œuvre de l'algorithme indispensable à notre "Agent d'Exécution" afin de déterminer, en temps réel, l'item à exécuter automatiquement*

Pour demeurer tout-à-fait complet dans le cadre de nos programmations orientées objet, nous proposons à suivre un exemple de nos principales hiérarchies de classes implémentées, résumé extrait de nos fichiers et bien représentatif en outre des démarches explicitées dans ce chapitre. En effet, et à l'image des très importantes documentations à mettre à la disposition de tout développeur intéressé par notre logiciel, il apparaît maintenant primordial de bien décrire les

relations existant entre les diverses classes d'objets composant notre agent. Nous nous penchons plus particulièrement, à ce niveau, sur celles directement liées au fonctionnement de nos modèles auto-adaptables (**Figure 5.2.2.6**). Par ailleurs, si nous retrouvons alors une majorité des données commentées auparavant, nous relevons aussi l'importante distinction centrale que nous avons dû opérer entre, d'un côté, les appels de commandes (*DoCommand*) et, de l'autre, les sélections d'actions (*OnSelectionChange*). Mais nous reviendrons sur ce sujet dans les parties à venir.

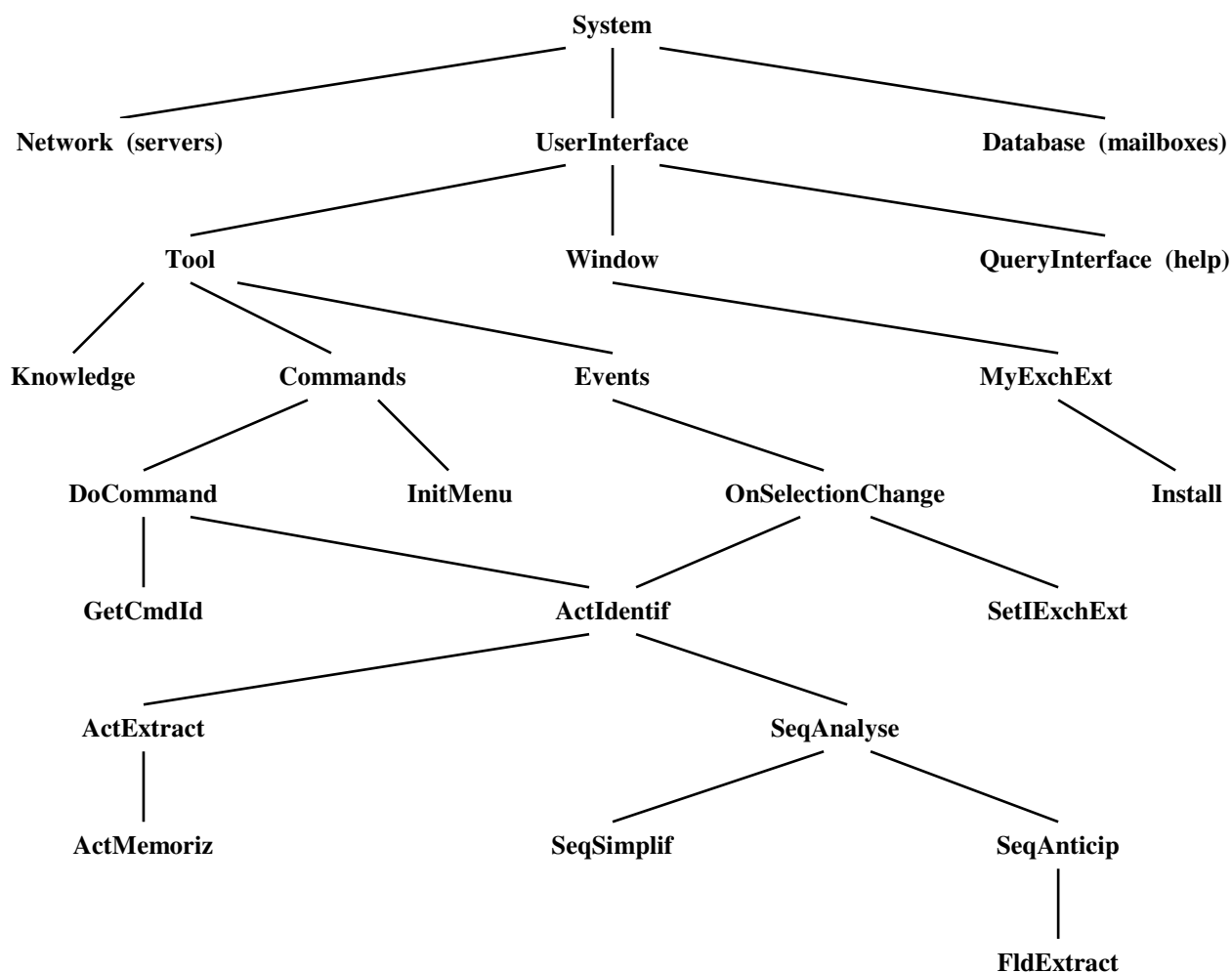


Figure 5.2.2.6 : *Extrait des principales hiérarchies de classes implémentées dans le cadre de la programmation "objet" des modèles auto-adaptables de notre "Agent d'Exécution" pris en exemple*

Pour conclure la présente étude du "code-source" de notre tout premier agent intelligent, nous nous devons de préciser, pour la suite de nos travaux, son concept de simplification auto-adaptable. Réellement indispensable dans un système d'interfaçage homme-machine qui se veut "intelligent", nous ne pouvions effectivement nous résoudre à automatiser certaines erreurs ou du moins incohérences de manipulation émanant de l'utilisateur. C'est pourquoi, nous avons placé ce processus, dans la chaîne et la chronologie des modules activés par notre "Agent d'Exécution", avant ceux de prédiction et d'anticipation. Caractérisée par une demande initiale de confirmation interactive, afin d'apporter une assurance certaine et toujours appréciable à l'interlocuteur, cette

simplification s'inscrit aussi très naturellement dans le cursus général des opérations de l'agent considéré ici, puisqu'elle est, en pratique, constamment suivie par l'exécution automatique de la séquence courante, dans le cas bien évidemment d'une acceptation de la part de l'utilisateur.

Si, par suite, l'**Annexe 4** nous détaillera alors encore plus avant (cf. notamment **A4.2.7**) les deux principaux cas de figure rencontrés au niveau de ces éventuelles erreurs exécutées de manière répétitive par nos utilisateurs, nous pouvons déjà rappeler qu'il s'agit là des boucles (toutes explorées par le booléen "testidem" dans la **Figure 5.2.2.3**) et des retours en arrière ("testundo" cette fois). De plus, si notre procédé de simplification ainsi développé tient compte en permanence de l'ensemble de nos principes d'auto-adaptation, il reste aussi complètement indépendant des processus le suivant (prédiction et anticipation), ce de par sa gestion à part des coefficients attachés à l'utilisateur courant (cf. "l'Agent de Comportement"). Pour exemple, nous pouvons ainsi remarquer qu'un refus de simplification d'une séquence d'actions n'entraîne en rien une impossibilité d'exécuter cette séquence de manière auto-adaptable ensuite...

5.2.3. L'Agent de Répétition

Arrivés à ce niveau, nous devons préciser, en conformité avec l'étude que nous venons de mener au sujet du premier de nos agents intelligents, que nous ne reviendrons pas, dans la suite de ce chapitre, sur les déclarations, chargements, constructions et autres installations nécessaires à la structure, au bon fonctionnement et à la gestion périphérique de nos entités logicielles. En effet, pour d'évidentes raisons de synthèse, d'efficacité et de pertinence concernant notre rapport de Thèse, seules les fonctions et modules "intelligents" vont maintenant focaliser notre attention, c'est-à-dire les implémentations présentant les procédés les plus significatifs de nos recherches.

Nous abordons alors ici le plus petit (en taille) de nos quatre agents, chargé d'apprendre et de reconnaître toutes les répétitions d'actions réalisées par nos différents utilisateurs courants. En conséquence, de ce second agent, et en complément aux informations et algorithmes que nous avons déjà mis en avant à son sujet (cf. notamment **4.2** et **4.3**), nous ne retenons que la plus remarquable de ses qualités, à savoir le principe de base de nos apprentissages et reconnaissances de répétitions d'actions (**Figure 5.2.3** sur la page suivante).

Une étude approfondie de ce nouvel extrait de nos implémentations met en évidence le fait que nous n'ayons pris les répétitions d'actions en compte qu'à partir d'un doublon avéré de chaque action, qu'il soit consécutif ou non. En outre, la répétition des messages eux-mêmes s'effectue en prenant en considération l'expéditeur de chaque *mail*, ceci selon un choix que nous avons cherché à effectuer de manière la plus pertinente et légitime possible. En effet, nous avons également ajouté, au cours de nos programmations successives, la prise en compte du sujet de chaque message, mais nous nous sommes vite résolus à ne plus considérer que l'expéditeur puisque le couple "(sujet, expéditeur)" n'est, somme toute, que trop rarement répété à l'identique pour permettre un apprentissage efficace. A remarquer toutefois que l'on peut aussi opter pour ne considérer, au contraire, que le sujet des messages, ou même des mots-clés issus de l'entête, du corps ou de la signature. En conséquence, nous avons travaillé avec les caractéristiques obtenues de l'expéditeur car, en comparaison avec le sujet déjà, c'est l'information concernant l'expéditeur qui apparaît la plus invariante et précise dans le temps, avec les aléas de la vie quotidienne...

L'**Annexe 4** montre par ailleurs que nos choix se sont avérés particulièrement judicieux, en notant que chaque message apparaît en définitive relativement unique de par l'ensemble de ses informations ; d'où l'extraction et la considération ciblées que nous avons opérées à leurs égards.

```

if (nouv)
{
    pos = fseek (fdstmp, 0, SEEK_SET);
    if (pos == 1)    goto error_return;
    nvcmt = 0;
    while (nouv && (fgets(chtmp,tailLect,fdstmp) != NULL))
    {
        strtmp = string(chtmp);
        if (strtmp.compare(0,tailleActTmp,strActTmp) == 0)
        {
            nvcmt++;
            if (nvcmt == 2)
                nouv = false;
        }
    }
    if (!nouv)
    {
        pos = fseek (fdrept, 0, SEEK_END);
        if (pos == 1)    goto error_return;
        strAct += " -> 2 \n";
    }
}
else
{
    pos = fseek (fdrept, 0, SEEK_SET);
    if (pos == 1)    goto error_return;
    while (wrcmt > 1)
    {
        if (fgets(chtmp,tailLect,fdrept) == NULL)
            goto error_return;
        wrcmt--;
    }
}
}
    
```

Figure 5.2.3 : *Introduction de l'ensemble de la mise en œuvre correspondant aux apprentissages et reconnaissances des répétitions d'actions, travaux réalisés par "l'Agent de Répétition"*

5.2.4. L'Agent de Séquençage

Ce nouvel agent intelligent est destiné à multiplier les capacités de l'entité précédente, en visant, comme son nom l'indique, le traitement beaucoup plus vaste et nettement plus délicat des séquençements d'actions. Les difficultés sont effectivement notables, notamment du fait que nous devons, en temps réel, conduire la constante mise-à-jour de plusieurs séquences différentes, prises simultanément car toute séquence peut être concernée par une nouvelle exécution d'action. Nous avons réussi, conformément au **Chapitre 4**, à élaborer des algorithmes intéressants, car peu gourmands, que ce soit en taille-mémoire ou encore en temps : nous apprécions ainsi, entre-autres, l'itération unique et la non-nécessité d'exécuter plusieurs passages successifs au sein des bases de données mises à contribution (cf. **Figure 5.2.4.1** à venir). En effet, réalisés à chaque parcours de notre base de connaissances, les différents algorithmes d'apprentissage et de reconnaissance des séquences demeurent rapides, efficaces, mais aussi surtout fiables et robustes (cf. **Annexe 4**).

Les investigations du présent agent traitent alors très bien les boucles pouvant apparaître relativement fréquemment au sein des séquences d'actions effectuées par les divers utilisateurs (cf. **4.1.2**). Cependant, dans un contexte de travail que nous souhaitons non perturbateur ainsi qu'aussi pertinent et agréable que possible pour l'utilisateur (pas trop d'informations apportées dans un même temps : cf. **3.2** à ce propos), nous avons choisi, à l'image des tests réalisés lors de nos implémentations successives et en accord avec ceux rapportés par l'**Annexe 4**, de ne retenir, à un instant donné et pour des actions bien regroupées, que les séquences, circulaires ou non, les plus longues, c'est-à-dire les plus significatives pour l'utilisateur (cf. la seconde **Figure 5.2.4.2**).

Tout reste en conséquence, et ce jusqu'au choix des plus profonds mécanismes de nos différents agents intelligents, pleinement orienté vers l'utilisateur humain.

```

while (stmp >= 2)
{ tailleStmprec = strStmp[s].size ();
  s++; scmptgen++;
  if (!sequssg)
  { rdstmp = 0;
    pos = fseek (fdstmp, 0, SEEK_SET);
    if (pos == 1) goto error_return;
    while (fgets(chtmp,tailLect,fdstmp) != NULL)
    { strtmp = string(chtmp);
      rdstmp++;
      if (rdstmp <= (rdcmpt-scmptgen))
      { if (strtmp.compare(0,m,seqUser) == 0)
        { strStmp[s] = strtmp;
          tailleStmp = strtmp.size ();
          sprecurt = 0;
        }
        else sprecurt++;
      } scmptgen += sprecurt;
    }
    pos = fseek (fdstmp, 0, SEEK_SET);
    if (pos == 1) goto error_return;
    while (fgets(chtmp,tailLect,fdstmp) != NULL)
    { strtmp = string(chtmp);
      if (strtmp.compare(0,tailleStmp,strStmp[s]) == 0)
      { posseq = true;
        for (k=(s-1) ; k>=0 ; k--)
        { if (fgets(chtmp,tailLect,fdstmp) != NULL)
          { strtmp = string(chtmp);
            while (strtmp.compare(0,m,seqUser) != 0)
            { if (fgets(chtmp,tailLect,fdstmp) == NULL) goto error_return;
              strtmp = string(chtmp);
            }
            if ((strtmp.compare(0,strStmp[k].size(),strStmp[k]) != 0))
            { posseq = false;
              k = -1;
            }
          }
        }
        if (posseq) stmp++; }
      }
    }
  }
  if (stmp >= 2) sequ = stmp; }

```

Figure 5.2.4.1 : *Mise en œuvre globale du canevas structurel (cf. 4.5.2) concernant nos procédés d'apprentissage automatique attachés aux séquencements d'actions et gérés par "l'Agent de Séquençage"*

5.2.5. L'Agent de Comportement

Dédié à l'appréhension et au traitement en temps réel des comportements de chacun des utilisateurs de la "Couche IA", ce quatrième agent se voit surtout chargé, dans un premier temps, de comprendre, comparer et synthétiser toutes les informations propres aux manipulations issues des utilisateurs (grâce bien entendu au travail préliminaire de notre "Agent d'Exécution"). Il nous permet en outre de grandement évaluer la qualité des fondements, du déroulement et des résultats décrivant nos différents algorithmes d'apprentissage automatique mis en place jusqu'ici.

```

if (sequ >= 2)
{
    wsprintf (chAct, " -> %d ", sequ);
    strAct = string(chAct);
    strAct += "\n";
    for (k=0 ; k<s ; k++)
    {
        tailleStmp = strStmp[k].size ();
        strStmp[k].erase ((tailleStmp-1), 1);
    }

    strActTmp = "";
    strActPrec = "begin";
    if (s == 2)
        strActPrec = strStmp[s-1];
    for (k=(s-2) ; k>=0 ; k--)
    {
        j = 0;
        while (strStmp[k].compare (j,1,":") != 0)
            j++;
        strStmp[k].erase (0, j+1);
        strActTmp += " |" + strStmp[k];
        if (k == 1)
            strActPrec = strStmp[s-1] + strActTmp;
    }

    strSeqTmp = strStmp[s-1] + strActTmp;
    strSeq = strSeqTmp + strAct;
    tailleStrSeq = strSeq.size ();
    }
    
```

Figure 5.2.4.2 : *Implémentation principale structurant nos concepts de reconnaissance auto-adaptable des séquençements d'actions*

Pour poursuivre l'étude de notre dernier "Agent de Comportement", et ce en centralisant, analysant et travaillant tous les rapports, pourcentages, comparaisons et taux intéressants, nous parvenons à établir et à stocker, grâce à de complètes écritures et incréments de coefficients concernés à chaque nouvelle action réalisée par l'utilisateur, un condensé de données statistiques. Ces dernières deviennent alors capables de nous apporter des connaissances très pertinentes sur le comportement courant de chaque utilisateur. Le **Tableau 5.2.5** à venir, ainsi que la **Figure 5.2.5.1** qui lui est directement associée, nous en fournissent par ailleurs de très utiles aperçus, riches en enseignements. L'implémentation de notre Modèle Auto-Adaptable de l'Utilisateur est initiée.

Le traitement et la synthèse de cet ensemble de caractéristiques et qualités remarquables, provenant d'une collaboration implicite entre les utilisateurs et notre système multi-agents, nous entraînent vers un très intéressant apport de connaissances dans le cadre de la prise en compte de nos utilisateurs respectifs.

De cette manière très bien et très justement renseignés à différents niveaux concernant directement l'homme face à la machine, nous pouvons à présent pleinement établir et mettre en œuvre notre modélisation de l'utilisateur, en nous aidant bien entendu grandement des travaux exposés auparavant dans ce mémoire, et notamment au sein du **Chapitre 3**. Nous allons donc opérer une adaptation de toutes les informations obtenues précédemment, une adéquation globale que nous visons aussi fine et efficace que possible, à l'ensemble de nos modèles auto-adaptables et à l'interfaçage homme-machine en particulier (cf. ainsi les **Figure 5.2.5.2** et **Figure 5.2.5.3** sur les prochaines pages à suivre).

Tableau 5.2.5 : *Récapitulatif des données statistiques rassemblées en temps réel, par notre "Couche IA", au sujet du comportement de chaque utilisateur et des qualités d'apprentissage de notre système multi-agents (cf. copies d'écran au cours de l'Annexe 4)*

Identifiant propre à l'utilisateur, catégorie(s) de modélisation de son comportement courant

- > Nombre et pourcentage d'actions distinctes par rapport au nombre total d'actions
Nombre d'actions de sélection et nombre d'appels de commande
 - > Nombre et pourcentage d'actions distinctes répétées
par rapport au nombre total d'actions répétées
Nombre d'actions de sélection répétées et nombre d'appels de commande répétés
 - > Pourcentage d'actions répétées par rapport à toutes celles effectuées
(légitimité de l'auto-adaptation)

 - > Nombre et pourcentage de séquences distinctes répétées
par rapport au nombre total de séquences répétées
Taille moyenne des séquences apprises
 - > Nombre de séquences auto-adaptables proposées,
en mode "Tell Me" (prédiction) et en mode "Do It" (anticipation)
Taille moyenne des séquences auto-adaptables proposées
 - > Nombre de séquences auto-adaptables exécutées,
en mode "Tell Me" (prédiction) et en mode "Do It" (anticipation)
Taille moyenne des séquences auto-adaptables exécutées
 - > Nombre total d'actions exécutées par auto-adaptation
Pourcentage d'actions auto-adaptables par rapport à toutes celles exécutées
(pertinence de l'auto-adaptation)

 - > Nombre de simplifications auto-adaptables de séquences proposées et exécutées
Pourcentage de simplifications exécutées par rapport aux simplifications proposées
(pertinence de la simplification)
 - > Taille moyenne des séquences avant et après les simplifications auto-adaptables exécutées
Pourcentage de réduction de la taille des séquences par application des
simplifications auto-adaptables (légitimité de la simplification)
-

Pour conclure alors cette partie très importante de nos recherches, nous nous devons encore de préciser, voire de fortement insister, que nous avons choisi d'établir l'ensemble de nos critères auto-adaptables de manière à ce qu'ils permettent aux huit espaces de notre modèle de l'utilisateur d'être toujours tous accessibles grâce aux multiples coefficients mis en place. Et si nous pouvons déjà, à ce sujet, revenir au niveau de la partie 3.3, l'Annexe 4 nous en apportera de plus une preuve intangible (cf. Figure A4.2.8 notamment).

Dans ce but essentiel, il nous a ainsi fallu constamment chercher à équilibrer, tout en restant bien entendu pleinement cohérent et juste en matière de relation de nos modèles avec la réalité, l'ensemble des considérations propres à chacune des tendances comportementales (autant de caractéristiques plausibles pour les deux sens décrivant chaque axe de notre méthode).

```

nb_acts_distct = nb_acts_distct_rept + (nb_acts - nb_acts_rept);

rts_acts = ((float)nb_acts_distct / (float)nb_acts) * 100;
rts_acts_int = rts_acts;
while (rts_acts >= 1)
    rts_acts--;
rts_acts_dec = rts_acts * 100;

if (nb_acts_rept != 0)
    rts_acts_rept = ((float)nb_acts_distct_rept / (float)nb_acts_rept) * 100;
else
    rts_acts_rept = 0;
rts_acts_rept_int = rts_acts_rept;
while (rts_acts_rept >= 1)
    rts_acts_rept--;
rts_acts_rept_dec = rts_acts_rept * 100;

if (nb_acts != 0)
    rts_aa = ((float)nb_acts_rept / (float)nb_acts) * 100;
else
    rts_aa = 0;
rts_aa_int = rts_aa;
while (rts_aa >= 1)
    rts_aa--;
rts_aa_dec = rts_aa * 100;

avs_learn = avs_learn / nb_sequ;
avs_learn_int = avs_learn;
while (avs_learn >= 1)
    avs_learn--;
avs_learn_dec = avs_learn * 100;

if (nb_sequ != 0)
    rts_sequ = ((float)nb_distct_sequ / (float)nb_sequ) * 100;
else
    rts_sequ = 0;
rts_sequ_int = rts_sequ;
while (rts_sequ >= 1)
    rts_sequ--;
rts_sequ_dec = rts_sequ * 100;
    
```

Figure 5.2.5.1 : *Premières parties de "code" chargées d'établir l'ensemble des statistiques propres au module des caractéristiques et qualités remarquables des manipulations courantes, processus contrôlé par "l'Agent de Comportement"*

Nous aboutissons finalement à ce qui apparaît comme étant un des points innovants les plus intéressants de nos recherches, à savoir la prise en compte, et l'exploitation intelligente surtout, des constats personnalisés que nous avons pu obtenir au niveau du statut et des types de comportement attribués à chacun de nos utilisateurs. En définitive, ces données toutes nouvelles ont contribué à catalyser, pour ces derniers et toujours en temps réel, l'ensemble de leurs futures actions auto-adaptables (cf. **4.5.3.2**).

S'en suivent enfin deux aperçus bien fournis, premières copies d'écrans (**Figure 5.2.5.4**) présentant dans ces lignes une introduction aux nombreuses autres fenêtres d'interface issues des exécutions réelles de notre système auto-adaptable (cf. **Annexe 4**). Nous découvrons ainsi, plus particulièrement, des résultats obtenus de notre modèle d'apprentissage et de reconnaissance des séquencements d'actions, puis diverses informations calculées par notre modèle comportemental attaché aux manipulations courantes de chacun de nos utilisateurs.

```

if (nb_acts > 500) persaxes[Exprm]++;   else persaxes[Neoph]++;
if (rts_acts_int > 20) persaxes[Neoph]++;   else persaxes[Exprm]++;
if ((nb_acts_cmd == 0) || ((nb_acts_slc / nb_acts_cmd) > (1/2)))
{ persaxes[Demnd]++;   persaxes[Neoph]++; }
else { persaxes[Affrm]++;   persaxes[Exprm]++; }
if ((rts_acts_rept_int > 0) && (rts_acts_rept_int < 20)) persaxes[Affrm]++;
else persaxes[Demnd]++;

if ((nb_rept_cmd == 0) || ((nb_rept_slc / nb_rept_cmd) > (1/2))) persaxes[Demnd]++;
else persaxes[Affrm]++;
if (rts_aa_int > 80)   { persaxes[Affrm]++;   persaxes[Exprm]++; }
else { persaxes[Demnd]++;   persaxes[Neoph]++; }
if (nb_sequ > 100)   { persaxes[Affrm]++;   persaxes[Exprm]++; }
else { persaxes[Demnd]++;   persaxes[Neoph]++; }
if ((rts_sequ_int == 0) || (rts_sequ_int > 20)) persaxes[Demnd]++;
else persaxes[Affrm]++;
if (avs_learn_int > 4) persaxes[Affrm]++;   else persaxes[Demnd]++;

if (nb_sequ_prop > 50) persaxes[Exprm]++;   else persaxes[Neoph]++;
if ((avs_learn_int != 0) && (avs_prop > avs_learn_int))
{ persaxes[Demnd]++;   persaxes[Exprm]++; }
else { persaxes[Affrm]++;   persaxes[Neoph]++; }
if ((nb_sequ_prop != 0) && ((nb_sequ_exec / nb_sequ_prop) > (4/5)))
    persaxes[Demnd]++;   else persaxes[Affrm]++;
if ((nb_sequ_exec != 0) && ((nb_sequ_exec_predict / nb_sequ_exec) > (3/4)))
    persaxes[Affrm]++;   else persaxes[Demnd]++;
if ((avs_prop != 0) && (avs_exec > avs_prop))
{ persaxes[Demnd]++;   persaxes[Exprm]++; }
else { persaxes[Affrm]++;   persaxes[Neoph]++; }

if (nbacts_exec > 50) persaxes[Exprm]++;   else persaxes[Neoph]++;
if (rts_aa_exec > 20) persaxes[Demnd]++;   else persaxes[Affrm]++;
if ((nb_simpl_prop != 0) && ((nb_simpl_prop / nb_distct_sequ) > (1/5)))
    persaxes[Neoph]++;   else persaxes[Exprm]++;
if (rts_exec_simpl > 50)   { persaxes[Demnd]++;   persaxes[Neoph]++; }
else { persaxes[Affrm]++;   persaxes[Exprm]++; }
if (rts_simpl > 50)   { persaxes[Demnd]++;   persaxes[Neoph]++; }
else { persaxes[Affrm]++;   persaxes[Exprm]++; }

rescarct = persaxes[Affrm] - persaxes[Demnd];
resexprn = persaxes[Exprm] - persaxes[Neoph];

if (rescarct > 0)
    if (resexprn > 0)
        perstype = "Controller   ( Affirmer and Experienced )";
    else if (resexprn < 0)
        perstype = "Promoter     ( Affirmer and Neophyte )";
    else perstype = "Affirmer     ( Controller or Promoter )";
else
    if (rescarct < 0)
        if (resexprn > 0)
            perstype = "Analyzer   ( Asker and Experienced )";
        else if (resexprn < 0)
            perstype = "Facilitater ( Asker and Neophyte )";
        else perstype = "Asker     ( Analyzer or Facilitater )";
    else
        if (resexprn > 0)
            perstype = "Experienced ( Controller or Analyzer )";
        else perstype = "Neophyte   ( Promoter or Facilitater )";

```

Figure 5.2.5.2 : *Lignes les plus significatives de l'implémentation concernant les investigations menées par le module attaché au traitement des connaissances de chacun de nos utilisateurs*

```

if (rescarct > 0)
  if (resexprn > 0)
    { // perstype = "Controller (Affirmer and Experienced)";
      seuil_wany = 0; seuil_refus = 5; seuil_predict = 7; seuil_anticip = 14; }
  else
    if (resexprn < 0)
      { // perstype = "Promoter (Affirmer and Neophyte)";
        seuil_wany = 1; seuil_refus = 3; seuil_predict = 5; seuil_anticip = 10; }
      else
        { // perstype = "Affirmer (Controller or Promoter)";
          seuil_wany = 0; seuil_refus = 4; seuil_predict = 6; seuil_anticip = 12; }
    else
      if (rescarct < 0)
        if (resexprn > 0)
          { // perstype = "Analyzer (Asker and Experienced)";
            seuil_wany = 2; seuil_refus = 3; seuil_predict = 6; seuil_anticip = 12; }
          else
            if (resexprn < 0)
              { // perstype = "Facilitater (Asker and Neophyte)";
                seuil_wany = 3; seuil_refus = 2; seuil_predict = 5; seuil_anticip = 10; }
              else
                { // perstype = "Asker (Analyzer or Facilitater)";
                  seuil_wany = 3; seuil_refus = 2; seuil_predict = 5; seuil_anticip = 11; }
            else
              if (resexprn > 0)
                { // perstype = "Experienced (Controller or Analyzer)";
                  seuil_wany = 1; seuil_refus = 4; seuil_predict = 6; seuil_anticip = 10; }
                else
                  { // perstype = "Neophyte (Promoter or Facilitater)";
                    seuil_wany = 2; seuil_refus = 2; seuil_predict = 4; seuil_anticip = 8; }

```

Figure 5.2.5.3 : *Principale étape demandée au module de gestion des seuils auto-adaptés, à savoir, pour ce processus achevant le travail réalisé par notre ultime agent intelligent, l'ajustement dynamique et en temps réel des caractéristiques de la relation homme-machine (cf. 4.5.3.2)*

5.2.6. Conclusion et enseignements

Une contrainte de notre modèle d'interface intelligente était la recherche d'une méthode efficace entraînant un processus d'implémentation et de fonctionnement global discret pour nos utilisateurs. Associés à des entités tendant vers une compétitivité maximale vis-à-vis du travail en temps réel, nos différents agents intelligents se sont alors avérés très pertinents, comme nous le démontre l'**Annexe 4** et puisqu'une seule récursivité notable est à signaler au sein de notre "Couche IA". Il s'agit d'ailleurs plus précisément de celle correspondant à la détermination des séquences d'actions à exécuter de manière auto-adaptable (**Figure 5.2.2.5**). Issue de l'utilisation des concepts *MAPI* (cf. **5.1.2**), cette procédure résursive a été indispensable pour une application concrète et valide du déroulement de tous nos algorithmes d'apprentissage automatique.

Ainsi, notre système multi-agents aboutit à un outil d'assistance auto-adaptable, attentif en permanence à ne ralentir ou à ne pénaliser, en aucun cas, l'efficacité de l'utilisateur, que ce soit en temps comme en espace-mémoire. Le principe du séquençement d'actions demeure par conséquent un choix judicieux pour nos programmes et autres implémentations ayant vu le jour : toujours discrets mais pleinement actualisés en temps réel, nos agents intelligents apparaissent en effet aujourd'hui de plus en plus appréciés par l'utilisateur...

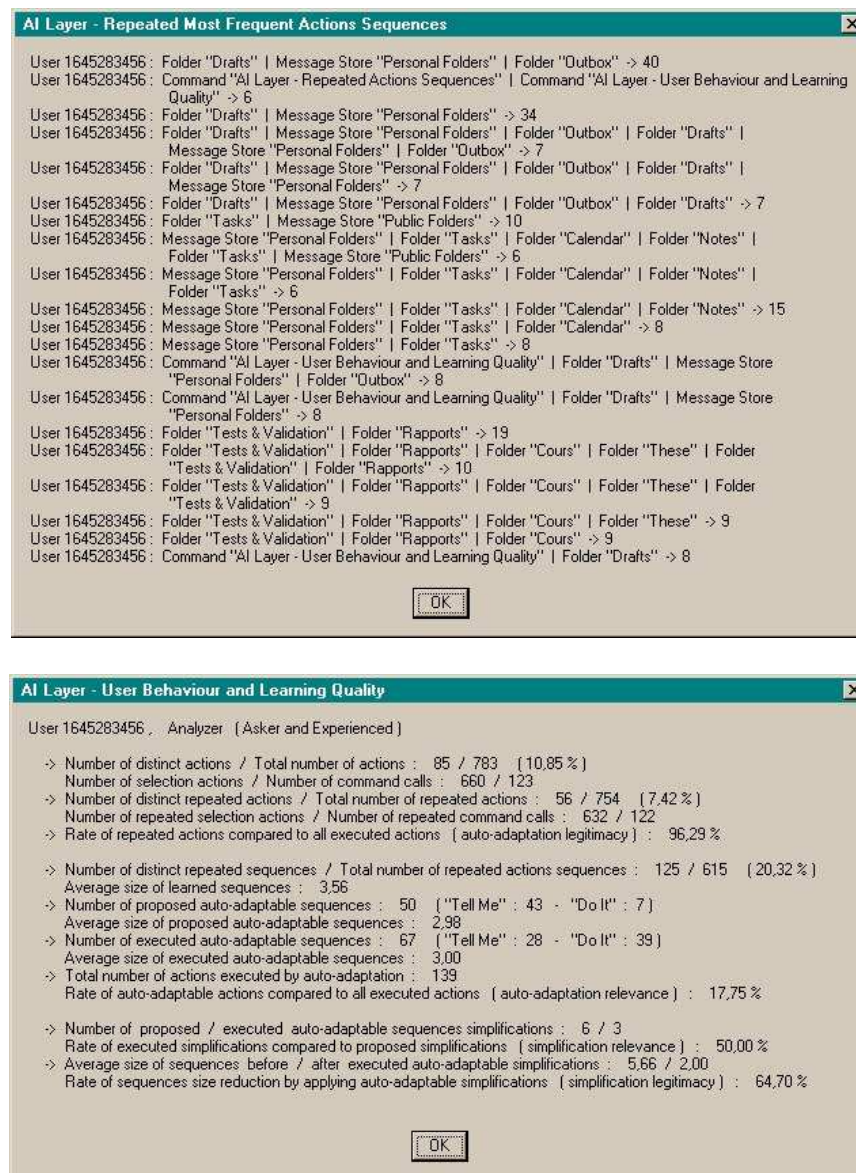


Figure 5.2.5.4 : *Copies d'écrans faisant état de divers résultats obtenus lors d'utilisations de nos nouveaux agents intelligents, et plus particulièrement ici à propos de nos deux dernières entités intitulées "Agent de Séquençage" et "Agent de Comportement"*

Dans un second temps, de nombreuses difficultés sont rapidement apparues, aussi bien au niveau de la recherche et de la conception d'un système multi-agents et surtout indépendant, qu'en ce qui concerne toute la mise en œuvre et les implémentations concrètes de nos théories, de nos procédés auto-adaptables et nouveaux algorithmes d'apprentissage automatique. Et à tout ceci se sont ajoutées bien d'autres complications, dès lors que nous nous sommes penchés sur la recherche de qualité et de complétude maximales pour chacun de nos algorithmes, d'efficacité et de rapidité des traitements de tous les instants aussi, et ce afin de conférer une réelle légitimité et une pertinence matérielle à chacune des utilisations successives de notre "Couche IA".

Toutefois, ces difficultés de mise en œuvre et programmation, notamment distribuées, ne se sont pas arrêtées à ce stade puisque de très nombreux problèmes techniques, pour la plupart même jamais imaginés au sein des départements de recherche et de développement d'ALCATEL, ont été rencontrés et bien entendu tous levés au cours de nos travaux sur l'AUM. Pour exemple, nous avons entre-autres longuement dû nous employer à découvrir et appliquer ensuite diverses et multiples informations (allant jusqu'à des données indispensables mais encore inconnues et inexploitées dans les systèmes de l'entreprise), voire même plusieurs fois finement ruser pour aboutir à une implémentation réellement efficace de nos auto-adaptations...

Sans oublier les conflits d'implémentation, de compilation et même d'exécution qui sont régulièrement apparus, non seulement entre beaucoup de méthodes (nouvelles et anciennes), mais également entre les agents intelligents eux-mêmes. D'où l'utilité et la nécessité plusieurs fois avancée (cf. notamment 4.5.3.2 à ce sujet) de la mise en place de totales coopérations et coordinations complètes entre toutes nos entités auto-adaptables.

Entièrement programmés sous forme d'extensions, de même que tous indépendants des applications logicielles et des interfaces-utilisateur traitées par notre "Couche IA", nos agents intelligents sont aujourd'hui aisément portables et maintenables, de quelconques modifications apportées à l'un ou l'autre de ces entités n'entraînant en rien des interventions supplémentaires sur les autres agents (cf. notamment 5.1.3). De plus, notre système multi-agents est bien entendu aussi "multi-utilisateurs" : plusieurs utilisateurs peuvent effectivement travailler en même temps avec notre "Couche IA" puisque leurs actions et séquences d'actions respectives se distinguent par l'identifiant-même de chaque session d'utilisateur. Par extension, un même utilisateur peut travailler avec plusieurs fenêtres ouvertes, et ce sans conséquence au niveau de l'ensemble de nos concepts et procédés intelligents.

Pour conclure, nous disposons maintenant, pour un même utilisateur et une application logicielle donnée, de processus intelligents et indépendants, bien regroupés au sein d'un système global. Une séquence d'action effectuée dans une fenêtre précise sera ainsi apprise, répertoriée et reconnue au travers de cette même fenêtre, mais également en ce qui concerne toutes les autres interfaces ouvertes à cet instant par l'utilisateur concerné.

Ainsi, nous allons nous pencher sur la validation complète des modèles auto-adaptables. Si les tests dits "unitaires" permettent alors de vérifier que nos unités de programme, modules informatiques ou encore procédures répondent bien à leurs spécifications respectives, les tests "système" et "d'acceptation", nettement plus globaux, sont ensuite réalisés pour confirmer que l'ensemble des besoins logiciels initiaux ont été satisfaits.

Excepté pour de très petits programmes, il n'est en outre pas réaliste de vouloir tester un logiciel comme une seule unité. Si nous avons construit notre système intelligent en assemblant successivement de plus petites entités ou sous-systèmes, ces derniers correspondent aussi à des assemblages de modules, eux-mêmes assemblages de procédures. C'est pourquoi, de la même façon que pour les processus de conception et programmation, les tests se feront rigoureusement par étapes, chaque étape constituant la suite logique de l'étape précédente.

En résumé, les rapports de tests à venir vont attester d'une mise en œuvre correcte de nos implémentations en utilisant des données similaires aux données réelles, ce tout en observant les résultats obtenus afin de détecter des anomalies et en déduire l'existence d'éventuelles erreurs de programmation. Nous verrons par ailleurs que la connaissance détaillée de la structure d'un logiciel peut être, à terme, extrêmement utile pour construire des tests judicieux et performants. Il conviendra donc d'établir un environnement de travail associant, à la fois, le programmeur et des personnes extérieures à la réalisation-même du système.

5.3. Validation de nos modèles auto-adaptables

5.3.1. Introduction

En achevant les présents travaux de recherche, notre outil informatique, auto-adaptable, baptisé également « Assistant Individuel de Télécommunication », se doit encore d'être évalué, validé et testé au niveau de l'ensemble de ses composantes. Nous avons alors choisi de partager les tests indispensables à une implémentation de nos modèles en quatre étapes successives. En effet, après avoir défini les configurations de plate-forme et d'environnement logiciel, nous nous sommes attelés à formaliser et valider les critères d'évaluation et de performances des diverses caractéristiques des prototypes développés. S'en sont déduites les justifications, la conception et les vérifications proprement dites de scénarios de tests adaptés et documentés, ce afin d'aboutir à la mise en œuvre intégrale d'une étude suivie de nos concepts comportementaux sur un panel d'utilisateurs aussi variés et représentatifs que possible.

Cette population d'individus, issue inévitablement du groupe *ALCATEL* pour d'évidentes raisons d'intérêts industriels dues à la vive concurrence qui caractérise actuellement le domaine des télécommunications, s'est composée d'une dizaine de développeurs en informatique, mais que nous avons surtout réussi à mobiliser en provenance d'équipes bien distinctes comme nous l'explicitons nettement plus en détail un peu plus loin. Et nous notons également à ce niveau que l'ensemble des évaluations, des justifications, des vérifications, des scénarios et bien entendu des résultats, est finalement rapporté au sein d'une **Annexe 4** très complète au sujet de ces travaux primordiaux de validation, surtout pour un outil informatique et logiciel présentant autant de facettes auto-adaptables et interconnectées.

5.3.2. Initialisation de la validation du prototype développé

5.3.2.1. Détermination des contraintes d'environnement

Nos agents intelligents et modèles auto-adaptables dédiés aux différents comportements d'utilisateurs face à un terminal de communication accédant à *Internet*, ont tous été implémentés dans le cadre de l'application *ALCATEL Unified Messaging 2.4 Build 2.4.5.0*.

En faisant appel à un PC et à un serveur de communication (*ALCATEL 4400 PBX*) lui-même associé à un poste téléphonique (fixe ou encore mobile), l'*AUM* se définit actuellement par un produit de messagerie unifiée fonctionnant à partir du système d'exploitation *Microsoft NT Server 4.0 Build 1381 (Service Pack 6)*. S'appuyant en outre sur le serveur de messagerie électronique *Microsoft Exchange Server 5.5 Build 2653.22*, l'*AUM* assure une interaction complète avec ses divers utilisateurs grâce à l'application *Microsoft Outlook*. Nous nous sommes alors efforcés de mettre en œuvre notre « Assistant Individuel de Télécommunication » afin qu'il soit totalement disponible dans les versions les plus récentes de cette précédente application, à savoir *Outlook'97*, *Outlook'98*, mais aussi et surtout *Outlook 2000*. En allant encore plus avant, nous pouvons même affirmer, juste avant d'achever le présent mémoire, que l'ensemble de nos différents modèles s'avèrent tous aussi s'adapter pleinement aux nouvelles émulations d'*Outlook 2002* (ou *Outlook XP*), aujourd'hui déjà disponibles au sein d'*ALCATEL*.

Pour résumer, nous disposons de la configuration matérielle ci-dessous (**Figure 5.3.2.1**).

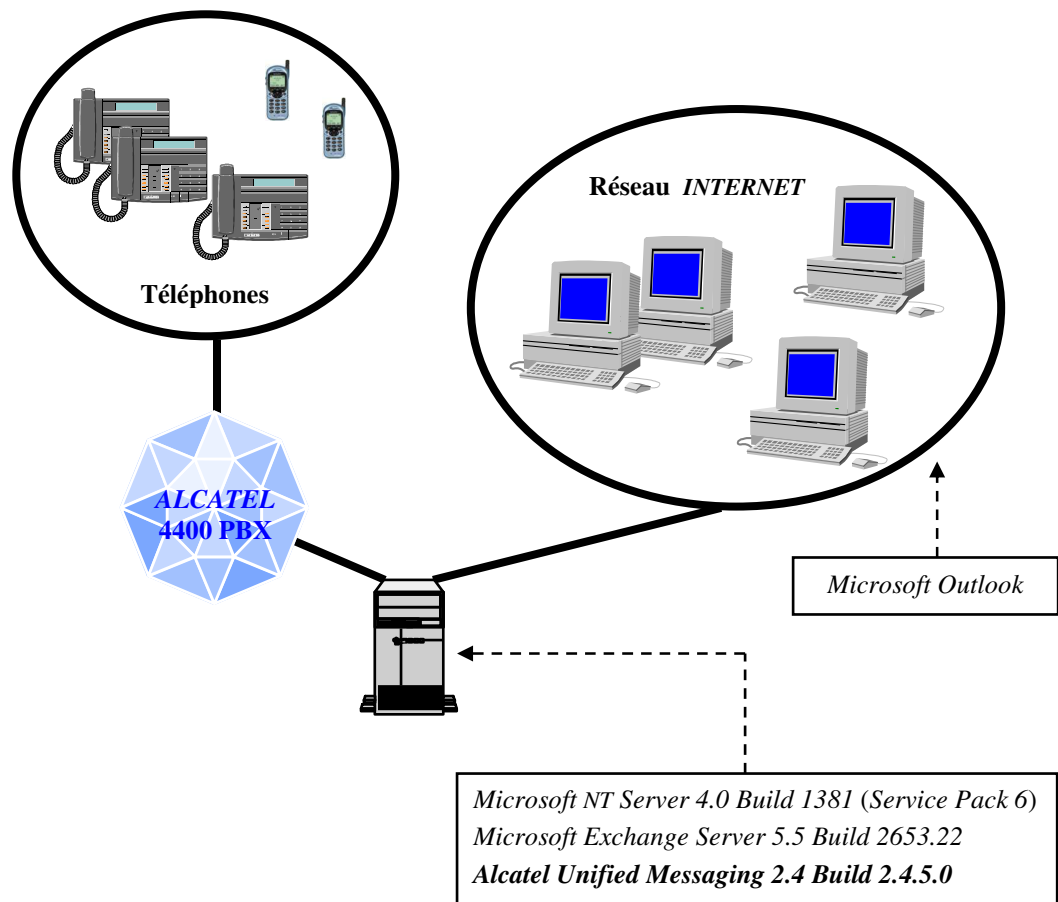


Figure 5.3.2.1 : Configuration physique mise en oeuvre pour les tests de l' AUM

5.3.2.2. Intégration du cycle de validation d'ALCATEL

En nous appuyant sur le dernier rapport interne traitant de la spécification des tests applicables à l'AUM [Augst, 2000], mais aussi bien entendu sur l'expérience des équipes de développement d'ALCATEL en matière de validation approfondie de ses logiciels informatiques, nous constatons que cette organisation est nécessaire dans le but d'offrir des solutions globales aux multiples besoins des différents clients d'ALCATEL. S'inscrivant au sein de la dernière phase de planification de la production d'un produit, immédiatement après le développement de celui-ci et juste avant le début de sa mise en fabrication industrielle, le processus de validation correspond à une revue technique complète, constituée de nombreux scénarios de tests variés ("alpha-tests"). Ces derniers seront eux-mêmes prolongés par des expérimentations réelles chez plusieurs clients ("beta-tests"), mais en dehors de la responsabilité de l'entité *Recherche & Développement*. La validation est ainsi essentiellement destinée à qualifier, au sens propre du terme, tous les systèmes de composants d'un produit, avant d'élaborer physiquement ce dernier (**Figure 5.3.2.2** à suivre).

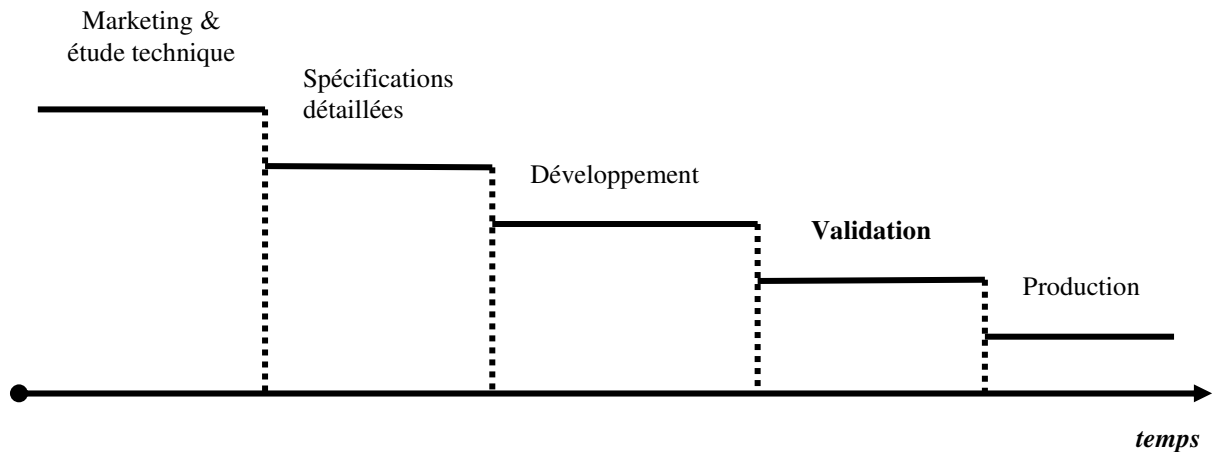


Figure 5.3.2.2 : *Echelonnement général simplifié du cursus d'élaboration des produits d'ALCATEL*

Si l'AUM est bien exploitable de manière fixe (ordinateur personnel ou PC) comme en déplacement (téléphone mobile), nous nous sommes avant tout intéressés, en accord avec les besoins actuels d'ALCATEL en matière d'interface intelligente, à l'interfaçage homme-machine disponible sur le PC, que ce soit au bureau en utilisation professionnelle ou également en privé. Conformément à la **Figure 5.3.2.1**, le PC étudié est alors directement connecté au serveur de Messagerie Unifiée, via l'*Intranet* local ou *Internet* à l'aide du protocole associé TCP/IP. Ainsi, la validation de nos prototypes s'est révélée très encourageante puisque les tests fonctionnels et l'ensemble des manipulations réelles effectuées sur les applications d'ALCATEL se sont conclus de manière tout-à-fait positive et sans aucun accroc (cf. **Annexe 4**).

5.3.3. Etude suivie sur un panel d'utilisateurs représentatifs

5.3.3.1. Descriptif et mise en œuvre des caractéristiques de l'étude

Les travaux de l'**Annexe 4**, notamment les scénarios appliqués à nos prototypes, doivent également être "vulgarisés". Il s'agit de multiplier les "alpha-tests" afin d'aboutir aux "beta-tests" dédiés à la véritable clientèle, aux utilisateurs visés par nos modèles auto-adaptables. Pour cela, nous proposons d'étudier nos concepts comportementaux sur un panel de divers utilisateurs. Notre population de test est ainsi constituée d'une dizaine de développeurs en informatique, issus directement de plusieurs équipes distinctes d'ALCATEL.

En effet, nos utilisateurs se divisent en trois équipes d'horizons divers : quatre individus composent tout d'abord notre groupe de "Messagerie Unifiée", trois personnes représentent les "Applications CTI" (*Computer Telephony Integration*), et deux autres enfin relèvent du propre domaine de la "Validation". Initiée dès la fin du mois d'Août 2001, cette "étude-utilisateurs" s'est déroulée sur deux mois complets et très représentatifs de la réalité quotidienne, puisque sources également, en pleine période de rentrée, d'activités bien soutenues pour les individus concernés.

Enfin, nos agents intelligents ayant été implémentés en entités exécutables et portables (cf. **5.1**), l'installation de notre étude a été aisée puisqu'il nous a concrètement suffi d'ajouter ces quatre fichiers en extension aux applications de messagerie choisies pour entretenir nos tests. Nous avons alors testé nos algorithmes sur des laps de temps relativement longs et fournis, très proches somme toute d'utilisations complètes attachées à la réalité du marché de tels logiciels. Nous avons pris conscience également de leurs bons fonctionnements respectifs, que ce soit au niveau de l'apprentissage et du traitement des séquences, ou à leurs applications au quotidien.

Et, si nous avons choisi d'intégrer à nos programmes de nombreux calculs implicites de statistiques résumant l'activité de chacun de nos utilisateurs, le concept d'une fenêtre d'interface dédiée à ces recueils d'informations a été développée, notamment au travers du quatrième agent intelligent chargé de répertorier et de définir le comportement propre à nos utilisateurs courants. Cela entraîne, en définitive, une visualisation graphique rapide et globale de tous les résultats inhérents à nos processus d'apprentissages auto-adaptables. Suit donc l'étude des aboutissements personnalisés et des manipulations remarquables issues des membres de notre population de test.

5.3.3.2. Dépouillement des données recueillies et analyse des résultats

L'étude-utilisateurs s'est voulue, à l'image des principes qualifiant tous nos mécanismes d'auto-adaptation, pleinement indépendante, discrète et implicite, ne modifiant et n'indisposant en rien les manipulations de l'utilisateur. En effet, toutes nos procédures permettent de collecter chacune des transactions entre l'homme et l'interface de manière totalement automatique. Nous nous sommes également efforcés de mettre en œuvre plusieurs types de traitement et analyse des données collectées durant les tests, comme bien sûr une approche globale chargée d'appréhender les informations dans leur ensemble et dans leurs contextes respectifs [Bernard, 1998]. Une autre démarche encore, plutôt comparative celle-ci, s'est attachée à considérer plus précisément l'effet de nos procédés auto-adaptables sur les divers utilisateurs étudiés.

Si la première de nos investigations a ainsi permis d'étudier le comportement des entités, homme ou machine, et quel que soit la situation dans laquelle se trouve le système, elle a ensuite contribué à vérifier que toutes les composantes des modèles soient bien capables de correspondre et de s'adapter aux désirs, attentes et préférences de chaque utilisateur. De leur côté, les diverses comparaisons menées au sujet de l'adéquation entre nos systèmes et les manipulations opérées, ont clairement différencié les enseignements inhérents aux acceptations positives et négatives des utilisateurs. Ainsi, lors d'un affichage propre à une interaction auto-adaptable, si l'un de ceux-ci répond par la négative, ou même agit ensuite de façon différente à ce que propose le message en question, ce comportement est révélateur, du moins indicateur, de la non-pertinence du message diffusé pour aider cet homme dans le contexte où il se trouve : l'interaction aurait pu ne pas se faire, sans que cela n'ait d'incidence sur le comportement futur de l'utilisateur. Mais, dans le cas contraire, si l'utilisateur accepte et autorise nos systèmes à effectuer des actions de manière auto-adaptable, nous pouvons alors légitimement en déduire, et retenir, que l'assistance a été efficace, améliorant effectivement le travail et les manipulations de l'homme face à sa machine.

En définitive, en nous appuyant donc sur une étude-utilisateurs très fournie, les résultats comme le comportement de tous nos modèles auto-adaptables semblent réellement correspondre aux multiples et diverses attentes de nos utilisateurs. Voici alors, à venir dès la page suivante, un aperçu complet des caractéristiques les plus notables que nous avons pu recueillir tout au long de nos conséquentes démarches de tests (**Tableau 5.3.3.2**).

Tableau 5.3.3.2 : *Récapitulatif des informations quantitatives les plus significatives, obtenues de l'étude-utilisateurs effectuée entre la fin Août et la début Novembre 2001 avec l'ensemble de nos agents intelligents*

Utilisateurs Données	Equipe " Messagerie Unifiée "				Equipe " Applications CTI "			Equipe " Validation "	
	①	②	③	④	⑤	⑥	⑦	⑧	⑨
Nombre total d'actions effectuées	3 822	5 702	4 041	3 615	2 844	6 690	4 356	6 424	7 047
Pourcentage d'actions distinctes par rapport au total effectué	2,82 %	2,89 %	4,03 %	4,53 %	6,75 %	2,21 %	3,25 %	3,30 %	3,61 %
Nombre de sélections	3 198	3 828	3 192	2 958	2 088	4 509	3 204	5 308	5 940
Nombre d'appels de commandes	624	1 874	849	657	756	2 181	1 152	1 116	1 107
Nombre total d'actions répétées	3 774	5 660	3 948	3 480	2 724	6 624	4 316	6 352	6 786
Pourcentage d'actions répétées par rapport à toutes celles effectuées (légitimité de l'auto-adaptation)	98,74 %	99,26 %	97,69 %	96,26 %	95,78 %	99,01 %	99,08 %	98,87 %	96,29 %
Nombre total de séquences répétées	1 176	2 928	1 380	1 056	1 116	1 404	1 644	1 488	3 690
Pourcentage de séquences distinctes par rapport au total effectué	14,28 %	8,60 %	16,95 %	6,81 %	18,27 %	20,94 %	15,08 %	27,41 %	20,32 %
Taille moyenne des séquences travaillées	2,96	2,72	2,66	3,13	3,20	2,71	2,66	3,36	3,56

Le nombre total d'actions effectuées est très disparate selon les individus abordés : il va jusqu'à varier du simple au double. Ces résultats ne sont pas non plus similaires au sein de chaque équipe de travail appréhendés. Mais, cela nous montre toutefois déjà que notre population a été choisie de manière relativement réfléchiée et judicieuse, puisque chacun de ses membres semble présenter des habitudes de travail et autres besoins de messagerie tout-à-fait personnels.

Et les enseignements de s'enchaîner ensuite très rapidement, en relevant en outre que l'utilisation des pourcentages apparaît aussi avantageuse et appréciée qu'inévitable. En effet, vu que les quantités d'actions manipulées par chacun de nos utilisateurs sont relativement éloignées, seul un rapport d'activité établi dans une même base d'unité comme le pourcentage peut nous permettre de comparer, et par suite de généraliser, toutes nos données. Nous découvrons ainsi que tous les pourcentages décrivant le nombre d'actions distinctes dans l'ensemble total de celles effectuées sont inférieurs à 7% (4,48% \pm 2,27 pour être précis) : la légitimité, la nécessité actuelle

d'un traitement des répétitions d'actions est alors, dès ce niveau, pleinement prouvée et justifiée. En outre, tous les individus de la population de test réalisent beaucoup plus de sélections (accès aux répertoires, désignation et lecture de messages, prise de connaissance des attachements, ...) que d'appels de commandes telles que les impressions, les écritures de nouveaux messages, tous les changements concernant le bureau de travail, ou encore les demandes d'informations, ... La **Figure 5.3.3.2** à venir, associée au tableau précédent, entérine ces constats de façon plus visuelle.

En conséquence, pour nos utilisateurs, l'utilisation des applications testées apparaît plus passive qu'active, et sujette à davantage de renseignements que de réalisations. Ceci correspond pour certains jusqu'à un pourcentage inférieur à 20% d'appels de commandes par rapport aux actions de sélection, et n'atteint, en tous les cas, pas une seule fois les 50% représentatifs de la moyenne. Par extension, un utilisateur de tels logiciels de messagerie électronique attend donc généralement beaucoup d'informations de la machine, indications provenant du monde extérieur ou concordant avec des attentes de rangement, de classification ou gestion de bases de données. Ces résultats supplantent alors fortement les actions de pure exécution inhérentes à un tel outil informatique, mais confortent aussi grandement nos propres travaux par le fait qu'il est à présent prouvé qu'il nous faut, en permanence, apporter un soutien à l'homme. Nous devons le suivre, le comprendre et l'assister dans son travail, voire faire un maximum de manipulations à sa place, réaliser ses actions fastidieuses et répétitives. Pour résumer, l'utilisateur ne doit, dans le cadre des diverses applications qu'il aborde au quotidien, demeurer que "simple" producteur ou rapporteur de données, ce pour communiquer le plus aisément possible avec ses multiples relations. Nos automatisations et auto-adaptations sous-jacentes se doivent ainsi de réussir à s'imposer à lui de manière de plus en plus présente, forte et efficace.

Tout cela est ensuite corroboré par les très impressionnants pourcentages correspondant au nombre d'actions répétitives apparaissant par rapport à l'ensemble des actions effectuées : aucun de ces rapports ne descend effectivement sous la barre des... 95%, augmentant même, pour la plupart, avec le nombre d'actions effectuées par chaque utilisateur ! C'est pourquoi, si plus nous réalisons d'actions, plus nous en répétons (ce qui paraît quelque peu réaliste, vu que chaque application traitée possède un nombre conséquent mais tout-de-même fini de possibilités, à part bien sûr ce qui concerne les messages entrants), nous ne pouvons que convenir ici, encore une fois, de la nécessité actuelle d'appréhender de manière intelligente l'ensemble des habitudes, itératives, des utilisateurs. Sans perdre de vue toutefois que la difficulté réside toujours dans le fait que chacun d'entre-eux en effectue bien entendu des différentes !...

Suite à l'exécution de ces multiples actions répétitives, chaque individu en arrive alors à répéter des séquences entières d'actions, dont la moyenne des tailles, dépassant souvent la valeur de 3 ($3,11 \pm 0,45$ exactement pour l'ensemble de l'étude), varie avec chacun des styles de travail, mais tend tout-de-même régulièrement vers des séquences composées de 6 voire 7 actions, donc consécutivement répétées. Par contre, si le nombre de séquences d'actions répétées demeure très disparate en fonction de l'utilisateur considéré, aucune autre relation remarquable n'est à établir avec les données précédentes et, en particulier, le nombre d'actions réalisées ou répétées.

Tout ceci appuie encore une fois la nécessité des fondements de la "Couche IA" puisque, à part une unique exception (due vraisemblablement à un travail orienté par l'individu vers une lecture massive de messages présentant des expéditeurs et autres sujets totalement éloignés), tous les utilisateurs présentent un pourcentage de séquences distinctes, par rapport à celles effectuées, inférieur à 21%, voire même nettement inférieur à ce taux pour deux d'entre-eux. Les répétitions de séquences d'actions, à l'image d'ailleurs de celles relevées précédemment pour les actions en général, apparaissent donc très nombreuses et appellent de manière très forte nos diverses auto-adaptations capables entre-autres de réaliser, en substitution ou du moins en assistance à chaque utilisateur, les fréquentes itérations pouvant vite devenir insipides...

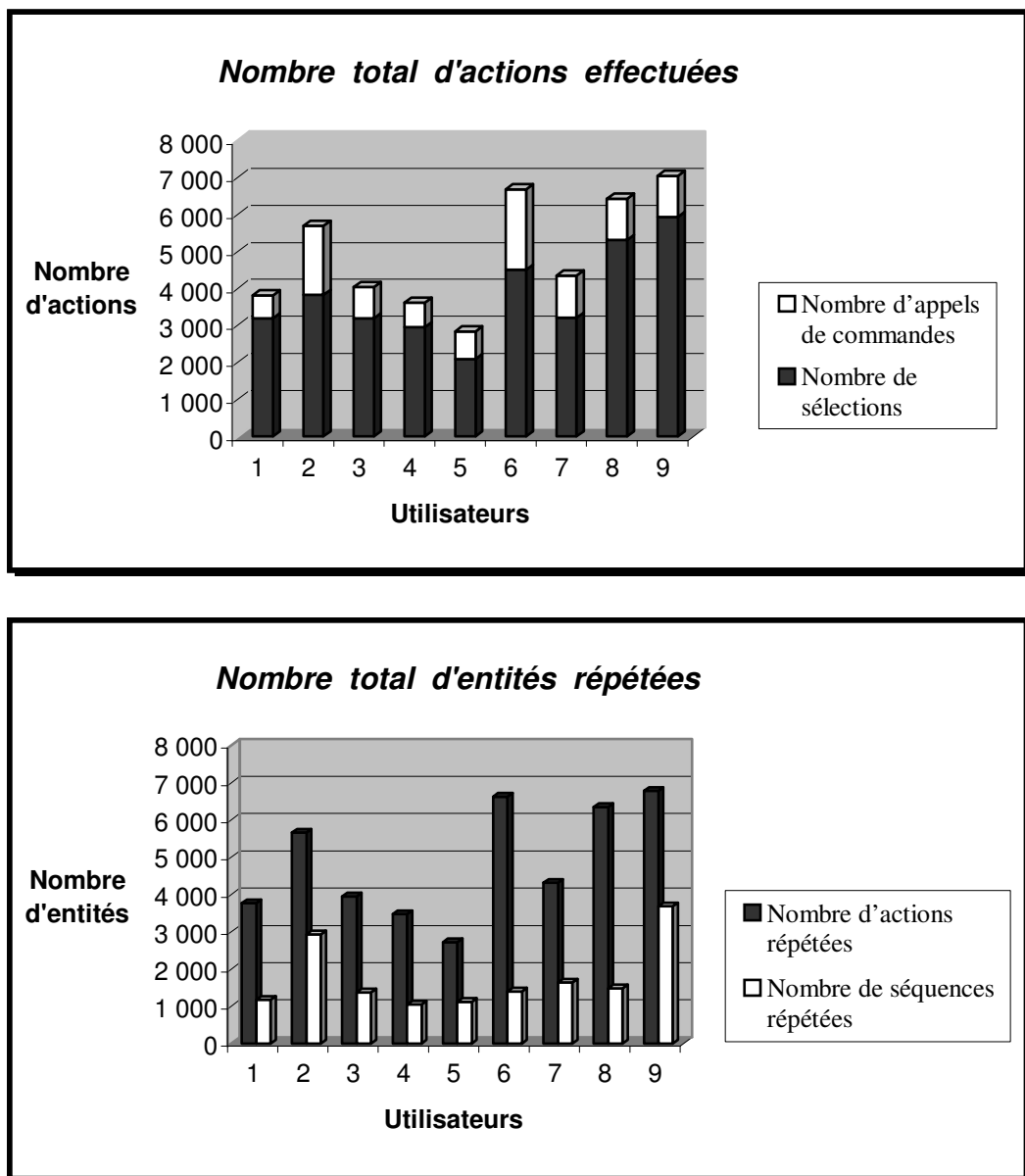


Figure 5.3.3.2 : Graphiques récapitulatifs des principales données obtenues

Enfin, si nous n'avons pas jugé utile de rapporter, au **Tableau 5.3.3.2**, les différentes données relatives à l'exécution et à la simplification auto-adaptables des séquences de chaque utilisateur, c'est qu'elles ne sont, à ce niveau, pas vraiment apparues pleinement significatives. En effet, galvanisés par notre nouvel outil intelligent et peut-être aussi par excès de zèle de leur part (que nous ne pouvons raisonnablement pas leur reprocher, au contraire !...), il s'est avéré que les protagonistes de ces tests ont tous presque systématiquement répondu par l'affirmative à chaque proposition d'exécution automatique d'une séquence remarquable ! A défaut alors d'être représentative, notre étude-utilisateurs a néanmoins démontré, à l'image de l'**Annexe 4**, que nos quatre agents auto-adaptables étaient en pleine possession de leurs moyens. Et nous sommes sûrs que l'avenir nous permettra d'obtenir des informations plus significatives de ces ultimes facettes de nos modèles intelligents.

Quant aux procédés de simplification auto-adaptable, aucune erreur ou incohérence de manipulation, pas même un retour en arrière rapide, n'a été détecté par nos systèmes. Mais il est également vrai que notre population de test était, sans autre possibilité envisageable comme nous l'avons explicité au cours de l'introduction à cette étude (cf. **5.3.3.1**), composée d'informaticiens aguerris... Ceci demeurant sans réelle importance puisque les complètes validations recueillies au sein de l'**Annexe 4** ont pleinement prouvé que notre simplification auto-adaptable fonctionne de façon utile et légitime, comme grandement pertinente et efficace. De cette étude approfondie, il ressort alors aussi très justement, à l'image du constat émanant du procédé de simplification abordé à l'instant, que chacun de nos individus se voit qualifié d'utilisateur "expérimenté". En effet, il s'avère qu'ils parviennent à contrôler très aisément leurs manipulations respectives et les mécanismes auto-adaptables inhérents proposés par notre "Couche IA", qu'ils peuvent également tous bien les analyser et surtout sereinement les organiser.

5.4. Conclusion

Nous avons adjoint à nos nouveaux et différents agents intelligents de nombreuses idées ayant entraîné bien des modules d'implémentation avancés. Cependant, nous retenons aussi que les problèmes liés à la recherche de représentations adéquates ainsi que de traitements efficaces, pertinents, utiles et adaptés aux séquences d'actions, réalisées de manière préférentielle par nos utilisateurs, ont, chacun à leur tour, présenté un certain nombre de difficultés de taille.

En conséquence, et concernant bien entendu des résistances souvent très intéressantes à étudier, nous avons œuvré à améliorer les diverses facettes de ces multiples points névralgiques, théoriques et pratiques, à l'image de toutes les informations détaillées dans ce chapitre. Ainsi, nous avons par exemple rencontré plusieurs difficultés de choix au niveau de la découverte, de l'utilisation et du traitement des tous récents langages de messagerie électronique tels que *COM* et *MAPI*, de la gestion de mémoire en temps réel, des travaux propres à la portabilité de nos entités logicielles, de la compréhension des nouveaux attraits intelligents, de l'application aussi intelligente des fondements de l'interface homme-machine...

C'est pourquoi, nous ne pouvons manquer de retenir que, suite à l'intégration réussie de nos programmes informatiques exécutables au sein des terminaux de communication d'*ALCATEL* et donc de l'*AUM* en particulier, aucune des nombreuses validations alors mises en œuvre ne nous a apporté de test négatif. En effet, si les nombreux "alpha-tests" se sont caractérisés par une indispensable étape de vérification technique de nos diverses implémentations, l'étude-utilisateurs nous a, de surcroît, apporté une validation vraiment complète et une vision globale très réaliste, de l'ensemble de nos fonctionnalités logicielles devenues auto-adaptables pour tout utilisateur.

Ces travaux d'implémentation et de validation apportent par suite une preuve irréfutable que notre nouvelle "Couche d'Intelligence Artificielle" correspond bien à un système totalement indépendant des applications courantes qu'il se charge progressivement "d'humaniser" : nos modèles auto-adaptables constituent une entité à part qu'il suffit "d'insérer" entre l'interface et les différentes applications d'un terminal de communication. En outre, nous avons pleinement pu détailler et concrétiser chacun de ces nombreux modèles théoriques qui composent finalement notre interface intelligente de télécommunication. Enfin, nous nous sommes réellement attachés à découvrir, étudier, maîtriser et entièrement appliquer, ce à l'ensemble de nos travaux, tous les différents principes de développement et de validation en usage au sein de la société *ALCATEL*.

Ainsi, notre "Couche IA", logicielle et indépendante, est également devenue adaptable à d'autres plate-formes dédiées à des applications directement intéressées par les mécanismes de modélisation de l'utilisateur, d'apprentissage automatique de séquençements d'actions comme la prédiction, l'anticipation, ou encore l'extraction et la simplification d'erreurs. L'importance de tels systèmes multi-agents, ou même simplement basés sur les agents logiciels, est ainsi liée, de nos jours, au besoin croissant de coordonner des opérations distribuées, ainsi que de simuler des procédés toujours plus complexes de façon décentralisée. Cependant, ils permettent en outre d'accompagner le développement d'une infrastructure de communication, comme de modéliser et de réaliser des systèmes à un niveau d'abstraction de plus en plus élevé.

En conclusion, nous sommes en présence d'un nouveau rapport conceptuel et matériel de l'homme face à la machine, au niveau des manipulations personnelles de celui-ci comme de la quantité d'informations qui les représente. Ceci se trouve corroboré par l'utilisation totale de nos processus d'extraction, d'apprentissage, d'automatisation, de prédiction ainsi que d'anticipation et de simplification enfin. Cette intégration par l'homme, dans des systèmes de type nouveau, de processus intelligents conduit à ne plus faire, mais à faire faire et à laisser faire par des machines dans lesquelles nous avons incorporé du savoir-faire et auxquelles nous fixons de bien nombreux objectifs, de plus en plus diversifiés.

Par extension, les technologies de l'information ne sont ainsi plus des modernisations classiques mais des technologies de rupture qui modifient fondamentalement la relation de l'homme au travail [Verroust, 1997]. Ceci progresse alors, de nos jours, vers la fusion de trois activités humaines bien disparates, à savoir la production directe ou indirecte, la recherche et le développement, de même que l'enseignement et la communication des savoirs.

Les temps de réponse d'un système interactif, tel celui que nous avons conçu, s'avèrent être, finalement, le facteur le plus important influençant l'opinion que se façonne l'utilisateur à propos de ce système. Ainsi, s'il considère que les temps de réponse sont régulièrement excessifs, l'utilisateur garde en effet une relative mauvaise image de l'outil, quelles que soient les diverses fonctionnalités qui lui sont en outre offertes.

Cela dit, il n'est pas non plus possible de réellement définir un temps de réponse acceptable et "standard", ce déjà puisque l'attente d'un utilisateur varie bien évidemment avec la complexité de chaque opération demandée au système. Sur ces points bien précis, si les résultats d'efficacité que nous avons listés et étudiés auparavant apparaissent tous bien représentatifs, ils apportent aussi des caractéristiques de taille pour la présente synthèse de nos agents intelligents implémentés, et la mise en œuvre de nos modèles auto-adaptables. Ainsi, ils se traduisent par d'intéressants gains de productivité, d'économie de temps de travail et de manipulations répétitives en accord avec chaque utilisateur pris en compte, ainsi que de réduction de charges au niveau-même de l'interface du terminal traité.

Conclusion et Perspectives

*Un des éléments-clés du futur de l'information est
qu'un utilisateur va recevoir plus de données
provenant de sources plus nombreuses.*

Steve Ballmer

[Vladyslav, 2001-2]

(P.D.G. de *Microsoft*)

En un peu plus que vingt-cinq ans, les réseaux d'ordinateur ont évolué de systèmes expérimentaux "exotiques" aux parties omniprésentes et indispensables de la vie quotidienne. Aucune autre invention simple n'a eu un impact plus spectaculaire sur la forme et les fonctions de la technologie informatique, ou un effet plus grand sur les vies ordinaires de la population mondiale. Les réseaux enjambent des continents, des ponts d'océans, et s'étendent dans le ciel à une telle mesure qu'il n'y a, sur la planète, aucune place trop éloignée pour être inaccessible par téléphone, *e-mail*, ou *fax*.

Pleinement tournés vers l'avenir, et dans un contexte de recherche aussi vaste et fourni, nous nous sommes alors attachés à travailler avec des applications de communication actuelles, très complexes puisque fournissant un ensemble conséquent de fonctionnalités complémentaires, innovantes pour transférer toutes sortes de données, et donc interagir avec d'autres applications. En effet, si ces dernières exigent des services de systèmes de messagerie sous-jacents, que ce soit pour le stockage d'un message et d'un destinataire, la récupération, l'adressage ou encore le transport de messages, elles peuvent aussi aller jusqu'à soutenir des caractéristiques avancées, comme des formes électroniques, des capacités de recherche avancées et des dossiers publics (conteneurs de messages qui sont accessibles par plus d'un utilisateur simultanément).

Cependant, nous avons souhaité poursuivre nettement plus loin nos travaux, en ajoutant, à ces nouvelles applications de télécommunication, un maximum des qualités reconnues à notre domaine informatique de prédilection, à savoir de l'Intelligence Artificielle. Par la procuration de techniques de gestion de la connaissance (de l'acquisition à l'exploitation) qui, appliquées au dialogue homme-machine, permettent une meilleure prise en compte de l'utilisateur et facilite ses multiples interventions (détections, explications, anticipations, etc), l'Intelligence Artificielle nous a permis de fournir, à chaque utilisateur, des mécanismes capables d'assurer une souplesse et une intelligence de tous les instants. Les deux derniers qualificatifs ont alors été introduits dans le fonctionnement de l'interface, devenue "intelligente" grâce à nos divers modèles d'auto-adaptation (règles, heuristiques, modélisation de la connaissance, architecture multi-agents, etc).

Par conséquent, l'ensemble de nos diverses contributions est à la fois ciblé et très étendu, conformément aux nombreux objectifs que nous nous étions aussi fixés dès le début de la Thèse. En effet, nos travaux de recherche se sont distingués, non seulement par le fait qu'ils aient été liés aux différentes catégories de mécanismes intelligents et autres processus d'apprentissage automatique, mais aussi parce que nous avons veillé à ce qu'ils soient toujours orientés vers tous les domaines existant aujourd'hui et pouvant apporter leurs avantages respectifs à notre but de modéliser l'utilisateur et son propre interfaçage avec les terminaux de communication actuels.

De multiples résultats ont ainsi été obtenus, et ce dans plusieurs directions différentes, comme dans le cadre tout d'abord de nos études bibliographiques et de nos nouvelles définitions comparatives, établies au niveau des interfaces intelligentes et des terminaux de communication. Si nombreuses ont été ensuite nos idées quant à l'intégration de concepts innovants au sein de ces entités dédiées à la télécommunication avec et entre les hommes, les spécifications formelles des scénarios correspondants, orientés vers les principales caractéristiques composant les bases de l'Intelligence Artificielle, nous ont permis d'asseoir solidement la légitimité de nos recherches. De ces deux premiers chapitres indispensables à leur justification, nous avons alors pu concevoir et élaborer d'importantes et complètes méthodes de représentation des connaissances internes aux terminaux, correspondant à leurs mécanismes de raisonnement. S'en est suivie la description de nos premiers processus, intelligents car auto-adaptables, d'extraction et de mémorisation en bases de données des différentes actions ou événements issus du travail de chaque utilisateur, que ce soit au niveau de la récente gamme des *WebTouch* ou sur l'*ALCATEL Unified Messaging*.

Par suite, la découverte et la gestion des actions effectuées de manière répétitive par chacun des utilisateurs ont établi les premières statistiques d'utilisations concernant l'ensemble des fonctionnalités proposées par chaque application traitée par nos systèmes. Et l'apprentissage automatique (symbolique comme statistique) des différents séquençements d'actions réalisés et répétés par l'homme face à la machine, a entraîné une étude approfondie, ainsi que l'obtention de plusieurs définitions innovantes attachées à la construction finale d'un modèle comportemental, incrémental, évolutif et surtout auto-adaptable puisque pleinement dédié à chacun des utilisateurs. Ce Modèle Auto-Adaptable de l'Utilisateur est alors associé à d'autres procédés intelligents : un modèle de prédiction du comportement de l'utilisateur, une anticipation auto-adaptable de ses nombreuses actions ensuite, ceci pour aboutir à une simplification automatisée et temps réel de certaines séquences d'actions remarquables, effectuées par l'utilisateur (boucles et redondances d'informations, mais aussi et surtout erreurs et incohérences de manipulation).

En définitive, quatre nouveaux agents intelligents, correspondant à une implémentation multi-agents indépendante, portable et aisément maintenable, voient le jour grâce, notamment, à l'utilisation des objets logiciels *COM* et *MAPI*, de multiples validations théoriques et pratiques à l'image de nos divers scénarios et de notre étude-utilisateurs réalisée sur un panel bien diversifié. Structurée par des modèles auto-adaptables dédiés à l'utilisateur de terminaux de communication, même "grand public", la Thèse a donc abouti à la mise en œuvre totalement réussie de concepts consistant à proposer aux utilisateurs, du néophyte au plus expérimenté, un système exécutable complet, intègre et autonome, facile aussi à joindre à toute application, de messagerie entre-autres.

Les forces de tels concepts sont alors d'apporter l'efficacité, la rapidité et la convivialité au travail grâce à la convergence d'outils de spécification formelle, d'apprentissage automatique et donc d'auto-adaptation. S'y ajoutent les systèmes de coordination et de synchronisation des divers agents logiciels intégrés, comme ceux d'interfaçage homme-machine, de modélisation de l'utilisateur, d'extraction et d'analyse des connaissances, ainsi que de gestion en temps réel de bases de données de grande taille. Ensemble, ils confèrent tous finalement d'appréciables qualités à notre nouveau modèle d'interface intelligente baptisé, entre-autres, "Couche IA" au regard de son indépendance logicielle associée à une insertion matérielle aisée. En relation directe avec nos objectifs initiaux, celle-ci transforme, assiste et améliore le contact quotidien de l'utilisateur avec sa ou ses machines respectives, de même qu'avec la messagerie électronique. En effet, grâce à tout un panel d'offres et de procédés personnalisés, nous avons pu relier le travail de l'utilisateur courant à une base de données ou d'actions le concernant, dans le but d'adapter l'interface alors manipulée en fonction de ses centres d'intérêts, sa façon de travailler et de revenir régulièrement au sein de certains aspects de sa communication avec la machine. Orienté en permanence vers des utilisations et des applications futures de nos concepts devenus auto-adaptables, nous avons établi

et concaténé l'ensemble de nos domaines et connaissances, en matière tout d'abord de théories propres à l'Intelligence Artificielle, comme ensuite dans le sens des technologies informatiques et télématiques les plus prometteuses à l'heure actuelle.

En outre, nos différents travaux se sont intégrés, au plein milieu d'une époque charnière dans l'histoire des télécommunications, à savoir celle de la convergence voix-données. Celle-ci devient effectivement aujourd'hui un besoin réel et grandissant pour l'utilisateur, ainsi qu'une possibilité technique réalisable depuis peu. Pour exemple, l'*Internet* est en train de restructurer en profondeur nos façons de travailler, tout en relevant que nous n'avons jamais eu autant besoin de communiquer. Il y a donc de nos jours une nécessité d'implémentation efficace d'applications concrètes destinées à simplifier la vie, à augmenter la rapidité de circulation de l'information et de sa propre élaboration, comme de sa richesse et de sa convivialité.

Finalement, le côté pratique de nos recherches, directement appliquées dans l'industrie, nous a forgé une solide expérience en terme de spécification, de formalisation et d'implantations logicielles. Il nous a notamment été indispensable, dès le tout début de nos travaux, de découvrir et manipuler quotidiennement différents systèmes d'exploitation et logiciels spécifiques associés au développement informatique en entreprise. Par rapport aux terminaux de la concurrence, ce concept d'interface intelligente présente alors une originalité et une potentialité de saut et de transfert technologique. L'analyse du marché commercial montre aussi un intérêt grandissant des clients dans la capacité et l'efficacité d'utilisation d'un tel matériel. Nous prenons d'ailleurs aussi pleinement conscience de l'engouement général suscité actuellement par la mise en service, au sein de diverses équipes d'*ALCATEL*, de notre nouvel outil informatique intelligent.

Une telle personnalisation générale du savoir, sans cesse croissante puisque pleinement incrémentale et évolutive, ainsi qu'une accumulation d'informations devenues intelligentes et une nécessité de traitements réellement adaptés en conséquence, ne conduisent heureusement pas inéluctablement à un émiettement de la pensée. Bien au contraire, à l'image d'ailleurs des divers et nombreux contextes que nous avons jugés enrichissants de considérer, il apparaît aujourd'hui clairement que l'on doit établir, à propos de la conception et de la manipulation des interfaces homme-machine du futur, des ponts entre une grande variété de domaines. Nous pensons plus particulièrement à ceux de l'informatique, de l'automatique, de l'ergonomie, de la biologie et de la biochimie (étude des cellules concernant le fonctionnement du cerveau humain), de la psychologie (étude des réactions conscientes et inconscientes de l'individu) et de la physiologie, de la linguistique, de la philosophie, de la logique, de l'anthropologie, de l'éthologie et de la sociologie, de l'économie enfin et de ses marchés... En définitive, à la convergence de toutes ces disciplines, une nouvelle science naît, certes encore lentement : la science cognitive.

S'il existe ainsi de multiples contextes où l'interfaçage entre l'homme et sa machine joue un rôle essentiel, cela appelle et entraîne cependant des problématiques sensiblement différentes. En effet, dans la liste représentative des approches originales susceptibles d'être rencontrées dans les années à venir, ce au sujet de la notion montante que représente alors l'interface intelligente, nous pouvons déjà relever ici le concept d'interface "hypermédia" qui permettra à l'utilisateur de se déplacer, suivant ses besoins et connaissances, dans un réseau de concepts et d'idées. Nous pouvons aussi remarquer l'interface dite "écologique", où il s'agit de présenter l'information de telle sorte que l'utilisateur ne doive que très rarement ressortir de ses processus les plus courants.

Par extension, les formidables progrès accomplis récemment dans la miniaturisation des microprocesseurs, la couverture planétaire des réseaux informatiques ou l'irrésistible chute des prix constituent un terreau propice au changement : les avancées du numérique annoncent donc de profondes mutations suscitant à la fois enthousiasme, terreur et résignation. A nous alors de poursuivre et d'espérer avancer nos recherches, dans le but de toujours améliorer les capacités,

les attraits, la pertinence aussi, voire même les coûts caractérisant les interactions intelligentes que nous avons contribué à faire naître entre l'homme et ses appareils artificiels. Peut-être réussirons-nous ainsi à gommer ces derniers constats ou sentiments encore un peu trop négatifs, émanant des hommes à l'encontre du monde de l'informatique. En effet, dans cette continuité et cette persévérance qui se doivent de caractériser en permanence toute recherche efficace, nous nous devons d'apporter de nouveaux éléments et d'énoncer de solides perspectives de travail qui seront susceptibles d'apparaître quant à nos modèles et prototypes d'auto-adaptation.

C'est pourquoi, nous pouvons tout d'abord remarquer que la catégorisation de chacun de nos utilisateurs peut certainement être encore bien affinée. Pour exemple, nous pourrions ainsi envisager d'intégrer à notre Modèle Auto-Adaptable de l'Utilisateur d'autres études ou méthodes d'analyse des comportements. Nous pensons plus particulièrement au procédé intitulé "*Coping & Stress Profile*" qui pourrait apporter une vision plus "chaleureuse" et "familiale" à nos concepts par la classification des informations en quatre domaines innovants, issus des préoccupations de la vie courante ("Personnel, Travail, Couple, Famille").

Dans un second temps, nous pouvons ajouter à cela l'application de nos modèles auto-adaptables et de leurs implémentations respectives à de nouveaux terminaux de communication, actuellement en passe d'être développés au sein d'ALCATEL. En effet, l'adaptabilité dont nous avons dû faire preuve, notamment lorsque le *Département Recherche* a été dissout sur notre site d'Ilkirsch, nous incite vraiment à en faire de même au niveau de nos agents logiciels. De plus, l'adaptabilité ne serait rien s'il n'y avait pas non plus la diversité. C'est pourquoi, bien que notre "Assistant Individuel de Télécommunication" relève à ce niveau essentiellement du domaine de l'Intelligence Artificielle, nous restons, en ce sens, fervents adeptes de la théorie selon laquelle cette discipline, pour être toujours efficace, ne peut être séparée de celle des bases de données qui constituent inévitablement ses ressources, de même que de l'utilisation intensive et avantageuse de la programmation orientée objet, COM et MAPI en tête.

Par conséquent, nous serons également par suite très intéressés par l'adaptation de nos recherches aux réseaux en général : collaboration entre agents intelligents d'interface au travers de terminaux de communication différents (interfaçage homme-homme, mais aussi machine-machine), productions et gestions d'avatars propres aux modèles auto-adaptables de chacun des utilisateurs, recherche d'informations personnalisées sur *Internet* en fonction des différents traits de caractères déterminés pour tout utilisateur connecté, ou encore assistance des utilisateurs pour un accès efficace, confortable et ergonomique à toute information multimédia. L'application de nos recherches à l'enseignement, et précisément à l'Enseignement Assisté par Ordinateur (EAO), présente enfin une combinaison entre les domaines de l'Intelligence Artificielle, de la gestion de bases de données et surtout des réseaux, ce dans un environnement distribué comme *Internet*. Si leur apparition dans le paysage éducatif semble maintenant inéluctable, les enseignants virtuels concurrenceront-ils alors les professeurs réels ? Seule certitude à ce sujet, si nos modèles auto-adaptables apparaissent bien profitables au monde de l'éducation, l'enseignant humain pourra se concentrer sur le coeur de son métier : la pédagogie, au sens noble du terme.

D'un côté plus matériel encore, nous pouvons aussi grandement nous appuyer sur les résultats obtenus jusqu'alors pour orienter plus particulièrement nos recherches vers l'adjonction de nouvelles technologies apparaissant de nos jours pleines d'avenir. En intégrant en effet à nos concepts une appréhension de l'utilisateur par caméra ou en détournant de leurs utilités premières les récentes versions de visiophones développés notamment par ALCATEL, la qualité de dialogue et d'interaction entre l'homme et la machine s'en trouverait inévitablement améliorée. De telles captures, en temps réel, d'images correspondant aux contextes des actions successives effectuées par l'utilisateur courant, directement associées aux multiples traitements réalisés en parallèle par

l'ensemble de nos agents auto-adaptables, seraient ainsi capables de permettre une complète étude intelligente de chacune de nos attitudes quotidiennes. Et si celle-ci ne se révélerait pas trop intrusive et indiscreète envers ses interlocuteurs, nous pourrions aboutir à une personnalisation sophistiquée, et fidèle aussi, concernant nos multiples comportements, qualités et autres objectifs.

Nos présentes perspectives engageant nos modèles auto-adaptables s'avèrent par suite d'autant plus judicieuses qu'une vraie motivation de la part de l'apprenant, donc placé au cœur d'un tel dispositif, demeure indispensable pour aboutir à un résultat positif : la pratique a en effet montré qu'un échec au sujet d'une telle formation individuelle est inévitable sans un encadrement sérieux et une bonne communication permanente.

Dans le vaste monde que représente aujourd'hui celui des télécommunications et de ses terminaux, nous pouvons aussi imaginer, sans être utopiques finalement, pouvoir citer d'ultimes perspectives en tentant d'associer à nos travaux une révolution en matière de représentation des connaissances manipulées. Nous pensons en effet ici à l'avènement des modeleurs d'informations et à l'établissement du nouveau "design" qui les caractérise. Vue entièrement cartographique, générée dynamiquement et de façon totalement interactive pour explorer toute masse de données textuelles ou autres, cette approche très visuelle et novatrice permettrait en définitive d'afficher, de manière réellement personnalisée et pertinente, l'ensemble des résultats issus du travail de nos divers agents intelligents et auto-adaptables.

Directement liées alors à la composition et à la gestion d'objets innovants sur les écrans, l'adjonction d'une telle vitrine conviviale, de même que l'amélioration conséquente du domaine de l'ergonomie ainsi proposée à l'utilisateur, réussiraient certainement définitivement à mettre en valeur l'entière population de nos agents logiciels, capables rappelons-le de grandement assister leurs interlocuteurs dans leurs démarches respectives et quotidiennes.

Avant de conclure nos présents travaux de recherche, nous rappelons finalement que nos contributions ont toujours été pleinement orientées vers une collaboration et une interconnexion en temps réel caractérisant des entités auto-adaptables, elles-mêmes focalisées sur l'interfaçage homme-machine au sein de terminaux de communication, sur l'assistance et l'apprentissage automatiques dédiés aux utilisateurs, ainsi que sur la modélisation intelligente de chacun d'eux.

Ainsi, toutes les diverses perspectives que nous venons également de mettre en évidence, et qui introduisent donc la poursuite future de nos travaux, s'associent à nos résultats dans le but principal, voire primordial pour l'avenir, de chercher en permanence à améliorer l'entente et la communication entre l'homme et ses machines. Pour tout ceci, il apparaît alors incontournable de réussir à terme à faire travailler ensemble processus autonomes et apprentissages automatiques, ce pour aboutir à des matériels de communication très intelligents et graduellement assimilables aux propres qualités humaines.

Bibliographie

L'ensemble de nos supports bibliographiques a bien entendu été entièrement orienté et dédié à la définition, à la conception ainsi que, finalement, à l'implémentation complète d'un nouveau concept d'intelligence pouvant caractériser une machine lors de sa communication avec un humain.

Il s'en est alors réellement distingué plusieurs catégories, à l'image d'ailleurs des définitions taxinomiques que nous avons établies en **1.2.3** dès le premier chapitre de ce mémoire. En effet, rappelons brièvement que nous avons notamment mis en avant, à ce niveau, cinq dominantes majeures entraînant une représentation véritablement exhaustive de notre bibliographie, à commencer par les modélisations et les conceptions d'interfaces intelligentes, suivies des agents intelligents et des systèmes multi-agents, des modèles-utilisateurs et des comportements auto-adaptatifs, des apprentissages automatiques aussi, et enfin des terminaux de communication, du point de vue logiciel comme matériel.

Ceci exposé, nous nous devons également de retenir ici que nos recherches bibliographiques, effectuées et ajustées de manière approfondie tout au long de la Thèse, se veulent récentes par convention puisque construites de 1993 (5 années avant le début de nos travaux) à aujourd'hui. L'éventail complet de nos références concerne alors plus précisément tous les types d'articles, livres, tutoriels informatiques, diaporamas, supports de cours, compte-rendus, rapports techniques et même parfois sources brutes de programmes, toutes ces références ayant été obtenues non seulement dans les différentes bibliothèques strasbourgeoises et alliées, mais surtout grâce aux nombreux moteurs de recherche mis à disposition par le réseau *Internet*.

Voici donc, sur les pages à suivre, la liste de toutes les principales références bibliographiques, au nombre d'ailleurs de 215 exactement et classées cette fois-ci par ordre alphabétique, que nous avons été amenées à découvrir et à étudier au cours de nos divers travaux de recherche concernant la présente Thèse.

[**Adam, Kolski & Vergison, 1998**]

Adam E., Kolski C., Vergison E., *Méthode Adaptable Basée sur la Modélisation de Processus pour l'Analyse et l'Optimisation de Systèmes Coopératifs dans l'Entreprise*, LAMIH, Université de Valenciennes et du Hainaut-Cambrésis, France, Research and Technology, SOLVAY SA, Bruxelles, Belgique, 6^{ème} Colloque d'Ergonomie et Informatique Avancée (ERGO IA'98), Biarritz, Ecole Supérieure des Technologies Industrielles Avancées / Centre de ressource technologique Innovation-Logiciel-Système (ESTIA/ILS), Bidart, France, pp. 270-279, 1998.

[**Akoulchina & Ganascia, 1997**]

Akoulchina I., Ganascia J., *SATELIT-Agent : an Adaptive Interface Based on Learning Interface Agents Technology*, LAFORIA-IBP-CNRS (LIP6), Université de Paris-VI, France, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 21-32, 1997.

[**Alberganti, 1999**]

Alberganti M., *Les "Profs en Silicium" au Banc d'Essai*, Article de Presse Spécialisée, Le Monde Interactif, France, 1999,
http://www.lemonde.fr/article_impression/0,2322,24510,00.html .

[**Alcatel, 1999-1**]

Alcatel, *"Web Touch One" : Around the World*, Guide Utilisateur Interactif, Colombes, France, Référence 3BN61927 FABG, Décembre 1999,
<http://www.webtouch.alcatel.com/> .

[**Alcatel, 1999-2**]

Alcatel, *Terminaux du Futur : Accès aux Nouveaux Services pour Tous*, Optimiser le Réseau pour les Nouveaux Services Internet, Recherche et Technologies, Centre de Recherche Corporate, Marcoussis, Paris, France, Référence 3CL 003510002 TQZZF, Ed. 02, pp. 10-11, 1999.

[**Alcatel, 1999-3**]

Alcatel, *Unified Messaging : User Guide*, Paris, France, Référence 3BA_19775_FR, Première Edition, 85 p., 1999.

[**Alcatel, 2000**]

Alcatel University (Brest), *Alcatel Unified Messaging*, Guide du Participant Expert, Diaporamas et Supports de Cours, Alcatel Business Systems, Services & Distribution Division, Lannion, France, Référence AUM0T80F_GP, Ed. 02, 330 p., 2000.

[**Alcatel, 2001**]

Alcatel, *Accéder Facilement à Internet*, Newlink, Le Magazine International d'Alcatel, Paris, France, Vol. 9, N° 1, p. 7, 2001.

[**Ambrosini, Cirillo & Micarelli, 1997**]

Ambrosini L., Cirillo V., Micarelli A., *A Hybrid Architecture for User-Adapted Information Filtering on the World Wide Web*, Dipartimento di Informatica e Automazione, Università di Roma Tre, Italy, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 59-61, 1997.

[**Amer, Sethi & al., 1994**]

Amer P., Sethi A., Fecko M., Uyar M., *Formal Design and Testing of Army Communication Protocols Based on Estelle*, Computer and Information Science Department, University of Delaware, Department of Electrical Engineering, City College of New York, Proceedings of the 1st ARL/ATIRP Conference, College Park, pp. 107-114, 1994.

[**Amyot, 1994**]

Amyot D., *Formalization of Timethreads Using LOTOS*, Thesis, University of Ottawa, Ontario, Canada, 213 p., 1994.

[**Amyot, Bordeleau & al., 1995**]

Amyot D., Bordeleau F., Buhr R., Logrippo L., *Formal Support for Design Techniques : a Timethreads-LOTOS Approach*, Telecommunication Software Engineering Research Group, Department of Computer Science, University of Ottawa, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, FORTE VIII : 8th International Conference on Formal Description Techniques, Chapman & Hall, pp. 57-72, 1995.

[**Amyot, Logrippo & Buhr, 1997**]

Amyot D., Logrippo L., Buhr R., *Spécification et Conception de Systèmes Communicants : une Approche Rigoureuse Basée sur des Scénarios d'Usage*, Groupe de Recherche en Télécommunications et Génie Logiciel, Université d'Ottawa, Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'97), Hermes, Paris, pp. 159-174, 1997.

[**Ardissono & Torasso, 2000**]

Ardissono L., Torasso P., *Dynamic User Modeling in a Web Store Shell*, Dipartimento di Informatica, Università di Torino, Italy, Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), Humboldt University, Berlin, Werner Horn, IOS Press, pp. 621-625, 2000.

[**Arnold & Gosling, 1996**]

Arnold K., Gosling J., *Le Langage Java*, International Thomson Publishing France, Paris, 330 p., 1996.

[**Artificial Life, 1999**]

Artificial Life, *To Release Alife-Smart-Text-Analyzer*, PR Newswire Boston, Java Industry Connection, Sun Microsystems, Palo Alto, California, United States of America, 2 p., 1999,
<http://industry.java.sun.com/javaneWS/stories/story2/0.1072.15081.00.html> .

[**Astor, 1999**]

Astor P., *Com One présente son Minitel Internet*, Article de Presse Spécialisée, Internet, Actualités, PC Direct, ZD Net France, Septembre 1999,
<http://news.zdnet.fr/story/0,,s2058843.00.html> .

[**Augst, 2000**]

Augst B., *Unified Messaging Test Specification*, Advanced Projects Team, Enterprise Business Unit, Alcatel Business Systems, Illkirch, France, Référence 3EU_59000_0000_QTZZA_04, 40 p., 2000.

[**Balakrishnan & Honavar, 1998**]

Balakrishnan K., Honavar V., *Intelligent Diagnosis Systems*, Artificial Intelligence Research Group, Department of Computer Science, Iowa State University, Ames, United States of America, Journal of Intelligent Systems, 41 p., 1998.

[**Balkenius, 1993**]

Balkenius C., *Natural Intelligence for Autonomous Agents*, Lund University Cognitive Science, Kungshuset, Sweden, Proceedings of the International Workshop on Mechatronical Computer Systems for Perception and Action, Hamstad University, 18 p., 1993.

[**Baral & Son, 1998**]

Baral C., Son T., *Relating Theories of Actions and Reactive Control*, Department of Computer Science, University of Texas, El Paso, United States of America, Linköping Electronic Articles in Computer and Information Science, Linköping University Electronic Press, Sweden, Vol. 3, N° 9, 74 p., 1998,
<http://www.ep.liu.se/ea/cis/1998/009/> .

[**Bastide, Taouil & al., 2000**]

Bastide Y., Taouil R., Pasquier N., Stumme G., Lakhil L., *Pascal : un Algorithme d'Extraction des Motifs Fréquents*, LIMOS, Université Blaise Pascal, Aubière, INRIA Lorraine, Nancy, I3S, Université de Nice - Sophia Antipolis, France, IFAUFB, Universität Karlsruhe, Allemagne, LIM, Université de la Méditerranée, Marseille, France, TSI, 32 p., 2000.

[**Belot, 1999**]

Belot N., *Les "Minitels Internet" Prêts pour l'An 2000*, Article de Presse Spécialisée, Téléphonateurs, Produits, PC Expert, Le Magazine de l'Actualité Micro-Informatique et de l'Internet, Paris, ZD Net France, Juillet 1999,
<http://www.zdnet.fr/prod/cgi-bin/affiche.pl?ID=9949> .

[**Benaki, Karkaletsis & Spyropoulos, 1997**]

Benaki E., Karkaletsis V., Spyropoulos C., *Integrating User Modeling into Information Extraction : the UMIE Prototype*, Institute of Informatics & Telecommunications, National Centre for Scientific Research "Demokritos", Athens, Greece, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 55-57, 1997.

[**Bengio & Ducharme, 1999**]

Bengio Y., Ducharme R., *Recherche sur l'Amélioration et Applications des Algorithmes d'Apprentissage*, Bibliothèque Interactive, Laboratoire d'Informatique des Systèmes Adaptatifs (LISA), Informatique et Recherche Opérationnelle, Faculté des Arts et des Sciences, Université de Montréal, Canada, 1999,
<http://www.iro.umontreal.ca/~lisa/> .

[**Benyon, 1993**]

Benyon D., *Adaptive Systems : a Solution to Usability Problems*, Computing Department, Open University, Milton Keynes, United Kingdom, Journal of User Modelling and User-Adapted Interaction, Kluwer, Vol. 3, pp. 1-22, 1993.

[**Benyon, 1996**]

Benyon D., *Domain Models for User Interface Design*, Computing Department, Open University, Milton Keynes, United Kingdom, Critical Issues in User Interface Systems Engineering, Springer-Verlag, 16 p., 1996.

[**Benyon, 1998**]

Benyon D., *Employing Intelligence at the Interface*, Computing Department, Open University, Milton Keynes, United Kingdom, Intelligent Interface Technology, Handbook of UI chapter, 22 p., 1998, <http://www.dcs.napier.ac.uk/~dbenyon/publ.html> .

[**Benyon & Murray, 1993-1**]

Benyon D., Murray D., *Adaptive Systems : from Intelligent Tutoring to Autonomous Agents*, Computing Department, Open University, Milton Keynes, Social and Computer Sciences research group, Department of Sociology, University of Surrey, Guildford, United Kingdom, Knowledge-based Systems, 52 p., 1993.

[**Benyon & Murray, 1993-2**]

Benyon D., Murray D., *Applying User Modelling to Human-Computer Interaction Design*, Computing Department, Open University, Milton Keynes, Social and Computer Sciences research group, Department of Sociology, University of Surrey, Guildford, United Kingdom, AI Review, pp. 43-69, 1993.

[**Bernard, 1998**]

Bernard F., *La Validation du Modèle Utilisateur d'un Système Adaptatif d'Aides Vocales pour une Application Multimédia et Multimodale Destinée au Grand Public*, Division Ergonomie, Direction des Ressources Humaines, SNCF, Paris, 6^{ème} Colloque d'Ergonomie et Informatique Avancée (ERGO IA'98), Biarritz, Ecole Supérieure des Technologies Industrielles Avancées / Centre de ressource technologique Innovation-Logiciel-Système (ESTIA/ILS), Bidart, France, pp. 91-100, 1998.

[**Bessières, 1999**]

Bessières H., *France Télécom Boude le Webphone, Jugé Trop Cher*, Télécoms, 01 Réseaux, N° 70, p. 45, Paris, France, Septembre 1999.

[**Bianchi, Berthouze & Kato, 1999**]

Bianchi N., Berthouze L., Kato T., *Understanding Subjectivity : An Interactionist View*, Intelligent Systems Division, Electrotechnical Laboratory, Department of Industrial and System Engineering, Chuo University, Tokyo, Japan, User Modeling : Proceedings of the Seventh International Conference (UM'99), Banff, Canada, Springer Wien New York, 10 p., 1999.

[**Bigus & Bigus, 1998**]

Bigus Jo., Bigus Je., *Constructing Intelligent Agents with Java*, A Programmer's Guide to Smarter Applications, IBM, Wiley Computer Publishing, New York, 379 p., 1998.

[**Bits & Pixels, 1998**]

Bits & Pixels, *Neural Classification Filter*, Search Enhancement Technology, Austin, Texas, United States of America, 1998, <http://www.bitpix.com/search/ncf.htm> .

[**Blandet & Bohrer, 1998**]

Blandet T., Bohrer P., *Workshop VRML*, Session de Novembre 1998 avec Tutoriel Interactif, Laboratoire de Réalité Virtuelle, Ecole Nationale Supérieure de Physique de Strasbourg, Département "Design", Ecole Supérieure des Arts Décoratifs de Strasbourg, France, 1998, <http://www.esad-stg.org/vrml> .

[**Bonner, 1997**]

Bonner J., *The Challenge to Design Intelligent Consumer and Domestic Product Interfaces*, Institute of Design, Teesside University, Cleveland, The Reality of Intelligent Interface Technology Workshop, Napier University, Edinburgh, 3 p., 1997.

[**Bordage & Saiz, 2001**]

Bordage F., Saiz J., *La Gestion des E-Mails Professionnels*, Dossier de Presse Spécialisée, Décision Micro & Réseaux n° 474, 01 Net, Groupe Tests, Vivendi Universal Publishing, Paris, France, pp. 47-51, 2001.

[**Bradshaw, 1997**]

Bradshaw J., *Software Agents*, The Boeing Company, Fred Hutchinson Cancer Research Center, Seattle, Washington, AAAI Press, Massachusetts Institute of Technology, Cambridge, United States of America, 481 p., 1997.

[**Brazier, Jonker & Treur, 1998**]

Brazier F., Jonker C., Treur J., *Principles of Compositional Multi-Agent System Development*, Artificial Intelligence Group, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, The Netherlands, Proceedings of the IFIP'98 Conference on Information Technology and Knowledge Systems, IT&KNOWS'98, Cuena, Chapman and Hall, 14 p., 1998.

[**Brazier, Jonker & al., 1998**]

Brazier F., Jonker C., Jüngen F., Treur J., *Distributed Scheduling to Support a Call Centre : a Co-operative Multi-Agent Approach*, Artificial Intelligence Group, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, Cambridge Technology Partners, Amsterdam, The Netherlands, Applied AI Journal, 25 p., 1998.

[**Bruillard, 1997**]

Bruillard E., *Les Machines à Enseigner*, Hermès, Paris, 319 p., 1997.

[**Brusilovsky & Schwarz, 1997**]

Brusilovsky P., Schwarz E., *User as Student : towards an Adaptive Interface for Advanced Web-Based Applications*, Human-Computer Interaction Institute, Department of Psychology, Carnegie Mellon University, Pittsburgh, United States of America, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 177-188, 1997.

[**Caglayan & Harrison, 1997**]

Caglayan A., Harrison C., *Agent Sourcebook*, A Complete Guide to Desktop, Internet and Intranet Agents, Charles River Analytics, IBM, Wiley Computer Publishing, New York, 349 p., 1997.

[**Calder, 1998**]

Calder M., *What Use are Formal Design and Analysis Methods to Telecommunications Services ?*, Department of Computing Science, University of Glasgow, Scotland, Feature Interactions in Telecommunications and Software Systems, IOS Press, pp. 23-31, 1998.

[**Carlson, 1999**]

Carlson Learning Company, *Personal Profile System*, Disc and Care Profile, Product Research and Development, Minneapolis, Minnesota, United States of America, 1999,
http://www.carlsonlearning.com/products/op_pps.htm.

[**Carolan, Collins & al., 1997**]

Carolan T., Collins B., Donohue S., McEnerney H., *Intelligent Agents*, Final Year Network Technologies Group Project, Department of Computer Science, Trinity College, University of Dublin, Ireland, 13 p., 1997,
<http://ntrg.cs.tcd.ie/cs4/agents/main3.html>.

[**Carrez, 1998**]

Carrez F., *Agent Technologies and Applications : an Overview*, Diaporama, Séminaire "Agents Intelligents", Software Department, Corporate Research Center, Alcatel, Illkirch, France, 135 Diapositives, 1998.

[**CCI, 2001**]

Chambre de Commerce et d'Industrie de Strasbourg et du Bas-Rhin, *Les Entreprises Alsaciennes Très "Net"*, Article de Presse Spécialisée, Le Point Eco n° 202, Dossier Nouvelle Economie, Strasbourg, France, pp. 11-18, 2001,
http://www.strasbourg.cci.fr/point_economique/202/statik/dossier.htm.

[**CeBIT, 2001**]

welt-Centrum für Büro-, Informations- und kommunikationsTechnik, *CeBIT News*, The Official Newspaper of CeBIT Hannover, Conference Service, Deutsche Messe AG, World's No.1, Hannover, Germany, 149 p., 22.3.2001,
http://www.cebit.de/homepage_e.

[**Chabbert, 1994**]

Chabbert D., *Autocommutateur : Alcatel Joue l'Ouverture*, Technologies, L'Usine Nouvelle, Paris, France, N° 2440, pp. 48-49, 1994.

[**Chene, Tijus & Poitrenaud, 1998**]

Chene D., Tijus C., Poitrenaud S., *Une Méthode de Conception des Interfaces Basée sur la Catégorisation : Théorie, Méthode et Outils Informatiques*, Centre National d'Etude des Télécommunications, Cesson Sévigné, Laboratoire de Cognition et Activités Mentales Finalisées, Université de Saint-Denis Paris VIII, 6^{ème} Colloque d'Ergonomie et Informatique Avancée (ERGO IA'98), Biarritz, Ecole Supérieure des Technologies Industrielles Avancées / Centre de ressource technologique Innovation-Logiciel-Système (ESTIA/ILS), Bidart, France, pp. 206-215, 1998.

[**Cohu-Weill, 2001**]

Cohu-Weill V., *Internet : pour quoi faire ?*, Article de Presse, Dernières Nouvelles d'Alsace n° 63, Multimédia, Strasbourg, France, Mars 2001.

[**Cole & Stumme, 2000**]

Cole R., Stumme G., *CEM - A Conceptual Email Manager*, School of Information Technology, Griffith University, Gold Coast Campus, Australia, Fachbereich Mathematik, Technische Universität Darmstadt, Germany, 15 p., 2000.

[**COM One, 2000**]

COM One Telecom, *Bienvenue sur @max.com*, COM One Group, Cestas, France, 2000, <http://www.com1-telecom.com/fr/index.html>.

[**Couch, Simon & al., 2000**]

Couch A., Simon C., Diebolt F., Liddell D., *Smoke Test Procedure : Alcatel Version*, Business Applications Department, Enterprise Solutions Division, Alcatel, Illkirch, France, 2000.

[**Courrier, 2001**]

Courrier S., *L'Oeil Rivé sur le Web*, Article de Presse Spécialisée, Science Net, Science & Vie n° 1008, Excelsior Publications, Paris, France, p. 135, 2001, <http://www.iisvr.com/icom.html>.

[**Coutaz, 1998**]

Coutaz J., *Interfaces Homme-Machine : "le Futur ne Manque pas d'Avenir"*, Conférence Invitée, CLIPS, IMAG, Grenoble, 6^{ème} Colloque d'Ergonomie et Informatique Avancée (ERGO IA'98), Biarritz, Ecole Supérieure des Technologies Industrielles Avancées / Centre de ressource technologique Innovation-Logiciel-Système (ESTIA/ILS), Bidart, France, pp. 43-55, 1998.

[**Crampes, 1997**]

Crampes J., *Interfaces Graphiques Ergonomiques : Conception et Modélisation*, IUT de Toulouse-Blagnac, Informatique, Technosup, Ellipses, Paris, France, 185 p., 1997.

[**Créance, 2001**]

Créance V., *Le Design*, Diaporama, Forum d'Échange, Brand Department, Mobile Phone Division, Amphithéâtre Graham Bell, Alcatel, Illkirch, France, 35 Diapositives, 2001.

[**Crosnier-Mangeat, 2001**]

Crosnier-Mangeat J., *FreePlanning*, Dossier de Presse, Agence de Relations Presse Spécialisée dans les NTIC, Trait d'Union, C.M. Associés, Paris, France, 2001, <http://www.traitdunioncm.com/freeplanning/dp-freeplanning.html>.

[**Davison & Hirsh, 1998**]

Davison B., Hirsh H., *Predicting Sequences of User Actions*, Department of Computer Science, Rutgers, The State University of New Jersey, New Brunswick, AAAI-98 / ICML'98 Workshop on Predicting the Future : AI Approaches to Time-Series Analysis, Madison, AAAI Press, United States of America, 8 p., 1998.

[**De Jong, 1999**]

De Jong K., *Evolutionary Computation for Discovery*, Knowledge Discovery, Viewing Wisdom From All Perspectives, Communications of the ACM, United States of America, N° 11, Vol. 42, pp. 51-53, 1999.

[**De la Cruz & Thaler, 1996**]

De la Cruz I., Thaler L., *Inside MAPI*, Microsoft's Guide to Designing and Developing Robust Messaging and Workflow Components, Microsoft Programming Series, Microsoft Press, Redmond, Washington, United States of America, 591 p., 1996.

[**Derouet, 1999**]

Derouet T., *Le Successeur du Minitel Est Arrivé*, Article de Presse Spécialisée, Tendances, Actualités, PC Expert, Le Magazine de l'Actualité Micro-Informatique et de l'Internet, Paris, France, p. 36, Janvier 1999.

[**Derouet & Flores, 1998**]

Derouet T., Flores J., *Alcatel One Touch Com*, Article de Presse Spécialisée, Tendances, Premiers Essais, Actualités, PC Expert, Le Magazine de l'Actualité Micro-Informatique et de l'Internet, Paris, France, pp. 36/68, Juillet-Septembre 1998.

[**Desmarais, Maluf & Liu, 1995**]

Desmarais M., Maluf A., Liu J., *User-Expertise Modeling with Empirically Derived Probabilistic Implication Networks*, Centre de Recherche Informatique de Montréal, Canada, Department of Computing Studies, Hong Kong Baptist University, 37 p., 1995.

[**Dhénin & Lichter, 2000**]

Dhénin C., Lichter A., *Comment créer la proximité sur le Net*, Article de Presse Spécialisée, Informatiques Magazine n° 97 et 98, Personnalisation, Dossier Entreprise Vnunet, VNU New Média, France, 19 p., 2000, <http://www.vnunet.fr/VNU2/ent/dossiers/perso.htm>.

[**Divoux, 2001**]

Divoux P., *UML et la Modélisation Objet*, Diaporama, Séminaire "Veille Technologique", Départements d'Informatique, Institut Universitaire de Technologie Strasbourg Sud, Université Louis Pasteur, Pôle API, Illkirch, France, 2001.

[**Encarnação & Stoev, 1999**]

Encarnação L., Stoev S., *An Application-Independent Intelligent User Support System Exploiting Action-Sequence Based User Modelling*, Fraunhofer Center for Research in Computer Graphics, Providence, United States of America, Department of Computer Science, University of Tübingen, Germany, User Modeling : Proceedings of the Seventh International Conference (UM'99), Banff, Canada, Springer Wien New York, 10 p., 1999.

[**Errore, 1997**]

Errore S., *An Innovative Approach for WWW Adaptive Interfaces : a CORINTO Project*, CONSORZIO Ricerca Nazionale Tecnologia Oggetti, Bari, Italy, The Reality of Intelligent Interface Technology Workshop, Napier University, Edinburgh, 5 p., 1997.

[**Faci & Logrippo, 1994**]

Faci M., Logrippo L., *Specifying Features and Analysing Their Interactions in a LOTOS Environment*, Protocols Research Group, Department of Computer Science, University of Ottawa, Ontario, Canada, Proceedings of the 2nd International Workshop on Feature Interactions in Telecommunications Systems, Amsterdam, IOS Press, pp. 136-151, 1994.

[**Finance, 2000**]

Finance J., *L'Informatique : Boîte à Outils des Différentes Sciences et Technologies ou Partenaire Scientifique à Part Entière ? De la Richesse des Interactions de l'Informatique avec les Différents Secteurs de la Connaissance Scientifique et Technologique*, Diaporama de Séminaire, Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA), Université Henri Poincaré, Nancy, Ecole Doctorale des Sciences pour l'Ingénieur, Ecole Nationale Supérieure de Physique de Strasbourg, Pôle API, Illkirch, France, 2000.

[**Fink, Kobsa & Schreck, 1997**]

Fink J., Kobsa A., Schreck J., *Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features : User Modeling, Privacy and Security Issues*, Human-Computer Interaction Research Division (HCI), Institute for Applied Information Technology (FIT), German National Research Center for Information Technology (GMD), Sankt Augustin, Proceedings of the 4th International Conference on Intelligence in Services and Networks, Como, Italy, pp. 459-467, 1997.

[**Fleming & Cohen, 1999**]

Fleming M., Cohen R., *User Modeling in the Design of Interactive Interface Agents*, Department of Computer Science, University of Waterloo, Ontario, User Modeling : Proceedings of the Seventh International Conference (UM'99), Banff, Canada, Springer Wien New York, 10 p., 1999.

[**Flores, 1998**]

Flores J., *WebPhone Siemens : le Web à Domicile*, Un Terminal Internet Tout-En-Un, Article de Presse Spécialisée, Tendances, Actualités, PC Expert, Le Magazine de l'Actualité Micro-Informatique et de l'Internet, Paris, France, p. 33, Juillet 1998.

[**Ganascia, 1993**]

Ganascia J., *L'Intelligence Artificielle*, Dominos, Flammarion, France, 126 p., 1993.

[**Ganascia, 1996**]

Ganascia J., *Charade & Fils : Evolutions, Applications et Extensions*, Article de Synthèse, Equipe ACASA, LAFORIA-CNRS (LIP6), Université Pierre et Marie Curie, Paris-VI, France, 23 p., 1996,
http://www-rtcd.lip6.fr/~ganascia/recherche_jggfr.html .

[**Garbay & Maitre, 1997**]

Garbay C., Maitre H., *Interaction Système/Environnement pour l'Interprétation des Signaux et des Images*, Rapport de Synthèse Version V, TIMC, Grenoble, ENST, Paris, Action Inter-PRC 10.2 GDR-PRC ISIS & CHM, Opération commune en liaison avec le GT5 (fusion) et le GT2 (raisonnement), 33 p., 1997.

[**Gardarin, 1999**]

Gardarin G., *Internet/Intranet et Bases de Données*, Eyrolles, Paris, France, 264 p., 1999.

[**Garrido, Brena & Sycara, 1996**]

Garrido L., Brena R., Sycara K., *Cognitive Modeling and Group Adaptation in Intelligent Multi-Agent Meeting Scheduling*, Centro de Inteligencia Artificial, Tecnológico de Monterrey, Mexico, The Robotics Institute, Carnegie Mellon University, Pittsburgh, United States of America, Proceedings of the First Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems, LANIA & MIA-UV, pp. 55-72, 1996.

[**Gavrilova & Voinov, 1997**]

Gavrilova T., Voinov A., *An Approach to Mapping of User Model to Corresponding Interface Parameters*, Intelligent Computer Technologies Department, Computer Science Faculty of St Petersburg State Technical University, Artificial Intelligence Laboratory, Institute for High Performance Computing, St Petersburg, Russia, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 24-29, 1997.

[**Gemlas, 2000**]

Gemlas C., *E-Learning : Entreprises Apprenantes, du Mythe à la Réalité*, Article de Presse Spécialisée, Informatiques Magazine n° 115, Intra-Muros, Dossier Entreprise Vnunet, VNU New Média, France, 10 p., 2000,
<http://www.vnunet.fr/doss/ent/nnelearning.htm> .

[**Giguet, 1998**]

Giguet E., *Méthode pour l'Analyse Automatique de Structures Formelles sur Documents Multilingues*, Thèse de Doctorat, Spécialité Informatique, Groupe de Recherche En Informatique, Image, Instrumentation de Caen (GREYC), Département d'Informatique, Université de Caen, Basse-Normandie, France, 230 p., 1998.

[**Goldberg, 1994**]

Goldberg D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, University of Alabama, Addison Wesley, United States of America, 432 p., 1994.

[**Goodman, 1998**]

Goodman P., *NevProp4 : Artificial Neural Network Software for Statistical Prediction*, Departments of Internal Medicine, Electrical Engineering and Computer Science, Washoe Medical Center, University of Nevada, Reno, United States of America, 1998,
<http://www.scs.unr.edu/nevprop/> .

[**Green, Cunningham & Somers, 1998**]

Green S., Cunningham P., Somers F., *Agent Mediated Collaborative Web Page Filtering*, Department of Computer Science, Trinity College, Broadcom Éireann Research, Dublin, Ireland, Proceedings of Second International Workshop on Cooperative Information Agents, Lecture Notes in Artificial Intelligence, Klusch & Weiß, Springer Verlag, 11 p., 1998.

[**Grunst, Oppermann & Thomas, 1996**]

Grunst G., Oppermann R., Thomas C., *Adaptive and Adaptable Systems*, Human-Computer Interaction Research Division, German National Research Center for Information Technology (GMD), Sankt Augustin, Germany, Computers as Assistants – A New Generation of Support Systems, Hillsdale, Lawrence Erlbaum Associates, pp. 29-46, 1996.

[**Gurney, 1996**]

Gurney K., *An Introduction to Neural Networks*, Psychology Department, University of Sheffield, United Kingdom, UCL Press, London, United Kingdom, 10 Chapters., 100 p., 1996,
<http://www.shef.ac.uk/psychology/gurney/notes/index.html> .

[**Haddad, 1999**]

Haddad L., *L'Ère Informatique*, Article de Presse Spécialisée, Science & Vie n° 982, Encyclopédie XX^e Siècle, Excelsior Publications, Paris, France, pp. 147-159, 1999.

[**Harrington, Banks & Santos, 1996**]

Harrington R., Banks S., Santos E., *Development of an Intelligent User Interface for a Generic Expert System*, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson, Proceedings of the Midwest Artificial Intelligence and Cognitive Science Conference, Bloomington, 6 p., 1996.

[**Haton & Smaïli, 1996**]

Haton M., Smaïli K., *Séminaire Apprentissage*, Recueil de Séminaire, 1^{ère} Journée Apprentissage, Equipe RFIA, Centre de Recherche en Informatique de Nancy (CRIN), INRIA Lorraine, Université Henri Poincaré Nancy I, Institut National Polytechnique de Lorraine, Université de Nancy II, Bâtiment LORIA, Campus Scientifique, Vandœuvre lès Nancy, INRA, Champenoux, France, 74 p., 1996.

[**Höök, 1997**]

Höök K., *Steps to Take Before Intelligent User Interfaces Become Real*, Swedish Institute of Computer Science, Kista, Sweden, The Reality of Intelligent Interface Technology Workshop, Napier University, Edinburgh, 8 p., 1997.

[**Ibrahim, 1995**]

Ibrahim B., *Une Nouvelle Etape dans la Convivialité : les Logiciels Auto-Educatifs*, Département d'Informatique, Université de Genève, Suisse, La Revue de l'Association pour l'Enseignement Public et Informatique, Informatique et Technologies Modernes dans l'Enseignement et la Formation, Paris, N° 77, pp. 137-142, 1995.

[**Ibrahim, 1997**]

Ibrahim B., *Use of HTML Forms for Complex User Interfaces to Server-Side Applications*, Computer Science Department, University of Geneva, Switzerland, International Journal of Human Computer Studies, Special Issue on Innovative Applications of the World Wide Web, Academic Press, N° 46, pp. 761-771, 1997.

[**Ikonicoff, 1999**]

Ikonicoff R., *La Communication Globale : la Naissance de l'Internet*, Article de Presse Spécialisée, Science & Vie n° 987, Encyclopédie XX^e Siècle, Excelsior Publications, Paris, France, pp. 89-95, 1999.

[**Ingrand, Chatila & al., 1996**]

Ingrand F., Chatila R., Alami R., Robert F., *PRS : a High Level Supervision and Control Language for Autonomous Mobile Robots*, LAAS-CNRS, Toulouse, France, Proceedings of the 1996 IEEE International Conference on Robotics and Automation (ICRA'96), Minneapolis, United States of America, pp. 137-151, 1996.

[**ISRF, 1999**]

Internet Screenphone Reference Forum, *Working Document-Release 0.933*, External Draft, Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT), Rennes, France, 46 p., 1999, <http://www.ccett.fr/isrf/>.

[**Jardini & Schillio, 2000**]

Jardini J., Schillio F., *L'E-Recrutement*, Conférence, Cabinet de recrutement on-line futurestep.fr, Service Information-Orientation-Emploi, Université Louis Pasteur, Amphithéâtre AT9, L'Atrium, Campus de l'Esplanade, Strasbourg, France, 2000, http://www.futurestep.com/cndt16fr/sign_in/welcome.asp.

[**Joachims, Freitag & Mitchell, 1997**]

Joachims T., Freitag D., Mitchell T., *WebWatcher : a Tour Guide for the World Wide Web*, Proceedings of the 1997 International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, California, United States of America, pp. 297-312, 1997.

[**Julhiet, 1993**]

Julhiet B., *Persona*, Support Pédagogique, Alcatel Business Systems, Illkirch, France, ADK, Version 1, 113 p., 1993.

[**Katz, 1998**]

Katz L., *A la Recherche de l'Impossible Interface*, Article de Presse Spécialisée, InfoPC d'Avril 98, Chronique, Tendances, France, p. 17, 1998.

[**Ketterlin, 1995**]

Ketterlin A., *Découverte de Concepts Structurés dans les Bases de Données*, Thèse en Sciences, Spécialité Informatique, Laboratoire des Sciences de l'Image, d'Informatique et de Télé-détection, Département d'Informatique, Université Louis-Pasteur, Strasbourg, France, N° d'ordre : 2205, 205 p., 1995.

[**Ketterlin, 1997**]

Ketterlin A., *Apprentissage Symbolique Supervisé*, Document de Cours, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télé-détection, Université Louis Pasteur, Strasbourg, France, 17 p., 1997.

[**Koda & Maes, 1996**]

Koda T., Maes P., *Agents with Faces : The Effect of Personification*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, United States of America, Proceedings of 5th IEEE International Workshop on Robot and Human Communication, Tsukuba, Japan, pp. 189-194, 1996.

[**Kodratoff, 1994**]

Kodratoff Y., *Induction and the Organization of Knowledge*, Machine Learning : a Multistrategy Approach, Tecuci et Michalski Editions, Morgan-Kaufmann, San Francisco, California, United States of America, Vol. 4, pp. 85-106, 1994.

[**Kohonen, Kaski & Lappalainen, 1997**]

Kohonen T., Kaski S., Lappalainen H., *Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM*, Neural Networks Research Centre, Helsinki University of Technology, Finland, Neural Computation, Massachusetts Institute of Technology, pp. 1321-1344, 1997, <http://websom.hut.fi/websom/doc/instructions.html> .

[**Kolski, 1997**]

Kolski C., *Interfaces Homme-Machine : Application aux Systèmes Industriels Complexes*, Equipe "Informatique Industrielle et Communication Homme-Machine", Laboratoire d'Automatique et de Mécanique Industrielles et Humaines, Université de Valenciennes et du Hainaut-Cambrésis, Collection Informatique, 2^e édition revue et augmentée, Hermès, Paris, 477 p., 1997.

[**Lachiche, 2001**]

Lachiche N., *Apprentissage Automatique : Découverte de Connaissances dans les Bases de Données*, Diaporama et Support de Cours, CNAM, Strasbourg, France, 46 Diapositives, 2001.

[**Lambert, Riera & al., 1998**]

Lambert M., Riera B., Martel G., Evsukoff A., Gentil S., *Conception et Evaluation d'Interfaces d'Aide à la Décision pour la Supervision*, LAMIH, Université de Valenciennes et du Hainaut-Cambrésis, Laboratoire d'Automatique de Grenoble, ENSIEG, 6^{ème} Colloque d'Ergonomie et Informatique Avancée (ERGO IA'98), Biarritz, Ecole Supérieure des Technologies Industrielles Avancées / Centre de ressource technologique Innovation-Logiciel-Système (ESTIA/ILS), Bidart, France, pp. 130-140, 1998.

[**Lashkari, Metral & Maes, 1994**]

Lashkari Y., Metral M., Maes P., *Collaborative Interface Agents*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, United States of America, Proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle, Vol. 1, 6 p., 1994.

[**Lauzon, 1999**]

Lauzon V., *Modèles Statistiques comme Algorithmes d'Apprentissage et MMCCs - Prédiction de Séries Financières*, Mémoire de Maîtrise Informatique, Département d'Informatique et de Recherche Opérationnelle, Faculté d'Arts et Sciences, Université de Montréal, Canada, 1999.

[**Leclercq, 2001**]

Leclercq S., *Les Terminaux Nomades : un Marché Chahuté*, Dossier de Presse Spécialisée, Computer Reseller News n° 77, Technologies (e-solutions), VNU Publications France, Suresnes, pp. 33-44, 2001.

[**Leduc & Léonard, 1993**]

Leduc G., Léonard L., *Comment Rendre LOTOS Apte à Spécifier des Systèmes Temps Réel ?*, Université de Liège, Institut d'Electricité Montefiore, Liège, Belgique, CFIP'93 Ingénierie des Protocoles, Hermès, Paris, pp. 407-425, 1993.

[**Lennard & Parkes, 1997**]

Lennard A., Parkes A., *Facilitating a Fully Adaptive, Intelligent Interface*, Computing Department, Lancaster University, The Reality of Intelligent Interface Technology Workshop, Napier University, Edinburgh, 8 p., 1997.

[**Lichtner, 2000**]

Lichtner A., *Trois Applications de Messagerie Unifiée*, Article de Presse Spécialisée, Informatiques Magazine n° 107, Télécommunications, Dossier Entreprise Vnunet, VNU New Média, France, 6 p., 2000, <http://www.vnunet.fr/VNU2/ent/dossiers/messagerie.htm> .

[**Lieberman, 1995**]

Lieberman H., *Letizia : an Agent that Assists Web Browsing*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, United States of America, Proceedings of the 1995 International Joint Conference on Artificial Intelligence, Montreal, Canada, 6 p., 1995.

[**Lieberman, 1997**]

Lieberman H., *Autonomous Interface Agents*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, ACM Conference on Human-Computer Interface (CHI'97), Atlanta, United States of America, 8 p., 1997.

[**Lieberman, 1998**]

Lieberman H., *Integrating User Interface Agents with Conventional Applications*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, Proceedings of the ACM Conference on Intelligent User Interfaces, San Francisco, United States of America, Knowledge-Based Systems Journal, 8 p., 1998.

[**Lippold & Pomian, 1995**]

Lippold B., Pomian J., *AVIS : une Méthode d'Analyse de la Cohérence des Documents Techniques*, Nemesia, Paris, Proceedings of the Fifteenth International Conference IA'95, Language Engineering 95, Montpellier, pp. 301-312, 1995, <http://www.nemesia.com/publications/GLING95.html> .

[**Lorenzo & Otero, 2000**]

Lorenzo D., Otero R., *Learning to Reason about Actions*, Computer Science Departments, University of Corunna, Spain, University of Texas, El Paso, United States of America, Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), Humboldt University, Berlin, Werner Horn, IOS Press, pp. 316-320, 2000.

[**Luckin & Du Boulay, 1999**]

Luckin R., Du Boulay B., *Capability, Potential and Collaborative Assistance*, School of Cognitive & Computing Sciences, University of Sussex, Brighton, United Kingdom, User Modeling : Proceedings of the Seventh International Conference (UM'99), Banff, Canada, Springer Wien New York, 10 p., 1999.

[**MacIntyre & Feiner, 1996**]

MacIntyre B., Feiner S., *Future Multimedia User Interfaces*, Columbia University, Multimedia Systems, Springer-Verlag, 4(5), pp. 250-268, 1996.

[**Maes, 1994**]

Maes P., *Modeling Adaptive Autonomous Agents*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, United States of America, Artificial Life Journal, Langton, Vol. 1, N° 1&2, 37 p., 1994.

[**Maglio & Barrett, 1997**]

Maglio P., Barrett R., *How to Build Modeling Agents to Support Web Searchers*, IBM Almaden Research Center, San Jose, United States of America, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 5-16, 1997.

[**Marshall & Greenwood, 1997**]

Marshall P., Greenwood S., *Multi-Agents to Provide Adaptivity in an Intelligent Interface to a Medical Database*, Intelligent Systems Research Group, School of Computing & Mathematical Sciences, Oxford Brookes University, United Kingdom, Proceedings of the Seventh International Conference on Human-Computer Interaction, Design of Computing Systems : Cognitive Considerations, San Francisco, United States of America, Elsevier, Amsterdam, Netherlands, Vol. 1, pp. 769-772, 1997.

[**Mataric, 1994**]

Mataric M., *Interaction and Intelligent Behavior*, Philosophy Doctorate Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 189 p., 1994.

[**Maulsby & Witten, 1997**]

Maulsby D., Witten I., *Learning Agents : from User Study to Implementation*, Medicine Wheel Network Design, Calgary, Canada, Department of Computer Science, University of Waikato, Hamilton, New Zealand, IEEE Computer, N° 30, Vol. 11, pp. 36-44, 1997.

[**Mellor, Tockey & al., 1998**]

Mellor S., Tockey S., Arthaud R., Leblanc P., *Software-platform-independent, Precise Action Specifications for UML*, Shlaer-Mellor Method - BridgePoint, Project Technology, Rockwell Collins, Verilog, UML'98 : Beyond the Notation – International Workshop, ESSAIM, Mulhouse, France, 11 p., 1998, <http://www.projtech.com/pubs/uml98.html> .

[**Michalski, Bratko & Kubat, 1998**]

Michalski R., Bratko I., Kubat M., *Machine Learning and Data Mining : Methods and Applications*, Artificial Intelligence / Database, George Mason University, Fairfax, United States of America, University of Ljubljana, Slovenia, University of South Western Louisiana, Lafayette, United States of America, John Wiley & Sons, Chichester, England, 456 p., 1998.

[**Michel, 2000**]

Michel C., *Analyse et Modélisation des Séquences Biologiques*, Diaporama, Séminaire de "Bio-Informatique", Département d'Informatique, Université de Technologie de Belfort-Montbéliard (UTBM), Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT), Ecole Nationale Supérieure de Physique de Strasbourg, Pôle API, Illkirch, France, 2000.

[**Mitchell, 1999**]

Mitchell T., *Machine Learning and Data Mining*, Center for Automated Learning and Discovery, School of Computer Science, Carnegie-Mellon University, Pittsburgh, General Domains, Knowledge Discovery, Viewing Wisdom From All Perspectives, Communications of the ACM, United States of America, N° 11, Vol. 42, pp. 31-36, 1999.

[**Mohseni & Stewart, 1997**]

Mohseni P., Stewart T., *JavaBeans Developer's Guide*, M&T Books, Henry Holt and Company, New York, United States of America, 311 p., 1997.

[**Motoda, Washio & al., 1997**]

Motoda H., Washio T., Kayama T., Yoshida K., *Extracting Behavioral Patterns from Relational History Data*, The Institute of Scientific and Industrial Research, Osaka University, Advanced Research Laboratory, Hitachi, Japan, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, 11 p., 1997.

[**Moukas, 1997**]

Moukas A., *User Modeling in a Multi-Agent Evolving System*, Software Agents Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, United States of America, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, 11 p., 1997.

[**Murphy & McTear, 1997**]

Murphy M., McTear M., *Learner Modelling for Intelligent CALL*, School of Information and Software Engineering, University of Ulster, North Ireland, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 301-312, 1997.

[**Murray, Schell & Willis, 1997**]

Murray J., Schell D., Willis C., *User Centered Design in Action : Developing an Intelligent Agent Application*, IBM Corporation, Research Triangle Park, 15th Annual International Conference on Computer Documentation Conference Proceedings (SIGDOC'97), Crossroads in Communication, Salt Lake City, ACM, New York, United States of America, pp. 181-188, 1997.

[**Nachi, 1995**]

Nachi M., *Interfaces et Simulations dans les Systèmes d'E.A.O. : Modélisation et Filtrage d'Interfaces*, Thèse en Sciences, Automatique Théorique et Systèmes, UFR Scientifique d'Orsay, Université de Paris XI (Paris-Sud), N° 95PA112312, 1995.

[**Nangle, Cunningham & Evans, 1998**]

Nangle B., Cunningham P., Evans R., *PALS : Personal Assistants which Learn for Intelligent Call Screening*, Computer Science Department, Trinity College, Broadcom Eireann Research, Kestrel House, Clanwilliam Place, Dublin, Ireland, 12 p., 1998.

[**Nemec-Poncik, 2001-1**]

Nemec-Poncik M., *Un Outil de Messagerie Unifiée*, Article de Presse Spécialisée, Computer Reseller News n° 75, Technologies (e-solutions), VNU Publications France, Suresnes, p. 46, 2001.

[**Nemec-Poncik, 2001-2**]

Nemec-Poncik M., *Un PDA pour Vendre des Applications Métier*, Article de Presse Spécialisée, Computer Reseller News n° 78, Périphériques, Mieux Vendre, Technologies, VNU Publications France, Suresnes, pp. 44-45, 2001.

[**Net Perceptions, 2001**]

Net Perceptions, *Retail Revelations - Intelligent Demand Generation*, Edina, Minnesota, United States of America, 2001,
<http://www.NetPerception.com/>.

[**Neuman, 2000**]

Neuman F., *Mon Annuaire Web à Moi*, Article de Presse Spécialisée, Actualités, Science & Vie Micro, Vnunet, France, 2000,
<http://www.vnunet.fr/svm/actu/article.htm?date=2000-02-17&numero=4147>.

[**Nevill-Manning, 1996**]

Nevill-Manning C., *Inferring Sequential Structure*, Philosophy Doctorate Thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 212 p., 1996.

[**Nevill-Manning & Witten, 1997**]

Nevill-Manning C., Witten I., *Identifying Hierarchical Structure in Sequences : a Linear-Time Algorithm*, Department of Computer Science, University of Waikato, Hamilton, New Zealand, Journal of Artificial Intelligence Research, Vol. 7, pp. 67-82, 1997,
<http://dna.stanford.edu/sequitur/>.

[**Nevill-Manning & Witten, 1998**]

Nevill-Manning C., Witten I., *Inferring Lexical and Grammatical Structure from Sequences*, Biochemistry Department, Stanford University, California, Department of Computer Science, University of Waikato, Hamilton, New Zealand, Proceedings of Compression and Complexity of Sequences, Carpentieri & al., Positano, Italy, IEEE Computer Society Press, Los Alamitos, California, pp. 265-274, 1998.

[**Nicolas, 1993**]

Nicolas J., *École sur l'Apprentissage Automatique*, Support de Cours, IRISA/INRIA-Rennes, PRC-GDR Intelligence Artificielle, Saint-Raphaël, France, 174 p., 1993.

[**Oppermann & Thomas, 1996**]

Oppermann R., Thomas C., *Supporting Learning as an Iterative Process in a Social Context*, Human-Computer Interaction Research Division, German National Research Center for Information Technology (GMD), Sankt Augustin, Germany, Proceedings of the European Conference on Artificial Intelligence in Education, Lisboa, pp. 150-155, 1996.

[**Palanque & Bastide, 1994**]

Palanque P., Bastide R., *Petri Net Based Design of User-Driven Interfaces Using the Interactive Cooperative Objects Formalism*, LIS, Université Toulouse I, France, Eurographics Workshop on "Design, Specification and Verification of Interactive Systems", Carara, Italy, 15 p., 1994.

[**Palanque & Bastide, 1995**]

Palanque P., Bastide R., *Spécifications Formelles pour l'Ingénierie des Interfaces Homme-Machine*, LIS, Université Toulouse I, France, Revue des Techniques et Sciences Informatiques, Vol. 14, pp. 473-500, 1995.

[**Paliouras, Karkaletsis & al., 1999**]

Paliouras G., Karkaletsis V., Papatheodorou C., Spyropoulos C., *Exploiting Learning Techniques for the Acquisition of User Stereotypes and Communities*, Institute of Informatics and Telecommunications, Division of Applied Technologies, National Centre for Scientific Research, Greece, User Modeling : Proceedings of the Seventh International Conference (UM'99), Banff, Canada, Springer Wien New York, 10 p., 1999.

[**Parekh, 1998**]

Parekh R., *Constructive Learning : Inducing Grammars and Neural Networks*, Doctoral Dissertation for the degree of Doctor of Philosophy, Department of Computer Science, Iowa State University, Ames, 247 p., 1998.

[**Perret du Cray, 2000**]

Perret du Cray B., *La Machine à Penser*, Article de Presse Spécialisée, Science & Vie Micro n° 185, Dossier Cerveau, La Connexion Informatique, Vnunet, France, 11 p., 2000, <http://www.vnunet.fr/VNU2/svm/dossiers/machinepenser.htm> .

[**Petrelli, De Angeli & Convertino, 1999**]

Petrelli D., De Angeli A., Convertino G., *A User-Centered Approach to User Modeling*, Cognitive and Communication Technology Division, ITC-IRST, Department of Psychology, University of Trieste, Italy, User Modeling : Proceedings of the Seventh International Conference (UM'99), Banff, Canada, Springer Wien New York, 10 p., 1999.

[**Pichon & Julienne, 1999**]

Pichon A., Julienne M., *Cerveau : les Clés de l'Intelligence*, Article de Presse Spécialisée, Eurêka n° 41, Bayard Presse, Paris, France, pp. 8-11, 1999.

[**Plante, Dumas & Plante, 2000**]

Plante P., Dumas L., Plante A., *Nomino*, Département de Linguistique, Faculté des Sciences Humaines, Université du Québec, Montréal, Nomino Technologies, Canada, 2000, <http://www.ling.uqam.ca/nomino/> .

[**Pohl, 1996**]

Pohl W., *Learning About the User – User Modeling and Machine Learning*, Department of Mathematics and Computer Science, Essen Universität, Germany, ICML'96 Workshop Machine Learning meets Human-Computer Interaction, Bari, pp. 29-40, 1996.

[**Pohl, 1997**]

Pohl W., *LaboUr – Machine Learning for User Modeling*, Human-Computer Interaction Research Department, German National Research Center for Information Technology (GMD), Sankt Augustin, Germany, Proceedings of the Seventh International Conference on Human-Computer Interaction (HCI International'97), Design of Computing Systems : Cognitive Considerations, San Francisco, United States of America, Elsevier Science Publishers, Amsterdam, Netherlands, Vol. 2, pp. 27-30, 1997.

[**Porto, Fogel & Fogel, 1995**]

Porto V., Fogel D., Fogel L., *Alternative Neural Network Training Methods*, Evolutionary Programming, Orincon Corporation, Natural Selection, IEEE Expert, United States of America, pp. 16-21, 1995.

[**Puerta, 1993**]

Puerta A., *The Study of Models of Intelligent Interfaces*, Medical Computer Science Group, Knowledge Systems Laboratory, Stanford University, Proceedings of the 1993 International Workshop on Intelligent User Interfaces, Orlando, Florida, pp. 71-78, 1993.

[**Puerta, Eriksson & al., 1993**]

Puerta A., Eriksson H., Gennari J., Musen M., *Beyond Data Models for Automated User Interface Generation*, Medical Computer Science Group, Knowledge Systems Laboratory, Departments of Medicine and Computer Science, Stanford University, Proceedings of the HCI'94, People and Computers, Glasgow University, 14 p., 1993.

[**Pynadath, Tambe & al., 2000**]

Pynadath D., Tambe M., Arens Y., Chalupsky H., Gil Y., Knoblock C., Lee H., Lerman K., Oh J., Ramachandran S., Rosenbloom P., Russ T., *Electric Elves : Immersing an Agent Organization in a Human Organization*, Information Sciences Institute and Computer Science Department, University of Southern California, Marina del Rey, Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents, United States of America, 5 p., 2000.

[**Quaresma & Lopes, 1997**]

Quaresma P., Lopes J., *Modeling Agents in Dialogue Systems*, Departamento de Informatica, Universidade Nova de Lisboa, Portugal, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 101-103, 1997.

[**Ramscar, Pain & Lee, 1997**]

Ramscar M., Pain H., Lee J., *Do We Know What the User Knows, and Does It Matter ? The Epistemics of User Modelling*, Department of Artificial Intelligence, Human Communication Research Centre, Department of Architecture, University of Edinburgh, Scotland, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 429-431, 1997.

[**Rational, 1997**]

Rational Software Corporation, *Unified Modeling Language Notation Guide*, Document interactif, UML Resource Center, Documentation Resources, Rational Rose, Version 1.1, 1997, <http://www.rational.com/uml/resources/documentation/notation/index.jtmpl> .

[**Rhee, Kim & al., 1997**]

Rhee P., Kim Y., Sim B., Zhoo Z., Park D., *Intelligent User Interface for Intelligent Multimedia Repository*, Computer Science and Engineering Department, Inha University, Inchon, South Korea, Proceedings of the SPIE, Multimedia Storage and Archiving Systems, The International Society for Optical Engineering, Dallas, United States of America, Vol. 3229, pp. 113-123, 1997.

[**Rogerson, 1997**]

Rogerson D., *Inside COM*, Microsoft's Component Object Model, Microsoft Programming Series, Microsoft Press, Redmond, Washington, United States of America, 376 p., 1997.

[**Russell & Norvig, 1995**]

Russell S., Norvig P., *Artificial Intelligence, A Modern Approach*, Prentice-Hall International Editions, Simon & Schuster Company, Englewood Cliffs, New Jersey, United States of America, 932 p., 1995.

[**Ruvini, 2000**]

Ruvini J., *Assistance à l'Utilisation d'un Environnement Interactif: Apprentissage des Habitudes de l'Utilisateur*, Thèse de Doctorat en Informatique, Information, Structures et Systèmes, UFR Sciences de Montpellier, Sciences et Techniques du Languedoc, Université de Montpellier II, 263 p., 2000.

[**Saloña, Vives & Gómez, 1993**]

Saloña A., Vives J., Gómez S., *An Introduction to LOTOS*, LOTOS Tutorial, Departamento Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, DIT-UPM, 112 p., 1993.

[**Sandel, 1997**]

Sandel O., *Modèle Neuronal d'Extraction des Routes dans une Image de Télédétection*, Rapport de Stage, D.E.A. Informatique, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur, Strasbourg, France, 65 p., 1997.

[**Sandel, Korczak & al., 1998**]

Sandel O., Korczak J., Chrétien J., Geyer C., *Le Concept "d'Assistant Conversationnel"*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Alcatel, Illkirch, France, Référence 3EU_59000_0070_PAZZB_01, 11 p., 16/10/1998.

[**Sandel, Korczak & al., 1999-1**]

Sandel O., Korczak J., Chrétien J., Geyer C., *La Première Définition Qualitative, Hiérarchique et Taxinomique des Interfaces Intelligentes*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Alcatel, Illkirch, France, Référence 3EU_59000_0071_PAZZB_01, 28 p., 12/02/1999.

[**Sandel, Korczak & al., 1999-2**]

Sandel O., Korczak J., Chrétien J., Geyer C., *La Première Définition du Concept "d'Assistant Individuel de Télécommunication"*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Alcatel, Illkirch, France, Référence 3EU_59000_0072_PAZZB_01, 10 p., 18/02/1999.

[**Sandel, Korczak & al., 1999-3**]

Sandel O., Korczak J., Chrétien J., Geyer C., *La Mise en Valeur de l'Auto-Adaptation dans l'Assistant Individuel de Télécommunication*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Alcatel, Illkirch, France, Référence 3EU_59000_0073_PAZZB_01, 27 p., 17/03/1999.

[Sandel, Korczak & al., 1999-4]

Sandel O., Korczak J., Chrétien J., Geyer C., *La Première Définition Qualitative et Comportementale d'un Modèle Auto-Adaptable de l'Utilisateur*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Alcatel, Illkirch, France, Référence 3EU_59000_0074_PAZZB_01, 20 p., 12/04/1999.

[Sandel, Korczak & al., 1999-5]

Sandel O., Korczak J., Chrétien J., Geyer C., *La Spécification Formelle des Actions des Scénarios Auto-Adaptables de l'Assistant Individuel de Télécommunication*, Diaporama de Séminaire, Groupe de Recherche en Intelligence Artificielle, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Alcatel, Illkirch, France, Référence 3EU_59000_0075_PAZZB_01, 44 Diapositives, 21/07/1999.

[Sandel, Korczak & al., 2000]

Sandel O., Korczak J., Chrétien J., Geyer C., Dos-Santos D., *Fonctionnalité "e-mail" d'ALCATEL : Représentation et Prédiction des Actions d'un Premier Modèle d'Application*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Equipe de Recherche Technologique en Informatique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Private Networks Department, Corporate Research Center, Advanced Projects Team, Business Applications Department, Enterprise Solutions Division, Alcatel, Illkirch, France, Référence 3EU_59000_0078_PAZZB_01, Edition 2.0, 46 p., 15/02/2000.

[Sandel, Korczak & al., 2001]

Sandel O., Korczak J., Dos-Santos D., Marchand J., *Modèle d'Interface Intelligente pour Terminaux de Communication : Validation et Tests Appliqués à "l'ALCATEL Unified Messaging"*, Rapport de Recherche, Groupe de Recherche en Intelligence Artificielle, Equipe de Recherche Technologique en Informatique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Université Louis Pasteur - Strasbourg I, Advanced Projects Team, Business Applications Department, Enterprise Solutions Division, Alcatel, Illkirch, France, Référence 3EU_59000_0004_PEZZB_01, 44 p., 11/05/2001.

[Savidis, Akoumianakis & Stephanidis, 1997]

Savidis A., Akoumianakis D., Stephanidis C., *Software Architecture for Transformable Interface Implementations : Building User-Adapted Interactions*, Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), Science and Technology Park of Crete, Heraklion, Crete, Greece, Proceedings of the Seventh International Conference on Human-Computer Interaction (HCI International'97), Design of Computing Systems : Cognitive Considerations, San Francisco, United States of America, Elsevier Science Publishers, Amsterdam, Netherlands, Vol. 1, pp. 453-456, 1997.

[Scerri, Pynadath & Tambe, 2001]

Scerri P., Pynadath D., Tambe M., *Adjustable Autonomy in Real-World Multi-Agent Environments*, Information Sciences Institute and Computer Science Department, University of Southern California, Marina del Rey, Proceedings of the International Conference on Autonomous Agents (Agents'01), United States of America, 8 p., 2001.

[Schlimmer & Hermens, 1993]

Schlimmer J., Hermens L., *Software Agents : Completing Patterns and Constructing User Interfaces*, School of Electrical Engineering & Computer Science, Washington State University, Pullman, United States of America, Journal of Artificial Intelligence Research, Vol. 1, pp. 61-89, 1993.

[Schlimmer & Wells, 1996]

Schlimmer J., Wells P., *Quantitative Results Comparing Three Intelligent Interfaces for Information Capture : A Case Study Adding Name Information into an Electronic Personal Organizer*, School of Electrical Engineering & Computer Science, Washington State University, Pullman, AllPen Software, Los Gatos, United States of America, Journal of Artificial Intelligence Research, Vol. 5, pp. 329-349, 1996.

[Sebban & Venturini, 1999]

Sebban M., Venturini G., *Apprentissage Automatique*, Hermes Science Publications, Paris, France, 174 p., 1999.

[Segal & Kephart, 1999]

Segal R., Kephart J., *SwiftFile*, Thomas J. Watson Research Center, IBM, United States of America, 1999, <http://alphaworks.ibm.com/aw.nsf/techmain/swiftfile> .

[**Sentient Machine Research, 1999**]

Sentient Machine Research, *DataDetective*, Technical Specifications 2.0, B3I, CIME Group, Vauhallan, France, Amsterdam, The Netherlands, 11 p., 1999,
<http://www.smr.nl> .

[**Sheth, 1994**]

Sheth B., *A Learning Approach to Personalized Information Filtering*, Master of Science in Computer Science and Engineering, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 76 p., 1994.

[**Shneiderman, 1997**]

Shneiderman B., *Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces*, Human-Computer Interaction Laboratory, Department of Computer Science, Institute for Systems Research, University of Maryland, College Park, Proceedings of the ACM International Workshop on Intelligent User Interfaces'97, New York, pp. 33-39, 1997.

[**Sorin & Colaïtis, 1999**]

Sorin C., Colaïtis F., *Interactions Homme-Machine Avancées : Nouvelles Interfaces, Nouveaux Usages*, Diaporama, Direction des Interactions Humaines, Branche Développement, France Télécom / CNET, Colloque sur la Prospective en Télécommunications, Nouveaux Usages, Réseau National de Recherche en Télécommunications, Paris, France, 17 Diapositives, 1999.

[**Sors, 2000**]

Sors A., *Une Souris Psychologue*, Article de Presse Spécialisée, Science Net, Science & Vie n° 991, Excelsior Publications, Paris, France, p. 154, 2000.

[**Stefanuk, 1997**]

Stefanuk V., *Embedding User Models in Intelligent Interfaces*, Institute for Information Transmission Problems, Russian Academy of Science, Moscow, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 18-23, 1997.

[**Stémart, 1997**]

Stémart I., *Contribution à la Définition et à la Mise en Oeuvre d'un Dialogue Adaptatif pour la C.A.O.*, Thèse en Informatique, Automatique Théorique et Systèmes, Université de Metz, N° 97METZ0001, 1997.

[**Stephanidis, 1997**]

Stephanidis C., *User Interfaces for All : Developing Interfaces for Diverse User Groups*, Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), Heraklion, Crete, Greece, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 443-444, 1997.

[**Stock, Strapparava & Zancanaro, 1997**]

Stock O., Strapparava C., Zancanaro M., *User Modelling in a Multimodal Dialogue System for Information Access*, Istituto per la Ricerca Scientifica e Tecnologica, Povo/Trento, Italy, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 42-47, 1997.

[**Sun, 1998**]

Sun Microsystems, *How Java Technology Works in Screen Phones*, Contents of Java Technologies, Java Platform, Palo Alto, California, United States of America, 1998,
<http://www.sun.com/java/javaworks/screenphone.jhtml> .

[**Teboul, 1998**]

Teboul L., *SNiFF+ Utilisateur*, Diaporama et Support de Cours, TakeFive Software, Vélizy, France, Version Java, 50 Diapositives, 32 p., 1998.

[**Thees & Gotzhein, 1998**]

Thees J., Gotzhein R., *The « eXperimental Estelle Compiler » - Automatic Generation of Implementations from Formal Specifications*, University of Kaiserslautern, Germany, Proceedings and Slides of the 2nd Workshop on Formal Methods in Software Practice (FMSP'98), Ardis, Clearwater Beach, Florida, United States of America, 18 p., 1998.

[**Thorel, 1999**]

Thorel J., *Le Webphone de Matra en vente à la Fnac*, Article de Presse Spécialisée, Business, Actualités, PC Direct, ZD Net France, Décembre 1999,
<http://news.zdnet.fr/story/0..s2059332.00.html> .

[**Turenne, 2000**]

Turenne N., *Apprentissage Statistique pour l'Extraction de Concepts à partir de Textes. Application au Filtrage d'Informations Textuelles*, Thèse en Sciences, Spécialité Informatique, Laboratoire d'Informatique et d'Intelligence Artificielle (LIIA), Ecole Nationale Supérieure des Arts et Industries de Strasbourg (ENSAIS), UFR Mathématiques-Informatique, Université Louis-Pasteur de Strasbourg, France, N° d'ordre: 3632, 265 p., 2000.

[**Umap, 2000**]

Umap Universal, *Le Premier Logiciel de Visualisation de l'Information*, Version 2.0, TriVium, Paris, France, 2000,

<http://www.umap.com/> .

[**UMEC, 2000**]

Universal Microelectronics Company, *UMEC Internet Screenphone ISP2001*, Taichung, Taiwan, 2000,

<http://www.umec-web.com/lsp2001/lsp2001.htm> .

[**Valtech, 1999**]

Valtech, Halmaert F. (Formateur), *Développer avec Java*, Diaporama et Support de Cours, Département Recherche, Alcatel Business Systems, Illkirch, France, Version 1.3, 358 p., 1999,

<http://www.valtech.com> .

[**Varma, Wood & Agogino, 1996**]

Varma A., Wood W., Agogino A., *A Machine Learning Approach to Automated Design Classification, Association and Retrieval*, Department of Mechanical Engineering, California University, Berkeley, Artificial Intelligence in Design'96, Stanford, United States of America, Kluwer Academic Publishers, Dordrecht, Netherlands, pp. 429-445, 1996.

[**Vassileva, 1997**]

Vassileva J., *A New View of Interactive Human-Computer Environments*, Department of Technical Computer Science, Federal Armed Forces University, Munich, Germany, User Modeling : Proceedings of the Sixth International Conference (UM'97), Springer Wien New York, pp. 433-435, 1997.

[**Verroust, 1997**]

Verroust G., *Histoire, Epistémologie de l'Informatique et Révolution Technologique*, Résumés de Cours, Maîtrise Sciences & Techniques Hypermédia, 2^{ème} Année, Département Hypermédiat UFR-6, Université Paris VIII, Saint-Denis, France, 1997,

<http://www.ac-versailles.fr/epi/articles/qverrou.htm> .

[**Vladyslav, 2001-1**]

Vladyslav F., *L'eVilla de Sony, un Terminal Internet Audacieux*, Article de Presse Spécialisée, Actualités, Terminal Internet, Science & Vie Micro, Vnunet, France, 2001,

<http://www.vnunet.fr/svm/actu/article.htm?numero=7823> .

[**Vladyslav, 2001-2**]

Vladyslav F., *Microsoft dévoile sa "Secrétaire Intuitive"*, Article de Presse Spécialisée, Recherche & Développement, Science & Vie Micro, Vnunet, France, 2001,

<http://www.vnunet.fr/svm/actu/article.htm?numero=7185> .

[**Weinfeld, 2000**]

Weinfeld M., *Introduction aux Réseaux de Neurones Formels*, Diaporama de Séminaire, Laboratoire d'Informatique de l'Ecole Polytechnique (LIX), Palaiseau, Ecole Doctorale des Sciences pour l'Ingénieur, Ecole Nationale Supérieure de Physique de Strasbourg, Pôle API, Illkirch, France, 2000.

[**Wemmert, 2000**]

Wemmert C., *Classification Hybride Distribuée par Collaboration de Méthodes Non Supervisées*, Thèse en Sciences, Spécialité Informatique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Département d'Informatique, Université Louis-Pasteur de Strasbourg, France, N° d'ordre : 3683, 176 p., 2000.

[**Whitehead, 1998**]

Whitehead E., *Advanced Human-Computer Interaction*, Teaching Material in Computer Information Systems Design (5 Chapters), School of Information Systems, Faculty of Technology, Kingston University, 63 p., 1998,

<http://infosys.king.ac.uk/ISSchool/BID/bid4/ahci/AHCIHome.html> .

[**Wolff, 1999**]

Wolff F., *Analyse Contextuelle des Gestes de Désignation en Dialogue Homme-Machine*, Thèse, Département de Formation Doctorale en Informatique, Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA), Université Henri Poincaré - Nancy 1, 204 p., 1999.

[**Yang, Parekh & Honavar, 1998**]

Yang J., Parekh R., Honavar V., *DistAI : an Inter-Pattern Distance-Based Constructive Learning Algorithm*, Artificial Intelligence Research Group, Department of Computer Science, Iowa State University, Ames, United States of America, Proceedings of the International Joint Conference on Neural Networks, Anchorage, Alaska, 41 p., 1998.

[**Yang, Renshou & Fanlun, 1997**]

Yang Z., Renshou T., Fanlun X., *A Proposed Framework for Intelligent Systems Based on Multi-Agent Conception*, Department of Automation, University of Sciences and Technologies of China, The Institute of Intelligent Machines, Academia Sinica, Hefei, Anhui, 1997 IEEE International Conference on Intelligent Processing Systems, Beijing, China, IEEE Society, New York, United States of America, Vol. 1, pp. 923-927, 1997.

[**Ye, 1997**]

Ye N., *Neural Networks Approach to User Modeling and Intelligent Interface : a Review and Reappraisal*, Department of Mechanics Engineering, Illinois University, Chicago, International Journal of Human-Computer Interaction, Ablex Publishing, United States of America, Vol. 9, pp. 3-23, 1997.

[**Yunfei, Renshou & al., 1997**]

Yunfei X., Renshou T., Haiying L., Fanlun X., *A Developing Platform for Intelligent System Based on Multi-Agent*, Department of Automation, University of Sciences and Technologies of China, The Institute of Intelligent Machines, Academia Sinica, Hefei, Anhui, 1997 IEEE International Conference on Intelligent Processing Systems, Beijing, China, IEEE Society, New York, United States of America, Vol. 1, pp. 915-919, 1997.

Liste des figures et copies d'écrans

FIGURE 1.1.1.2 :	<i>PHOTOGRAPHIES DES DERNIERS TERMINAUX DE COMMUNICATION DE SONY ET INTEL, PLEINEMENT DEDIES AU RESEAU INTERNET</i> -----	10
FIGURE 1.1.1.3 :	<i>DERNIERS SERVEURS DE COMMUNICATION ET STANDARDS TELEPHONQUES NUMERISES "ALCATEL 4400"</i> -----	11
FIGURE 1.1.2.2 :	<i>PHOTOGRAPHIE DU TERMINAL INTERNET "ICOM" DE LA SOCIETE INTERACTIVE IMAGING SYSTEMS (IIS)</i> -----	14
FIGURE 1.2.3.1 :	<i>ARBRE PRINCIPAL DÉTAILLANT LA RACINE ET LE PREMIER NIVEAU DE NOTRE NOUVELLE DÉFINITION TAXINOMIQUE DÉDIÉE AUX INTERFACES INTELLIGENTES</i> -----	28
FIGURE 1.2.3.2 :	<i>PREMIER ARBRE DU SECOND NIVEAU DE NOTRE DÉFINITION TAXINOMIQUE DÉDIÉE AUX INTERFACES INTELLIGENTES ET EXPOSANT LEURS DIFFÉRENTES MODÉLISATIONS ET CONCEPTIONS REMARQUABLES</i> -----	28
FIGURE 1.2.3.3 :	<i>ARBRE DU SECOND NIVEAU DE NOTRE DÉFINITION TAXINOMIQUE DES INTERFACES INTELLIGENTES, DÉTAILLANT LES AGENTS INTELLIGENTS ET SYSTÈMES MULTI-AGENTS</i> -----	28
FIGURE 1.2.3.4 :	<i>TROISIÈME ARBRE DU SECOND NIVEAU DE NOTRE DÉFINITION TAXINOMIQUE ET SPÉCIFIANT LES MODÈLES D'UTILISATEURS ET COMPORTEMENTS AUTO-ADAPTATIFS</i> -----	29
FIGURE 1.2.3.5 :	<i>QUATRIÈME ARBRE DU SECOND NIVEAU DE NOTRE DÉFINITION TAXINOMIQUE DÉDIÉE AUX INTERFACES INTELLIGENTES ET DÉTAILLANT L'ENSEMBLE DES CONCEPTS ENVISAGEABLES POUR TRAITER L'APPRENTISSAGE AUTOMATIQUE DES SÉQUENCES D' ACTIONS ÉMANANT DE CHACUN DE NOS UTILISATEURS</i> -----	29
FIGURE 1.2.3.6 :	<i>DERNIER ARBRE DE NOTRE DÉFINITION TAXINOMIQUE DÉDIÉE AUX INTERFACES INTELLIGENTES FAISANT ÉTAT DES DIFFÉRENTES INFORMATIONS APPORTÉES PAR LES TERMINAUX DE COMMUNICATION, D'ALCATEL COMME DE LA CONCURRENCE</i> -----	29
FIGURE 1.2.4.3 :	<i>EXEMPLE D'UNE CARTE WEBSOM, BIEN APPLIQUE AU RESEAU INTERNET ET FOCALISE SUR LE DOMAINE DE L'INTELLIGENCE ARTIFICIELLE ("AI" SUR LA PRESENTE CARTE)</i> -----	36
FIGURE 1.3.2 :	<i>MODE EXPERIMENTAL ET PREMIERS CONCEPTS OPERATOIRES ENVISAGEABLES POUR LA MISE EN OEUVRE COMPLETE DE NOTRE MODELE D'INTERFAÇAGE INTELLIGENT</i> -----	57
FIGURE 2.2.2.2 :	<i>SCHÉMA ARCHITECTURAL DE L'INTERFACE AUTO-ADAPTABLE</i> -----	68
FIGURE 2.3.4.1 :	<i>SYNTHESES GRAPHIQUES DE LA DESCRIPTION FONCTIONNELLE DE NOS CONCEPTS AUTO-ADAPTABLES</i> -----	77
FIGURE 2.3.4.2 :	<i>DESCRIPTION FONCTIONNELLE DE NOS CONCEPTS AUTO-ADAPTABLES</i> -----	78-82
FIGURE 3.2.3.1 :	<i>SCHÉMA GLOBAL DE FONCTIONNEMENT DE NOTRE MODÈLE D'UTILISATEUR</i> -----	99
FIGURE 3.3.1 :	<i>SCHÉMA GÉNÉRAL DES INTERACTIONS ENTRE COMPORTEMENTS INDIVIDUELS ET CATÉGORIES D'UTILISATEURS PROPOSÉES PAR LA MÉTHODE "PERSONA"</i> -----	102
FIGURE 3.3.2.1 :	<i>CARACTÉRISTIQUES GÉNÉRALES DE LA MÉTHODE "PERSONA" ADAPTÉE DANS UN PREMIER TEMPS À L'INTERFAÇAGE HOMME-MACHINE</i> -----	103
FIGURE 3.3.2.2 :	<i>CARACTÉRISTIQUES PRINCIPALES DE LA MÉTHODE "PERSONA" ADAPTÉE À L'INTERFAÇAGE HOMME-MACHINE "GRAND PUBLIC" SUR TERMINAUX DE COMMUNICATION D'ALCATEL, AVEC MISE EN OEUVRE DE NOTRE NOUVEL "AXE DE LA CONNAISSANCE"</i> -----	104
FIGURE 3.3.3.2 :	<i>LISTE DES QUALITÉS PRINCIPALES ISSUE DE LA MÉTHODE "PERSONA" ET COMPOSANT LA PREMIÈRE APPROCHE EXHAUSTIVE DE NOTRE PROTOTYPE "D'INTERVIEW COMPORTEMENTALE"</i> -----	106
FIGURE 3.3.3.3 :	<i>LISTE CONTRACTÉE DES QUALITÉS ENTRAÎNANT UNE SECONDE APPROCHE MOINS INTRUSIVE DE NOTRE "INTERVIEW COMPORTEMENTALE"</i> -----	107

FIGURE 4.1.1 :	<i>MODE OPERATOIRE DE MISE EN OEUVRE DE NOTRE CONCEPT D'AUTO-ADAPTATION</i> -----	111
FIGURE 4.1.2 :	<i>EXEMPLES DE SEQUENCES ET DE GRAMMAIRES LES REPRODUISANT</i> -----	113
FIGURE 4.2.1.1 :	<i>SYNTHESES GRAPHIQUES DE LA REPRESENTATION DES ACTIONS EXECUTABLES</i> -----	116
FIGURE 4.2.1.2 :	<i>AUTOMATE HIERARCHIQUE DE TOUTES LES MANIPULATIONS REALISABLES PAR CHAQUE UTILISATEUR SUR L'APPLICATION "E-MAIL"</i> -----	117
FIGURE 4.4.3 :	<i>STRUCTURE HIERARCHIQUE REPRESENTATIVE DE NOS SEQUENCES D' ACTIONS</i> -----	129
FIGURE 4.5.3.2 :	<i>MODELISATION CHRONOLOGIQUE DES FONCTIONS "TELL ME" ET "DO IT"</i> -----	152
FIGURE 4.5.3.3 :	<i>SCHÉMA FONCTIONNEL DU MODULE INTELLIGENT DE PRÉDICTION ET D'ANTICIPATION DE SÉQUENCES RÉPÉTÉES D' ACTIONS</i> -----	154
FIGURE 5.1.3.1 :	<i>SCHÉMA RÉCAPITULATIF DES RÉSULTATS OBTENUS PAR IMPLÉMENTATION AU NIVEAU DES CARACTÉRISTIQUES RESPECTIVES DE NOS AGENTS INTELLIGENTS</i> -----	165
FIGURE 5.1.3.2 :	<i>SCHÉMA FONCTIONNEL DE L' ENSEMBLE DES AGENTS INTELLIGENTS COMPOSANT NOTRE " COUCHE IA "</i> -----	167
FIGURE 5.2.2.1 :	<i>DÉBUT DE NOTRE PROCÉDÉ D' EXTRACTION DES CONNAISSANCES, MIS EN OEUVRE, AU SEIN DE "L' AGENT D' EXÉCUTION", EN RELATION DIRECTE AVEC CHAQUE CHANGEMENT DE SÉLECTION ISSUE DE L' INTERFACE-UTILISATEUR</i> -----	169
FIGURE 5.2.2.2 :	<i>MODULE DE SIMPLIFICATION AUTO-ADAPTABLE DES SÉQUENCES D' ACTIONS, L' IMPLÉMENTATION DE LA DÉTECTION D' ÉVENTUELLES ERREURS DE MANIPULATION ÉMANANT DE L' UTILISATEUR</i> -----	170
FIGURE 5.2.2.3 :	<i>MISE EN OEUVRE DE NOS CORRECTIONS AUTO-ADAPTABLES AU SUJET DES ERREURS DE MANIPULATION DÉTECTÉES À PARTIR DU TRAVAIL DE CHAQUE UTILISATEUR</i> -----	171
FIGURE 5.2.2.4 :	<i>LANCEMENT DE LA PHASE D' ANTICIPATION ET D' EXÉCUTION AUTO-ADAPTABLES D' UNE SÉQUENCE D' ACTIONS, DÉTERMINÉE PAR NOTRE SYSTÈME MULTI-AGENTS ET SURTOUT PLEINEMENT DÉSIRÉE PAR L' UTILISATEUR COURANT</i> -----	171
FIGURE 5.2.2.5 :	<i>MISE EN ŒUVRE DE L' ALGORITHME INDISPENSABLE À NOTRE "AGENT D' EXÉCUTION" AFIN DE DÉTERMINER, EN TEMPS RÉEL, L' ITEM À EXÉCUTER AUTOMATIQUEMENT</i> -----	172
FIGURE 5.2.2.6 :	<i>EXTRAIT DES PRINCIPALES HIÉRARCHIES DE CLASSES IMPLÉMENTÉES DANS LE CADRE DE LA PROGRAMMATION "OBJET" DES MODÈLES AUTO-ADAPTABLES DE NOTRE "AGENT D' EXÉCUTION" PRIS EN EXEMPLE</i> -----	173
FIGURE 5.2.3 :	<i>INTRODUCTION DE L' ENSEMBLE DE LA MISE EN OEUVRE CORRESPONDANT AUX APPRENTISSAGES ET RECONNAISSANCES DES RÉPÉTITIONS D' ACTIONS, TRAVAUX RÉALISÉS PAR "L' AGENT DE RÉPÉTITION"</i> -----	175
FIGURE 5.2.4.1 :	<i>MISE EN ŒUVRE GLOBALE DU CANEVAS STRUCTUREL CONCERNANT NOS PROCÉDÉS D' APPRENTISSAGE AUTOMATIQUE ATTACHÉS AUX SÉQUENCEMENTS D' ACTIONS ET GÉRÉS PAR "L' AGENT DE SÉQUENÇAGE"</i> -----	176
FIGURE 5.2.4.2 :	<i>IMPLÉMENTATION PRINCIPALE STRUCTURANT NOS CONCEPTS DE RECONNAISSANCE AUTO-ADAPTABLE DES DIVERS SÉQUENCEMENTS D' ACTIONS</i> -----	177
FIGURE 5.2.5.1 :	<i>PREMIÈRES PARTIES DE "CODE" CHARGÉES D' ÉTABLIR L' ENSEMBLE DES STATISTIQUES PROPRES AU MODULE DES CARACTÉRISTIQUES ET QUALITÉS REMARQUABLES DES MANI- PULATIONS COURANTES, PROCESSUS CONTRÔLÉ PAR "L' AGENT DE COMPORTEMENT"</i> -----	179
FIGURE 5.2.5.2 :	<i>LIGNES LES PLUS SIGNIFICATIVES DE L' IMPLÉMENTATION CONCERNANT LES INVESTIGATIONS MÉNÉES PAR LE MODULE ATTACHÉ AU TRAITEMENT DES CONNAISSANCES DE CHACUN DE NOS UTILISATEURS</i> -----	180
FIGURE 5.2.5.3 :	<i>PRINCIPALE ÉTAPE DEMANDÉE AU MODULE DE GESTION DES SEUILS AUTO-ADAPTÉS, À SAVOIR, POUR CE PROCESSUS ACHEVANT LE TRAVAIL RÉALISÉ PAR NOTRE ULTIME AGENT INTELLIGENT, L' AJUSTEMENT DYNAMIQUE ET EN TEMPS RÉEL DES CARACTÉRISTIQUES DE LA RELATION HOMME-MACHINE</i> -----	181

FIGURE 5.2.5.4 :	<i>COPIES D'ÉCRANS FAISANT ÉTAT DE DIVERS RÉSULTATS OBTENUS LORS D'UTILISATIONS DE NOS NOUVEAUX AGENTS INTELLIGENTS, ET PLUS PARTICULIÈREMENT À PROPOS DE NOS DEUX DERNIÈRES ENTITÉS INTITULÉES "AGENT DE SÉQUENÇAGE" ET "AGENT DE COMPORTEMENT"</i> -----	182
FIGURE 5.3.2.1 :	<i>CONFIGURATION PHYSIQUE MISE EN OEUVRE POUR LES TESTS DE L'AUM</i> -----	185
FIGURE 5.3.2.2 :	<i>ECHELONNEMENT GÉNÉRAL SIMPLIFIÉ DU CURSUS D'ÉLABORATION DES PRODUITS D'ALCATEL</i> -----	186
FIGURE 5.3.3.2 :	<i>GRAPHIQUES RÉCAPITULATIFS DES PRINCIPALES DONNÉES OBTENUES</i> -----	190
FIGURE A1.1.1 :	<i>TERMINAL DE COMMUNICATION "WEB TOUCH ONE" D'ALCATEL, DEDIE A L'ACCES FACILE A L'INTERNET POUR LE "GRAND PUBLIC" ET NOTAMMENT MIS EN OEUVRE EN PARTENARIAT AVEC THOMSON MULTIMEDIA</i> -----	227
FIGURE A1.1.2 :	<i>TERMINAL DE COMMUNICATION "WEBPHONE @NET" DE SIEMENS, TOUT PREMIER CONCURRENT DECLARE DU "WEB TOUCH ONE" D'ALCATEL</i> -----	228
FIGURE A1.1.3 :	<i>TERMINAL "TEL@PHONE" DE MATRA NORTEL COMMUNICATIONS</i> -----	229
FIGURE A1.1.4 :	<i>TERMINAL DE COMMUNICATION "WEB AUDIO PHONE" DE SAMSUNG</i> -----	229
FIGURE A1.1.5 :	<i>TERMINAL DE COMMUNICATION "@MAX" DE COM ONE</i> -----	230
FIGURE A1.1.6 :	<i>TERMINAL DE COMMUNICATION "ISP 2001" D'UMEC</i> -----	231
FIGURE A1.2.1 :	<i>REPRÉSENTATION ILLUSTRÉE DU SCHÉMA CONCEPTUEL COMPLET DE "L'ALCATEL UNIFIED MESSAGING"</i> -----	233
FIGURE A1.3.2.1 :	<i>SYNTHESE DES "ENTREES/SORTIES" DE L'APPLICATION "E-MAIL"</i> -----	239
FIGURE A1.3.2.2 :	<i>SYNTHESES GRAPHIQUES DE LA DESCRIPTION FONCTIONNELLE DE L'APPLICATION "E-MAIL" ETUDIEE</i> -----	240
FIGURE A1.3.2.3 :	<i>DESCRIPTION FONCTIONNELLE DE L'APPLICATION "E-MAIL"</i> -----	241-244
FIGURE A3.1.2 :	<i>SCHÉMA DÉTAILLÉ DE L'ARCHITECTURE EN COUCHES DE MAPI, AVEC SPÉCIFICATION, EN PARTICULIER, DE SES ASSOCIATIONS AVEC COM</i> -----	265
FIGURE A3.2.1 :	<i>EXTRAIT PRÉSENTANT L'INTÉGRATION DES DIFFÉRENTES LIBRAIRIES AINSI QUE LE DÉVELOPPEMENT DE PROTOTYPAGES DE FONCTIONS, TOUS DEUX INDISPENSABLES À L'EXÉCUTION DE NOTRE AGENT D'EXÉCUTION PRIS EN EXEMPLE</i> -----	266
FIGURE A3.2.2.1 :	<i>EXTRAIT CORRESPONDANT À LA DÉCLARATION ET À LA DÉFINITION DES DIFFÉRENTES MÉTHODES "OBJET" COMPOSANT LA PREMIÈRE CLASSE IMPORTANTE DE NOTRE SYSTÈME INTELLIGENT ET MULTI-AGENTS</i> -----	267
FIGURE A3.2.2.2 :	<i>EXTRAIT MONTRANT LES DÉCLARATIONS DES DIFFÉRENTES MÉTHODES "OBJET" COMPOSANT LES DEUX CLASSES PRINCIPALES DE NOTRE SYSTÈME MULTI-AGENTS, RESPECTIVEMENT DÉDIÉES AU TRAITEMENT DES COMMANDES ET DES ÉVÉNEMENTS CARACTÉRISANT CHACUN DE NOS UTILISATEURS</i> -----	268
FIGURE A3.2.3.1 :	<i>FONCTION PRINCIPALE DE NOTRE PREMIER AGENT INTELLIGENT</i> -----	269
FIGURE A3.2.3.2 :	<i>"CODE" CHARGÉ DE DÉCRIRE LE CONSTRUCTEUR PRINCIPAL DE NOTRE CLASSE DE BASE NOMMÉE "MYEXCHEXT"</i> -----	269
FIGURE A3.2.3.3 :	<i>INSTALLATION DE LA CLASSE GÉRANT NOTRE "AGENT D'EXÉCUTION" ET LANÇANT NOTAMMENT, DÈS LE DÉMARRAGE DE NOTRE "COUCHE IA", LA TOUTE PREMIÈRE DES EXTRACTIONS D' ACTIONS RÉALISÉES PAR L'UTILISATEUR</i> -----	270
FIGURE A3.2.3.4 :	<i>PROCÉDURE GÉNÉRALE D'INSTALLATION DE TOUTE EXTENSION DE COMMANDE OU AJOUT D'UNE FONCTIONNALITÉ NOUVELLE À NOTRE SYSTÈME MULTI-AGENTS</i> -----	271
FIGURE A3.2.3.5 :	<i>MÉTHODE CHARGÉE DE TRAITER LA PRISE EN COMPTE, PAR NOTRE SYSTÈME D'AGENTS INTELLIGENTS, D'UN ÉVÉNEMENT CORRESPONDANT À UN CHANGEMENT DE SÉLECTION, PAR L'UTILISATEUR COURANT, D'UN ITEM QUELCONQUE SUR L'ÉCRAN</i> -----	272

FIGURE A3.3.1.1 :	<i>AFFICHAGE EN TEMPS RÉEL DES INFORMATIONS CORRESPONDANT À LA SÉLECTION PAR L'UTILISATEUR DES ICÔNES DÉDIÉS À NOTRE "COUCHE IA" (CODE DE L'ICÔNE OU DE LA LIGNE DU MENU DÉROULANT ATTACHÉS À NOTRE PREMIER "AGENT D'EXÉCUTION")</i>	----- 273
FIGURE A3.3.1.2 :	<i>EXTRAIT DU "CODE-SOURCE" NÉCESSAIRE AUX AFFICHAGES DE PROPOSITIONS OU INFORMATIONS INTERACTIVES, APPORTÉES EN TEMPS RÉEL À L'UTILISATEUR (EN ACCORD AVEC LE MODULE DE PRÉDICTION ET D'ANTICIPATION DES SÉQUENCES)</i>	----- 274
FIGURE A3.3.2.1 :	<i>DÉCLARATIONS DE DIFFÉRENTES MÉTHODES "OBJET" COMPOSANT L'INTERFAÇAGE DE NOTRE SYSTÈME MULTI-AGENTS AVEC LES APPLICATIONS TRAITÉES PAR CE DERNIER</i>	----- 275
FIGURE A3.3.2.2 :	<i>CONSTANTES, RENOMMAGES DE CONTEXTES ET AUTRES INTÉGRATIONS DE NOUVEAUX FICHIERS-SYSTÈMES NÉCESSAIRES AU TRAITEMENT PAR NOTRE SYSTÈME MULTI-AGENTS DES VERSIONS SUCCESSIVES D'UNE MÊME APPLICATION LOGICIELLE (DE MESSAGERIE ÉLECTRONIQUE, PUISQU'IL S'AGIT PLUS PRÉCISÉMENT DE "MICROSOFT OUTLOOK")</i>	----- 276
FIGURE A4.2.1.1 :	<i>DÉVELOPPEMENT DES DIFFÉRENTS CHAMPS DE MENU CORRESPONDANT AUX PRINCIPALES FONCTIONNALITÉS DE LA "COUCHE IA" INTÉGRÉES À L'AUM</i>	----- 281
FIGURE A4.2.1.2 :	<i>REPRÉSENTATIONS GRAPHIQUES ET DESCRIPTIFS TEXTUELS DES ICÔNES ATTACHÉS À LA "COUCHE IA"</i>	----- 282
FIGURE A4.2.1.3 :	<i>RÉSULTATS OBTENUS PAR NOS FONCTIONNALITÉS AUTO-ADAPTABLES SUITE AU DÉMARRAGE DE LA "COUCHE IA"</i>	----- 282
FIGURE A4.2.2 :	<i>RÉSULTATS OBTENUS LORS DU TEST DE NOTRE FONCTIONNALITÉ DE GESTION DES ACTIONS EFFECTUÉES DE MANIÈRE RÉPÉTITIVE</i>	----- 283
FIGURE A4.2.3.1 :	<i>RÉSULTATS OBTENUS LORS DU PREMIER TEST DE L'APPRENTISSAGE AUTOMATIQUE DES RÉPÉTITIONS DE SÉQUENCES D' ACTIONS</i>	----- 284
FIGURE A4.2.3.2 :	<i>SÉQUENCE RÉCAPITULATIVE DE L'ENSEMBLE DES DIFFÉRENTES ACTIONS EFFECTUÉES LORS DES TROIS PREMIERS SCÉNARIOS DE NOS "ALPHA-TESTS" ET EXTRACTION VISUELLE DES SÉQUENCES CACHÉES À L'AFFICHAGE</i>	----- 285
FIGURE A4.2.3.3 :	<i>RÉSULTATS OBTENUS LORS DU TEST COMPLÉMENTAIRE DE L'APPRENTISSAGE AUTOMATIQUE DES RÉPÉTITIONS DE SÉQUENCES D' ACTIONS</i>	----- 286
FIGURE A4.2.4.1 :	<i>RÉSULTATS OBTENUS PAR LE MODÈLE AUTO-ADAPTABLE DE L'UTILISATEUR SUITE AUX QUATRE PREMIERS SCÉNARIOS COMPOSANT LES "ALPHA-TESTS"</i>	----- 287
FIGURE A4.2.4.2 :	<i>COPIES D'ÉCRANS CHRONOLOGIQUES MONTRANT LES RÉSULTATS OBTENUS LORS DU TEST DE LA PRÉDICTION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR</i>	----- 289
FIGURE A4.2.5.1 :	<i>COPIES D'ÉCRANS MONTRANT CHRONOLOGIQUEMENT LES RÉSULTATS OBTENUS LORS DU PASSAGE DE LA PRÉDICTION À L'ANTICIPATION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR</i>	----- 291
FIGURE A4.2.5.2 :	<i>ÉTAT DES PREMIERS RÉSULTATS OBTENUS AU NIVEAU DE L'APPRENTISSAGE AUTOMATIQUE DES SÉQUENCES RÉPÉTÉES LORS DU TEST DE L'ANTICIPATION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR</i>	----- 292
FIGURE A4.2.5.3 :	<i>RÉSULTATS OBTENUS LORS DU TEST COMPLÉMENTAIRE DE L'ANTICIPATION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR</i>	----- 293
FIGURE A4.2.6 :	<i>RÉSULTATS OBTENUS LORS DU TEST DE REDÉMARRAGE DE NOTRE PROTOTYPE AVEC DES BASES DE DONNÉES DÉJÀ EXISTANTES ET BIEN REMPLIES</i>	----- 295
FIGURE A4.2.7.1 :	<i>RÉSULTATS SIGNIFICATIFS OBTENUS LORS DU TEST DE LA SIMPLIFICATION AUTO-ADAPTABLE DE SÉQUENCES D' ACTIONS CONTENANT AU MOINS DEUX FOIS UNE MÊME ACTION</i>	----- 297
FIGURE A4.2.7.2 :	<i>RÉSULTATS SIGNIFICATIFS OBTENUS LORS DU DERNIER TEST, À SAVOIR CELUI DE LA SIMPLIFICATION AUTO-ADAPTABLE DE SÉQUENCES D' ACTIONS CONTENANT L' ACTION COMMAND "UNDO" </i>	----- 299
FIGURE A4.2.8 :	<i>COPIES D'ÉCRANS DÉMONTRANT FINALEMENT QUE TOUTES LES CATÉGORIES D'UTILISATEURS MISES EN AVANT PAR LA MÉTHODE "PERSONA" SONT RÉELLEMENT ACCESSIBLES PAR NOTRE PROTOTYPE D'INTERFACE INTELLIGENTE AUTO-ADAPTABLE</i>	----- 301

Liste des tableaux

TABLEAU 1.1.2.2 :	<i>RECAPITULATIF DE TOUTES LES CARACTERISTIQUES TECHNIQUES ET CONTRAINTES D'UTILISATION EMANANT DES DIFFERENTS TERMINAUX DE COMMUNICATION DE TYPE "WEBPHONE" QUE NOUS AVONS RELEVES ET ETUDIES</i> -----	15
TABLEAU 1.1.3.3 :	<i>RECAPITULATIF DE TOUTES LES CARACTERISTIQUES ET CONTRAINTES TECHNIQUES EMANANT DES DIFFERENTES APPLICATIONS DE "MESSAGERIE UNIFIEE"</i> -----	20
TABLEAU 1.2.2.3 :	<i>DETAIL ET MISE EN VALEUR DES DIFFERENTES CATEGORIES D'INTERFACES INTELLIGENTES ISSUES DE NOTRE DEFINITION D'UNE HIERARCHIE ADAPTEE A L'INTERFAÇAGE HOMME-MACHINE</i> -----	27
TABLEAU 1.2.4.2 :	<i>CARACTERISATION HIERARCHIQUE DES REFERENCES D'INTERFACES INTELLIGENTES REMARQUABLES TRAITANT DES AIDES AUTOMATIQUES ET TUTORIELS INFORMATIQUES</i> -----	33
TABLEAU 1.2.4.3 :	<i>CARACTERISATION HIERARCHIQUE DES REFERENCES D'INTERFACES INTELLIGENTES REMARQUABLES TRAITANT DE L'IMPLICATION MASSIVE DES TECHNIQUES D'EXPLOITATION DE CONNAISSANCES</i> -----	37
TABLEAU 1.2.4.4 :	<i>CARACTERISATION HIERARCHIQUE DES REFERENCES D'INTERFACES INTELLIGENTES REMARQUABLES TRAITANT DE L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR</i> -----	41
TABLEAU 1.2.4.5 :	<i>CARACTERISATION HIERARCHIQUE DES REFERENCES D'INTERFACES INTELLIGENTES REMARQUABLES TRAITANT DE L'AVENEMENT DES AGENTS DITS "INTELLIGENTS"</i> -----	46
TABLEAU 1.2.4.6 :	<i>CARACTERISATION HIERARCHIQUE DES REFERENCES D'INTERFACES INTELLIGENTES REMARQUABLES TRAITANT DE LEURS APPLICATIONS RESPECTIVES EN ENTREPRISE</i> -----	52
TABLEAU 1.3.1 :	<i>RECAPITULATIF COMPLET DE NOS CARACTERISATIONS HIERARCHIQUES APPLIQUEES A L'ENSEMBLE DES REFERENCES D'INTERFACES INTELLIGENTES MIS EN AVANT PAR NOS ETATS DE L'ART</i> -----	53
TABLEAU 2.1.2.1 :	<i>RECAPITULATIF DES PREMIERS MECANISMES INTELLIGENTS ISSUS DES REGLES DE FONCTIONNEMENT DE NOTRE CONCEPT DE L'ASSISTANT CONVERSATIONNEL</i> -----	64
TABLEAU 4.2.3 :	<i>EXEMPLES DE SEQUENCES D' ACTIONS FREQUENTES</i> -----	119
TABLEAU 4.3.2 :	<i>EXTRAIT DE LA BASE DE DONNEES DES ACTIONS</i> -----	121
TABLEAU 4.3.3.1 :	<i>RECAPITULATIF DES SEQUENCES PRESENTEES PAR LE TABLEAU A1.4.1, ET VUES POUR CHACUN DES UTILISATEURS LISTES AUX DIFFERENTES DATES CONSIDEREES</i> -----	123
TABLEAU 4.3.3.2 :	<i>EXEMPLES DE DECOMPOSITION FINE POUR DEUX DES SEQUENCES D' ACTIONS PRESENTEES DANS LE TABLEAU A1.4.1</i> -----	124
TABLEAU 4.3.3.3 :	<i>EXTRACTION ET COMPTABILISATION DES DIFFERENTES REPETITIONS D' ACTIONS ET DE SEQUENCES D' ACTIONS ISSUES DES EXEMPLES DU TABLEAU 4.3.3.1</i> -----	124
TABLEAU 4.4.1 :	<i>LISTE DES ACTIONS POUVANT FAIRE OFFICE "D' ACTIONS-CLES"</i> -----	126
TABLEAU 4.4.2.1 :	<i>LISTE DES DIFFERENTES INTERACTIONS DE DEPENDANCE POUVANT ETRE EXTRAITES DES REPETITIONS RESPECTIVES DE SEQUENCES D' ACTIONS PAR LE TABLEAU 4.3.3.3</i> -----	127
TABLEAU 4.4.2.2 :	<i>LISTE DES DIFFERENTES RELATIONS DE DEPENDANCE POUVANT ETRE EXTRAITES DE MANIERE ABSTRAITE DE L'AUTOMATE HIERARCHIQUE DE LA FIGURE 4.2.1.2</i> -----	128
TABLEAU 4.4.5 :	<i>EXTRAIT DE GENERALISATION DE NOS REPETITIONS DE SEQUENCES D' ACTIONS</i> -----	131
TABLEAU 5.2.5 :	<i>RÉCAPITULATIF DE L'ENSEMBLE DES DONNÉES STATISTIQUES RASSEMBLÉES EN TEMPS RÉEL, PAR NOTRE "COUCHE IA", AU SUJET DU COMPORTEMENT DE CHAQUE UTILISATEUR ET DES QUALITÉS D' APPRENTISSAGE DE NOTRE SYSTÈME MULTI-AGENTS</i> -----	178

TABLEAU 5.3.3.2 :	<i>RECAPITULATIF DES INFORMATIONS QUANTITATIVES LES PLUS SIGNIFICATIVES, OBTENUES DE L'ETUDE-UTILISATEURS EFFECTUEE ENTRE LA FIN AOUT ET LA DEBUT NOVEMBRE 2001 AVEC L'ENSEMBLE DE NOS AGENTS INTELLIGENTS -----</i>	188
TABLEAU A1.4.1 :	<i>EXEMPLES D' ACTIONS, DE SEQUENCES D' ACTIONS, MAIS EGALEMENT DE REPETITIONS D' ACTIONS ET DE REPETITIONS DE SEQUENCES D' ACTIONS -----</i>	245
TABLEAU A1.4.2 :	<i>GLOSSAIRE RECAPITULATIF DE NOS DIVERSES PROPRIETES D' ACTIONS ETABLIES -----</i>	247
TABLEAU A2.1 :	<i>RÉCAPITULATIF DE L' ENSEMBLE DES OPÉRATEURS DE LOTOS MANIPULÉS AU COURS DES SPÉCIFICATIONS FORMELLES DE NOS MODÈLES AUTO-ADAPTABLES -----</i>	249
TABLEAU A4.1.1 :	<i>RÉSULTATS DES TESTS CONCERNANT LES LANCEMENTS D' APPLICATIONS-----</i>	277
TABLEAU A4.1.2 :	<i>RÉSULTATS DES TESTS CONCERNANT L' E-MAIL -----</i>	278
TABLEAU A4.1.3 :	<i>RÉSULTATS DES TESTS CONCERNANT LE FAX -----</i>	278
TABLEAU A4.1.4 :	<i>RÉSULTATS DES TESTS CONCERNANT LE VOICE-MAIL-----</i>	278
TABLEAU A4.1.5 :	<i>RÉSULTATS DES TESTS CONCERNANT LES REDIRECTIONS DE MESSAGES-----</i>	278
TABLEAU A4.1.6 :	<i>RÉSULTATS DES TESTS CONCERNANT LES ACCÈS CONCURRENTS-----</i>	278
TABLEAU A4.1.7 :	<i>RÉSULTATS DES TESTS CONCERNANT LA NOTIFICATION-----</i>	279
TABLEAU A4.1.8 :	<i>RÉSULTATS DES TESTS CONCERNANT LES PLANNINGS -----</i>	279
TABLEAU A4.1.9 :	<i>RÉSULTATS DES TESTS CONCERNANT LA SÉCURITÉ -----</i>	279
TABLEAU A4.1.10 :	<i>RÉSULTATS DES TESTS CONCERNANT LES RÉPERTOIRES DE TRAVAIL -----</i>	279
TABLEAU A4.1.11 :	<i>RÉSULTATS DES TESTS CONCERNANT LES LANGUES-----</i>	279
TABLEAU A4.1.12 :	<i>RÉSULTATS DES TESTS CONCERNANT NOS MODÈLES AUTO-ADAPTABLES-----</i>	280

Liste des définitions

DEFINITION 1.2.2.2 :	<i>ENUMERATION COMPLETE DES MULTIPLES CARACTERISTIQUES PERMETTANT LA QUALIFICATION EXHAUSTIVE DE L'INTELLIGENCE D'UNE INTERFACE</i> -----	25
DEFINITION 1.2.2.3 :	<i>SPECIFICATION DU CALCUL FONCTIONNEL REGULANT NOTRE NOUVEAU CONCEPT D'HIERARCHIE DES INTERFACES INTELLIGENTES</i> -----	26
DEFINITION 3.2.2 :	<i>DEFINITION INITIALE DE NOTRE "MODELE AUTO-ADAPTABLE DE L'UTILISATEUR" CHARGEE D'ETABLIR LES DIVERSES CARACTERISTIQUES REMARQUABLES D'UN UTILISATEUR EN COMMUNICATION AVEC UNE MACHINE</i> -----	98
DEFINITION 3.2.3.2 :	<i>NOUVELLE DEFINITION PLUS FONCTIONNELLE DES DIFFERENTES QUALITES REMARQUABLES D'UN UTILISATEUR EN INTERACTION AVEC UNE MACHINE</i> -----	101
DEFINITION 4.2.3.1 :	<i>REPRESENTATION CANONIQUE D'UNE ACTION</i> -----	118
DEFINITION 4.2.3.2 :	<i>REPRESENTATION CANONIQUE D'UNE SEQUENCE D' ACTIONS</i> -----	118
DEFINITION 4.2.3.3 :	<i>SECONDE REPRESENTATION DES SEQUENCES D' ACTIONS</i> -----	120
DEFINITION 4.2.3.4 :	<i>TROISIEME REPRESENTATION DES SEQUENCES D' ACTIONS</i> -----	120
DEFINITION 4.4.2.1 :	<i>RELATION DE DEPENDANCE ENTRE ACTIONS</i> -----	126
DEFINITION 4.4.2.2 :	<i>PROPRIETE D'INDEPENDANCE D'UNE ACTION</i> -----	126
DEFINITION 4.4.2.3 :	<i>RELATION DE DEPENDANCE TRANSITIVE ENTRE ACTIONS</i> -----	129
DEFINITION 4.4.4 :	<i>ALGORITHME D'EXTRACTION DES REPETITIONS D' ACTIONS ET DE SEQUENCES D' ACTIONS A PARTIR D'UNE BASE DE DONNEES</i> -----	130
DEFINITION 4.5.1.4 :	<i>DEFINITION FONCTIONNELLE DE NOTRE TRI</i> -----	134
DEFINITION 4.5.2.4 :	<i>ALGORITHME DE DETERMINATION DES REPETITIONS DE SEQUENCES D' ACTIONS SELON L'APPROCHE SYMBOLIQUE</i> -----	146
DEFINITION 4.5.2.5 :	<i>ALGORITHME DE DETERMINATION DES REPETITIONS DE SEQUENCES D' ACTIONS SELON L'APPROCHE STATISTIQUE</i> -----	150
DÉFINITION A2.2 :	<i>SPÉCIFICATIONS FORMELLES EFFECTIVES DE NOS SCÉNARIOS AUTO-ADAPTABLES</i> -----	250
DÉFINITION A2.3 :	<i>SPÉCIFICATIONS FORMELLES EXHAUSTIVES DE NOTRE NOUVEAU CONCEPT DE TRI APPLIQUÉ À L'EXEMPLE DU COURRIER ÉLECTRONIQUE</i> -----	256

Liste des scénarios

SCENARIO 2.3.1.1 :	<i>NOUVELLE FONCTIONNALITE GENERALE TRAITANT DE L'ABOUTISSEMENT OBLIGATOIRE D'UNE INFORMATION</i> -----	69
SCENARIO 2.3.1.2 :	<i>PREMIERE FONCTIONNALITE TRAITANT DE L'ABOUTISSEMENT OBLIGATOIRE D'UNE INFORMATION PAR UNE UTILISATION AUTO-ADAPTABLE DES MOYENS DE COMMUNICATION</i> ---	69
SCENARIO 2.3.1.3 :	<i>SECONDE FONCTIONNALITE TRAITANT DE L'ABOUTISSEMENT OBLIGATOIRE D'UNE INFORMATION PAR UNE INTERRUPTION AUTOMATIQUE FACE A UN APPAREIL INADEQUAT</i> -----	69
SCENARIO 2.3.1.4 :	<i>TROISIEME FONCTIONNALITE TRAITANT DE L'ABOUTISSEMENT OBLIGATOIRE D'UNE INFORMATION PAR UN CHOIX AUTO-ADAPTABLE DE L'OUTIL COURANT LE PLUS ADEQUAT</i> ----	70
SCENARIO 2.3.1.5 :	<i>NOUVELLE FONCTIONNALITE GENERALE TRAITANT DE LA MISE EN FORME DES HABITUDES DE L'UTILISATEUR</i> -----	70
SCENARIO 2.3.1.6 :	<i>PREMIERE FONCTIONNALITE TRAITANT DE LA MISE EN FORME DES HABITUDES D'UTILISATEUR AU NIVEAU DE LA LECTURE DES MESSAGES ARRIVES</i> -----	71
SCENARIO 2.3.1.7 :	<i>SECONDE FONCTIONNALITE TRAITANT DE LA MISE EN FORME DES HABITUDES AU NIVEAU DE L'ELABORATION D'UNE REPONSE AUTO-ADAPTABLE</i> -----	71
SCENARIO 2.3.1.8 :	<i>TROISIEME FONCTIONNALITE TRAITANT DE LA MISE EN FORME DES HABITUDES AU NIVEAU DE LA DETERMINATION D'UN MOYEN PREFERENTIEL</i> -----	72
SCENARIO 2.3.1.9 :	<i>NOUVELLE FONCTIONNALITE TRAITANT DU FILTRAGE DES MOYENS D'INFORMATION</i> -----	72
SCENARIO 2.3.1.10 :	<i>FONCTIONNALITE TRAITANT DE L'INFORMATION AUTO-ADAPTABLE DE L'UTILISATEUR</i> -----	73
SCENARIO 2.3.1.11 :	<i>FONCTIONNALITE TRAITANT DES STATISTIQUES ISSUES DU TRAVAIL DE L'UTILISATEUR</i> -----	73
SCENARIO 2.3.2.1 :	<i>NOUVELLE FONCTIONNALITE GLOBALE TRAITANT D'UNE MISE A JOUR AUTO-ADAPTABLE DES INFORMATIONS RECUEILLIES</i> -----	74
SCENARIO 2.3.2.2 :	<i>ULTIME FONCTIONNALITE GLOBALE TRAITANT DE TRIS AUTO-ADAPTABLES DES DIVERSES INFORMATIONS RECUEILLIES</i> -----	75
SCÉNARIO A4.2.1 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" CONCERNANT LE LANCEMENT ET L'INITIALISATION DE LA "COUCHE IA"</i> -----	281
SCÉNARIO A4.2.2 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" CONCERNANT LA GESTION DES ACTIONS EFFECTUÉES DE MANIÈRE RÉPÉTITIVE</i> -----	283
SCÉNARIO A4.2.3.1 :	<i>DESCRIPTIF D'UN PREMIER "ALPHA-TEST" CONCERNANT L'APPRENTISSAGE AUTOMATIQUE DES RÉPÉTITIONS DE SÉQUENCES D' ACTIONS</i> -----	284
SCÉNARIO A4.2.3.2 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" COMPLÉMENTAIRE CONCERNANT L'APPRENTISSAGE AUTOMATIQUE DES RÉPÉTITIONS DE SÉQUENCES D' ACTIONS ET VALIDANT LE CAS PARTICULIER INHÉRENT À L'AFFICHAGE DES SÉQUENCES SECONDAIRES INCLUES DANS DES SÉQUENCES PLUS GRANDES</i> -----	286
SCÉNARIO A4.2.4 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" CONCERNANT LA PRÉDICTION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR</i> -----	288
SCÉNARIO A4.2.5.1 :	<i>DESCRIPTIF D'UN PREMIER "ALPHA-TEST" CONCERNANT L'ANTICIPATION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR</i> -----	291
SCÉNARIO A4.2.5.2 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" SUPPLÉMENTAIRE CONCERNANT L'ANTICIPATION AUTO-ADAPTABLE DU COMPORTEMENT DE L'UTILISATEUR ET TRAITANT DU CARACTÈRE NON INTRUSIF DE NOTRE PROTOTYPE À CE SUJET</i> -----	293
SCÉNARIO A4.2.6 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" CONCERNANT LE REDÉMARRAGE DU PROTOTYPE AVEC DES BASES DE DONNÉES CONSÉQUENTES</i> -----	294
SCÉNARIO A4.2.7.1 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" CONCERNANT LA SIMPLIFICATION AUTO-ADAPTABLE DES SÉQUENCES D' ACTIONS CONTENANT AU MOINS DEUX FOIS UNE MÊME ACTION</i> -----	296
SCÉNARIO A4.2.7.2 :	<i>DESCRIPTIF D'UN "ALPHA-TEST" CONCERNANT LA SIMPLIFICATION AUTO-ADAPTABLE DES SÉQUENCES D' ACTIONS CONTENANT L' ACTION COMMAND "UNDO" </i> -----	298

Annexes

Dans le but d'être, en définitive, totalement exhaustifs au sujet de nos propos rapportés tout au long de ce mémoire de Thèse, nous nous proposons à présent d'en compléter le contenu par l'intégration à nos travaux de recherche de quatre annexes successives, indispensables par ailleurs à une compréhension efficace de l'ensemble des résultats que nous avons pu obtenir.

Voici par conséquent, en résumé introductif à ces ultimes données, un aperçu succinct caractérisant les domaines principaux et respectifs abordés par nos chapitres annexés à venir :

- L'**Annexe 1** décrit les états de l'art détaillés que nous avons consacrés aux terminaux et logiciels de communication, en assurant ainsi le dégagement et la définition, tous deux indispensables à nos travaux sur les mécanismes d'apprentissage automatique, de l'ensemble de nos modèles d'application. Cette première annexe s'achève de plus en proposant d'intéressants tableaux de récapitulation au sujet des diverses données abordées et des nombreuses définitions successivement élaborées tout au long de nos recherches.
- L'**Annexe 2** se voit chargée en second lieu de nous détailler l'ensemble des spécifications formelles que nous avons développées tout au long de nos années de Thèse afin d'initier, et de pleinement valider par la même occasion, chacun de nos scénarios et modèles auto-adaptables élaborés.
- L'**Annexe 3** correspond par suite à une description concrète, de même qu'à une analyse commentée des points les plus intéressants et les plus significatifs, du contenu de tous les fichiers et modules informatiques composant les structures de nos travaux d'implémentation ayant conduit à l'établissement final de notre système multi-agents et multi-utilisateurs nommé "Couche IA".
- L'**Annexe 4**, enfin, revient et décrit, résultats complets à l'appui, toutes les phases successives qui ont contribué à valider, selon l'ensemble notamment des procédés industriels utilisés au sein des entreprises du Groupe *ALCATEL*, chacun de nos agents intelligents ainsi que tous les modules intégrés et nécessaires à un fonctionnement efficace de notre innovante "Couche IA". Par extension, cette dernière annexe aboutit finalement à la validation exhaustive de notre entier modèle d'interface intelligente.

Annexe 1 :

Etude détaillée des terminaux et logiciels de communication ; dégagement et définition des différents modèles d'application

Si nous avons tenu, dès le **Chapitre 1** de ce mémoire de Thèse, à adjoindre la présente étude à nos travaux bibliographiques traitant de l'ensemble des divers terminaux et logiciels de communication actuels, nous ne pouvons par suite omettre, et afin de demeurer véritablement exhaustifs dans nos investigations, de préciser et pleinement compléter ces états de l'art par le détail de nos modèles d'application en résultant. Essentiels en outre à la ligne directrice de nos raisonnements jalonnant tout le **Chapitre 4**, l'extraction et la définition de telles modélisations nous permettent en effet de grandement spécifier l'important contexte inhérent à nos mécanismes d'apprentissage automatique, ainsi que bien sûr à nos concepts auto-adaptables et multi-agents.

Cependant, ceci exposé, nous désirons encore insister ici très fortement sur le fait que nous avons également jugé adéquat, pour d'évidentes raisons de synthèse et de pertinence, de clore cette première annexe en mettant en avant plusieurs intéressants tableaux de récapitulation, tant au sujet des diverses données abordées et manipulées qu'en ce qui concerne toutes les définitions successivement élaborées au cours de nos recherches. Ces tableaux de synthèse ont par ailleurs régulièrement été cités en référence à nos multiples explications, et notamment à celles mises justement en évidence au sein du quatrième chapitre.

A1.1. Découverte et état de l'art du "Webphone"

A1.1.1. Le "Webphone" selon *ALCATEL*

Présenté tout d'abord sous le nom "*d'Internet Screenphone*", ce terminal de communication a été le tout premier prétendant au trône de notre *Minitel* [Derouet, 1999].

S'agissant plus précisément d'un téléphone doté d'un écran tactile de 7,5 pouces en diagonale et offrant une résolution de 640x480 points en 256 couleurs, une luminosité réglable aussi ainsi qu'un mode en "économiseur d'écran" [Alcatel, 1999-1], cette nouvelle machine propose de plus un carnet d'adresses intégré, à environ mille entrées distinctes dans sa version la plus récente, permettant d'appeler directement son interlocuteur. Nous pouvons alors déjà noter à ce niveau que, dans la liste des coordonnées, peuvent figurer aussi bien noms, numéros de téléphone, adresses *e-mail* de même qu'adresses *Internet*.

Au coeur de l'appareil, on trouve ensuite un processeur *Motorola MPC 823* cadencé à 66 MHz, 32 Mo de mémoire vive, 8 Mo de mémoire de type "Flash", et 2 Mo de mémoire morte ou non détruite à l'extinction du Webphone. En outre, le navigateur *Java* embarqué, qui pour l'heure supporte les standards *Internet* de base comme *HTTP 1.1* et *HTML 3.2*, peut encore largement évoluer, comme d'ailleurs le reste des diverses applications proposées dont certaines focaliseront une grande partie de notre énergie dans la suite de ce rapport de Thèse.

Original et pratique, l'accès à la page de démarrage personnalisable de ce terminal s'effectue aussi par simple appui sur une touche. Si l'accent est ainsi constamment mis sur la convivialité, l'ergonomie et un grand confort d'utilisation [Alcatel, 1999-1], la navigation comme la saisie sont en plus assurées par un clavier alphanumérique, lequel se replie discrètement sous l'appareil et se voit associé également à un lecteur de carte à puce [Belot, 1999].

Le client de messagerie, aussi écrit en *Java*, supporte les protocoles *SMTP*, *POP3* et *IMAP4*. Sans oublier enfin que les messages peuvent être rédigés hors connexion et comporter des fichiers attachés textuels ou multi-médias (formats *Wav*, *Gif*, *Jpeg*, ...). D'ailleurs, dès réception d'un message nouveau et sans que l'appareil ne soit forcément en mode actif, une diode clignotante signale cette arrivée à l'utilisateur.

Finalement commercialisé au tout début de l'automne 1999 sous le nom "d'ALCATEL Web Touch One", ce premier terminal de communication *Internet* est bien entendu disponible aujourd'hui dans les magasins spécialisés, mais aussi dans ceux de la grande distribution afin d'atteindre au maximum une clientèle avant tout "grand public".

Présentant alors notamment un bouton entièrement dédié à l'accès immédiat au réseau *Internet* ainsi qu'un autre concernant exclusivement la fonction *Minitel*, ce nouvel outil *ALCATEL*, encore plus complet et ergonomique qu'auparavant, continue cependant d'axer ses principales capacités sur un téléphone performant aux fonctions les plus avancées. Nous pouvons en effet notamment citer ici l'option "mains libres", une boîte vocale, l'identification de l'appelant, la mise en attente, etc [Alcatel, 1999-1].

Comme nous pouvons de plus très aisément le remarquer sur la photographie présentée ci-contre (**Figure A1.1.1**), l'agréable apparence compacte, associée à une taille concrète de 28,4 x 21,8 x 12,1 centimètres et un poids minimal de 3,1 kilogrammes (bloc d'alimentation compris), permet également à un tel terminal de communication de s'adapter à toutes les pièces d'une maison particulière.



Figure A1.1.1 : Terminal de communication "Web Touch One" d'ALCATEL, dédié à l'accès facile à l'Internet pour le "grand public" et notamment mis en oeuvre en partenariat avec THOMSON Multimedia [Alcatel, 1999-1]

En définitive, nous pouvons retenir dès à présent que, créé pour que tout un chacun puisse profiter d'*Internet* sans contraintes, le "Web Touch One" d'ALCATEL permet d'accéder à une multitude de services, facilement, et instantanément : « c'est l'*Internet* au bout des doigts » [Alcatel, 1999-1].

A1.1.2. Le "WebPhone @net" de Siemens

Premier concurrent en date du précédent "Web Touch One" d'ALCATEL, le "WebPhone @net" de Siemens (**Figure A1.1.2**), bien entendu également prévu pour accéder simplement à *Internet*, présente lui une interface *Windows CE* sur un écran tactile de 640x480 pixels, cependant monochrome [Flores, 1998].



Figure A1.1.2 : Terminal de communication "WebPhone @net" de Siemens, tout premier concurrent déclaré du "Web Touch One" d'ALCATEL

Remplissant des fonctions de télécopie, de répondeur et surtout de téléphone sans fil pour ce qui le concerne, ce terminal *Internet* de Siemens assure également des transactions bancaires sécurisées grâce à son propre lecteur de carte à puce. Doté ensuite d'un modem intégré à 33,6 Kbps, il peut accueillir un clavier et être connecté à une imprimante ou à un ordinateur par le biais de ses ports internes, "série" et "parallèle".

De son côté, la mémoire de cette machine est capable de stocker cent messages électroniques, ou vingt *fax*. Quant à la mise à jour des logiciels intégrés, celle-ci est déjà prévue via le réseau *Internet*, en notant encore que ces applications comprennent un navigateur (compatible *HTML 3.2* et *Javascript*), un agenda électronique et des boîtes aux lettres faisant appel aux protocoles *SMTP* et *POP3*.

A1.1.3. Le "Tel@phone" de Matra Nortel Communications

D'origine et de conception françaises à l'image du "Web Touch One" d'ALCATEL, le "Tel@phone" de Matra Nortel Communications, ou terminal grand public *Internet*, est lui doté d'un écran couleur tactile de huit pouces de diagonale, et offre également trois modes de saisie distincts (stylet, doigt et clavier) ainsi qu'un lecteur de carte à puce (**Figure A1.1.3** à suivre).



Figure A1.1.3 : Terminal "Tel@phone" de Matra Nortel Communications

Après avoir essuyé un échec en tant que terminal *Internet* dans les entreprises, le "WebPhone" selon *Matra Nortel Communications* tente ainsi aujourd'hui une percée en grande distribution aux côtés du "WebTouch" d'*ALCATEL* [Thorel, 1999].

A noter cependant ici que, contrairement à *ALCATEL* justement qui a choisi de travailler avec le système d'exploitation *JavaOS*, *Matra Nortel Communications* a opté de son côté pour *MS Windows CE*, 20 Mo de mémoire vive (soit 12 Mo de moins que pour *ALCATEL*) et un clavier "azerty" (non alphanumérique) [Belot, 1999].

A1.1.4. Le "Web Audio Phone" de Samsung

Encore plus proche certainement du "Web Touch One" d'*ALCATEL* que le précédent "Tel@phone" de *Matra Nortel Communications* du fait notamment de l'utilisation du même système d'exploitation, le "Web Audio Phone" de *Samsung* (**Figure A1.1.4**) se distingue pourtant par 16 Mo seulement de mémoire vive (deux fois moins que dans le cas d'*ALCATEL*) et l'absence d'un accès direct à l'application du *Minitel*, mais aussi par la remarquable intégration d'une nouvelle carte *Ethernet* ainsi que la présentation d'un écran plus grand de 8,2 pouces [Belot, 1999].



Figure A1.1.4 : Terminal de communication "Web Audio Phone" de Samsung

A1.1.5. Le "@Max" de COM One

Après les premiers terminaux de communication du type "Webphone" déjà abordés plus haut, c'est au tour de *COM One* de se lancer dans la fabrication d'un tel "*Minitel Internet*".

Et, comme nous avons aussi grandement pû le constater jusqu'à présent, cette société insiste d'emblée sur le fait que le réseau *Internet* apporte, à notre époque, une indéniable et toute nouvelle dimension à la communication. Celui-ci apparaît en effet « toujours en perpétuel mouvement technologique, tant sur le plan de la structure des réseaux que des fonctionnalités multi-médias. Ainsi, et afin de parer à une rapide obsolescence, les terminaux se doivent aujourd'hui d'être évolutifs. » [COM One, 2000]

Reposant sur ce concept de tout premier ordre, "@Max" (prononcer "at-max") se veut être en conséquence un outil puissant et ouvert. Utilisant les dernières technologies industrielles de l'environnement PC, il se présente alors comme un terminal hybride permettant de se connecter aussi facilement sur les services en ligne du réseau IP (*Internet Protocol*) que sur ceux du réseau propriétaire de *France Télécom* [Astor, 1999].

"@Max" ouvre ainsi, à l'aube de l'an 2000, les portes du *Web* et du *Minitel*, tout en étant également compatible avec *Linux* et *MS Windows CE*, *Netscape* comme *MS Internet Explorer*, le câble ou encore les lignes téléphoniques classiques. Doté d'un processeur *NS Geode Media Gx1* cadencé à 266 MHz, de 32 Mo de mémoire vive, de 8 Mo de mémoire "Flash", ainsi que d'un écran à cristaux liquides de 16 millions de couleurs et de taille confortable avec ses 8,2 pouces, "@Max" est également équipé, entre-autres, du navigateur *Web* de *Spyglass (Voyager)* qui propose à l'utilisateur un accès simple et agréable à l'*Internet*, ainsi qu'un très bon confort de visualisation et une navigation rapide répondant aux standards actuels (*HTML*, multi-fenêtrage, *Java Script* et *Java*). Associé à un client *POP3 / SMTP*, ce nouveau terminal supporte en outre *Real Player* de même qu'un format de diffusion vidéo développé au sein-même de *COM One* et appelé *Viewsurf* par ses soins.

Disposant de la réception manuelle et automatique (en cas d'absence) de l'ensemble du courrier, *COM One* n'a pas non plus omis d'intégrer un voyant lumineux à son appareil pour prévenir tout personne de l'arrivée de nouveaux messages. Soucieux aussi de proposer un vrai terminal interactif, cette société a de plus intégré un clavier convivial de type ordinateur portable, un pointeur *TouchPad*, un port "série" et un port "parallèle" (compatible avec la plupart des imprimantes du marché) permettant enfin de connecter une unité de stockage, un lecteur de CD-Rom externe ou encore une caméra numérique.

De cette manière, nous constatons alors que *COM One* a conçu, pour le grand public mais aussi pour le monde professionnel, un terminal véritablement stylé, simple, convivial et intuitif. Nous en découvrons d'ailleurs ci-dessous une récente illustration (**Figure A1.1.5**).



Figure A1.1.5 : Terminal de communication "@Max" de COM One [COM One, 2000]

Puissant, "@Max" offre en conséquence toutes les fonctions avancées de navigation *Internet*, de courrier électronique, de *fax*, de *Minitel*, de répertoriage multi-critères et, bien entendu, de téléphonie ("mains-libres", présentation du numéro du correspondant, répondeur-enregistreur vocal avec avertissement en cas d'appels) [COM One, 2000]. Notons ici qu'il propose aussi une prise de contact facile avec l'*Internet*, mais sans jamais supprimer les avantages classiques des terminaux de communication actuels.

D'un autre côté, grâce à son architecture ouverte, "@Max" supporte les principaux systèmes d'exploitation du marché. De base, *COM One* a cependant sélectionné le système *QNX* : un système d'exploitation industriel *Unix* temps réel, puissant et multi-tâches. Construit ensuite autour d'une interface graphique conviviale, le terminal peut encore être personnalisé et mis à jour grâce à un simple téléchargement en ligne. Son lecteur de carte à puce ainsi que le logiciel de cryptage *SSL 3.0* ("*Secure Socket Layer*"), proposé en standard, permettent d'ailleurs à "@Max" d'offrir toutes les fonctions de paiement en ligne, de sécurité d'accès, de profil, d'authentification et de fidélisation "marketing" du client.

En définitive, il ressort aussi avant tout de la présente étude qu'en plus de l'alléchant marché "grand public" actuel, *COM One* semble également viser à terme celui des grandes entreprises désireuses de renouveler leur parc de *Minitel*. Sans oublier pour autant celui des applications verticales qui reposent sur l'utilisation d'un terminal dédié [Astor, 1999]...

A1.1.6. "L'Internet Screenphone" d'Universal Microelectronics Company

Pour ce dernier terminal de communication que nous nous jugeons propice de découvrir dans ces lignes, *UMEC* présente, en coopération avec l'imposant constructeur d'équipements et de logiciels de télécommunication qu'est *LUCENT Technologies (Bell Labs)*, "l'ISP 2001" ou "l'*Internet ScreenPhone Inferno Intelligent*". Entre-autres, le système d'exploitation et d'interconnexion de réseaux *Inferno 2.0* de *LUCENT Technologies* permet à ce nouvel appareil d'accéder de manière sécuritaire à l'*Internet*, afin de rassembler notamment des informations d'un simple clic pour l'utilisateur courant [UMEC, 2000].

Voici d'ailleurs dès à présent une vue générale de ce nouveau terminal dédié à *Internet* (**Figure A1.1.6**).



Figure A1.1.6 : Terminal de communication "ISP 2001" d'UMEC [UMEC, 2000]

Par suite, il nous reste à relever que "l'*Internet Screenphone*" d'*UMEC* supporte également deux lignes téléphoniques distinctes : une pour les appels téléphoniques et une autre, simultanée, pour la connexion *Internet*. En outre, avec un cryptage avancé *SSL 3.0* chargé de sauvegarder toute information sensible transmise au travers du domaine public de l'*Internet*, l'exécution intuitive et conviviale de cet "*Internet Screenphone*" est globalement associée à un répondeur digital, un téléphone "mains-libres", une identification des appelants et un complet carnet d'adresses interactif [UMEC, 2000].

Enfin, pêchant cependant encore certainement par la taille quelque peu invalidante de son écran tactile de 7,4 pouces seulement, cet ultime terminal présente tout-de-même divers logiciels intégrés et fonctions *Web* de grande qualité comme *HTML 3.2*, *Javascript*, un modem de 56 Kbps, *SMTP* et *POP3* pour la gestion des *e-mails*, un clavier rétractable ou encore un visionnement d'images efficace, qui fournissent tous à l'utilisateur un cadre de travail commode et tout-à-fait aisé.

A1.2. La "Messagerie Unifiée", second outil pour nos recherches

A1.2.1. La "Messagerie Unifiée" selon *ALCATEL* : "l'*AUM*"

A l'image de ce que nous avons déjà quelque peu abordé au cours du **Chapitre 1** de ce mémoire, "l'*ALCATEL Unified Messaging*" (*AUM*), ou plus précisément "l'*ALCATEL 4400 Unified Messaging*" en relation avec le serveur de télécommunication et le système de téléphonie qui lui sont associés (cf. **1.1.1.3**), regroupe pas moins de trois médias distincts issus directement de la messagerie : l'*e-mail*, le *voice mail* et la télécopie.

En effet, l'*AUM* se charge, par définition, "d'unifier" trois différents systèmes afin d'aboutir à une seule et unique application complète, réussissant ainsi à fortement diminuer le nombre d'outils de messagerie devant être constamment administrés. La **Figure A1.2.1** proposée à l'entame de la page suivante est d'ailleurs très explicite à ce sujet.

En remarquant alors que cette application logicielle travaille également de concert avec les fonctions de la base de données et le réseau unique d'administration pour stocker et manipuler l'ensemble des messages en une seule et même interface-utilisateur, nous découvrons que ce système d'*ALCATEL* présente ensuite d'autres riches et très intéressantes caractéristiques, comme un traitement avancé des différents moyens de communication, une uniformisation du format des messages et une amélioration du flux des informations [Alcatel, 2000]. De plus, il y est aussi pris en compte l'utilisation de la puissance et des capacités multi-tâches de *Microsoft Windows NT*, l'intégration bien entendu du système de messagerie électronique sur un réseau "TCP/IP", ainsi que l'interconnexion avec le serveur de communication *ALCATEL* "4400 PBX" via des cartes "*multiport voice/fax*".



Figure A1.2.1 : Représentation illustrée du schéma conceptuel complet de "l'ALCATEL Unified Messaging" [Alcatel, 1999-3]

Pour être tout-à-fait exhaustif dans notre actuelle présentation, nous nous devons également d'indiquer ici que la dernière version commercialisée de "l'ALCATEL Unified Messaging" correspond à la version 2.4 de l'application. Faisant alors appel à un PC et à un serveur de communication ("ALCATEL 4400 PBX") lui-même associé à tous les types de poste téléphonique, l'AUM se définit actuellement par un produit fonctionnant à partir du système d'exploitation *Microsoft NT Server 4.0 Build 1381* issu du *Service Pack 6* [Alcatel, 2000]. S'appuyant en outre sur le serveur de messagerie électronique *Microsoft Exchange Server 5.5 Build 2653.22*, elle assure enfin une complète interaction avec ses différents utilisateurs grâce à l'application *Microsoft Outlook*.

Et nous concluons à ce niveau en n'oubliant pas spécifier encore que, si l'AUM est bien exploitable de manière fixe (ordinateur personnel) comme en déplacement (téléphone mobile), les services d'ALCATEL y ont tout récemment inséré de nouvelles fonctionnalités, dites "d'Opératrice Automatique" et caractérisées par un filtrage ou un routage d'appel en fonction du calendrier, du numéro appelé, de la requête courante aussi du client concerné.

A1.2.2. "SmartPhone Messagerie Unifiée" de Novavox

« A l'origine, "SmartPhone Messagerie Unifiée" (SMU) est un serveur vocal au-dessus duquel on a développé une solution de messagerie unifiée destinée à une dizaine d'utilisateurs », explique le dirigeant de *Novavox*, Yvan Taieb [Lichtner, 2000].

Connu pour ses solutions de téléphonie, *Novavox* a en effet décidé de poursuivre et de compléter son offre en proposant des modules d'extension traitant de la messagerie unifiée et reposant sur *Microsoft Exchange*. De plus, "SmartPhone Messagerie Unifiée" fonctionne en architecture client-serveur, avec *Microsoft Exchange* pour ce qui concerne le serveur et *Microsoft Outlook* sur le poste "client". Totalement intégré dans *Exchange*, *SMU* permet ainsi de recevoir et d'envoyer, à partir d'une interface unique, des *e-mails*, *fax* et messages vocaux, qui rentrent alors aussi dans le circuit traditionnel d'information de l'entreprise considérée.

En outre, *SMU* met plus précisément en œuvre trois modules-clés différents : le serveur, l'administrateur et l'agent. Le module "serveur" complète tout d'abord les fonctions de messagerie électronique d'*Exchange* et propose, comme déjà établi plus haut, un serveur de *fax* ainsi qu'un système complet de messagerie vocale. Le module "administrateur", de son côté, permet de contrôler en permanence le statut de toutes les boîtes vocales. De surcroît, il autorise le paramétrage de toute la messagerie (taille de chaque boîte, temps de validité des messages, ...) et l'accès aux rapports de statistiques. Le module "agent", quant à lui, se charge de rajouter des extensions "clients" pour *Exchange* et *Outlook*.

Ainsi, après installation de l'ensemble des extensions précédemment définies, les messages et les *fax* sont, finalement, rapidement répertoriés dans la boîte de réception de chacun des utilisateurs. Très simple d'utilisation aussi, « une seule journée d'installation suffit pour mettre en place tous les paramètres de *SMU* », commente d'ailleurs Yvan Taieb.

Pour écouter les messages écrits par le biais du téléphone, *SMU* utilise alors les services d'une carte vocale *DSP* (*Digital Signal Processor*) - celle de *Dialogic* par exemple. A noter que cette carte intègre notamment une interface téléphonique analogique. En outre, à partir de la plate-forme *SMU*, *Novavox* propose également d'étendre à terme le système de messagerie unifiée par un développement d'applications *SVI* ("Serveur Vocal Interactif").

A1.2.3. "David Professional" de Tobit Software

Editeur allemand spécialisé dans les logiciels de communication, *Tobit Software* insiste d'emblée sur le fait que sa solution de messagerie unifiée "David Professional" (*Version 6.0*) se positionne aujourd'hui comme la plate-forme la plus complète du marché : messagerie, serveur SMS ou bien vocal, télécopie... [Lichtner, 2000]

A l'instar de "SmartPhone Messagerie Unifiée", "David Professional" s'appuie en effet sur une architecture en mode client-serveur et vient compléter les fonctions de messagerie et de *fax* de *Microsoft Exchange*, mais aussi de *Novell Groupwise*. Signalons d'ailleurs au passage que l'application est de plus livrée en option avec une passerelle de messagerie *Lotus Notes* et *SMTP*.

Et l'application se décline cette fois-ci en quatre modules : un serveur de *fax*, un serveur de messagerie vocale, un serveur SMS et un module d'administration, l'*InfoCenter* (gestion, statuts et distribution des messages, archivage, etc). A remarquer alors notamment que le serveur SMS compresse automatiquement tous les messages courts : « Il filtre toutes les informations redondantes et les caractères sans importance (blancs, lettres doubles, etc). Résultat, cette technologie permet de transmettre des messages en s'affranchissant de la limite maximale des 160 caractères pour ce type de messages », précise Dieter van Acken, responsable de la communication au sein de *Tobit Software*.

En dehors de ses fonctions de base précédemment exposées, "David Professional" intègre également d'intéressants services tels que la numérotation automatique, l'identification des appels, le routage, la gestion de documents ainsi qu'un module de publication automatique de documents baptisé *WebBox*. Ce module constitue l'un des points forts de l'application, car il permet de transformer les messages reçus en pages *Web*, lesquelles sont, ensuite, consultables à distance à partir de n'importe quel navigateur. Pour importer les messages issus des téléphones cellulaires, *WebBox* transforme aussi les messages au format *WML* (*Wireless Markup Language*) en *HTML*. Par ailleurs, la dernière version inclut un nouveau "client" qui autorise l'accès au *Web* en temps réel aux boîtes de réception à partir d'un navigateur.

Très complet, "David Professional" va donc presque déjà au-delà de sa mission première et actuelle de messagerie unifiée.

A1.2.4. "Enterprise Interaction Center" d'Interactive Intelligence

"Enterprise Interaction Center" (ou encore *EIC*), développé par la société américaine *Interactive Intelligence*, est un progiciel de messagerie unifiée qui repose entièrement sur un moteur de gestion des interactions en langage *Java* [Lichtner, 2000].

EIC combine plus précisément, et au sein d'un serveur *Windows NT*, les fonctions d'un autocommutateur (PBX) avec celles d'un distributeur automatique d'appels (ACD), d'un répondeur vocal, d'un serveur de télécopies, d'une messagerie vocale et, enfin, d'un centre de téléphonie intégrée (CTI). En outre, *EIC* se compose de deux modules généraux distincts : d'un côté, le serveur d'interaction et, de l'autre, l'interaction administrateur.

Le serveur d'interaction comprend une multitude de composants dont un système complet de distribution automatique d'appels avec un routage intelligent, la messagerie unifiée pour fédérer tous les types de messages en s'appuyant sur *Microsoft Exchange*, un serveur de télécopies et le composant *Interaction Designer*. Ce dernier est un outil de développement orienté objet qui permet d'automatiser les processus d'appels téléphoniques, les *fax* et les *e-mails*, qu'ils soient entrants ou sortants. Cet outil permet en fait de définir une série d'étapes qui sont utilisables ensuite en mode "run-time" (temps réel) sous la forme "d'applets" (petites applications indépendantes) écrites bien évidemment en *Java*.

Enfin, pour être tout-à-fait complet à ce stade de notre étude, notons également que l'*EIC* intègre un Serveur Vocal Interactif (SVI) qui autorise l'accès en consultation et en mise à jour des informations de la base de données à un correspondant muni d'un téléphone à touches. A ce niveau, l'administrateur a la possibilité de générer des "scripts" (petits programmes ou listes d'instructions) SVI complets ainsi que des analyses personnalisées.

Le module d'administration, quant à lui, permet bien entendu de contrôler le système. L'administrateur peut, entre-autres, effectuer très simplement des modifications telles que créer des files d'attente, ajouter des utilisateurs, etc. Par ailleurs, *EIC* est entièrement compatible avec *Windows NT*, *Windows 95* et *Microsoft Exchange Server* : il est vu comme un nouveau composant de la suite *Microsoft BackOffice* et des applications *Office*. Dans cette optique, il utilise notamment le moniteur de performances de *Windows NT* et le système de gestion des utilisateurs de ce système d'exploitation.

A1.2.5. La "Messagerie Unifiée" selon *Freeplanning*

Logiciel de messagerie unifiée et de travail coopératif, "Freeplanning" est produit par la société française du même nom, créée à Paris par une équipe de quatre professionnels du *Web* et partenaire de *SUN Microsystems*. Permettant d'archiver et d'organiser l'ensemble de son courrier électronique à partir d'un assistant numérique personnel (PDA), d'un ordinateur portable ou encore d'un poste fixe, cet outil propose également à l'utilisateur de consulter ses données (rendez-vous, tâches, contacts, ...) ou ses messages (*mails*, *fax* et SMS) à distance, via son téléphone mobile et sa boîte vocale associée, mais aussi bien entendu via le réseau *Internet* [Nemec-Poncik, 2001-1].

Doté d'applications comme le partage d'agendas, les carnets d'adresses, la création de portails d'informations personnalisés, ou encore le stockage et le partage de fichiers, "Freeplanning" est ensuite capable de se synchroniser avec les logiciels déjà existants dans le poste-utilisateur (*Outlook* ou *Lotus*) et avec les assistants numériques personnels (PDA) de type *Palm*, *Psion* ou encore *Pocket PC*. Sans oublier que l'ensemble des rendez-vous pris en temps réel est, par la suite, mis à jour automatiquement sur le micro-ordinateur ou l'assistant personnel de l'utilisateur, et présenté à lui sur une seule page.

Alors destinés aux utilisateurs ou Internautes individuels comme à tous les types d'entreprises, les différents modules de "Freeplanning" permettent à chacun de visualiser succinctement les informations essentielles permettant par exemple d'organiser une journée [Crosnier-Mangeat, 2001]. De plus, le "planning" ou agenda de l'utilisateur courant est coordonné avec l'ensemble de ses interlocuteurs par un système simple de demande et d'acceptation de rendez-vous, et ce sur l'ensemble de ses ordinateurs fixes ou mobiles. Dans le même ordre d'idée, l'information importante de son éventuelle société apparaît régulièrement sur la page d'accueil de "Freeplanning", alors qu'un système d'alerte permet, par l'envoi d'*e-mails* ou de SMS, de le prévenir des événements importants qui sont susceptibles de survenir.

Une telle coordination des différents agendas de l'utilisateur est proposée par la fonction "synchronisation" de "Freeplanning". Cette fonction garantit entre-autres à l'utilisateur la non-duplication de ses bases d'adresses ainsi que la saisie unique de ses informations, lui permettant le suivi permanent de ses données, qu'il soit sédentaire ou plutôt nomade.

A tout cela, "Freeplanning" ajoute finalement des fonctions de travail en groupe qui laissent à l'utilisateur la liberté de gérer, pour un projet donné, des groupes de travail internes et externes à son entreprise, et d'en assurer ainsi la coordination (fichiers joints, agendas, retours d'informations, forum, ...). Sans oublier enfin que, grâce à la possibilité de gérer simultanément plusieurs bases de données existantes, chaque utilisateur peut aussi mettre en place une véritable campagne "d'e-marketing".

A1.3. Descriptions avancées des différents modèles d'application

A1.3.1. Spécification des processus "d'Entrées/Sorties"

A1.3.1.1. Introduction

Afin d'initier proprement l'étude de nos mécanismes d'apprentissage automatique sans contraintes supplémentaires, nous nous proposons d'emblée de prendre un exemple bien représentatif de nos modèles d'application et d'en approfondir l'ensemble des données prises en compte. Ainsi, nous allons à présent nous pencher plus particulièrement sur le cas du *WebTouch* d'*ALCATEL* (dernière génération du *Webphone*) et de ses intéressantes caractéristiques de messagerie électronique que nous avons déjà appréhendées lors du premier chapitre.

Dans un premier temps, il faut noter que cette application "*e-mail*", étudiée ici à partir de plusieurs versions de logiciels actuels d'*ALCATEL*, fait en priorité appel à trois différents fichiers ou "classes" *Java* pour gérer ses entrées (*FileClient*, *MailClient* et *Pop3Client*) ainsi qu'à deux autres pour ses sorties (*Mail* et *MailOut*).

Cependant, il ne faut pas oublier qu'à ce premier procédé de gestion des "Entrées/Sorties" s'en ajoute un second tout aussi important, surtout dans le cas présent de nos travaux sur les interfaces intelligentes. En effet, il s'agit de l'écoute et de la prise en compte, au niveau de l'interface homme-machine de chacun des logiciels hôtes considérés, des événements relatifs aux actions de l'utilisateur sur l'application, et plus précisément des séquences événementielles réalisées par l'utilisateur sur les boutons de l'interface manipulée (méthodes des *Listener Interfaces*).

Ces constats initiaux, que bien entendu nous nous empressons de détailler ci-dessous, vont nous permettre ensuite de décrire l'application "*e-mail*" de manière fonctionnelle, pour aboutir ainsi au dégagement d'un de nos modèles dits "d'application".

A1.3.1.2. "Entrées/Sorties" en JAVA

Le modèle "d'Entrée/Sortie" de *Java* manipule des "Streams" (flots d'octets) et/ou des "Reader/Writer" (flux de caractères) : la différence entre ces deux modèles se situe simplement au niveau du format des données qui passent dans le tube [Valtech, 99]. Remarquons aussi que les classes *InputStreamReader* et *OutputStreamWriter* permettent de faire le lien entre les deux modèles, c'est-à-dire de transformer un flux de caractères en un flux de données, et vice-versa.

Le processus des entrées et des sorties est alors géré en *Java* par deux outils complémentaires appelés "canal" et "filtre". Un canal représente une source de données (fichier, *pipe*, *socket*, tableau, ...) et correspond à une interface de bas-niveau permettant la lecture et l'écriture d'un octet ou d'un caractère. Le filtre, quant à lui, se connecte ensuite sur un canal et permet d'offrir une interface de haut-niveau pour lire et écrire un entier, un flottant ou encore une chaîne complète de caractères.

Le modèle “d’Entrée/Sortie” de *Java* ressemble ainsi beaucoup à un jeu d’assemblage pour... plombier : on branche des tuyaux sur d’autres tuyaux [Valtech, 99], avec une fonctionnalité propre qui se veut avoir été attribuée à chacun de ces tuyaux.

A1.3.1.3. Recensement des processus d’entrées

L’analyse complète de l’application “*e-mail*” étudiée ici nous a permis de clairement établir que cette dernière gère différentes entrées d’informations. De plus, nous avons également réussi à établir que l’ensemble de ces entrées, de même d’ailleurs qu’en ce qui concernent les sorties décrites ensuite, peut être répertorié en trois domaines distincts.

En effet, et en ce qui concerne tout d’abord les interactions avec le serveur de “*mails*”, nous recensons successivement la connexion au serveur, le protocole de synchronisation et autres normes de transfert des “*mails*”, la liste courante des “*mails*” de l’utilisateur ainsi que l’ensemble des données constituant le “*mail*” à afficher. Si les attributs et caractéristiques paramétriques de l’utilisateur résument à elles seules la configuration et le fonctionnement interne du logiciel hôte, la gestion des événements relatifs à l’utilisateur se compose enfin de deux catégories d’entrées bien fournies, à savoir les activations respectives des différents boutons de l’interface par l’utilisateur, puis les informations et données textuelles diverses constituant le “*mail*” à envoyer.

Tout ceci non sans oublier de préciser l’absolue nécessité, pour une compilation correcte de ces nombreux apports d’informations à l’application, d’importer également initialement les classes *java.io.BufferedReader* (mémoire des caractères lors de leur lecture), *java.io.FileReader* (lecture des informations à partir d’un fichier) ainsi que *java.io.InputStreamReader* (lecture depuis une entrée standard, comme un clavier par exemple).

A1.3.1.4. Recensement des processus de sorties

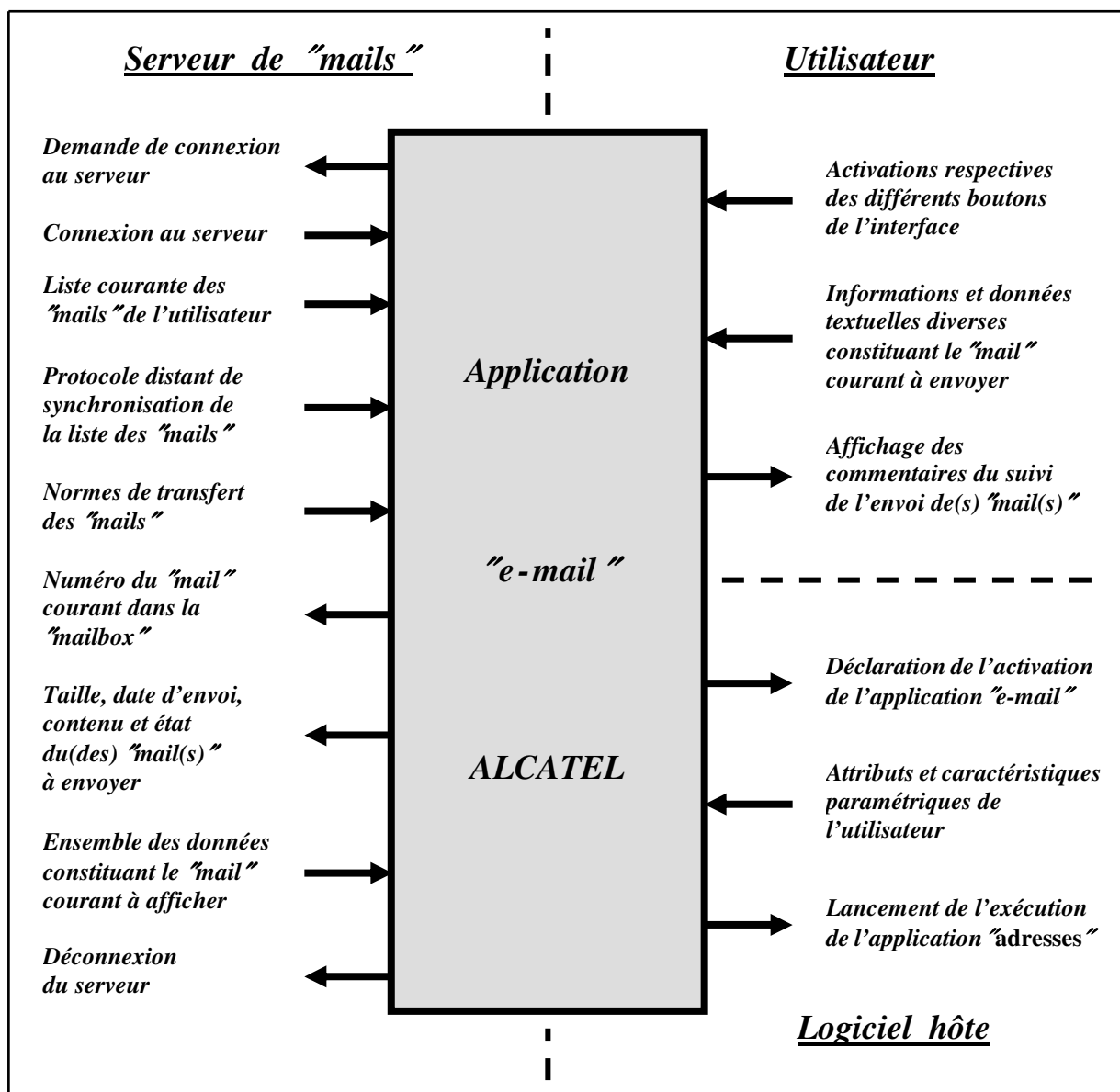
Quant à présent aux sorties de données, assimilées bien entendu à tout apport d’un résultat utilisable en entrée par d’autres applications, il ne faut pas omettre l’importation indispensable des classes *java.io.FileWriter* (écriture des informations dans un fichier) et *java.io.OutputStreamWriter* (écriture sur une sortie standard, comme un écran par exemple).

Suivant les trois mêmes domaines déjà annoncés précédemment pour les entrées, les sorties d’informations également chargées de caractériser notre application se regroupent alors, dans le cadre encore une fois des interactions avec le serveur de “*mails*”, autour des demandes de connexion et déconnexion du serveur, du numéro du “*mail*” courant dans la “*mailbox*”, ainsi que des tailles, dates d’envoi, contenus et états des “*mails*” à envoyer. D’un autre côté, c’est le logiciel hôte lui-même qui configure puis, non seulement véhicule la déclaration d’activation de l’application “*e-mail*”, mais aussi dirige le lancement de l’exécution de l’application “*adresses*”.

Finalement, chaque affichage des commentaires propres au suivi de l’envoi des “*mails*” permet de fournir, successivement, l’ensemble des événements relatifs à l’utilisateur.

A1.3.2. Schémas descriptifs du modèle de l'application "e-mail"

Nous pouvons maintenant clairement établir un premier schéma récapitulatif des nombreuses entrées et sorties de notre application "e-mail" étudiée. A noter alors que ce modèle présenté ci-dessous permet, une nouvelle fois, de bien mettre en valeur le fait que les entrées et sorties de l'application peuvent être répertoriées en trois domaines distincts (**Figure A1.3.2.1**).



- Légende :**
- - - Ligne séparatrice de deux domaines d' "entrée/sortie".
 - ← → Flèches caractérisant les différentes entrées et sorties respectives de l'application.
 - Envoi* Descriptif d'une entrée ou d'une sortie de l'application.

Figure A1.3.2.1 : Synthèse des "Entrées/Sorties" de l'application "e-mail"

Dans un second temps, voici finalement, sur les quatre pages à suivre (**Figure A1.3.2.3**), l'ensemble des détails fonctionnels exhaustifs de la structure hiérarchique de notre application "e-mail" étudiée, ce qui correspond alors exactement à la description des automates informels et indispensables au dégagement du modèle dit "d'application" recherché par nos travaux.

Cependant, il apparaît à ce niveau tout-de-même judicieux de prendre auparavant connaissance des différents formalismes et autres symboles manipulés au travers de cette figure (**Figure A1.3.2.2**).

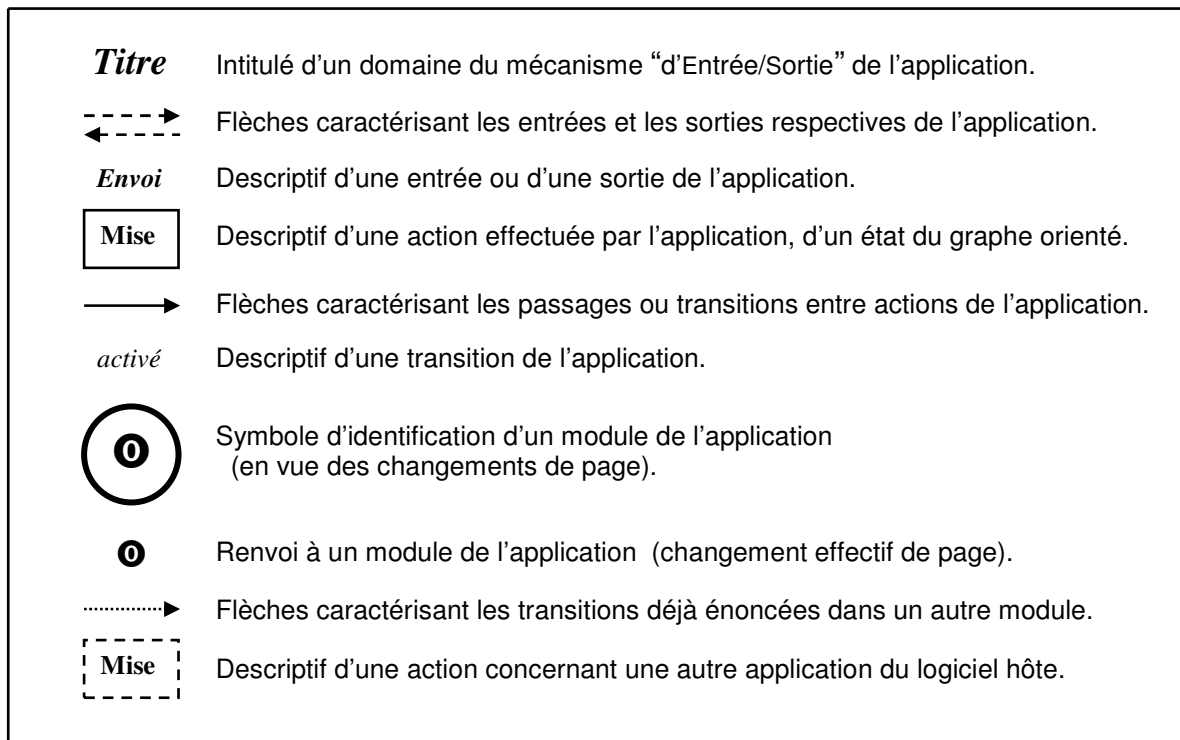


Figure A1.3.2.2 : Synthèses graphiques de la description fonctionnelle de l'application "e-mail" étudiée

En outre, concernant la présentation et la structuration globale de cette **Figure A1.3.2.3** à venir, nous pouvons également déjà constater qu'au niveau des entrées et des sorties de l'application "e-mail", celles qui sont décrites sur le côté gauche de chacune des quatre pages utilisées détaillent les interactions avec le serveur de "mails" (premier des trois domaines distincts "d'Entrée/Sortie" énoncés précédemment), alors que les côtés droits listent l'ensemble des relations demandées par l'application au sujet des deux autres domaines, à savoir les événements relatifs au logiciel hôte d'ALCATEL et à l'utilisateur concerné.

Bien entendu, et comme nous pouvons d'ailleurs très bien le constater au travers du légendage exposé ci-dessus (grâce notamment au détail des deux formalismes de changement de page), il ne faut pas non plus perdre de vue que différents liens entre ces quatre pages sont à considérer en permanence. Cela découle du fait que l'ensemble des modules de l'application étudiée, ainsi que de leurs interactions respectives, ne sont matériellement pas représentables sur une seule et même page !...

Interactions avec le serveur de "mails"

Gestion des événements relatifs au logiciel hôte et à l'utilisateur

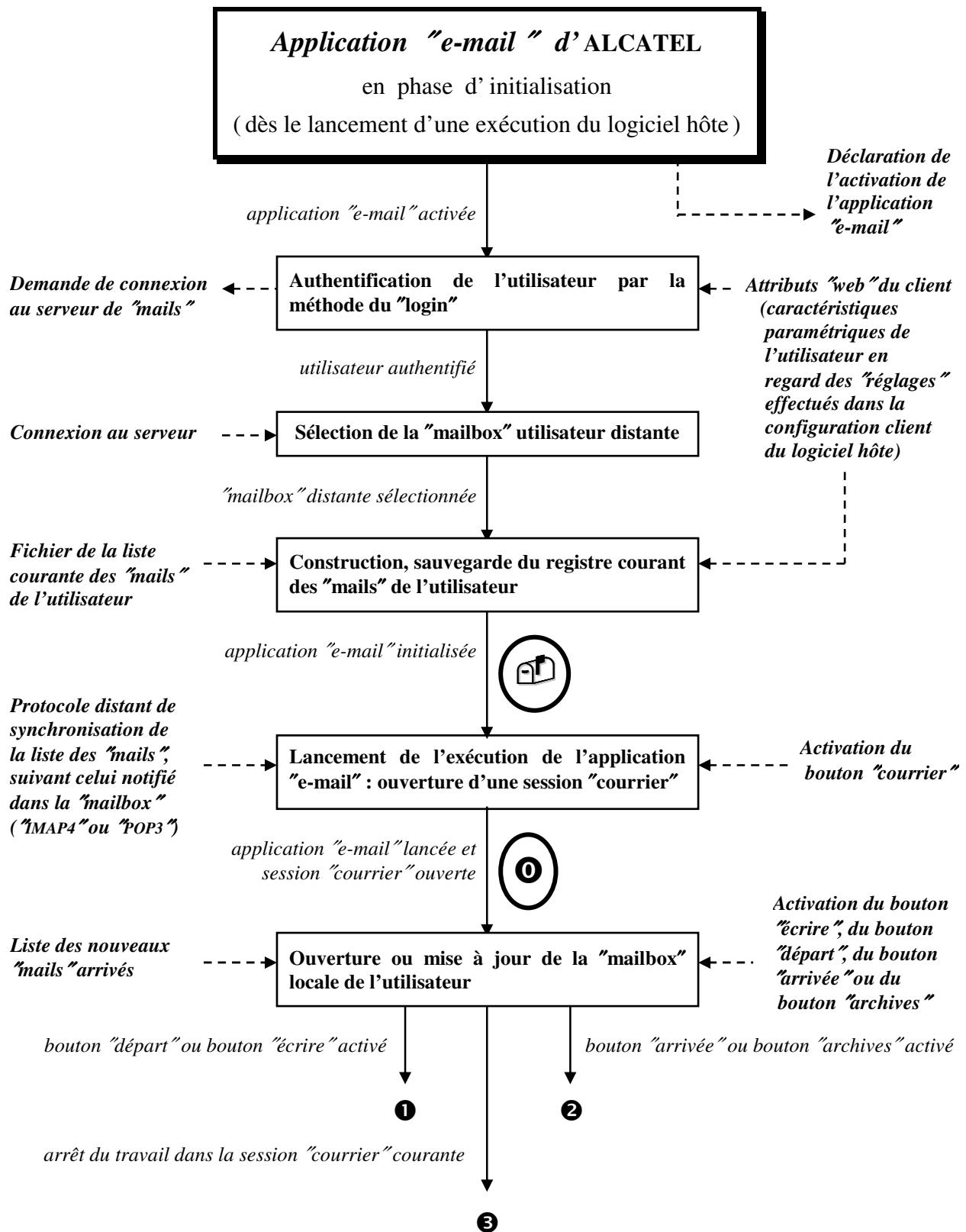


Figure A1.3.2.3 : Description fonctionnelle de l'application "e-mail"

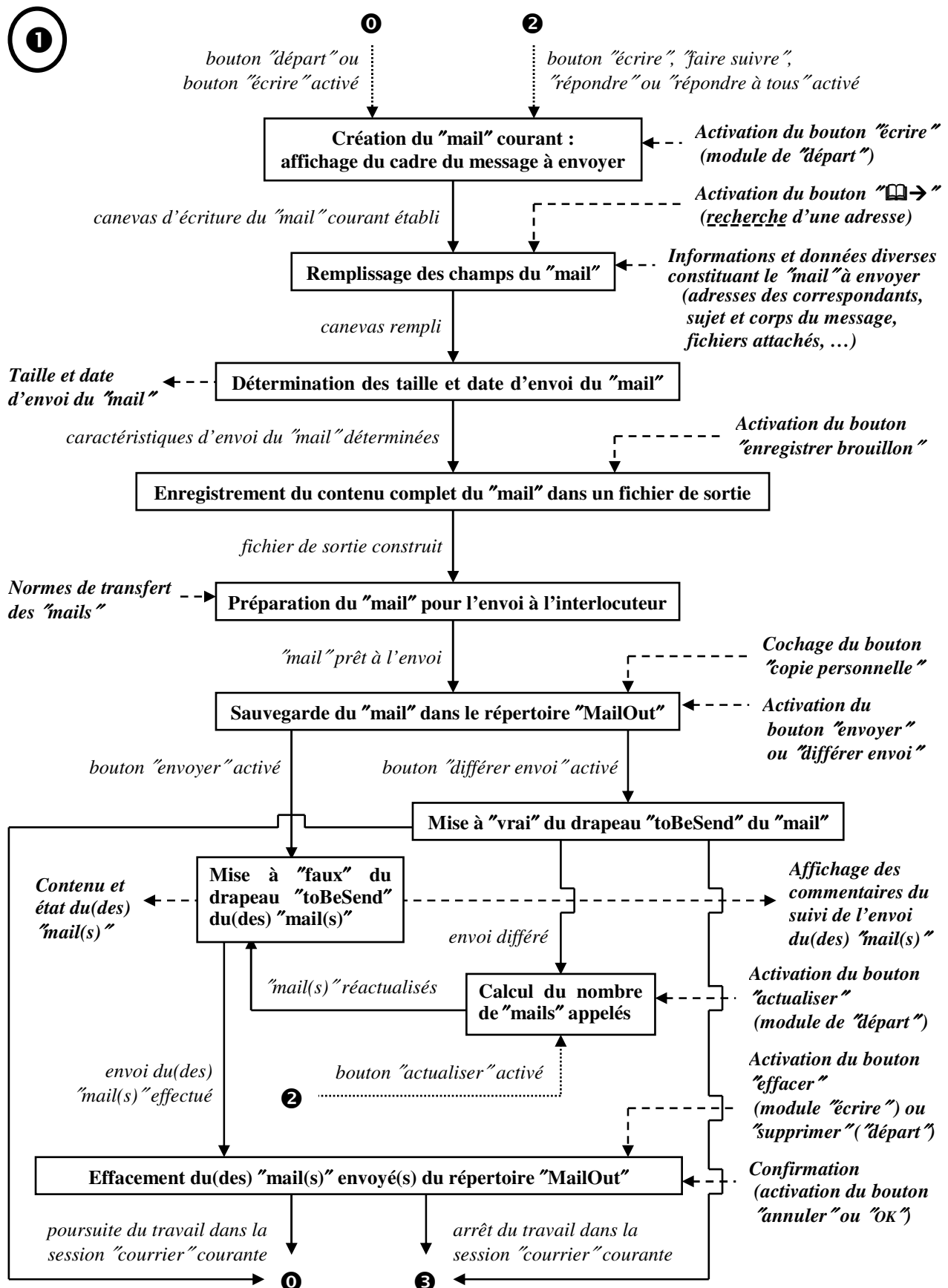


Figure A1.3.2.3 : Description fonctionnelle de l'application "e-mail"

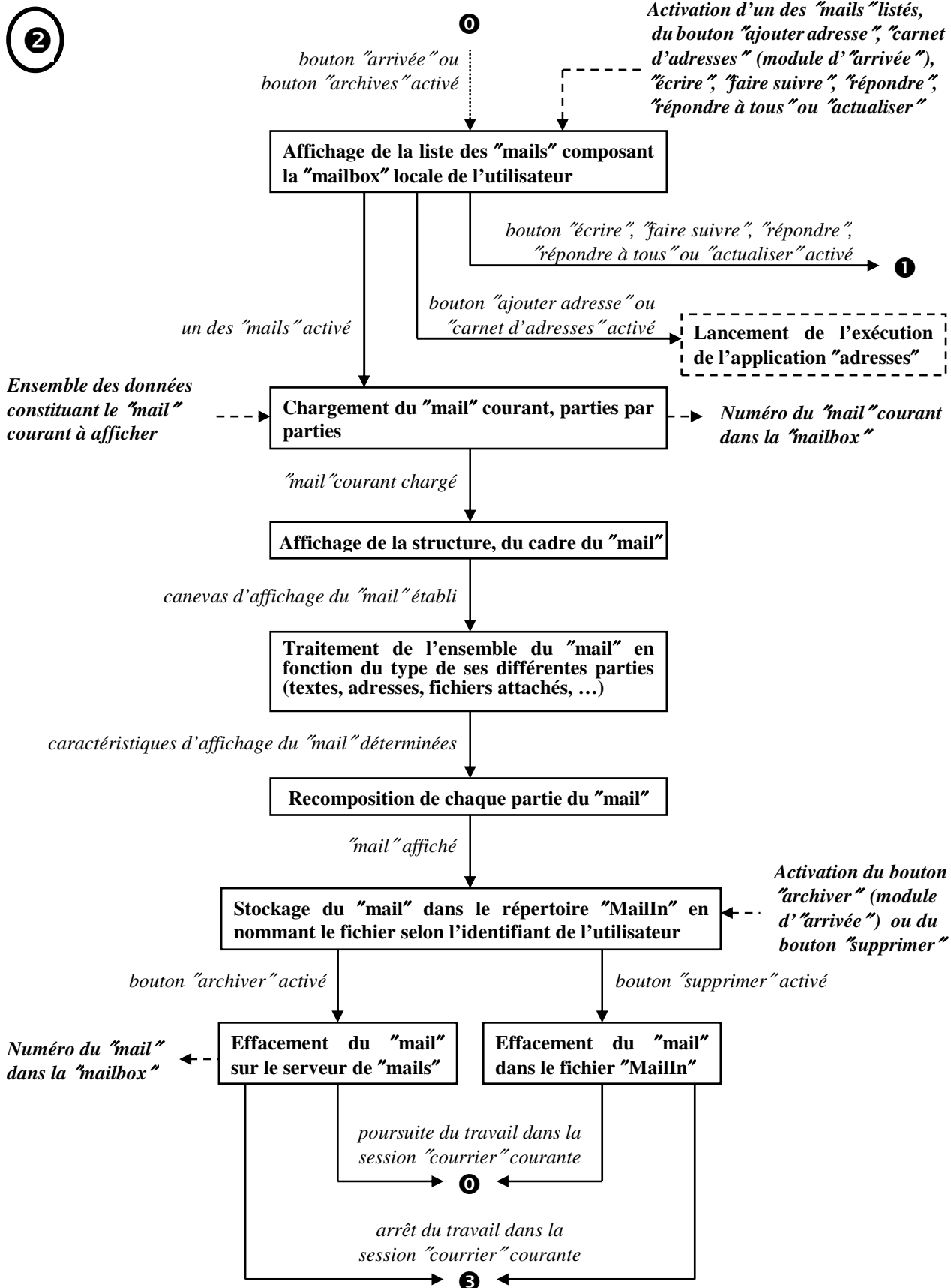


Figure A1.3.2.3 : Description fonctionnelle de l'application "e-mail"

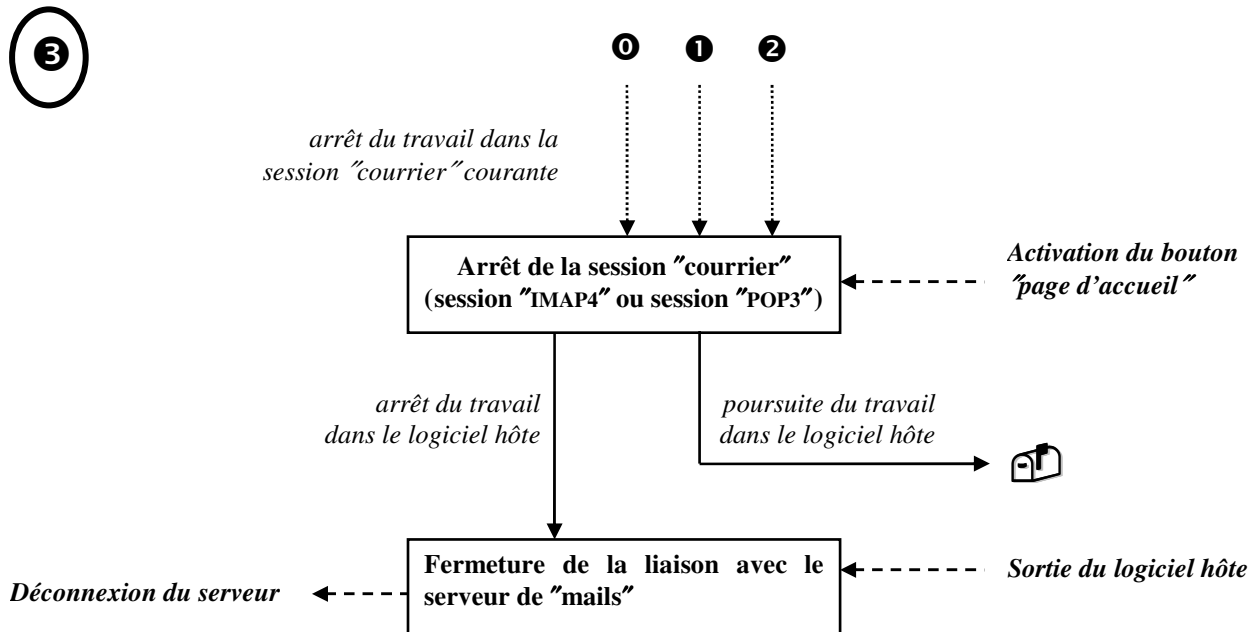


Figure A1.3.2.3 : Description fonctionnelle de l'application "e-mail"

Tout ceci établi, nous sommes maintenant à même de pouvoir encore spécifier plus précisément les différentes fonctionnalités de l'application décrites au travers de chacune des quatre dernières pages, et ce selon la manière informelle suivante :

- Page 241 : Initialisation et lancement de l'exécution de l'application.
- Page 242 : Composition et envoi d'un nouveau message.
- Page 243 : Affichage et traitement d'un message reçu.
- Page 244 : Arrêt du travail avec l'application.

Ainsi, ces ultimes précisions apportées, nous pouvons dès à présent conclure quant à l'étude initiale des premières informations indispensables à l'élaboration de notre concept d'auto-adaptation. Qualifiées de "Modèle d'Application", ces données nouvelles constituent en quelque sorte la "matière première" ou encore la "base fonctionnelle" au sein de laquelle nous allons pouvoir intégrer nos différents algorithmes d'apprentissage automatique.

A1.4. Tableaux récapitulatifs

Les deux derniers tableaux encore à suivre pour clore cette première annexe, dédiée à l'étude des terminaux et logiciels de communication ainsi qu'au dégagement de nos différents modèles d'application, sont directement rattachés à la bonne compréhension des mécanismes exposés lors du **Chapitre 4**. Ils proposent en effet, non seulement un panel représentatif des diverses données abordées et extraites par nos systèmes auto-adaptables, mais aussi un complet récapitulatif des nombreuses définitions successivement élaborées au cours de nos recherches (**Tableau A1.4.1** et **Tableau A1.4.2**).

Tableau A1.4.1 : *Exemples d'actions, de séquences d'actions, mais également de répétitions d'actions et de répétitions de séquences d'actions*

Numéro d'identification	Login de l'utilisateur	Intitulé de l'application	Exécution		Descriptif
			Date	Heure	
8898	osa	courrier	12/10/1999	08 : 06 : 16	courrier
8899	osa	courrier	12/10/1999	08 : 06 : 17	arrivée
8902	osa	courrier	12/10/1999	08 : 06 : 19	actualiser
8904	jc	courrier	12/10/1999	08 : 14 : 09	courrier
8905	osa	courrier	12/10/1999	08 : 14 : 10	\ml251101/
8906	jc	courrier	12/10/1999	08 : 14 : 58	écrire
8908	jc	courrier	12/10/1999	08 : 15 : 08	/all\
8920	osa	courrier	12/10/1999	08 : 19 : 53	répondre à tous
8921	osa	courrier	12/10/1999	08 : 19 : 59	/Re: Brainstorming\
8926	osa	courrier	12/10/1999	08 : 21 : 04	/Bonjour à tous !\
8927	jc	courrier	12/10/1999	08 : 21 : 07	/Réunion à 9h30...\
8928	osa	courrier	12/10/1999	08 : 21 : 08	/Message bien reçu :\
8930	jjk	courrier	12/10/1999	08 : 21 : 15	courrier
8931	jc	courrier	12/10/1999	08 : 21 : 18	/En salle intelligente.\
8932	osa	courrier	12/10/1999	08 : 21 : 23	/à quelle heure ?!!\
8933	jjk	courrier	12/10/1999	08 : 21 : 25	arrivée
8935	jjk	courrier	12/10/1999	08 : 21 : 26	actualiser
8936	osa	courrier	12/10/1999	08 : 21 : 27	/M'enfin !\
8937	jc	courrier	12/10/1999	08 : 21 : 30	/Jean\
8940	jc	courrier	12/10/1999	08 : 21 : 35	envoyer
8942	jjk	courrier	12/10/1999	08 : 21 : 40	\rp101002/
8944	osa	courrier	12/10/1999	08 : 24 : 00	copie personnelle
8945	osa	courrier	12/10/1999	08 : 24 : 01	envoyer
8950	osa	courrier	12/10/1999	08 : 27 : 12	actualiser
19738	jjk	courrier	23/11/1999	11 : 37 : 09	courrier
19740	jjk	courrier	23/11/1999	11 : 37 : 12	départ
19741	cge	courrier	23/11/1999	11 : 37 : 14	courrier
19742	jjk	courrier	23/11/1999	11 : 37 : 15	écrire
19745	osa	courrier	23/11/1999	11 : 37 : 16	courrier
19746	osa	courrier	23/11/1999	11 : 37 : 17	arrivée
19747	cge	courrier	23/11/1999	11 : 37 : 19	écrire
19749	jjk	courrier	23/11/1999	11 : 37 : 20	/sandel\
19750	osa	courrier	23/11/1999	11 : 37 : 29	actualiser
19751	osa	courrier	23/11/1999	11 : 37 : 30	\repedem16/
19752	osa	courrier	23/11/1999	11 : 37 : 32	répondre
19754	jjk	courrier	23/11/1999	11 : 37 : 33	/Bonjour Olivier !\
19755	cge	courrier	23/11/1999	11 : 37 : 35	/engi\

19756	osa	courrier	23/11/1999	11 : 37 : 38	/Message bien reçu :\
19758	jjk	courrier	23/11/1999	11 : 37 : 39	/Séminaire demain...\
19767	osa	courrier	23/11/1999	11 : 37 : 59	/c'est OK pour moi.\
19769	osa	courrier	23/11/1999	11 : 38 : 02	envoyer
19770	cge	courrier	23/11/1999	11 : 38 : 04	/Réunion... à 14h !!\
19771	jjk	courrier	23/11/1999	11 : 39 : 53	copie personnelle
19772	jjk	courrier	23/11/1999	11 : 40 : 05	envoyer
19775	osa	courrier	23/11/1999	11 : 40 : 08	\rep23111/\
19776	cge	courrier	23/11/1999	11 : 40 : 09	envoyer
19788	osa	courrier	23/11/1999	11 : 42 : 59	archiver
19790	jc	courrier	23/11/1999	11 : 43 : 05	courrier
19792	jc	courrier	23/11/1999	11 : 43 : 06	arrivée
19793	osa	courrier	23/11/1999	11 : 43 : 14	\repedem16/\
19795	jc	courrier	23/11/1999	11 : 43 : 15	actualiser
19796	osa	courrier	23/11/1999	11 : 43 : 18	supprimer
19798	osa	courrier	23/11/1999	11 : 43 : 29	OK
19815	jc	courrier	23/11/1999	11 : 46 : 49	\rp231131/\
56987	osa	courrier	24/11/1999	18 : 27 : 55	courrier
56988	osa	courrier	24/11/1999	18 : 27 : 58	arrivée
56992	osa	courrier	24/11/1999	18 : 29 : 53	actualiser
56995	osa	courrier	24/11/1999	18 : 30 : 05	\um241112/\
56996	osa	courrier	24/11/1999	18 : 30 : 08	ajouter adresse
57038	osa	courrier	24/11/1999	18 : 37 : 09	écrire
57039	osa	courrier	24/11/1999	18 : 47 : 49	copie personnelle
57087	osa	courrier	24/11/1999	18 : 47 : 55	envoyer
127916	osa	courrier	26/11/1999	13 : 19 : 48	courrier
127917	osa	courrier	26/11/1999	13 : 19 : 53	arrivée
127935	jc	courrier	26/11/1999	13 : 27 : 05	répondre à tous
127936	osa	courrier	26/11/1999	13 : 27 : 06	actualiser
127938	jc	courrier	26/11/1999	13 : 27 : 09	/Re: Réunion PND\
127981	jc	courrier	26/11/1999	13 : 39 : 58	copie personnelle
127987	jc	courrier	26/11/1999	13 : 40 : 55	/En salle intelligente.\
127988	osa	courrier	26/11/1999	13 : 40 : 58	écrire
127989	jc	courrier	26/11/1999	13 : 40 : 59	/Jean\
127990	jc	courrier	26/11/1999	13 : 41 : 02	envoyer
127996	jc	courrier	26/11/1999	13 : 41 : 08	actualiser
127998	osa	courrier	26/11/1999	13 : 41 : 09	retour
128015	osa	courrier	26/11/1999	13 : 41 : 15	\dr261102/\
128016	osa	courrier	26/11/1999	13 : 42 : 25	faire suivre
128018	osa	courrier	26/11/1999	13 : 42 : 48	recherche
128019	osa	courrier	26/11/1999	13 : 42 : 53	/pl, dds\
128026	osa	courrier	26/11/1999	13 : 43 : 32	/Poursuite du travail\
128075	osa	courrier	26/11/1999	13 : 54 : 43	envoyer
128078	osa	courrier	26/11/1999	13 : 54 : 55	page d'accueil

Tableau A1.4.2 : *Glossaire récapitulatif de nos diverses propriétés d'actions établies*

Intitulé(s) de la propriété	Critère(s) de notation	Référence(s) de la définition correspondante
activation d'un bouton d'interface	intitulé du bouton	4.2.3.1
désignation spatiale d'un "mail"	\ identifiant exact du "mail" /	4.2.3.1
entrée au clavier d'une donnée textuelle	/ chaîne entière de caractères \	4.2.3.1
tête, début, ouverture	<p style="text-align: center;"><i>a ; B</i></p> <p>où</p> <ul style="list-style-type: none"> - <i>a</i> ↔ action de tête, de début, d'ouverture de la séquence - <i>B</i> ↔ dernière(s) action(s) éventuelle(s) de la séquence 	<p style="text-align: center;">4.2.3.2 4.2.3.3 4.2.3.4</p>
queue, fin, fermeture	<p style="text-align: center;"><i>A ; b</i></p> <p>où</p> <ul style="list-style-type: none"> - <i>A</i> ↔ première(s) action(s) éventuelle(s) de la séquence - <i>b</i> ↔ action de queue, de fin, de fermeture de la séquence 	<p style="text-align: center;">4.2.3.2 4.2.3.3 4.2.3.4</p>
interne, parcours	<p style="text-align: center;"><i>a ; B ; c</i></p> <p>où</p> <ul style="list-style-type: none"> - <i>a</i> ↔ action de tête, de début, d'ouverture de la séquence - <i>B</i> ↔ action(s) interne(s) ou de parcours de la séquence - <i>c</i> ↔ action de queue, de fin, de fermeture de la séquence 	<p style="text-align: center;">4.2.3.2 4.2.3.3 4.2.3.4</p>
dépendance	<p style="text-align: center;"><i>A ; b ; c ; D</i></p> <p>où</p> <ul style="list-style-type: none"> - <i>A</i> ↔ première(s) action(s) éventuelle(s) de la séquence - <i>b</i> ↔ "action-mère" de <i>c</i> - <i>c</i> ↔ action dépendante de <i>b</i> - <i>D</i> ↔ dernière(s) action(s) éventuelle(s) de la séquence 	4.4.2.1
indépendance	<p style="text-align: center;"><i>a ; B</i></p> <p>où</p> <ul style="list-style-type: none"> - <i>a</i> ↔ action indépendante - <i>B</i> ↔ dernière(s) action(s) éventuelle(s) de la séquence 	4.4.2.2
dépendance transitive	<p style="text-align: center;"><i>A ; b ; C ; d ; E</i></p> <p>où</p> <ul style="list-style-type: none"> - <i>A</i> ↔ première(s) action(s) éventuelle(s) de la séquence - <i>b</i> ↔ "action-mère" - <i>C</i> ↔ action(s) interne(s) de la séquence - <i>d</i> ↔ action dépendante de manière transitive de <i>b</i> - <i>E</i> ↔ dernière(s) action(s) éventuelle(s) de la séquence 	4.4.2.3

Annexe 2 :

Spécification formelle de nos modèles auto-adaptables

En appoint au second chapitre de ce mémoire, mais également en premier complément de celui traitant de nos procédés et modèles d'apprentissage automatique mis en oeuvre tout au long de la Thèse, nous présentons ci-après l'ensemble des formalisations caractérisant nos divers scénarios, tous dédiés à notre innovant concept de l'auto-adaptation.

A l'image de l'ensemble de nos différents arguments concernant le présent sujet et déjà longuement explicités plus haut, nous nous devons alors de rappeler ici que nous avons choisi de privilégier le langage de spécification *LOTOS* (*Language Of Temporal Ordering Specification*) pour nos travaux de recherche. Nous proposons ainsi de précéder les différentes pages suivantes, entièrement orientées vers la mise en oeuvre de nos modèles formels, par un résumé, un récapitulatif initial et complet de l'ensemble des opérateurs constamment utilisés par *LOTOS*.

Et si nos fichiers de "code-source", tels que nous les avons pleinement étudiés au sein du **Chapitre 5** (se référer plus précisément à la partie **5.2**) et de la troisième annexe à ce rapport, correspondent bien évidemment à une succession d'instructions élémentaires destinées à être méticuleusement exécutées par une machine, tous ces programmes informatiques se doivent d'être au préalable décrits dans des langages formels [Ganascia, 1993]. Retenons par conséquent, en dernière introduction à la présente et très importante annexe, qu'un langage formel est défini mathématiquement à l'aide d'un "alphabet", constitué ensuite de signes et de règles de formation des expressions, elles-mêmes appelées, par analogie avec les langages naturels, des "règles de grammaire". Le tout circonscrit finalement un ensemble de mots, c'est-à-dire de séquences de signes construites conformément à ces dernières règles de grammaire.

A2.1. Récapitulatif des différents opérateurs de *LOTOS*

Comme précisé en introduction à cette annexe, voici, sur la page à venir et dans le but de pouvoir apprécier et étudier exhaustivement l'ensemble de nos spécifications formelles relatées ensuite, un résumé complet des outils mis en avant et manipulés en permanence par le langage *LOTOS*, au travers de sa syntaxe (**Tableau A2.1**).

Tableau A2.1 : *Récapitulatif de l'ensemble des opérateurs de LOTOS manipulés au cours des spécifications formelles de nos modèles auto-adaptables*

<i>Symbole</i>	<i>Nom</i>	<i>Rôle(s)</i>
?	Entrée	Récupération d'une donnée, d'une valeur
!	Sortie	Transmission d'un résultat interne obtenu
;	Préfixe d'action	Séparateur d'actions successives, séquençement d'actions
[]	Choix	Séparateur d'alternatives, disjonction d'actions
[...] →	Garde	Condition préfixée de lancement d'actions, extensible à la construction d'une liste de cas
	Intercalage	Concurrence, parallélisme sans interaction
[[...]]	Synchronisation partielle	Composition concurrente avec interaction (passage de valeurs par événements) à partir de plusieurs portes de processus
	Synchronisation totale	Parallélisme entre toutes les portes spécifiées

A2.2. Spécification formelle de nos actions auto-adaptables

Tout cela établi, et en confirmation donc de la dernière partie des recherches effectuées au **Chapitre 2** de ce mémoire, nous pouvons à présent mettre en oeuvre, sur les diverses pages à suivre, l'ensemble des formalismes inhérents à nos scénarios auto-adaptables décrits jusqu'alors (**Définition A2.2**).

Définition A2.2 : *Spécifications formelles effectives de nos scénarios auto-adaptables*

specification Assistant_Individuel_de_Telecommunication [Utilisateur, Reseau] : **noexit**

library

NaturalNumber, Boolean, String, Set

endlib

type natinfo **is** Nat, String

sorts NatInfo

opns

analyse_information (_) : String → NatInfo

endtype (* natinfo *)

type natseuil **is** Nat, String

sorts NatSeuil

opns

definition_automatique_seuil (_) : String → NatSeuil

endtype (* natseuil *)

type stringinfo **is** String, Nat

sorts StringInfo

opns

suivi_communication : → StringInfo

explication_actions , explication_statistiques : → StringInfo

recherche_information (_) , elaboration_reponse (_) : String → StringInfo

habitudes_message (_ , _) : String , String → StringInfo

ecriture_message (_ , _) : String , String → StringInfo

etablissement_statistiques (_) : Nat → StringInfo

endtype (* stringinfo *)

type stringoutil **is** String

sorts StringOutil

opns

choix_automatique_outil (_) : String → StringOutil

endtype (* stringoutil *)

type stringstock **is** String, Nat, Bool

sorts StringStock

opns

noeuds_hierarchiques (_) : StringStock → Nat

reorganisation_classification (_) : StringStock → Bool

endtype (* stringstock *)

type setinfos **is** Set, StringStock

sorts SetInfos

opns

nouveaux_messages , hierarchie_consultation : → SetInfos

modification_hierarchie_consultation (_) : String → SetInfos

modification_reponse_automatique (_) : String → SetInfos

classification_information (_ , _) : String, StringStock → SetInfos

endtype (* setinfos *)

type setoutils **is** Set, Bool, String

sorts SetOutils

opns

preferences_telematiques : → SetOutils

modification_preferences_telematiques : → SetOutils

ajout_configuration (_) , basculement_contexte (_) : SetOutils → Bool

modification_configuration (_ , _) : SetOutils , SetOutils → Bool

outils_existants (_) , outils_encore_disponibles (_) : String → SetOutils

endtype (* setoutils *)

type setapplis **is** SetInfos, Bool

sorts SetApplis

opns

assimilation_information (_ , _) : String, SetApplis → Bool

endtype (* setapplis *)

behaviour

```
(
  (
    (
      Envoi_Information [Utilisateur, Reseau]
      [] Consultation_Messages [Utilisateur]
      [] Besoin_Seuil [Utilisateur]
      [] Filtrage_Explicite [Utilisateur]
      [] Recherche_Classification [Utilisateur]
    )
    | [Utilisateur] |
    Operation [Utilisateur]
  )
  || Arrivee_Information [Utilisateur, Reseau]
)
```

where

```

process Envoi_Information [Utilisateur, Reseau] : exit :=
(
  Utilisateur ? interlocuteur :String ? information :String
  [(interlocuteur <> "") and (information <> "")]
; (
  Reseau ! outils_existants (interlocuteur) :SetOutils
  ||| Utilisateur ! preferences_telematiques :SetOutils
)
; Lancement_Communication [Utilisateur, Reseau] (interlocuteur, information)
; Utilisateur ! modification_preferences_telematiques :SetOutils

where
process Lancement_Communication [Utilisateur, Reseau]
(personne :String, message :String) : exit :=
(
  Reseau ! outils_encore_disponibles (personne) :SetOutils
; Utilisateur ! choix_automatique_outil (message) :StringOutil
; Reseau ? retour_communication :Nat
  [(retour_communication >= 0) and (retour_communication < 4)]
; (
  [retour_communication = 0]
  → Lancement_Communication [Utilisateur, Reseau]
  (personne, message)
[] [retour_communication = 1]
  → (
    Reseau ? outils_non_vocaux :Bool
    ; (
      [outils_non_vocaux = true]
      → Lancement_Communication [Utilisateur, Reseau]
      (message)
[] [outils_non_vocaux = false]
      → Elaboration_Message [Utilisateur, Reseau]
      (personne, message)
    )
  )
[] [retour_communication = 2]
  → Elaboration_Message [Utilisateur, Reseau] (personne, message)
[] [retour_communication = 3]
  → Reseau ! suivi_communication :StringInfo
)

where
process Elaboration_Message [Utilisateur, Reseau]
(categorie_personne :String, texte :String) : exit :=
(
  Utilisateur ! habitudes_message
  (categorie_personne, texte) :StringInfo
; Reseau ! ecriture_message (categorie_personne, texte) :StringInfo
) endproc (* Elaboration_Message *)
) endproc (* Lancement_Communication *)
)
endproc (* Envoi_Information *)

```

```
process Consultation_Messages [Utilisateur] : exit :=
(
  Utilisateur ! nouveaux_messages :SetInfos
; Utilisateur ! hierarchie_consultation :SetInfos
; Classement_Messages [Utilisateur]

where
  process Classement_Messages [Utilisateur] : exit :=
  (
    Utilisateur ? encore_message :Bool
  ; [encore_message = true]
    → (
      Utilisateur ? message :StringInfo
    ; Utilisateur ! modification_hierarchie_consultation
      (message) :SetInfos
    ; Utilisateur ? reponse_automatique :Bool
    ; (
      [reponse_automatique = false]
      → Classement_Messages [Utilisateur]
    [] [reponse_automatique = true]
      → (
        Utilisateur ! elaboration_reponse
          (message) :StringInfo
        ; Utilisateur ! modification_reponse_automatique
          (message) :SetInfos
        ; Classement_Messages [Utilisateur]
      )
    )
  )
)
endproc (* Classement_Messages *)
)
endproc (* Consultation_Messages *)
```

```
process Besoin_Seuil [Utilisateur] : exit :=
(
  Utilisateur ? type_seuil :String
  [type_seuil ∈ {"information", "reorganisation", "interets"}]
; (
  [type_seuil = "information"]
  → ( Utilisateur ? seuil_information :NatSeuil ! seuil_information
      [seuil_information > 0]
    [] Utilisateur ! definition_automatique_seuil (type_seuil) :NatSeuil
      [type_seuil = "information"]
  )
)
)
```

```

    [] [type_seuil = "reorganisation"]
      → ( Utilisateur ? seuil_reorganisation :NatSeuil ! seuil_reorganisation
          [seuil_reorganisation > 0]
          [] Utilisateur ! definition_automatique_seuil (type_seuil) :NatSeuil
            [type_seuil = "reorganisation"]
          )
    [] [type_seuil = "interets"]
      → ( Utilisateur ? seuil_interets :NatSeuil ! seuil_interets
          [seuil_interets > 0]
          [] Utilisateur ! definition_automatique_seuil (type_seuil) :NatSeuil
            [type_seuil = "interets"]
          )
  )
)
endproc (* Besoin_Seuil *)

```

```

process Recherche_Classification [Utilisateur] : exit :=
(
  Utilisateur ? information :String [information <> ""]
  ; Utilisateur ! recherche_information (information) :StringInfo
)
endproc (* Recherche_Classification *)

```

```

process Filtrage_Explicite [Utilisateur] : exit :=
(
  Utilisateur ? type_filtrage :String ? configuration :SetOutils
  [type_filtrage ∈ {"modification_configuration",
                    "adjonction_configuration", "changement_contexte"}]
  ; (
    [type_filtrage = "modification_configuration"]
    → ( Utilisateur ? ancienne_configuration :SetOutils
        ; Utilisateur ! modification_configuration
          (ancienne_configuration, configuration) :Bool
        )
    [] [type_filtrage = "adjonction_configuration"]
    → Utilisateur ! ajout_configuration (configuration) :Bool
    [] [type_filtrage = "changement_contexte"]
    → Utilisateur ! basculement_contexte (configuration) :Bool
  )
) endproc (* Filtrage_Explicite *)

```

```

process Operation [Utilisateur] : exit :=
(
  Utilisateur ? besoin_connaissances :Bool [besoin_connaissances = true]

```

```

; (
  ( Utilisateur ? deduction_filtrage :Bool ? configuration :SetOutils
    ; [deduction_filtrage = true]
    → Utilisateur ! basculement_contexte (configuration) : Bool
  )
  [] Utilisateur ! explication_actions :StringInfo
  [] ( Utilisateur ? accord_statistiques :Bool ? instant :Nat [instant >= 0]
    ; [accord_statistiques = true]
    → ( Utilisateur ! etablisement_statistiques (instant) :StringInfo
      ; Utilisateur ! explication_statistiques :StringInfo
    )
  )
) )
) endproc (* Operation *)

```

```

process Arrivee_Information [Utilisateur, Reseau] : exit :=
(
  Reseau ? information :String [information <> "" ]
  ; Utilisateur ! analyse_information (information) :NatInfo
  ; (
    (
      Utilisateur ? applications_concernees :SetApplis
      [applications_concernees <> ∅]
      ; Utilisateur ! assimilation_information
      (information, applications_concernees) :Bool
    )
    ||| (
      Utilisateur ? outil_stockage :StringStock [outil_stockage <> "" ]
      ; Utilisateur ! classification_information
      (information, outil_stockage) :SetInfos
      ; Gestion_Arborescence [Utilisateur] (outil_stockage)
    )
  )
)
where
process Gestion_Arborescence [Utilisateur] (support :StringStock) : exit :=
(
  Utilisateur ! noeuds_hierarchiques (support) :Nat
  ; Utilisateur ? besoin_reorganisation :Bool
  ; [besoin_reorganisation = true]
  → ( Utilisateur ! reorganisation_classification (support) :Bool
    ; Gestion_Arborescence [Utilisateur] (support)
  )
)
endproc (* Gestion_Arborescence *)
)
endproc (* Arrivee_Information *)

```

```

endspec (* Assistant_Individuel_de_Telecommunication *)

```

A2.3. Spécification du concept de tri : exemple de "l'e-mail"

Pour clore finalement cette seconde annexe attachée à l'élaboration de plusieurs de nos diverses spécifications formelles, nous nous proposons encore de revenir cette fois au niveau des importants travaux menés au **Chapitre 4** de ce mémoire, à propos notamment de nos nouveaux modèles d'application et de l'implémentation exhaustive des différents prototypes que nous avons produits. En effet, nous avons mis en oeuvre, et plus précisément au **Paragraphe 4.5.1.4**, un tout nouveau concept de tri des messages, entièrement dédié aux fonctionnalités de courrier électronique prises en exemple et que nous nous sommes chargés de rendre auto-adaptables.

C'est pourquoi, comme clairement avancé alors, voici maintenant également le détail complet des formalismes ayant permis d'entériner, de manière théorique et tout-à-fait rigoureuse, ce modèle du tri, primordial pour nos recherches actuelles et futures (**Définition A2.3**).

Définition A2.3 : *Spécifications formelles exhaustives de notre nouveau concept de tri appliqué à l'exemple du courrier électronique*

specification Tri_Email [Entree , Sortie] : **noexit**

library

NaturalNumber, Boolean, String, Set

endlib

type setinfos **is** Set

sorts SetInfos

opns

valeur_vide : → SetInfos

// unique générateur de base

endtype (* setinfos *)

type table **is** String, SetInfos, Bool, Nat

sorts Table

opns

nouvelle_table : → Table

// 1^{er} générateur de base

adjonction_table (_ , _ , _) : Table, String, SetInfos → Table

// 2^e générateur de base

adjonction_table_triee (_ , _ , _) : Table, String, SetInfos → Table

suppression_table (_ , _) : Table, String → Table

modification_table (_ , _ , _) : Table, String, SetInfos → Table

test_presence_element (_ , _) : Table, String → Bool

recherche_informations (_ , _) : Table, String → SetInfos

test_table_vide (_) : Table → Bool

test_table_triee (_) : Table → Bool

longueur_table (_) : Table → Nat

eqns**forall** $t : \text{Table}$, $i, j : \text{String}$, $x, y : \text{SetInfos}$ **ofsort** Table
$$\begin{aligned} & \text{adjonction_table_triee} (\text{nouvelle_table} , i , x) \\ & = \text{adjonction_table} (\text{nouvelle_table} , i , x) \\ j < i & \Rightarrow \text{adjonction_table_triee} (\text{adjonction_table} (t, i, x) , j , y) \\ & = \text{adjonction_table} (\text{adjonction_table} (t, i, x) , j , y) \\ j > i & \Rightarrow \text{adjonction_table_triee} (\text{adjonction_table} (t, i, x) , j , y) \\ & = \text{adjonction_table} (\text{adjonction_table_triee} (t, j, y) , i , x) \\ \\ i == j & \Rightarrow \text{suppression_table} (\text{adjonction_table} (t, i, x) , j) = t \\ i <> j & \Rightarrow \text{suppression_table} (\text{adjonction_table} (t, i, x) , j) \\ & = \text{adjonction_table} (\text{suppression_table} (t, j) , i , x) \\ \\ \text{modification_table} (t, i, x) & = \text{adjonction_table} (\text{suppression_table} (t, i) , i , x) \end{aligned}$$
ofsort Bool
$$\begin{aligned} & \text{test_presence_element} (\text{nouvelle_table} , i) = \text{false} \\ i == j & \Rightarrow \text{test_presence_element} (\text{adjonction_table} (t, i, x) , j) = \text{true} \\ i <> j & \Rightarrow \text{test_presence_element} (\text{adjonction_table} (t, i, x) , j) \\ & = \text{adjonction_table} (\text{test_presence_element} (t, j) , i , x) \\ \\ \text{test_table_vide} (\text{nouvelle_table}) & = \text{true} \\ \text{test_table_vide} (\text{adjonction_table} (t, i, x)) & = \text{false} \\ \\ \text{test_table_triee} (\text{nouvelle_table}) & = \text{true} \\ \text{test_table_triee} (\text{adjonction_table} (\text{nouvelle_table}, i, x)) & = \text{true} \\ j < i & \Rightarrow \text{test_table_triee} (\text{adjonction_table} (\text{adjonction_table} (t, i, x) , j , y)) \\ & = \text{test_table_triee} (\text{adjonction_table} (t, i, x)) \\ j > i & \Rightarrow \text{test_table_triee} (\text{adjonction_table} (\text{adjonction_table} (t, i, x) , j , y)) = \\ & \text{false} \end{aligned}$$
ofsort SetInfos
$$\begin{aligned} & \text{recherche_informations} (\text{nouvelle_table} , i) = \text{valeur_vide} \\ i == j & \Rightarrow \text{recherche_informations} (\text{adjonction_table} (t, i, x) , j) = x \\ i <> j & \Rightarrow \text{recherche_informations} (\text{adjonction_table} (t, i, x) , j) \\ & = \text{recherche_informations} (t, j) \end{aligned}$$
ofsort Nat
$$\begin{aligned} & \text{longueur_table} (\text{nouvelle_table}) = 0 \\ & \text{longueur_table} (\text{adjonction_table} (t, i, x)) = \text{longueur_table} (t) + 1 \end{aligned}$$
endtype (* table *)

behaviour

```
(  
  Adjonction_Email_Trie [ Entree , Sortie ]  
  [] Suppression_Email [ Entree , Sortie ]  
  [] Modification_Email [ Entree , Sortie ]  
  [] Test_Presence_Email [ Entree , Sortie ]  
  [] Recherche_Informations_Email [ Entree , Sortie ]  
  [] Test_Table_Vide [ Entree , Sortie ]  
  [] Test_Table_Trie [ Entrée , Sortie ]  
  [] Longueur_Table [ Entrée , Sortie ]  
)
```

where

```
process Adjonction_Email_Trie [Entree, Sortie] : exit :=  
(  
  (  
    Entree ? f :Table  
    | [Entree] |  
    Entree ? serverInput_Indice :String ? serverInput_Valeur :SetInfos  
    [(serverInput_Indice <> "") and (serverInput_Valeur <> valeur_vide)]  
  )  
  ; [ test_presence_element (f, serverInput_Indice) == false ] // nouvel email  
    → Sortie ! adjonction_table_trie (f, serverInput_Indice, serverInput_Valeur) :Table  
)  
endproc (* Adjonction_Email_Trie *)
```

```
process Suppression_Email [Entree, Sortie] : exit :=  
(  
  (  
    Entree ? f :Table  
    | [Entree] |  
    Entree ? serverInput_Indice :String [(serverInput_Indice <> "")]  
  )  
  ; [ test_presence_element (f, serverInput_Indice) == true ] // email existant  
    → Sortie ! suppression_table (f, serverInput_Indice) :Table  
)  
endproc (* Suppression_Email *)
```

```
process Modification_Email [Entree, Sortie] : exit :=  
(  
  (  
    Entree ? f :Table  
    | [Entree] |  
    Entree ? serverInput_Indice :String ? serverInput_Valeur :SetInfos  
    [(serverInput_Indice <> "") and (serverInput_Valeur <> valeur_vide)]  
  )  
  ; [ test_presence_element (f, serverInput_Indice) == true ]           // email existant  
    → Sortie ! modification_table (f, serverInput_Indice, serverInput_Valeur) :Table  
)  
endproc (* Modification_Email *)
```

```
process Test_Presence_Email [Entree, Sortie] : exit :=  
(  
  (  
    Entree ? f :Table  
    | [Entree] |  
    Entree ? serverInput_Indice :String [(serverInput_Indice <> "")]  
  )  
  ; Sortie ! test_presence_element (f, serverInput_Indice) :Bool  
)  
endproc (* Test_Presence_Email *)
```

```
process Recherche_Informations_Email [Entree, Sortie] : exit :=  
(  
  (  
    Entree ? f :Table  
    | [Entree] |  
    Entree ? serverInput_Indice :String [(serverInput_Indice <> "")]  
  )  
  ; Sortie ! recherche_informations (f, serverInput_Indice) :SetInfos  
)  
endproc (* Recherche_Informations_Email *)
```

```
process Test_Table_Vide [Entree, Sortie] : exit :=  
(  
  Entree ? f :Table  
  ; Sortie ! test_table_vide (f) :Bool  
)  
endproc (* Test_Table_Vide *)
```

```
process Test_Table_Trie [Entree, Sortie] : exit :=  
(  
  Entree ? f :Table  
  ; Sortie ! test_table_trie (f) :Bool  
)  
endproc (* Test_Table_Trie *)
```

```
process Longueur_Table [Entree, Sortie] : exit :=  
(  
  Entree ? f :Table  
  ; Sortie ! longueur_table (f) :Nat  
)  
endproc (* Longueur_Table *)
```

```
endspec (* Tri_Email *)
```

Annexe 3 :

Détails d'implémentation de tous nos modèles auto-adaptables

En totale conformité, et surtout en complément annoncé et indispensable à la première moitié de notre **Chapitre 5** étudiées auparavant dans ce mémoire, nous ne pouvions omettre à ce niveau, et ce afin de demeurer véritablement exhaustifs dans nos présentes investigations, de revenir et de préciser l'ensemble des diverses caractéristiques de programmation, significatives comme innovantes, ressortant de l'élaboration des structures de notre système intelligent et multi-agents implémenté.

Ainsi, dans ce but, nous désirons encore insister ici très fortement sur le fait que nous nous attacherons bien entendu, au cours de cette nouvelle annexe à nos différents travaux de recherche, à ne détailler que les parties les plus intéressantes et les plus attrayantes de nos programmes. Dans le même ordre d'idée, nous avons également jugé adéquat, pour d'évidentes raisons de synthèse et de pertinence, de ne mettre en avant que les propriétés les plus remarquables et les plus instructives correspondant à nos agents logiciels successifs. Tout ceci sans oublier de rappeler enfin que tous ces commentaires sont bien entendu grandement liés aux multiples et déjà très complètes explications apportées jusqu'ici dans ce rapport au sujet de l'ensemble de nos travaux d'implémentation.

A3.1. Préliminaires techniques sur les langages *COM* et *MAPI*

A3.1.1. *COM*

Comme déjà introduit au cours du **Chapitre 5** (cf. la partie **5.1.2** pour davantage de précisions à ce sujet), *COM* est la fondation sur laquelle, dans le domaine précis des messageries électroniques, les technologies de *Microsoft* sont construites. Et parce que *COM* est, à lui seul, un langage indépendant, ses composants consistent en un code exécutable distribué, comme toute bibliothèque de liaison dynamique (*DLL*, *Dynamic Link Library*) ou comme des programmes exécutables. Les composants écrits selon *COM*, norme standardisée qui plus est, satisfont ainsi à toutes les exigences d'une architecture composée : les composants *COM* se lient dynamiquement et *COM* emploie des *DLLs* pour lier les composants dynamiquement. A noter d'ailleurs à ce niveau que les standards sont particulièrement importants aujourd'hui en matière de programmation, surtout quand des parties différentes sont développées par des gens différents dans des organisations différentes oeuvrant dans des pays différents ; sans standards, rien ne travaillerait ensemble [Rogerson, 1997].

Tout ceci maintenant établi, nous remarquons en outre ici que ces concepts d'implémentation avec composants inter-connectés qu'incarne *COM* s'avèrent en définitive très intéressants puisque la technologie dans l'industrie du logiciel progresse de nos jours beaucoup trop rapidement pour laisser nos applications végéter en attente d'une mise à jour. La solution présente est donc de séparer toute application en de petites applications distinctes ou composants, ces composants étant alors ensuite assemblés en temps réel pour former les applications finales. Et de tels composants peuvent être mis à jour indépendamment les uns des autres, permettant finalement à chaque application ainsi traitée d'évoluer sans contraintes, ni de temps, ni également d'espace.

Par conséquent, et sans oublier qu'il correspond actuellement à la base de l'approche de *Microsoft* pour l'informatique distribuée en général, nous retiendrons enfin que, combiné encore de manière adéquate à un autre langage nommé cette fois-ci *MAPI*, cet outil *COM* est une très bonne façon de fournir des applications orientées objet ou des services à d'autres applications, voire de personnaliser des fonctionnalités, présentes mais aussi futures.

A3.1.2. MAPI

MAPI correspond à un composant particulier de messagerie issu du standard de l'Architecture des Systèmes Ouverts de *Windows* (*WOSA, Windows Open Systems Architecture*), créé par *Microsoft* et ses partenaires pour faire des systèmes d'exploitation de la famille *Windows* une plate-forme complète et robuste pour des applications générales et verticales.

« Chaque jour, nous comptons davantage sur l'électronique pour manipuler nos besoins de communication les plus basiques », constate effectivement Dan Fay de l'équipe *Messaging Technical* formée au sein de *Microsoft* [De la Cruz & Thaler, 1996]. Et Dan Fay d'enchaîner immédiatement en précisant que « le courrier électronique, par exemple, devient une banalité tant dans les affaires qu'à la maison. Pour assurer que cette transition aux systèmes électroniques de plus en plus complexes continue efficacement, nous devons déterminer comment intégrer les myriades de systèmes disponibles pour le consommateur. Les consommateurs demanderont la compatibilité entre leurs applications de bureau et l'infrastructure de communication électronique. La messagerie multimédia - impliquant non seulement des messages à base de texte mais aussi des images et des sons - est d'ailleurs dans la bonne voie. »

Le futur apparaît en effet brillant à ce sujet, comme nous avons pu le retenir jusqu'à présent, mais comment le développeur peut-il atteindre ses buts face à tous les standards et autres applications que chacun des systèmes de messagerie aujourd'hui indépendants ont eux-mêmes créés ? Si *SMTP* et *POP* deviennent les protocoles par défaut pour les systèmes de messagerie, comment un créateur d'une application assure-t-il que son produit travaillera sans problèmes avec ceux des autres systèmes de messagerie et continuera à travailler dans l'avenir, compte-tenu des changements de protocoles, de caractéristiques, etc ? Pour résumer, comment pouvons-nous protéger notre investissement de développement inhérent à cette Thèse ?...

La réponse nous est alors pleinement apparue en découvrant *MAPI*. Bien que *MAPI* soit un peu complexe, c'est un réel et puissant outil pour développeurs : son architecture "client/serveur" est en effet une assurance contre l'imprévisibilité de l'avenir, et *MAPI* fournit aux développeurs la capacité et la flexibilité pour créer une nouvelle génération entière d'applications de messagerie et de flux de travail. En fait, si nous pouvons noter ici que *MAPI* est compliqué parce que la messagerie en elle-même est compliquée, c'est aussi la messagerie qui est compliquée parce que les technologies sous-jacentes sont difficiles et que les demandes placées en elles sont très importantes à l'époque que nous vivons [De la Cruz & Thaler, 1996].

MAPI est ainsi une très intéressante voie standard pour la messagerie électronique, une haie contre les incertitudes de l'avenir et l'idée de codifier les interfaces entre-elles. C'est ce dont *MAPI* se caractérise pleinement : un contrat entre des applications de haut niveau, pouvant par exemple être dédiées à l'utilisateur, et des composants de bas niveau de systèmes de messagerie qui promettent un haut degré d'interopérabilité pour tout logiciel compatible. Et l'architecture y est de plus conçue pour être toujours élastique face au changement d'exigences techniques. Parce que les composants d'un tel système sont tous modulaires et leurs interfaces explicites, ils peuvent être facilement remplacés, notamment par des changements au niveau de la technologie sous-jacente qu'est *COM*, à l'image de ce que nous avons déjà étudié auparavant. *MAPI* emploie plus précisément une architecture en couches, conçue ainsi pour remplir trois exigences de principe : interopérabilité entre composants, modularité et extensibilité.

En outre, bien que *MAPI* s'appelle une "interface de programmation pour application", ce n'est pas vraiment une application dans le sens traditionnel. Les composants de messagerie que les programmeurs connaissent couramment sont en effet essentiellement des bibliothèques de fonctions liées qui peuvent être appelées par toute autre application. Néanmoins, de leur côté, les interfaces entre composants *MAPI* ne sont pas définies comme des fonctions qu'une application peut simplement appeler. Au lieu de cela, chaque composant *MAPI* se rend en effet nettement plus efficace pour nos travaux de recherche puisque chacun d'eux expose une ou plusieurs interfaces de Modèle d'Objet Composant (*COM*) et appelle par la suite des méthodes précises sur ces interfaces, très bien ciblées en permanence.

Par extension, *MAPI* reste alors "ouvert" dans le sens que des systèmes écrits à l'aide de ce langage peuvent librement communiquer avec d'autres systèmes de messagerie. Intérieurement, des composants *MAPI* parlent l'un avec l'autre au travers d'un ensemble d'interfaces soigneusement spécifiées qui définissent seulement comment les divers composants interagissent, pas comment ils sont implémentés. Parce que les données internes sont cachées, des composants s'accommodant à *MAPI* sont par conséquent hautement interopérables, permettant aux utilisateurs finaux et administrateurs de configurer facilement le poste de travail d'un utilisateur avec des combinaisons variées de composants d'un ou plusieurs fournisseurs.

Ce degré d'indépendance, bien sûr, demande que la fonction de chaque composant soit bien définie, les implémentations hautement modulaires, et les interfaces externes complètement spécifiées. Mais, bien que les interfaces externes demandent ainsi d'être explicites au maximum, les besoins fonctionnels de chaque composant *MAPI* sont suffisamment généraux pour qu'ils puissent être étendus à de nouvelles applications ou à des technologies émergentes. D'où véritablement un très grand intérêt présenté une nouvelle fois par ce concept d'implémentation qu'est *MAPI* pour l'ensemble de notre présent Modèle d'Interface Intelligente.

Encore plus techniquement, et pour décrire enfin le cœur-même de tout système *MAPI*, l'architecture *MAPI* consiste en trois couches séparées, à savoir les applications "client", le sous-système *MAPI*, et les fournisseurs de service (*service providers*) qui communiquent avec chacun des systèmes de messagerie considérés [De la Cruz & Thaler, 1996]. Ceci introduit, et dans le but d'une description vraiment efficace au sujet de ce modèle en couches, la **Figure A3.1.2** à venir sur la page suivante nous en apporte de plus un très intéressant point de vue schématique.

L'architecture *MAPI* se différencie alors entre des consommateurs de services de messagerie et des producteurs de tels services. Du côté des consommateurs se trouvent les applications "client" qui envoient et reçoivent des messages ; du côté des producteurs se distinguent plusieurs systèmes d'arrière-fond comme les boîtes-aux-lettres, les bases de données de messages, les destinataires, les réseaux ou encore les programmes d'administration. Les producteurs fournissent en conséquence des accès aux fonctionnalités du système de messagerie au travers d'une ou plusieurs bibliothèques à liaison dynamique (*DLLs* déjà abordées plus haut), propres aux fournisseurs de service et qui peuvent être installées sur la machine du côté "client". Le sous-système *MAPI*, quant à lui, est une couche entre chaque consommateur et chaque producteur qui agit comme un courtier entre les divers composants en acheminant des requêtes du client au fournisseur de service correspondant. Le client voit ainsi toujours une interface cohérente indépendamment de quelle messagerie lui est connectée en arrière-fond, et les clients communiquent avec le système d'arrière-fond par l'interface "client" du sous-système *MAPI*.

En pratique, une motivation importante pour *MAPI* est le concept de la boîte universelle d'arrivée (*universal inbox*) qui est une aire sur le bureau interactif où l'utilisateur voit et manipule toute sa correspondance électronique : *e-mail*, *fax*, *voice-mail*, planning d'information, documents, etc. De cette manière, l'architecture *MAPI* rend possible l'inter-opérabilité totale, à savoir la capacité à passer des données sans discontinuité entre différents produits de fournisseurs, mais aussi à copier des informations d'un fournisseur à un autre.

Cela établi, n'oublions pas non plus que *MAPI* permet également de réaliser des systèmes de messagerie facilement extensibles. Un nouveau comportement peut alors en effet leur être ajouté en adjoignant ou remplaçant simplement un module de fournisseur de service au lieu de remettre à jour, de réécrire, ou même, pire, de remplacer chaque système tout entier. L'extensibilité permet donc aux systèmes d'être facilement maintenus et remis à niveau avec une dépense réellement minimale.

En outre, d'un point de vue beaucoup plus orienté vers l'interfaçage homme-machine, les utilisateurs finaux éprouvent aussi moins de rupture pour une rapide mise à niveau que dans le cas d'une migration complète vers un nouveau système, ce parce que les changements peuvent être accomplis par étapes : d'abord en installant les clients, puis les fournisseurs de service, et enfin en remplaçant les composants d'arrière-fond. Nous avons d'ailleurs eu pleinement l'occasion d'expérimenter et d'apprécier l'ensemble de ces qualités propres à *MAPI* lors de l'étude-utilisateur que nous avons pû mener à terme.

Conformément à tout ce que nous avons déjà introduit auparavant, les "clients" sont les plus "hauts" placés dans la chaîne de consommation des services de messagerie schématisée ci-contre, et occupent ainsi le niveau supérieur de l'architecture *MAPI*. Tout ceci en rappelant également que ces applications sont les seuls composants avec lesquels nos utilisateurs finaux sont amenés à interagir, ce au travers d'interfaces leur étant dédiées.

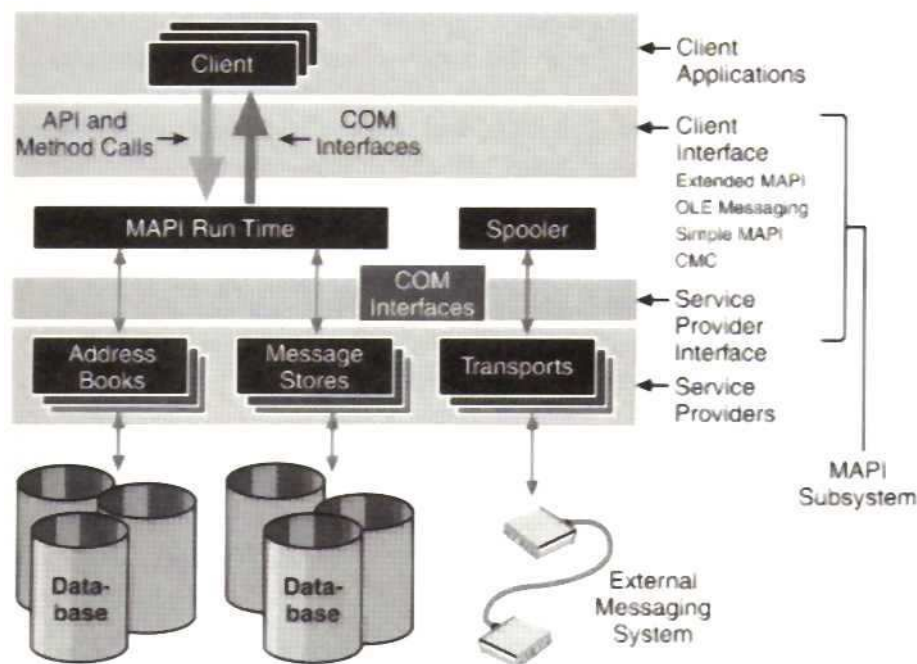


Figure A3.1.2 : Schéma détaillé de l'architecture en couches de MAPI, avec spécification, en particulier, de ses associations avec COM [De la Cruz & Thaler, 1996]

En définitive, pour être tout-à-fait complet à ce stade de nos explications, nous devons encore noter ici que MAPI prend finalement en compte trois principaux fournisseurs de service : un fournisseur de stock de messages (*message store provider*) pour créer, soumettre et stocker des messages (implémentation d'une base de données hiérarchique de répertoires et messages) ; un fournisseur de carnet d'adresses (*address book provider*) pour chercher des adresses électroniques de destinataires (implémentation de bases de données hiérarchiques de conteneurs, listes de distributions, et destinataires) ; et un fournisseur de transport (*transport provider*) pour traiter, de manière concrète, la transmission physique de messages sur plusieurs moyens.

A3.2. Déclarations et initialisation des procédures implémentées

A3.2.1. Prototypages des fonctions et méthodes inhérentes

Nous abordons cette troisième annexe avec les lignes de "code-source" correspondant simplement au premier de nos agents intelligents. Nommé "l'Agent d'Exécution" par nos soins, nous sommes bien sûr déjà rentrés, à son sujet, encore bien davantage dans les détails au cours des explications apportées au niveau du **Chapitre 5** précédemment développé dans ce mémoire.

Ainsi, nous découvrons déjà, sans plus attendre, la **Figure A3.2.1** présentée ci-dessous, destinée en priorité à nous permettre de prendre connaissance des différentes bibliothèques appelées par nos modules informatiques, ainsi que de plusieurs exemples de prototypes de fonctions intelligentes. Tous ces éléments de programmation seront bien entendu ensuite utilisés, voire même entièrement développés, par l'ensemble de notre système multi-agents.

```

#ifndef _CMDEXT_H_                                // an Application Agent's Library
#define _CMDEXT_H_

#include <WINDOWS.H>
#include <COMMCTRL.H>
#include <MAPIX.H>
#include <MAPIUTIL.H>
#include <MAPIFORM.H>
#include <INITGUID.H>
#include <MAPIGUID.H>
#include <EXCHEXT.H>

#include "RESOURCE.H"

/////////////////////////////////////////////////////////////////
// First functions prototypes

int ActIdentif (LPEXCHEXTCALLBACK);
void ActExtract (LPEXCHEXTCALLBACK, UINT, char);
void ActMemoriz (LPEXCHEXTCALLBACK, string, string, string);

void SeqAnalyse (LPEXCHEXTCALLBACK);
void SeqAnticip (LPEXCHEXTCALLBACK, FILE *, string, string, int);

OL::MAPIFolderPtr FldExtract (OL::_FoldersPtr);

```

Figure A3.2.1 : *Extrait présentant l'intégration des différentes bibliothèques ainsi que le développement de prototypes de fonctions, tous deux indispensables à l'exécution de notre Agent d'Exécution pris en exemple*

Nous pouvons alors relever, dès ce niveau de notre implémentation, que les bibliothèques dédiées aux outils *COM* et *MAPI* (cf. **5.1.2** pour davantage de précisions à ce sujet) apparaissent primordiales et s'imposent d'emblée dans nos programmations. A celles-ci s'ajoutent en outre, par extension, toutes les informations indispensables à l'ensemble des services de messagerie électronique traités, et notamment au fonctionnement externe des différents serveurs de courriers (cf. également **5.1.1**) appelés par nos modèles auto-adaptables.

En second lieu, nous constatons aussi que la présence de *MAPI* est déjà bien générale, puisque cette dernière est en effet remarquable d'emblée jusque dans les passages de paramètres des diverses fonctions de base composant nos différents agents intelligents.

A3.2.2. Déclarations des principales classes et méthodes "objet"

Nos différents agents intelligents sont, concrètement, implémentés et composés de trois classes essentielles. En effet, si la première (nommée "MyExchExt") correspond encore aux diverses fonctionnalités de base et de gestion proprement dite d'une extension logicielle dédiée (cf. 5.1.3 à ce propos), les deux suivantes sont focalisées sur le traitement complet des commandes et des événements provenant directement de l'ensemble des manipulations de l'utilisateur. Selon nos souhaits et besoins inhérents, ces classes demeurent ainsi pleinement attachées à chaque structuration ou fondement du vaste domaine de l'interfaçage homme-machine.

En notant finalement que les deux dernières de ces classes héritent aussi inévitablement de toutes les méthodes de la première (notamment puisque celle-ci est principalement orientée vers le fonctionnement et donc l'interaction de notre système multi-agents avec les applications), nous nous proposons à présent de nous plonger en détail dans la définition et la conception générale de ces trois entités primordiales (**Figure A3.2.2.1** ci-dessous et **Figure A3.2.2.2** à suivre).

```

////////////////////////////////////
// Forward declarations

class MyExchExt;
class MyExchExtCommands;
class MyExchExtUserEvents;

extern "C"
{
    LPEXCHEXT __declspec(dllexport) ExchEntryPoint(void);
}

class MyExchExt : public IExchExt
{
public:
    MyExchExt ();
    STDMETHODIMP QueryInterface
        (REFIID riid, LPVOID *ppvObj);
    inline STDMETHODIMP_(ULONG) AddRef
        () { ++m_cRef; return m_cRef; };
    inline STDMETHODIMP_(ULONG) Release
        () { ULONG ulCount = --m_cRef;
            if (!ulCount) { delete this; }
            return ulCount; };
    STDMETHODIMP Install (LPEXCHEXTCALLBACK pmechb,
        ULONG mecontext, ULONG ulFlags);

private:
    ULONG m_cRef;
    UINT m_context;

    MyExchExtCommands * m_pExchExtCommands;
    MyExchExtUserEvents * m_pExchExtUserEvents;
};

```

Figure A3.2.2.1 : *Extrait correspondant à la déclaration et à la définition des différentes méthodes "objet" composant la première classe importante de notre système intelligent et multi-agents*

```

class MyExchExtCommands : public IExchExtCommands
{
public:
    MyExchExtCommands ()
        { m_cRef = 0; m_context = 0;
          m_cmdid = 0; m_itbb = 0; m_itbm = 0; m_hWnd = 0; };
    STDMETHODCALLTYPE QueryInterface
        (REFIID riid, LPVOID * ppvObj);
    inline STDMETHODCALLTYPE AddRef
        () { ++m_cRef; return m_cRef; };
    inline STDMETHODCALLTYPE Release
        () { ULONG ulCount = --m_cRef;
          if (!ulCount) { delete this; }
          return ulCount; };
    STDMETHODCALLTYPE InstallCommands(LPEXCHEXTCALLBACK pmechb,
        HWND hwnd, HMENU hmenu,
        UINT FAR * cmdidBase, LPTBENTRY lptbeArray,
        UINT ctbe, ULONG ulFlags);
    STDMETHODCALLTYPE DoCommand(LPEXCHEXTCALLBACK pmechb, UINT mni);
    STDMETHODCALLTYPE InitMenu(LPEXCHEXTCALLBACK pmechb);
    STDMETHODCALLTYPE Help(LPEXCHEXTCALLBACK pmechb, UINT mni);
    STDMETHODCALLTYPE QueryHelpText(UINT mni, ULONG ulFlags, LPTSTR sz, UINT cch);
    STDMETHODCALLTYPE QueryButtonInfo(ULONG tbid, UINT itbb, LPTBUTTON ptbb,
        LPTSTR lpsz, UINT cch, ULONG ulFlags);
    STDMETHODCALLTYPE ResetToolbar(ULONG tbid, ULONG ulFlags);
    inline VOID SetContext
        (ULONG eecontext) { m_context = eecontext; };
    inline UINT GetCmdID() { return m_cmdid; };

private:
    ULONG m_cRef;
    ULONG m_context;
    UINT m_cmdid; // command identifier for menu extension command
    UINT m_itbb; // toolbar
    UINT m_itbm;
    HWND m_hWnd;
};

class MyExchExtUserEvents : public IExchExtUserEvents
{
public:
    MyExchExtUserEvents() { m_cRef = 0; m_context = 0;
        m_pExchExt = NULL; };
    STDMETHODCALLTYPE QueryInterface
        (REFIID riid, LPVOID * ppvObj);
    inline STDMETHODCALLTYPE AddRef
        () { ++m_cRef; return m_cRef; };
    inline STDMETHODCALLTYPE Release
        () { ULONG ulCount = --m_cRef;
          if (!ulCount) { delete this; }
          return ulCount; };
    STDMETHODCALLTYPE OnSelectionChange(LPEXCHEXTCALLBACK pmechb);
    STDMETHODCALLTYPE OnObjectChange(LPEXCHEXTCALLBACK pmechb);
    inline VOID SetContext
        (ULONG eecontext) { m_context = eecontext; };
    inline VOID SetIExchExt
        (MyExchExt * pExchExt) { m_pExchExt = pExchExt; };

private:
    ULONG m_cRef;
    ULONG m_context;
};

```

Figure A3.2.2.2 : *Extrait montrant les déclarations des différentes méthodes "objet" composant les deux classes principales de notre système multi-agents, respectivement dédiées au traitement des commandes et des événements caractérisant chacun de nos utilisateurs*

A3.2.3. Fonctions et méthodes pour le traitement global des actions

En prenant toujours comme exemple le support et l'entité complète de notre premier agent intelligent nommé "Agent d'Exécution" dont nous avons étudié au **Chapitre 5** les qualités spécifiques, nous découvrons maintenant la fonction que nous avons intitulée "*DLLMain*" et que nous avons désignée pour le chargement proprement dit de chacune de nos "Librairies de Liaison Dynamique", à savoir de chacun de nos différents agents logiciels (**Figure A3.2.3.1**).

S'en suivra immédiatement le "code" principal correspondant à un premier constructeur "objet", ici celui définissant plus précisément notre classe "*MyExchExt*" de gestion d'extension introduite précédemment (**Figure A3.2.3.2**).

```
// FUNCTION : DLLMain()
// Return Value
// TRUE - DLL successfully loads and LoadLibrary will succeed.
// FALSE - will cause an error message saying it cannot locate the extension DLL.
// Comments
// We only need to get a copy of the DLL's HINSTANCE.

BOOL WINAPI DllMain (HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    if (DLL_PROCESS_ATTACH == fdwReason)
        ghInstDLL = hinstDLL;
    return TRUE;
}
```

Figure A3.2.3.1 : *Fonction principale de notre premier agent intelligent*

```
// MyExchExt::MyExchExt ()
// Purpose
// Constructor.
// Initialize members and create supporting interface objects.
// Comments
// Each context gets its own set of interface objects.
// Furthermore, interface objects per context are kept track of
// and the interface methods are called in the proper context.

MyExchExt::MyExchExt ()
{
    m_cRef = 1; m_context = 0;
    m_pExchExtCommands = new MyExchExtCommands;
    m_pExchExtUserEvents = new MyExchExtUserEvents;
    // in "MyExchExtUserEvents" methods, I need a reference to "MyExchExt"
    m_pExchExtUserEvents->SetIExchExt (this);
}
```

Figure A3.2.3.2 : *"Code" chargé de décrire le constructeur principal de notre classe de base nommée "MyExchExt" (cf. A3.2.2 pour davantage de précisions à ce sujet)*

Ceci déjà bien établi, les deux extraits à venir, toujours pleinement issus de l'ensemble de notre "code-source", nous proposent maintenant un aperçu des méthodes directement dédiées à l'installation et à la prise en compte exhaustive des structures propres à nos différents agents intelligents.

En effet, nous découvrons tout d'abord la mise en œuvre nécessaire à l'initialisation et à la vérification d'une tenue fiable du contexte de travail courant, incluant d'emblée la première des extractions d'actions correspondant au lancement de notre "Couche IA" (**Figure A3.2.3.3**). Dans un second temps, la page suivante nous apporte tous les détails nous permettant de bien comprendre la procédure générale d'installation inhérente à chaque extension de commande, et donc indispensable aussi pour toute adjonction d'une fonctionnalité nouvelle à notre système multi-agents (**Figure A3.2.3.4**). A noter d'ailleurs encore ici que cette dernière méthode demeure en outre très importante pour garantir, et *a fortiori* valider, l'indépendance de chacun de nos modules informatiques et auto-adaptables.

```
// MyExchExt::Install()
// Parameters
//   pmechb - pointer to Extension callback function.
//   mecontext - context code at time of being called.
//   ulFlags - flag to say if install is for modal or not.
// Purpose
//   Called once for each new contexted that is entered.
//   Proper version number is checked here.
// Return Value
//   S_OK - object supported in the requested context.
//   S_FALSE - object is not supported in the requested context.

STDMETHODIMP MyExchExt::Install (LPEXCHEXTCALLBACK pmechb,
                                ULONG mecontext, ULONG ulFlags)
{
    ULONG ulBuildVersion;
    HRESULT hr;
    UINT cmdid = 0;

    m_context = mecontext;
    pmechb->GetVersion (&ulBuildVersion, EECBGV_GETBUILDVERSION);
    if (EECBGV_BUILDVERSION_MAJOR
        != (ulBuildVersion & EECBGV_BUILDVERSION_MAJOR_MASK))
        return S_FALSE;

    switch (mecontext)
    {
        case EECONTEXT_VIEWER :
        case EECONTEXT_SEARCHVIEWER :
            if (icon == false)
            {
                begin = true;
                ActExtract (pmechb, cmdid, 'o');
            }
            hr = S_OK;
            break;
        default : hr = S_FALSE;
                break;
    }
    return hr;
}
```

Figure A3.2.3.3 : *Installation de la classe gérant notre "Agent d'Exécution" et lançant notamment, dès le démarrage de notre "Couche IA", la toute première des extractions d'actions réalisées par l'utilisateur*

```

////////////////////////////////////
// MyExchExtCommands::InstallCommands()
//
// Parameters
// pmechb      - Callback Interface.
// hWnd       - window handle to main window of context.
// hMenu      - menu handle to main menu of context.
// lptbeArray - array of toolbar button entries.
// ctbe       - count of button entries in array.
// ulFlags    - reserved constant.
//
// Purpose
// This function is called when commands are installed
// for each context the extension services.
//
// Return Value
// S_FALSE means the commands have been handled.
//

STDMETHODIMP MyExchExtCommands::InstallCommands (LPEXCHEXTCALLBACK pmechb, HWND hWnd,
                                                HMENU hMenu, UINT FAR *pcmdidBase,
                                                LPTBENTRY lptbeArray, UINT ctbe,
                                                ULONG ulFlags)
{
    HRESULT hr = S_FALSE;
    HMENU hMenuTools;
    hWndMem = hWnd;

    if ((EECONTEXT_SEARCHVIEWER == m_context) || (EECONTEXT_VIEWER == m_context))
    {
        // install the new menu command, append to end of "Tools" menu
        m_hWnd = hWnd;
        pmechb->GetMenuPos (EECMDID_ToolsCustomizeToolbar, &hMenuTools, NULL, NULL, 0);
        AppendMenu (hMenuTools, MF_SEPARATOR, 0, NULL); // add a menu separator
                                                    // add our extension command
        AppendMenu (hMenuTools, MF_BYPOSITION | MF_STRING, *pcmdidBase,
                    "AI Layer - Extracted Last Actions");
        m_cmdid = *pcmdidBase;
        (*pcmdidBase)++;
    }

    if (EECONTEXT_VIEWER == m_context) // install the new toolbar button
    {
        // walk through the toolbars and find the standard toolbar
        int tbindx;
        HWND hwndToolbar = NULL;

        for (tbindx=ctbe-1 ; (int)tbindx>-1 ; --tbindx)
            if (EETBID_STANDARD == lptbeArray[tbindx].tbid)
            {
                hwndToolbar = lptbeArray[tbindx].hwnd;
                m_itbb = lptbeArray[tbindx].itbbBase++;
                break;
            }

        if (hwndToolbar) // add our button's bitmap to the toolbar's set of buttons
        {
            TBADDBITMAP tbab;
            tbab.hInst = ghInstDLL;
            tbab.nID = IDB_EXTBTN;
            m_itbm = SendMessage (hwndToolbar, TB_ADDBITMAP, 1, (LPARAM)&tbab);
        }
    }

    return hr;
}

```

Figure A3.2.3.4 : *Procédure générale d'installation de toute extension de commande ou ajout d'une fonctionnalité nouvelle à notre système multi-agents*

Pour clore cette nouvelle partie dédiée à nos travaux d'implémentation et avant tout destinée à mettre en évidence les diverses et principales charnières informatiques nécessaires à la définition des bases et de l'ensemble de la structuration de nos agents intelligents, la dernière copie d'écran exposée ci-dessous nous permet finalement de prendre connaissance d'une des méthodes déjà primordiales dans notre système (**Figure A3.2.3.5**).

En effet, cette entité est chargée plus précisément de gérer la prise en compte d'un événement particulier et, somme toute, également très fréquent, à savoir celui correspondant à un changement de sélection, par l'utilisateur courant, d'un item quelconque sur l'écran. Cette méthode apparaît bien en conséquence très importante pour la suite de nos programmations, puisque c'est notamment elle qui, grâce à une judicieuse utilisation de deux autres méthodes appelées "*QueryInterface*" et "*GetCmdId*", va nous permettre d'appréhender respectivement les interfaces de communication entre nos différentes extensions logicielles (ou agents intelligents) et les nombreux identifiants attachés aux actions et commandes réalisées par les utilisateurs. Ainsi, et comme nous l'étudierons d'ailleurs immédiatement au travers du prochain paragraphe, l'ensemble formé par ces fonctions successives vont grandement et efficacement nous assister dans l'apprentissage automatique des répétitions de séquences, mais aussi et surtout dans le traitement en temps réel et l'exécution auto-adaptable de chaque action extraite.

```
// MyExchExtUserEvents::OnSelectionChange ()
// Parameters
// pmech - pointer to Callback Object.
// Purpose
// This function is called when the selection in the UI is changed.
// Comments
// OnSelectionChange is called whenever the selection changes either
// within a pane or is changed between panes.

STDMETHODIMP_(VOID) MyExchExtUserEvents::OnSelectionChange (LPEXCHEXTCALLBACK pmech)
{
    MyExchExtCommands *pExchExtCommands = NULL;
    static LPENTRYID lpeid = NULL;
    HMENU hMenu;
    HRESULT hr;
    UINT cmdid;

    SeqAnalyse (pmech);
    if (m_context == EECONTEXT_SEARCHVIEWER)
        return;
    pmech->GetMenu (&hMenu);
    hr = m_pExchExt->QueryInterface (IID_IExchExtCommands, (LPVOID *)&pExchExtCommands);
    if (FAILED(hr))
        goto error_return;

    cmdid = pExchExtCommands->GetCmdID ();
    pExchExtCommands->Release ();
    pExchExtCommands = NULL;
    ActExtract (pmech, cmdid, 's');

error_return :
    if (lpeid != NULL)
        MAPIFreeBuffer (lpeid);
}
```

Figure A3.2.3.5 : *Méthode chargée de traiter la prise en compte, par notre système d'agents intelligents, d'un événement correspondant à un changement de sélection, par l'utilisateur courant, d'un item quelconque sur l'écran*

A3.3. Les procédés terminaux de notre "Couche IA"

A3.3.1. Les affichages

En relation directe et constante avec toutes les fonctionnalités intelligentes offertes aux utilisateurs de la "Couche IA" (cf. notamment notre schéma de la **Figure 5.1.3.2**), les affichages d'un tel système multi-agents, dédié qui plus est pleinement à l'interfaçage homme-machine, possèdent une place primordiale dans nos implémentations et engendrent une part toute aussi importante de l'ensemble du travail effectué par nos modèles auto-adaptables.

En effet, nos différents procédés d'interaction correspondent majoritairement à des mécanismes propres à la prise de connaissance et de décision, en temps réel par nos utilisateurs, des divers résultats et autres conséquences auto-adaptables, obtenus du déroulement et de l'application de nos agents intelligents à l'activité complète de chaque utilisateur sur le terminal considéré. Divisés d'emblée en deux catégories distinctes, ces procédures restent cependant toujours bien complémentaires de par, entre-autres, leurs opérations concertées en permanence. Ainsi, nous pouvons essentiellement retenir ici, non seulement la mise en œuvre d'un affichage des renseignements correspondant à la sélection par l'utilisateur lui-même des icônes ou champs de menu déroulant dédiés à notre "Couche IA" (cf. la **Figure A3.3.1.1** présentée ci-dessous avec, en supplément éventuel, la partie **A4.2.1**), mais également des affichages de propositions ou d'informations interactives, toutes apportées à l'utilisateur courant de manière bien évidemment auto-adaptable et en accord avec les différents modules de simplification, prédiction et anticipation des séquences (cf. **Figure A3.3.1.2** sur la page suivante).

```
// MyExchExtCommands::DoCommand()
// Return Value
//   Return S_OK to let the system know the command was handled.
//   Return S_OK on commands you are taking over from it.
//   Return S_FALSE to let the system know you want it to
//   carry out its command, even if you modify its action.
// Comments
//   Use this function to either respond to the command item
//   (menu or toolbar) added or modify an existing command.

STDMETHODIMP MyExchExtCommands::DoCommand (LPEXCHEXTCALLBACK pmech, UINT cmdid)
{ HRESULT hr = S_FALSE;
  if (m_cmdid != cmdid)
  { if (extr_cmd == 0)
    { ActExtract (pmech, cmdid, 'c');
      SeqAnalyse (pmech);
      return hr; }
    if ((m_context == EECONTEXT_VIEWER) || (m_context == EECONTEXT_SEARCHVIEWER))
    { ActExtract (pmech, cmdid, 'e');
      MessageBox (m_hWnd, actList.c_str(), " AI Layer - Extracted Last Actions", MB_OK);
      SeqAnalyse (pmech);
      hr = S_OK; }
    return hr; }
}
```

Figure A3.3.1.1 : *Affichage en temps réel des informations correspondant à la sélection par l'utilisateur des icônes dédiés à notre "Couche IA" (ici, code de l'icône ou de la ligne du menu déroulant attachés à notre premier "Agent d'Exécution")*

```

// auto-adaptable execution of the user current sequence of actions
if ((seuil >= seuil_predict) && (seuil < seuil_anticip))
{
    wsprintf (buffpred,
        "The following sequence : \n%s \nhas already been repeated %d times . \n\n
        Would you like this sequence to be now carried out automatically , \n
        since you have just executed the first action in this sequence ?",
        seqList.c_str(), nb_rep_sequ);

    messpred = string(buffpred);
    sht = MessageBox (m_hWnd, messpred.c_str(),
        "AI Layer - Prediction of Actions Sequence",
        MB_YESNO | MB_ICONQUESTION);
}
else
if ((seuil >= seuil_predict) && ((seuil % seuil_anticip) == 0))
{
    if (seuil == seuil_anticip)
        wsprintf (buffantic,
            "The following sequence : \n%s \nhas already been repeated %d times .
            \n\nWould you like this sequence to be now executed automatically ,\n
            as soon as the AI Layer detects the first action in this sequence ?",
            seqList.c_str(), nb_rep_sequ);

    else // permission to change opinion
    {
        wsprintf (buffantic,
            "The following sequence : \n%s \nhas already been repeated %d times .
            \n\nDo you still want this sequence to be executed automatically ?",
            seqList.c_str(), nb_rep_sequ);
        auto_anticip = true;
    }

    messantic = string(buffantic);
    sht = MessageBox (m_hWnd, messantic.c_str(),
        "AI Layer - Anticipation of Actions Sequence",
        MB_YESNO | MB_ICONEXCLAMATION);
}
}

```

Figure A3.3.1.2 : *Extrait du "code-source" nécessaire aux affichages de propositions ou informations interactives, apportées en temps réel à l'utilisateur (en accord par exemple ici avec le module de prédiction et d'anticipation des séquences)*

Par suite, un détail, de premier plan cela dit, est encore à relever de cette dernière copie d'écran avec le constat de la séparation, effectuée par nos agents intelligents au niveau des considérations respectives de nos interactions, entre prédiction ou prévision (semi-automatique avec proposition faite à l'utilisateur appelant une réponse de sa part) et anticipation, celle-ci correspondant en effet à un concept pleinement auto-adapté.

En outre, il est également à remarquer pour finir, que le premier de nos deux types d'affichage présentés dans cette partie se veut plus précisément explicite (commandé qu'il est toujours par une sélection, une réelle manipulation de l'utilisateur), alors que le second, le plus intéressant pour nous bien évidemment, demeure purement implicite (et surtout auto-adaptable) puisque proposé à l'utilisateur à chaque moment adéquat le caractérisant, de même qu'en fonction des nombreuses informations synthétisées à son égard.

A3.3.2. L'interface d'accès aux applications

En définitive, nous atteignons à ce stade de nos implémentations la toute dernière composante de notre imposante "Couche IA", à savoir celle capable finalement de rattacher et d'appliquer l'ensemble de nos travaux d'intelligence artificielle réalisés jusqu'alors aux divers logiciels traités par notre système multi-agents.

Un aperçu succinct mais bien significatif nous en est alors apporté par les deux copies d'écran encore à suivre, que ce soit dans un premier temps pour l'installation et l'initialisation d'un tel procédé d'échange informatique (**Figure A3.3.2.1** ci-contre), ou plus spécifiquement aussi dans le cadre de l'implémentation des constantes, renommages de contextes et autres intégrations de nouveaux fichiers-systèmes nécessaires à la prise en compte ainsi qu'au traitement des diverses versions successives d'une même application logicielle (**Figure A3.3.2.2** à venir sur la prochaine page).

```

#ifndef __OOMODEL_H__
#define __OOMODEL_H__

// IoutlookExtCallback
//
// Purpose:
// Interface implemented by Outlook client extension
// to access Outlook Object Model.
//

#undef INTERFACE
#define INTERFACE IOutlookExtCallback

DEFINE_GUID (IID_IOutlookExtCallback, 0x0006720D, 0, 0, 0xC0, 0, 0, 0, 0, 0, 0, 0x46);

DECLARE_INTERFACE_ (IOutlookExtCallback, IUnknown)
{
    BEGIN_INTERFACE

        // IUnknown methods
        STDMETHOD (QueryInterface) (THIS_ REFIID riid, LPVOID FAR *lppvObj) PURE;
        STDMETHOD_ (ULONG, AddRef) (THIS) PURE;
        STDMETHOD_ (ULONG, Release) (THIS) PURE;

        // IOutlookExtCallback methods
        STDMETHOD (GetObject) (THIS_ IUnknown FAR *FAR *ppunk) PURE;
        STDMETHOD (GetOfficeCharacter) (THIS_ void FAR *FAR *ppmsotfc) PURE;
    };

    typedef IoutlookExtCallback *LPOUTLEXTCALLBACK;

```

Figure A3.3.2.1 : *Déclarations de différentes méthodes "objet" composant l'interfaçage de notre système multi-agents avec les applications traitées par ce dernier*

```

// A version of the A4635 Outlook extension will be compiled for each Outlook version!
// The following predefined constants are defined in the project settings...
//     AILAYER_OL97   -> A4635OutExt97
//     AILAYER_OL98   -> A4635OutExt98
//     AILAYER_OL2000 -> A4635OutExt2000

// VC++ 5.0 "#import function" is different from VC++ 6.0...
// Use 2 differents namespaces for Office and Outlook Object Models!
#define OL_NAMESPACE    rename_namespace ("OL")
#define MSO_NAMESPACE  rename_namespace ("MSO")

#if defined (AILAYER_OL2000)
    #define OUTLOOK_VERSION    OUTLOOK_2000
    #import "C:\Program Files\Microsoft Office\Office\mso9.dll" MSO_NAMESPACE
    rename ("DocumentProperties", "DocProps")
    #import "C:\Program Files\Microsoft Office\Office\msoutl9.olb" OL_NAMESPACE
#elif defined (AILAYER_OL98)
    #define OUTLOOK_VERSION    OUTLOOK_98
    #import "C:\Tools\MsOffice\Office\mso97.dll" MSO_NAMESPACE rename ("Font",
    "IFont") rename ("DocumentProperties", "DocProps")
    #import "C:\Program Files\Microsoft Office\Office\msoutl85.olb" OL_NAMESPACE
    rename ("Font", "IFont") rename ("DocumentProperties", "DocProps")
#elif defined (AILAYER_OL97)
    #define OUTLOOK_VERSION    OUTLOOK_97
    #import "C:\WINNT\System32\stdole2.tlb" no_namespace exclude ("GUID",
    "DISPPARAMS", "EXCEPINFO", "OLE_HANDLE", "IFontDisp", "IPictureDisp",
    "IUnknown", "IDispatch", "IEnumVARIANT", "OLE_TRISTATE", "IFont", "IPicture",
    "IFontEventsDisp")
    #import "C:\WINNT\System32\fm20.dll"
    #import "C:\Tools\MsOffice\Office\mso97.dll" MSO_NAMESPACE
    #import "C:\Program Files\Microsoft Office\Office\msoutl8.olb" OL_NAMESPACE
#endif // AILAYER_OL2002 or AILAYER_OLXP too

// Alternatively, you can rename the interfaces so that
// you can build any version without changing your code :
#if defined (AILAYER_OL98)
    // These are defined differently in Outlook 2000
    // and in Outlook 98, so make them look the same.
    #define _NameSpacePtr    NameSpacePtr
    #define _FoldersPtr      FoldersPtr
    #define _ItemsPtr        ItemsPtr
    #define _ExplorerPtr     ExplorerPtr
    #define _InspectorPtr    InspectorPtr
    #define _CommandBarsPtr  CommandBarsPtr
#elif defined (AILAYER_OL97)
    // These are defined differently in Outlook 2000
    // and Outlook 97, so make them look the same.
    #define _NameSpacePtr    _INameSpacePtr
    #define _FoldersPtr      FoldersPtr
    #define _ItemsPtr        ItemsPtr
    #define _ExplorerPtr     ExplorerPtr
    #define _InspectorPtr    InspectorPtr
    #define _CommandBarsPtr  CommandBarsPtr
    #define _ApplicationPtr  _DApplicationPtr
    #define _Application     _DApplication
    #define _AppointmentItem _DAppointmentItem
    #define _ContactItem     _DContactItem
    #define _NoteItem        _DNoteItem
    #define _TaskItem        _DTaskItem
#endif // Rename

```

Figure A3.3.2.2 : *Constantes, renommages de contextes et autres intégrations de nouveaux fichiers-systèmes nécessaires au traitement par notre système multi-agents des différentes versions successives d'une même application logicielle (de messagerie électronique ici, puisqu'il s'agit plus précisément de "Microsoft Outlook")*

Annexe 4 :***Validation méthodique et exhaustive de nos prototypes***

Si les équipes de validation d'ALCATEL ont élaboré une très complète batterie "d'alpha-tests" appelant véritablement toutes les différentes facettes de l'AUM à être abordées, il nous faut préciser ici au préalable que les domaines pris en compte par nos présents travaux nous demandent de nous focaliser essentiellement sur l'étude des tests liés à l'utilisateur et plus généralement à l'interface qui lui est dévolue face à la machine : nous passons par conséquent toute exploration de validation dédiée à un travail d'administration du système.

A4.1. Intégration des méthodes de validation d'ALCATEL

Ainsi, en complément à la seconde et dernière partie du **Chapitre 5** de ce mémoire, nous pouvons à présent établir, et surtout valider, l'ensemble des tests industriels suivants appliqués à l'AUM ou ALCATEL Unified Messaging (**Tableaux A4.1.1 à A4.1.12**).

Tableau A4.1.1 : *Résultats des tests concernant les lancements d'applications*

General		
Test	Status	Remarks
Access the mailbox using <i>Outlook 97</i>	Ok	
Access the mailbox using <i>Outlook 98</i>	Ok	
Access the mailbox using <i>Outlook 2K</i>	Ok	
Access the mailbox using <i>Outlook Express</i>	Ok	
Access the mailbox using <i>Netscape Messenger</i>	Ok	

Tableau A4.1.2 : *Résultats des tests concernant l'e-mail*

Email		
Test	Status	Remarks
Send e-mail from a PC client	Ok	
Read e-mails from a PC client	Ok	
User can attach a fax in an e-mail by using standard <i>Outlook</i> image attachment	Ok	
User can attach voice message in an e-mail by using standard sound attachment	Ok	
Automatically route incoming messages by date, subject, sender or still topics with <i>Outlook</i> rules	Ok	

Tableau A4.1.3 : *Résultats des tests concernant le fax*

Fax		
Test	Status	Remarks
Send fax from <i>Outlook</i> to a fax machine	Ok	
Receive and view a fax in the <i>Outlook</i> inbox	Ok	
An e-mail to the sender notifies the status of a sent fax	Ok	
"Print-to-fax" allows faxing document from any windows application	Ok	

Tableau A4.1.4 : *Résultats des tests concernant le voice-mail*

Voice-mail with ViewMail Outlook		
Test	Status	Remarks
<i>ViewMail Outlook</i> allows listening voice-mails on PC	Ok	
Send voice-mail from a PC client	Ok	
Listen to voice-mail from a PC client using streaming	Ok	

Tableau A4.1.5 : *Résultats des tests concernant les redirections de messages*

Reply, Forward and message management		
Test	Status	Remarks
Reply to a voice-mail by a voice-mail on PC	Ok	
Reply to a voice-mail by an e-mail	Ok	
Forward a voice-mail from a PC	Ok	
Forward an e-mail with a voice attachment	Ok	
Forward an e-mail to an available fax machine	Ok	
Forward an e-mail with a fax attachment	Ok	

Tableau A4.1.6 : *Résultats des tests concernant les accès concurrents*

Concurrent Access		
Test	Status	Remarks
Leave a message to a subscriber who is consulting his mailbox	Ok	

Tableau A4.1.7 : *Résultats des tests concernant la notification*

Notification		
Test	Status	Remarks
Subscriber can set rules governing message notification	Ok	
Subscriber can configure his notification to fit specific schedules	Ok	

Tableau A4.1.8 : *Résultats des tests concernant les plannings*

Schedules		
Test	Status	Remarks
Holiday periods are set each year	Ok	
Schedules are set each day	Ok	

Tableau A4.1.9 : *Résultats des tests concernant la sécurité*

Security		
Test	Status	Remarks
Subscribers are assigned a class of service which control their feature settings and their permissions to perform specific administration functions	Ok	
Backup / Restore (NT)	Ok	

Tableau A4.1.10 : *Résultats des tests concernant les répertoires de travail*

Directory		
Test	Status	Remarks
If extension is busy, automatically transfer to the voice mailbox	Ok	
Distribution lists allow sending a same message to a previously defined set of subscribers	Ok	
Public distribution lists are available to all subscribers who are allowed to by their class of service	Ok	
Every subscriber is listed in the internal directory	Ok	
A subscriber can define personal distribution lists for his use	Ok	
Subscriber can list, add or delete recipients in his personal distribution list	Ok	
User can choose not to appear in the external directory	Ok	

Tableau A4.1.11 : *Résultats des tests concernant les langues*

Localization		
Languages		
Test	Status	Remarks
Conversations are supported in multiple languages	Ok	
One language is set as default, but subscribers can choose a preferred language to interact with the system	Ok	
Client and server in different languages	Ok	

Tableau A4.1.12 : Résultats des tests concernant nos modèles auto-adaptables

“Assistant Individuel de Télécommunication”		
Nouveaux Modèles Auto-Adaptables (Interface Intelligente)		
Test	Status	Remarks
Lancement et initialisation étendue (bases de données, menu et icônes) de tous les composants exécutables de la “Couche IA” (<i>AI Layer</i>)	Ok	
Extraction et répertoriage en temps réel des actions successives issues du travail de chaque utilisateur sur l’AUM	Ok	
Découverte et gestion des actions répétées par chaque utilisateur (statistiques d’utilisations concernant toutes les fonctionnalités de l’AUM)	Ok	
Apprentissage automatique et répertoriage des séquences d’actions effectuées de manière répétitive par chaque utilisateur	Ok	
Prédiction et anticipation des différentes séquences d’actions apprises de chacun des utilisateurs, avec exécution auto-adaptable de toutes les fonctionnalités mises à disposition par l’AUM	Ok	
Détermination et mise à jour permanente des différentes caractéristiques comportementales de chacun des utilisateurs, de même que des conséquences associées sur les paramètres de travail respectifs	Ok	
Simplification auto-adaptable de l’ensemble des séquences d’actions réalisées par chaque utilisateur sur l’AUM	Ok	

Ainsi, comme nous pouvons ici déjà le relever, les tests fonctionnels de nos prototypes, effectués sur les applications d’*ALCATEL* dans le cadre des procédures courantes de validation des logiciels de l’entreprise, se sont conclus de manière tout-à-fait positive et sans aucun accroc. Nous allons maintenant pouvoir confirmer toute cette validation de nos différents modèles auto-adaptables, ce en exposant ci-après, de manière approfondie, les “alpha-tests” mis en avant par notre **Tableau A4.1.12**.

A4.2. Mise en oeuvre des scénarios dédiés à nos modèles

A4.2.1. Lancement et initialisation de la “Couche IA”

En nous appuyant cette fois-ci sur la dernière procédure de test mise en place par les équipes de validation d’*ALCATEL* [Couch, Simon & al., 2000], nous allons à présent nous appliquer à concevoir plusieurs scénarios de tests qui vous nous permettent de valider l’ensemble des fonctionnalités implémentées dans notre nouvelle Interface Intelligente.

Toutefois, il est à noter au préalable qu’il est impératif, à ce stade de la mise en oeuvre de nos “alpha-tests”, de s’assurer que les différents répertoires de l’application sont bien dénués d’informations personnelles (messages, notes, rapports, ...) pouvant en effet modifier la généralité des scénarios et des copies d’écrans fournis dans la suite du rapport.

En outre, nous devons encore rappeler ici que nous avons choisi, comme déjà longuement discuté lors de nos derniers chapitres, de représenter toute action par son descriptif, ou intitulé, entouré de deux traits verticaux (| Folder **Inbox** | par exemple). Par extension, une séquence est ensuite composée d'actions, toutes décrites comme exposé précédemment et respectivement séparées par le symbole du point-virgule (par exemple : | Folder "**Inbox**" | ; | Message Store "**Personal Folders**" | ; | Folder "**Rapports**" |).

Ces différentes et indispensables précisions introduites, nous pouvons à présent nous pencher sur notre batterie d'expérimentations que nous allons bien entendu entamer par la vérification complète du bon déroulement des multiples initialisations associées au premier démarrage de notre logiciel (**Scénario A4.2.1 et Figure A4.2.1.1**). Nous n'oublierons pas non plus de proposer également, un peu plus tard, un second et nouveau lancement de notre prototype, après que l'ensemble de nos bases de données se soient quelque peu étoffées.

Scénario A4.2.1 : *Descriptif d'un "alpha-test" concernant le lancement et l'initialisation de la "Couche IA"*

1. Lancement de l'application *Microsoft Outlook* installée sur la machine utilisée et permettant d'accéder au serveur opérationnel de l'AUM.
2. Ouvrir le Menu "Tools" dans la *Barre des Menus* pour voir apparaître les différents champs correspondant aux principales fonctionnalités de la "Couche IA" intégrées à l'AUM.
3. Choisir le premier de ces champs intitulé "**AI Layer - Extracted Last Actions**", vérifier la nouvelle fenêtre apparue à l'écran et la fermer en cliquant sur son bouton "OK".

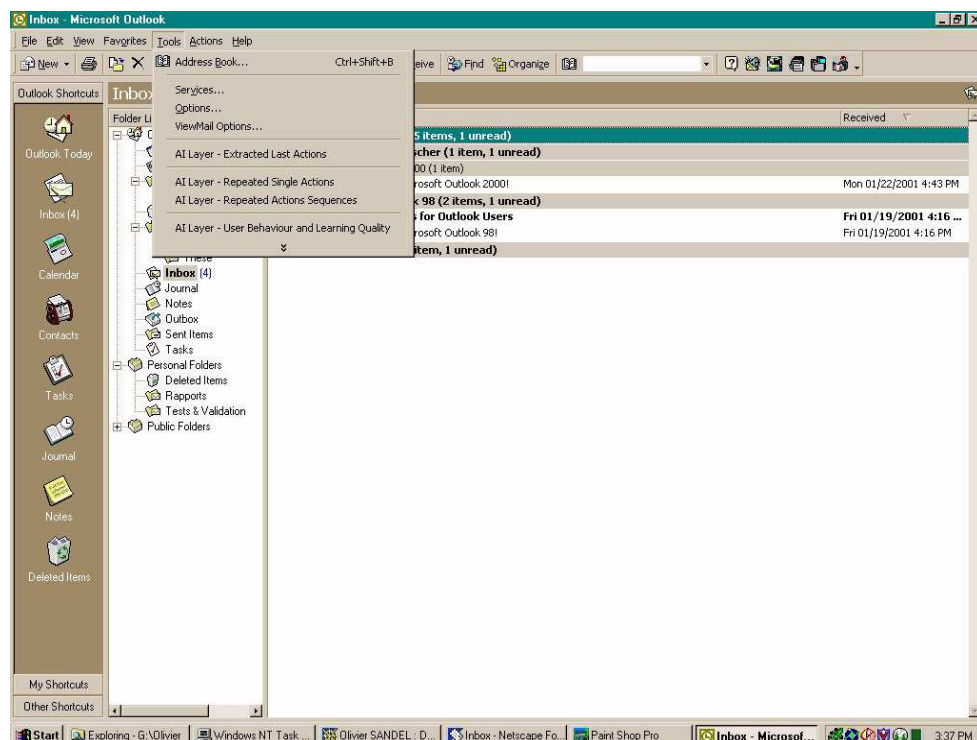


Figure A4.2.1.1 : *Développement des différents champs de menu correspondant aux principales fonctionnalités de la "Couche IA" intégrées à l'AUM*

Si l'ouverture de notre Assistant Individuel de Télécommunication ainsi réalisée autorise effectivement instantanément l'accès au menu correspondant à nos différents modèles auto-adaptables implémentés, elle permet aussi surtout de prendre connaissance de nos icônes associées à ces nouvelles fonctionnalités et dont nous rapportons ci-dessous intégralement les informations fournies aux utilisateurs après ce premier scénario (**Figures A4.2.1.2 et A4.2.1.3**).

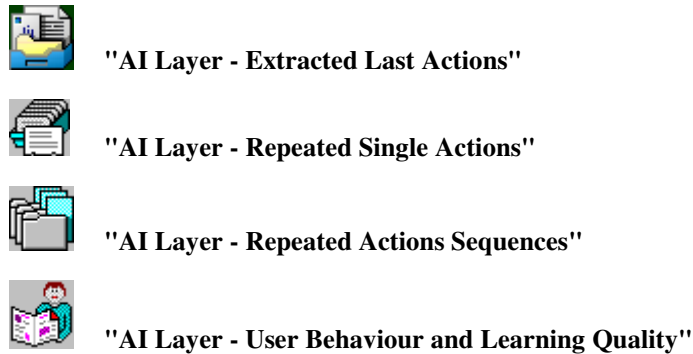


Figure A4.2.1.2 : *Représentations graphiques et descriptifs textuels des icônes attachés à la "Couche IA"*

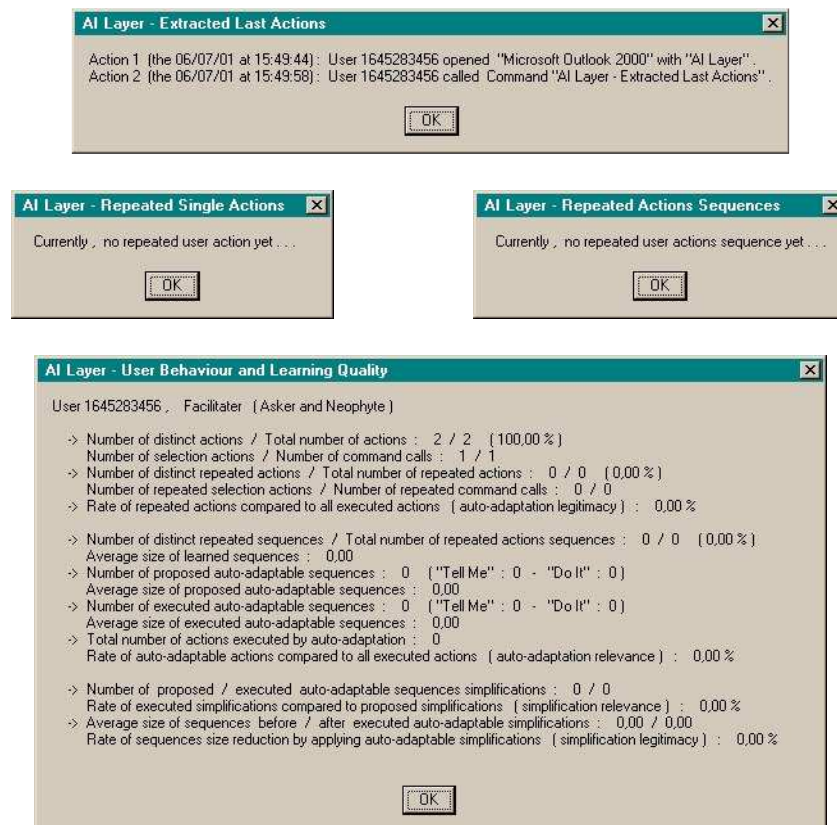


Figure A4.2.1.3 : *Résultats obtenus par nos fonctionnalités auto-adaptables suite au démarrage de la "Couche IA"*

Nous relevons alors notamment, en plus du fait que les initialisations successives se sont bien déroulées, que notre prototype a déjà correctement listé nos premières actions effectuées sur l'AUM, et ce en parfaite concordance avec le Modèle Auto-Adaptable de l'Utilisateur qui qualifie en effet très justement le nouvel utilisateur. Nous sommes à cet instant catégorisé en "Facilitant", à savoir en "Demandant" avide d'informations de toutes sortes mais également, bien entendu, en "Néophyte".

A noter encore ici, toujours au sujet de l'initialisation de la "Couche IA", que les différentes bases de données, attachées et indispensables à nos modèles auto-adaptables respectifs, se sont automatiquement créées et enregistrées dans un répertoire du système, dès ce premier lancement du prototype. Précisons également que les paramètres de cette nouvelle aire de stockage restent tout-à-fait accessibles et modifiables ultérieurement pour tout administrateur de l'AUM.

A4.2.2. Découverte et gestion des actions répétées

Après avoir ainsi réitéré la dernière opération du **Scénario A4.2.1** précédent sur tous les autres champs correspondant à notre Assistant Individuel de Télécommunication, et ce afin d'observer, à l'image de la **Figure A4.2.1.3**, les différentes informations déjà extraites par notre prototype dès le démarrage de l'AUM, nous poursuivons à présent nos investigations à l'aide des nouvelles opérations de validation ci-contre (**Scénario A4.2.2**).

Scénario A4.2.2 : *Descriptif d'un "alpha-test" concernant la gestion des actions effectuées de manière répétitive*

1. Réaliser à nouveau l'action | **Command "AI Layer - Extracted Last Actions"** | comme explicité auparavant. Vérifier que le contenu de la nouvelle fenêtre apparue à l'écran présente bien une ligne supplémentaire correspondant à cette dernière action réalisée et refermer en cliquant sur le bouton "OK".
2. Effectuer ensuite | **Command "AI Layer - Repeated Single Actions"** |, et constater que le contenu de la nouvelle fenêtre fait bien état des deux répétitions d'actions réalisées grâce au présent scénario. Refermer en cliquant sur le bouton "OK".

En effet, nous obtenons alors la confirmation souhaitée du bon fonctionnement du nouveau modèle testé ici, à savoir celui de l'extraction des actions répétées par chaque utilisateur de notre prototype (**Figure A4.2.2**).



Figure A4.2.2 : *Résultats obtenus lors du test de notre fonctionnalité de gestion des actions effectuées de manière répétitive*

A4.2.3. Apprentissage automatique des répétitions de séquences d'actions

De la même manière que jusqu'à présent, voyons maintenant quelles sont les différentes informations recueillies par notre Interface Intelligente lors de son travail au niveau de l'apprentissage automatique des répétitions de séquencements d'actions (**Scénario A4.2.3.1 et Figure A4.2.3.1**).

Scénario A4.2.3.1 : *Descriptif d'un premier "alpha-test" concernant l'apprentissage automatique des répétitions de séquences d'actions*

1. Faire | **Folder "Drafts" | ; | Message Store "Personal Folders" | ; | Folder "Outbox" |** .
2. Effectuer une seconde fois la nouvelle séquence précédente.
3. Faire ensuite | **Command "AI Layer - Extracted Last Actions" | ; | Command "AI Layer - Repeated Single Actions" |** .
4. Lancer alors l'action | **Command "AI Layer - Repeated Actions Sequences" |**, et constater que le contenu de la nouvelle fenêtre fait bien état des deux répétitions de séquences d'actions apprises de l'ensemble des actions effectuées jusqu'à présent. Refermer en cliquant sur le bouton "OK".
5. Effectuer aussi l'action | **Command "AI Layer - Repeated Single Actions" |**, et vérifier l'évolution de cette fenêtre. Refermer en cliquant sur le bouton "OK".

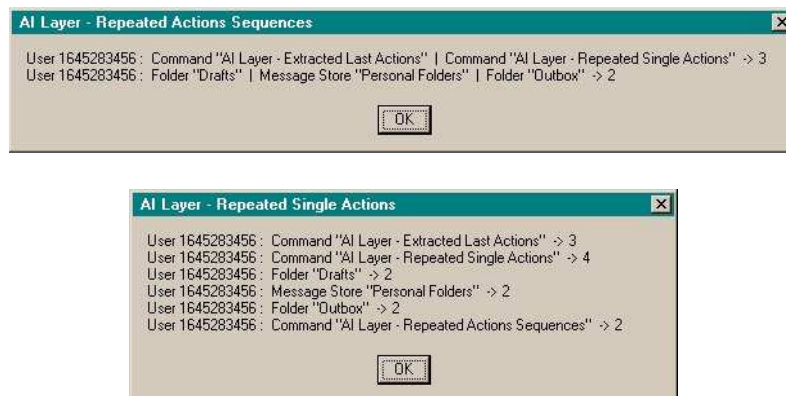


Figure A4.2.3.1 : *Résultats obtenus lors du premier test de l'apprentissage automatique des répétitions de séquences d'actions*

Ceci établi, nous pouvons cependant également remarquer ici que les séquences d'actions répétées, apprises automatiquement, répertoriées et affichées dans notre fenêtre "AI Layer - Repeated Actions Sequences" devraient en théorie être au nombre de quatre, et non pas seulement deux...

Explicitons alors pourquoi nous avons jugé nécessaire d'opérer cette amélioration.

Mais avant tout, revenons et précisons notre précédente observation.

Nous constatons en effet aisément, en récapitulant scrupuleusement l'ensemble des différentes actions qui se sont succédées depuis le début de nos présents "alpha-tests", que la séquence d'actions | Folder "Drafts" | ; | Message Store "Personal Folders" |, ainsi que celle notée | Message Store "Personal Folders" | ; | Folder "Outbox"|, ont bien été effectuées à deux reprises par nos soins, comme démontré ci-contre (**Figure A4.2.3.2**).

| "Microsoft Outlook 2000" with "AI Layer" | ; | Command "AI Layer - Extracted Last Actions" | ; | Command "AI Layer - Repeated Single Actions" | ; | Command "AI Layer - Repeated Actions Sequences" | ; | "AI Layer - User Behaviour and Learning Quality" | ; | Command "AI Layer - Extracted Last Actions " | ; | Command "AI Layer - Repeated Single Actions" | ; | Folder "Drafts" | ; | Message Store "Personal Folders" | ; | Folder "Outbox" | ; | Folder "Drafts" | ; | Message Store "Personal Folders" | ; | Folder "Outbox" | ; | Command "AI Layer - Extracted Last Actions" | ; | Command "AI Layer - Repeated Single Actions" | ; | Command "AI Layer - Repeated Actions Sequences" | ; | Command "AI Layer - Repeated Single Actions" |

Figure A4.2.3.2 : *Séquence récapitulative de l'ensemble des différentes actions effectuées lors des trois premiers scénarios de nos "alpha-tests" et extraction visuelle des séquences cachées à l'affichage*

Suite à ce constat, nous nous devons par conséquent de préciser dès à présent que, si les répétitions des précédentes séquences, composées chacune de deux actions, ne sont pas affichées dans la fenêtre mise en avant précédemment, cela correspond à un choix précis et réfléchi d'interfaçage : lors de l'implémentation du modèle dont il est question ici, nous avons en effet décidé de toujours privilégier, mais uniquement à l'affichage bien entendu, la séquence la plus grande réalisée (en terme, précisément, de nombre total d'actions contenues dans chaque séquence). Nous avons jugé cette commodité pleinement nécessaire à notre prototype afin que nos résultats issus de l'apprentissage automatique des répétitions de séquences n'apparaissent pas de manière trop touffue aux divers utilisateurs.

Dans le cas précis du **Scénario A4.2.3.1**, il apparaît effectivement inutile de montrer à l'utilisateur deux lignes successives où la seconde présente simplement une action supplémentaire rajoutée à la séquence de la ligne précédente. Cela ne fait, au regard de la grande quantité de séquences répétées que peut générer un même utilisateur en travaillant sur l'AUM, qu'accabler ce dernier avec des informations triviales et secondaires. Nous qualifierons d'ailleurs par suite ces informations de "parasites", puisque l'utilisateur sait pertinemment que, pour construire une séquence de quatre actions par exemple, cette dernière a d'abord été composée de deux actions, puis de trois et enfin de quatre : inutile donc d'afficher alors ces trois différentes étapes successives dans la fenêtre récapitulative, vu que seul le résultat le plus pertinent pour l'utilisateur mérite que l'on retienne son attention et surtout son temps.

De toutes manières, il est à noter finalement ici que, s'il nous est apparu profitable de ne pas afficher les différentes séquences parasites, elles sont bien évidemment entièrement apprises et comptabilisées par notre prototype et seront par conséquent, à leur tour, proposées à l'utilisateur en temps voulu, lorsqu'elles seront elles-mêmes les plus longues effectuées...

Tout ceci exposé, nous devons encore ajouter à ce niveau qu'il existe en fait une circonstance particulière où une séquence d'actions secondaire est tout-de-même soumise à l'affichage : c'est en effet le cas où cette séquence présente un nombre de répétitions déjà supérieur à celui de la séquence courante la plus grande.

Pour s'en convaincre, voici un scénario supplémentaire à ce sujet (**Scénario A4.2.3.2**), suivi d'une nouvelle copie d'écran corroborant nos présentes affirmations (**Figure A4.2.3.3**).

Scénario A4.2.3.2 : *Descriptif d'un "alpha-test" complémentaire concernant l'apprentissage automatique des répétitions de séquences d'actions et validant le cas particulier inhérent à l'affichage des séquences secondaires incluses dans des séquences plus grandes*

1. Réaliser | Folder "Calendar" | ; | Folder "Tasks" | ; | Folder "Outbox" |.
2. Puis | Folder "Calendar" | ; | Folder "Tasks" | ; | Message Store "Public Folders" |.
3. Et à nouveau | Folder "Calendar" | ; | Folder "Tasks" | ; | Folder "Outbox" |.
4. Appeler alors | Command "AI Layer - Repeated Actions Sequences" |, et retenir que le contenu de la fenêtre fait bien, cette fois, état de toutes les deux nouvelles répétitions de séquences apprises. Refermer en cliquant sur le bouton "OK".

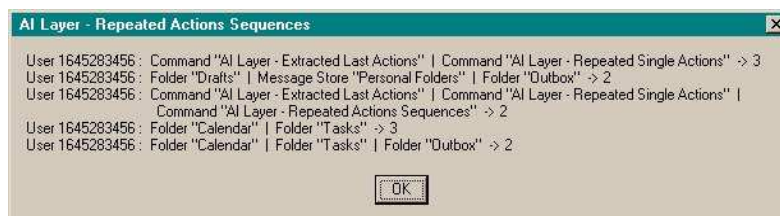


Figure A4.2.3.3 : *Résultats obtenus lors du test complémentaire de l'apprentissage automatique des répétitions de séquences d'actions*

Enfin, il reste à remarquer, toujours au sujet de la dernière copie d'écran ci-dessus, que la première séquence dont la répétition est affichée par le prototype se trouve également être incluse dans celle exposée en troisième position.

S'il est alors tout-à-fait correct, selon l'ensemble de nos principes détaillés auparavant, que ces deux différentes séquences soient répertoriées ici du fait qu'elles ne présentent pas le même nombre de répétitions, cela l'aurait été tout autant dans le cas contraire. En effet, la seconde séquence exposée par la fenêtre, et qui est donc apparue chronologiquement entre les deux séquences relatées précédemment, nous informe que la première séquence n'est pas, dans ce cas-là, une séquence "parasite" de la troisième.

A4.2.4. Prédiction auto-adaptable des séquences d'actions répétées

Conformément à la **Figure 5.1.3.1**, nous abordons à présent plus spécifiquement la validation de notre Modèle Comportemental Auto-Adaptable dédié à chacun des utilisateurs de l'AUM et de notre prototype intégré.

Nous pouvons alors tout d'abord remarquer que, en rapport avec le début de nos "alpha-tests" au **Paragraphe A4.2.1** et au vu de nos différentes manipulations qui s'en sont succédées, notre Modèle Auto-Adaptable de l'Utilisateur, que nous appelons par l'action | **Command "AI Layer - User Behaviour and Learning Quality"** |, s'est déjà chargé de modifier le statut de l'utilisateur que nous sommes.

En effet, nous sommes maintenant répertoriés en tant que "Néophyte" puisque le système nous situe, à juste titre, entre la catégorie "Promouvant" et celle de "Facilitant". Cela apparaît tout-à-fait justifié puisque nous convenons également n'avoir pas encore réalisé assez d'actions et de séquences d'actions, identiques ou non, pour passer du statut de "Facilitant" au statut de "Promouvant". Mais, nous sommes en outre suffisamment actifs du fait que nous avons déjà effectué, entre beaucoup d'autres caractéristiques prises en compte par notre prototype, 12 séquences et presque autant de sélections d'actions que d'appels de commandes, répétés ou non (**Figure A4.2.4.1**).

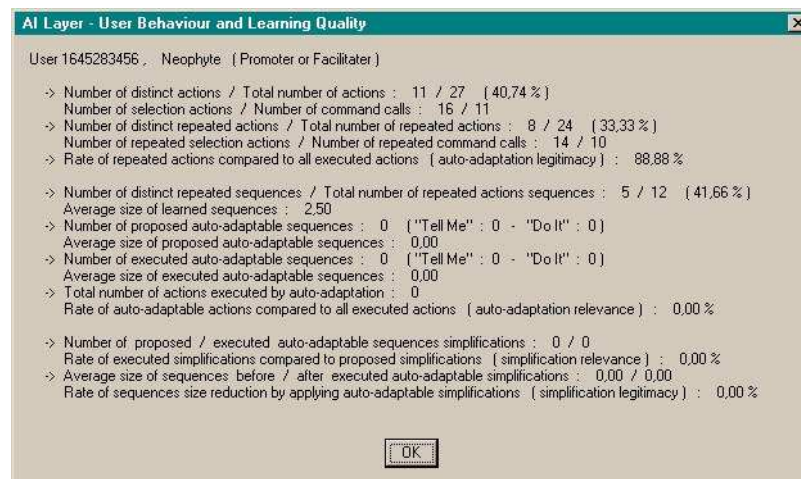


Figure A4.2.4.1 : Résultats obtenus par le Modèle Auto-Adaptable de l'Utilisateur suite aux quatre premiers scénarios composant les "alpha-tests"

Ceci établi, il va nous falloir maintenant élaborer un nouveau scénario pour découvrir le travail effectué par notre Assistant Individuel de Télécommunication à propos de la prédiction et de l'anticipation de nombreuses actions réalisées par l'utilisateur.

A ce sujet, rappelons dans un premier temps que, si nous effectuons, à plusieurs reprises, la même séquence d'actions, notre logiciel nous propose, après apprentissage automatique de ces diverses manipulations, d'exécuter lui-même, et seul, cette séquence. C'est ce que nous allons à présent vérifier au travers du scénario à venir (**Scénario A4.2.4**).

Scénario A4.2.4 : *Descriptif d'un "alpha-test" concernant la prédiction auto-adaptable du comportement de l'utilisateur*

1. Faire | Folder "Drafts" | ; | Message Store "Personal Folders" | ; | Folder "Outbox" | ; | Message Store "Public Folders" | .
2. Faire ensuite | Folder "Drafts" | ; | Message Store "Personal Folders" | ; | Folder "Outbox" | ; | Folder "Calendar" | ; | Folder "Drafts" | .
3. Valider alors l'exactitude de la nouvelle fenêtre apparue à l'écran en y répondant par l'affirmative. Vérifier l'exécution automatique, par le système, des actions successives composant la séquence plusieurs fois répétée auparavant.
4. Appeler alors l'action | Command "AI Layer - User Behaviour and Learning Quality" | , et constater que le contenu de la fenêtre a bien été mis à jour. Refermer en cliquant sur le bouton "OK".
5. Effectuer | Message Store "Personal Folders" | ; | Folder "Drafts" | . Opter encore une fois pour l'affirmative dans la nouvelle fenêtre affichée et valider l'exécution automatique de la séquence correspondante.
6. Appeler ensuite l'action | Command "AI Layer - Repeated Actions Sequences" | , et constater que le contenu de cette fenêtre a bien, lui-aussi, été correctement mis à jour. Refermer en cliquant sur le bouton "OK".
7. Sélectionner à nouveau | Folder "Drafts" | .
8. Prendre une nouvelle fois connaissance de la fenêtre apparue à l'écran, mais en y répondant cette fois-ci par la négative.
9. Faire alors de même (réponse négative) avec la nouvelle fenêtre affichée qui, suite à la réponse négative apportée précédemment, propose très justement de traiter une autre séquence, débutant bien sûr toujours par la même action courante et déjà suffisamment répétée aussi pour faire l'objet d'une exécution automatique. A noter qu'il s'agit là, plus particulièrement, d'une séquence incluse dans la précédente.
10. S'assurer que le prototype a bien compris les deux derniers souhaits et qu'il ne prend, par conséquent, aucune initiative auto-adaptable.
11. Effectuer | Folder "Contacts" | ; | Folder "Drafts" | .
12. Observer alors que le système n'affiche pas immédiatement de nouvelle proposition d'exécution automatique à propos des deux séquences traitées ci-dessus, et, par conséquent, ne devient pas trop intrusif pour l'utilisateur.

En tout premier commentaire, et si les différents résultats de ce scénario sont mis en avant par la **Figure A4.2.4.2** à venir, il nous faut d'abord préciser ici que, lorsque l'utilisateur accepte une proposition d'exécution automatique pour une séquence d'actions, les éventuelles autres séquences pour lesquelles il serait également possible d'envisager une telle exécution à ce moment précis (même action initiale) sont ignorées.

En effet, après la réalisation auto-adaptable acceptée par l'utilisateur à propos d'une séquence d'actions donnée, l'exécution automatique des autres ne présente plus aucun intérêt puisque l'action courante ne correspond alors plus à la première action composant ces séquences. Ceci est d'ailleurs très bien démontré, au cours du scénario ci-contre, par la différence de réaction dont fait preuve notre prototype entre, d'un côté, les points **3** et **5**, et puis, de l'autre, le point **9**.

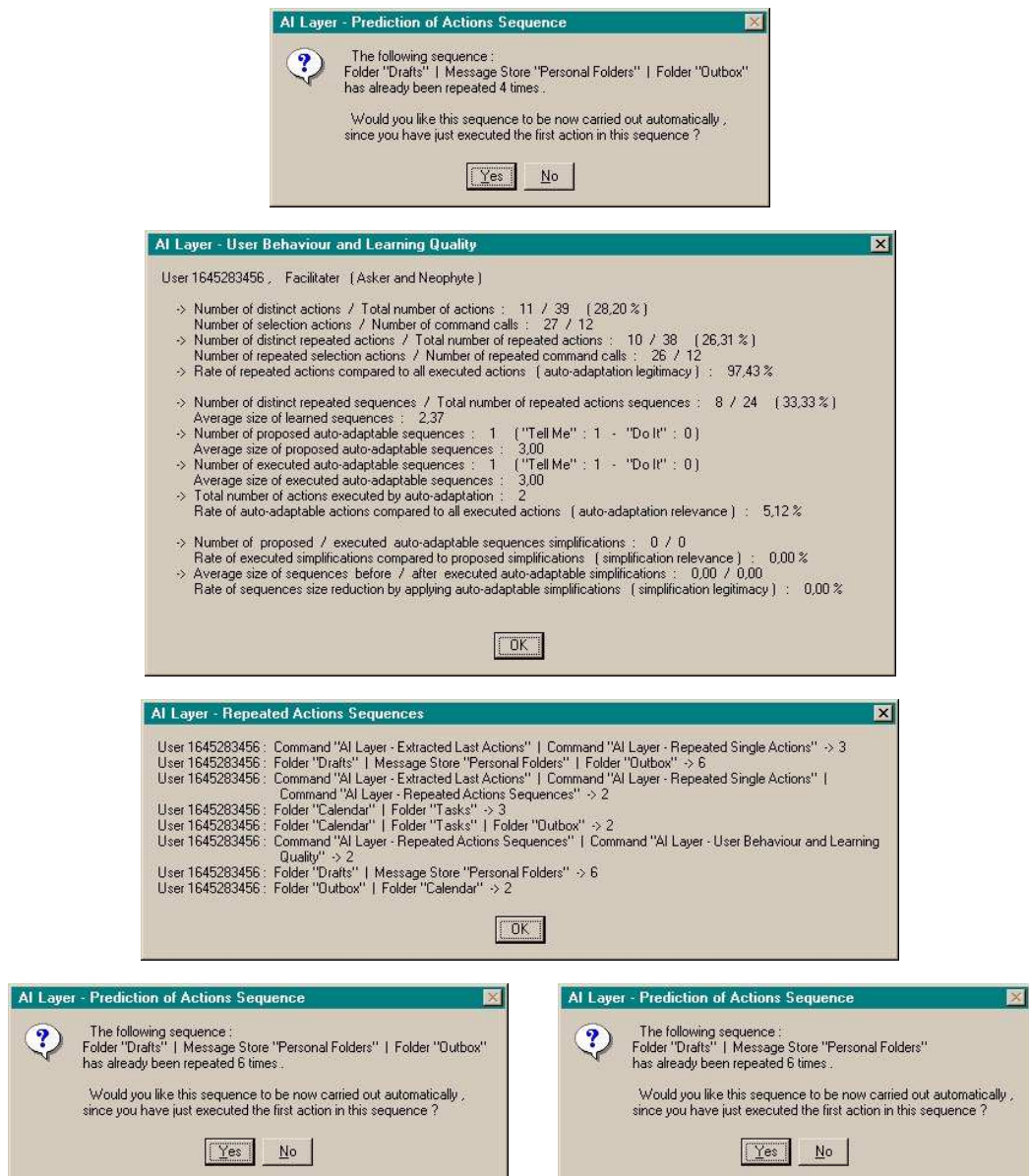


Figure A4.2.4.2 : *Copies d'écrans chronologiques montrant les résultats obtenus lors du test de la prédiction auto-adaptable du comportement de l'utilisateur*

En second lieu, nous pouvons constater que, suite à nos récentes manipulations, notre Modèle Auto-Adaptable de l'Utilisateur a réussi, au cours du précédent scénario, à très rapidement affiner la catégorisation de l'utilisateur que nous représentons.

Dès le point 3 en effet, du fait d'avoir choisi de laisser le système exécuter de manière automatique une première séquence d'actions, notre prototype a déduit que nous étions davantage coopératifs et conciliants avec lui, que dominants et affirmants : nous sommes donc, à juste titre, repassés en statut de "Facilitant". De plus, cette classification se confirme pleinement puisqu'elle corrobore également notre but actuel de découvrir, d'aborder ici une première fois toutes les différentes fonctionnalités du prototype : nous sommes par conséquent bien "Demandant" et "Néophyte".

Cela dit, le changement de catégorisation de l'utilisateur, à cet instant précis du **Scénario A4.2.4**, nous permet également de valider un point essentiel de notre prototype. En effet, l'utilisation et surtout l'exploitation de ces précédentes données apparaissent ici pour adapter au mieux la prédiction, et plus tard bien entendu aussi l'anticipation, des séquences d'actions aux diverses méthodes de travail de chacun des utilisateurs. De plus, une permanente mise à jour des caractéristiques comportementales correspondent déjà, elles-mêmes, à une intéressante fin en soit. Comme le montre par ailleurs une nouvelle fois très bien notre **Figure A4.2.4.2**, nous remarquons effectivement qu'entre les points **3** et **5** du scénario étudié ici, le seuil de déclenchement du processus de proposition d'une réalisation automatique de séquence par le système est passé de la valeur "4" à celle de "5". En redéfinissant (comme déjà explicité plus haut) le statut de l'utilisateur à la suite précisément de ce point **3**, notre Modèle Auto-Adaptable de l'Utilisateur a estimé que l'utilisateur courant, en phase de découverte assidue du logiciel, devait pouvoir bénéficier d'une marge supplémentaire de manipulation avant qu'il ne lui soit proposé d'automatiser certaines de ses opérations actuelles.

A noter alors encore à ce niveau que les différents seuils de ce nouveau système d'exploitation des données comportementales d'un utilisateur restent, une nouvelle fois, tout-à-fait accessibles et modifiables ultérieurement pour tout administrateur de l'AUM.

En conclusion, le système complet de seuillage ainsi implémenté et constamment mis à jour en fonction des souhaits successifs de l'utilisateur, mais aussi et surtout de la catégorie que le Modèle Auto-Adaptable de l'Utilisateur lui attribue, permet à cet utilisateur de ne pas être trop souvent assailli par des propositions du modèle de prédiction, ou plutôt d'être réellement sollicité de manière "intelligente", adaptée au mieux à ses propres besoins. Par conséquent, et notamment dans le cas précis du dernier scénario, notre Assistant Individuel de Télécommunication fera une nouvelle proposition d'exécution automatique de la séquence d'actions | Folder "Drafts" | ; | Message Store "Personal Folders" | ; | Folder "Outbox" | lorsque l'utilisateur aura, à nouveau, effectué plusieurs fois la même séquence (mais certes pas autant que pour la première fois).

C'est d'ailleurs, en partie, ce que nous allons développer et valider à présent.

A4.2.5. Anticipation auto-adaptable des séquences d'actions fréquentes

Nous aboutissons maintenant à la prise en compte du second volet de notre Modèle Comportemental Auto-Adaptable, déjà quelque peu introduit auparavant.

Il s'agit à présent, en effet, de justifier puis de vérifier le lancement de notre processus d'anticipation auto-adaptable. Lié bien entendu à une répétition relativement fréquente par l'utilisateur de certaines séquences d'actions, ce mécanisme devient aussi surtout indispensable lorsque cet utilisateur a, par le passé, plusieurs fois souhaité que la machine effectue ces séquences respectives à sa place, de manière automatique.

Dans cette optique, nous pouvons alors mettre en avant les deux scénarios à venir (**Scénario A4.2.5.1** et **Scénario A4.2.5.2**), séries de tests qui s'ajoutent aux précédentes déjà décrites dans ce rapport et pour lesquelles nous continuons de préciser les différents résultats significatifs obtenus (**Figure A4.2.5.1** et **Figure A4.2.5.2**).

Scénario A4.2.5.1 : *Descriptif d'un premier "alpha-test" concernant l'anticipation auto-adaptable du comportement de l'utilisateur*

1. Faire | **Message Store "Personal Folders"** | ; | **Folder "Outbox"** | ; | **Folder "Drafts "** | . Répondre par l'affirmative à la nouvelle proposition d'exécution automatique affichée par le système et vérifier cette dernière.
2. Faire ensuite | **Message Store "Public Folders"** | ; | **Folder "Drafts"** | , et répondre toujours au système par l'affirmative.
3. Sélectionner à nouveau | **Folder "Drafts"** | . Répondre par l'affirmative.
4. Faire | **Folder "Tasks"** | ; | **Folder "Drafts"** | . Répondre par l'affirmative.
5. Faire | **Folder "Calendar"** | ; | **Folder "Outbox"** | ; | **Folder "Calendar"** | ; | **Folder "Drafts"** | . Répondre par l'affirmative.
6. Sélectionner | **Folder "Drafts"** | . Répondre par l'affirmative.
7. Faire aussi | **Command "AI Layer - User Behaviour and Learning Quality"** | ; | **Folder "Drafts"** | . Répondre par l'affirmative.
8. Faire encore | **Message Store "Public Folders"** | ; | **Folder "Drafts"** | , et répondre par l'affirmative à la nouvelle fenêtre afin de déclencher le processus d'anticipation auto-adaptable correspondant.
9. Sélectionner alors simplement | **Folder "Drafts"** | , et vérifier que l'exécution de la séquence | **Folder "Drafts "** | ; | **Message Store "Personal Folders"** | ; | **Folder "Outbox"** | est maintenant réalisée par le système sans plus aucune demande d'acquiescement de sa part.
10. Confirmer enfin ce mécanisme d'anticipation auto-adaptable par | **Command "AI Layer - Repeated Actions Sequences"** | ; | **Folder "Drafts"** | , en remarquant en outre que nous commençons réellement à voir apparaître des séquences de tailles suffisamment intéressantes pour justifier l'élaboration de l'ensemble de nos Modèles Auto-Adaptables dédiés à l'interface homme-machine.

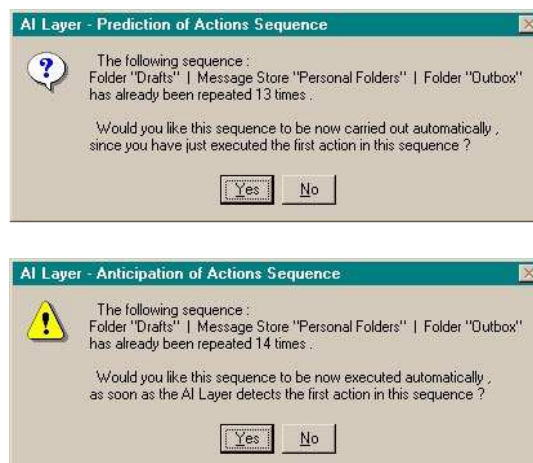


Figure A4.2.5.1 : *Copies d'écrans montrant chronologiquement les résultats obtenus lors du passage de la prédiction à l'anticipation auto-adaptable du comportement de l'utilisateur*

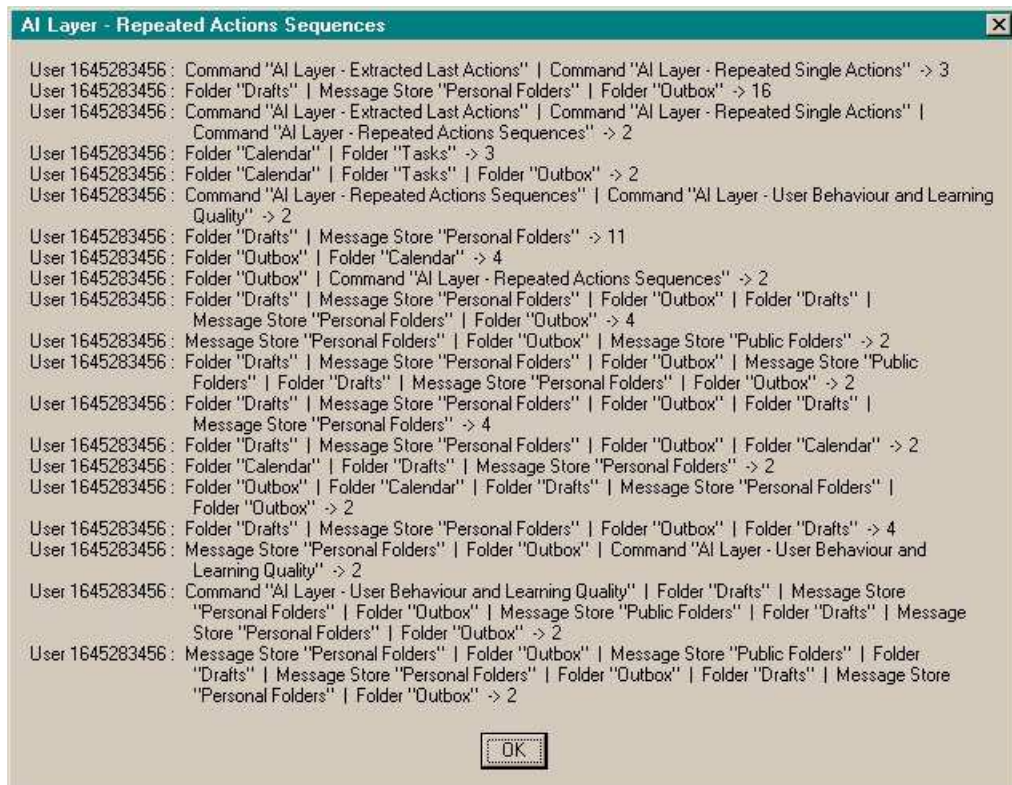


Figure A4.2.5.2 : *Etat des premiers résultats obtenus au niveau de l'apprentissage automatique des séquences répétées lors du test de l'anticipation auto-adaptable du comportement de l'utilisateur*

Cependant, le principe de notre présent modèle d'anticipation auto-adaptable ne peut se limiter à l'étape ci-dessus.

En effet, et conformément à une de nos principales préoccupations déterminées dès le tout début de la Thèse, notre prototype se doit également de se remettre constamment en question afin qu'il ne devienne pas trop intrusif et contrôlant vis-à-vis de ses utilisateurs. Par conséquent, nous l'avons doté d'une capacité à offrir à chaque utilisateur les possibilités, après que la machine ait effectué un certain nombre de fois une même séquence de manière auto-adaptable, de reprendre le contrôle des opérations et de décider si, oui ou non, il désire poursuivre le processus de traitement et d'anticipation auto-adaptable de la séquence concernée. De cette manière, nous tenons réellement à ne pas définitivement abandonner les consultations du "Tell Me" lorsque nous passons en mode "Do It" : nous nous devons de prendre en considération un éventuel changement d'avis de la part de chacun des utilisateurs.

C'est pourquoi, nous présentons ci-après un scénario complémentaire du présent "alpha-test" (**Scénario A4.2.5.2**), suivi des résultats correspondants et chargé de nous permettre de valider le côté non invasif de notre mécanisme d'anticipation auto-adaptable (**Figure A4.2.5.3**).

Scénario A4.2.5.2 : *Descriptif d'un "alpha-test" supplémentaire concernant l'anticipation auto-adaptable du comportement de l'utilisateur et traitant du caractère non intrusif de notre prototype à ce sujet*

1. Sélectionner, trois fois de suite, | **Folder "Drafts" |** .
2. Faire ensuite | **Folder "Calendar" |** ; | **Folder "Drafts" |** .
3. Sélectionner, mais seulement deux fois de suite cette fois-ci, | **Folder "Drafts" |** . Répondre par l'affirmative au système afin de poursuivre le mécanisme d'anticipation auto-adaptable lancé auparavant.
4. Confirmer encore une fois par | **Command "AI Layer - Repeated Single Actions" |** ; | **Folder "Drafts" |** .

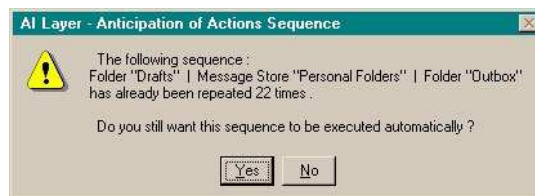


Figure A4.2.5.3 : *Résultats obtenus lors du test complémentaire de l'anticipation auto-adaptable du comportement de l'utilisateur*

Finalement, et afin de clore cette partie ayant permis la validation complète de notre modèle d'anticipation auto-adaptable, il nous faut encore préciser ici que, aussi bien pour le point **8** du **Scénario A4.2.5.1** que dans le cas du point **3** du **Scénario A4.2.5.2**, si l'utilisateur répond par la négative dans la fenêtre proposée par le système, cela a bien entendu pour conséquence d'arrêter le travail courant du présent mécanisme d'anticipation auto-adaptable et de réinitialiser le processus de prédiction correspondant.

Mais, dans ce cas et à l'image des détails déjà exposés lors du commentaire du **Scénario A4.2.4**, le système se charge de faire appel à son expérience acquise jusqu'alors : l'utilisateur n'aura ainsi pas à réitérer la séquence concernée autant de fois qu'au tout début pour se voir afficher à l'écran une nouvelle proposition de prédiction auto-adaptable de la part du système...

A4.2.6. Nouveau démarrage de la "Couche IA"

Comme précisé dès le début du **Paragraphe A4.2.1**, nous nous proposons à présent de fermer puis de relancer l'exécution de notre prototype d'auto-adaptation, afin de rendre clairement compte du bon redémarrage de celui-ci ainsi que d'un chargement correct et complet des différentes bases de données, déjà bien remplies (**Scénario A4.2.6**).

Scénario A4.2.6 : *Descriptif d'un "alpha-test" concernant le redémarrage du prototype avec des bases de données déjà conséquentes*

1. Sélectionner | **Command "Exit and Log Off" in Menu "File"** |.
2. Attendre que le travail de l'application *Microsoft Outlook* cesse complètement.
3. Puis relancer cette même application.
4. Sélectionner alors d'emblée | **Folder "Drafts"** |, et constater que notre modèle comportemental auto-adaptable, ainsi que son mécanisme courant d'anticipation, n'ont pas été altérés.
5. Confirmer en validant chaque affichage de la séquence | **Command "AI Layer - User Behaviour and Learning Quality"** | ; | **Command "AI Layer - Repeated Actions Sequences"** | ; | **Command "AI Layer - Repeated Single Actions"** | ; | **Command "AI Layer - Extracted Last Actions"** |.

Avant de découvrir, sur la page suivante, les différentes copies d'écran faisant état de ces dernières étapes (**Figure A4.2.6**), nous pouvons tout d'abord relever qu'il existe encore deux autres moyens d'arrêter le travail avec l'AUM et notre prototype, à savoir | **Command "Exit" in Menu "File"** | mais aussi | **Command "Exit" in the Window Top Right Corner** |. Cependant, celui que nous utilisons dans le scénario ci-dessus reste le plus exhaustif, donc le plus sûr, le plus efficace et surtout le plus propre au niveau du réseau et des bases de données.

En outre, n'oublions pas non plus de préciser ici que la réduction de l'application, appelée par | **Command "Reduce" in the Window Top Right Corner** |, ne ferme en aucun cas le système, mais met seulement ce dernier en veille : notre prototype reste alors entièrement opérationnel avec ses bases de données toutes ouvertes.

En second lieu, nous remarquons également que, et ce notamment en prenant connaissance de la copie d'écran correspondant aux différentes répétitions de séquences, nous avons choisi d'appliquer un tri [Sandel, Korczak & al., 1999-5] à chaque masse d'informations dépassant, à l'affichage, la taille d'un écran.

En effet, dans notre constante aspiration à être le moins invasif possible pour l'utilisateur, nous ne pouvions admettre de ne pas aider ce dernier à lire et à comprendre facilement les données fournies par notre système. Ces informations pouvant rapidement devenir conséquentes, et donc de lecture relativement gênante, stressante voire difficilement utilisable pour certains utilisateurs, nous avons rendu auto-adaptable chacun des affichages de notre prototype. Ainsi, celui-ci ne présente à ses utilisateurs que, dans le cas tout d'abord de l'extraction des connaissances, les dernières actions les plus récentes, et, pour ce qui est cette fois-ci des différentes répétitions d'actions ou de séquences d'actions, celles qui s'avèrent les plus importantes en nombre à l'instant courant.

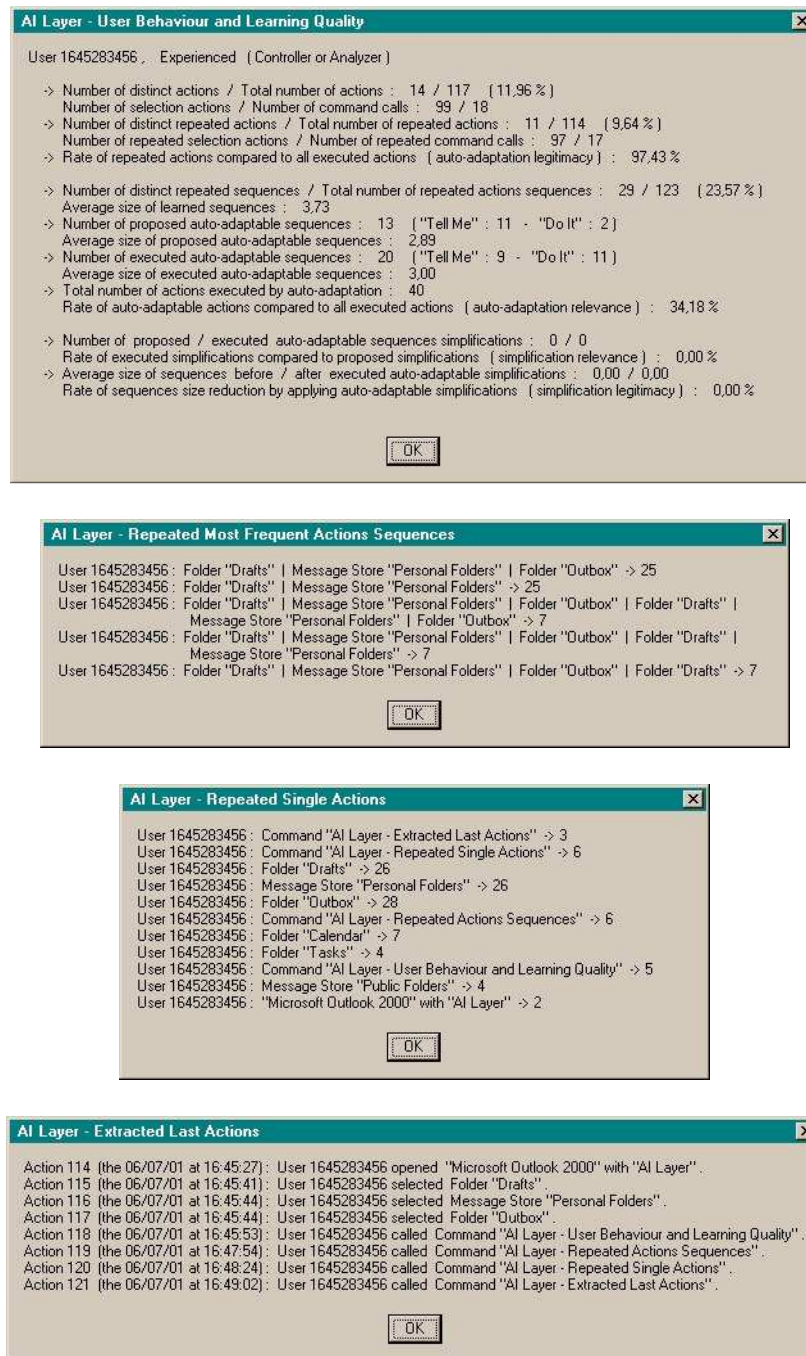


Figure A4.2.6 : Résultats obtenus lors du test de redémarrage de notre prototype avec des bases de données déjà existantes et bien remplies

Cette nouvelle qualité de tri adjointe à notre Assistant Individuel de Télécommunication confère par conséquent, à tout utilisateur, l'assurance d'une information, en temps réel, toujours efficace et aisée puisqu'au maximum de la pertinence.

A4.2.7. Simplification auto-adaptable des séquences d'actions effectuées

Ultime fonctionnalité implémentée et intégrée à notre modèle comportemental déjà scrupuleusement examiné jusqu'à maintenant, nous proposons finalement, au travers de cette simplification auto-adaptable, de traiter toutes les séquences d'actions laissant apparaître d'éventuelles erreurs de manipulations de la part de nos utilisateurs respectifs.

Nos investigations dans ce domaine ont alors spécifiquement mis en avant deux types de séquences pouvant en effet être simplifiées, corrigées de manière auto-adaptable : celles, tout d'abord, contenant au moins deux fois une même action et présentant par conséquent une certaine redondance d'informations, mais ensuite aussi les séquences qui incluent l'action | **Command "Undo"** | donnant lieu bien entendu, de la part de l'utilisateur, à un retour en arrière immédiat suite à une manipulation non satisfaisante.

C'est pourquoi, voici encore deux nouveaux scénarios qui vont nous permettre de valider l'ensemble des interventions de notre modèle de simplification auto-adaptable exploré dans cette partie (**Scénario A4.2.7.1 et Scénario A4.2.7.2, Figure A4.2.7.1 et Figure A4.2.7.2**).

Scénario A4.2.7.1 : *Descriptif d'un "alpha-test" concernant la simplification auto-adaptable des séquences d'actions contenant au moins deux fois une même action*

1. Réaliser | **Message Store "Personal Folders"** | ; | **Folder "Tasks"** | ; | **Folder "Calendar"** | ; | **Folder "Notes"** | ; | **Folder "Tasks"** | ; | **Message Store "Public Folders"** | .
2. Effectuer une seconde fois la nouvelle séquence précédente.
3. Sélectionner | **Folder "Drafts"** | .
4. Faire ensuite | **Message Store "Personal Folders"** | ; | **Folder "Tasks"** | ; | **Folder "Calendar"** | ; | **Folder "Notes"** | ; | **Folder "Tasks"** | ; | **Message Store "Public Folders"** | ; | **Command "AI Layer - Extracted Last Actions"** | .
5. Faire | **Message Store "Personal Folders"** | ; | **Folder "Tasks"** | ; | **Folder "Calendar"** | ; | **Folder "Notes"** | ; | **Folder "Tasks"** | ; | **Message Store "Public Folders"** | ; | **Command "AI Layer - User Behaviour and Learning Quality"** | .
6. Faire | **Message Store "Personal Folders"** | ; | **Folder "Tasks"** | ; | **Folder "Calendar"** | ; | **Folder "Notes"** | ; | **Folder "Tasks"** | ; | **Message Store "Public Folders"** | ; | **Command "AI Layer - Repeated Single Actions"** | .
7. Faire | **Message Store "Personal Folders"** | ; | **Folder "Tasks"** | ; | **Folder "Calendar"** | ; | **Folder "Notes"** | ; | **Folder "Tasks"** | ; | **Message Store "Public Folders"** | ; | **Command "AI Layer - Repeated Actions Sequences"** | .
8. Sélectionner finalement | **Message Store "Personal Folders"** | .
9. Valider alors l'exactitude de la nouvelle fenêtre apparue à l'écran en y répondant par l'affirmative. Vérifier ensuite que la nouvelle proposition de prédiction auto-adaptable, légitimement mise en avant par le système au vu du nombre des manipulations effectuées ci-dessus, tient bien compte du précédent souhait de simplifier la séquence courante ; accepter et valider son exécution automatique.
10. Confirmer par | **Command "AI Layer - User Behaviour and Learning Quality"** | .

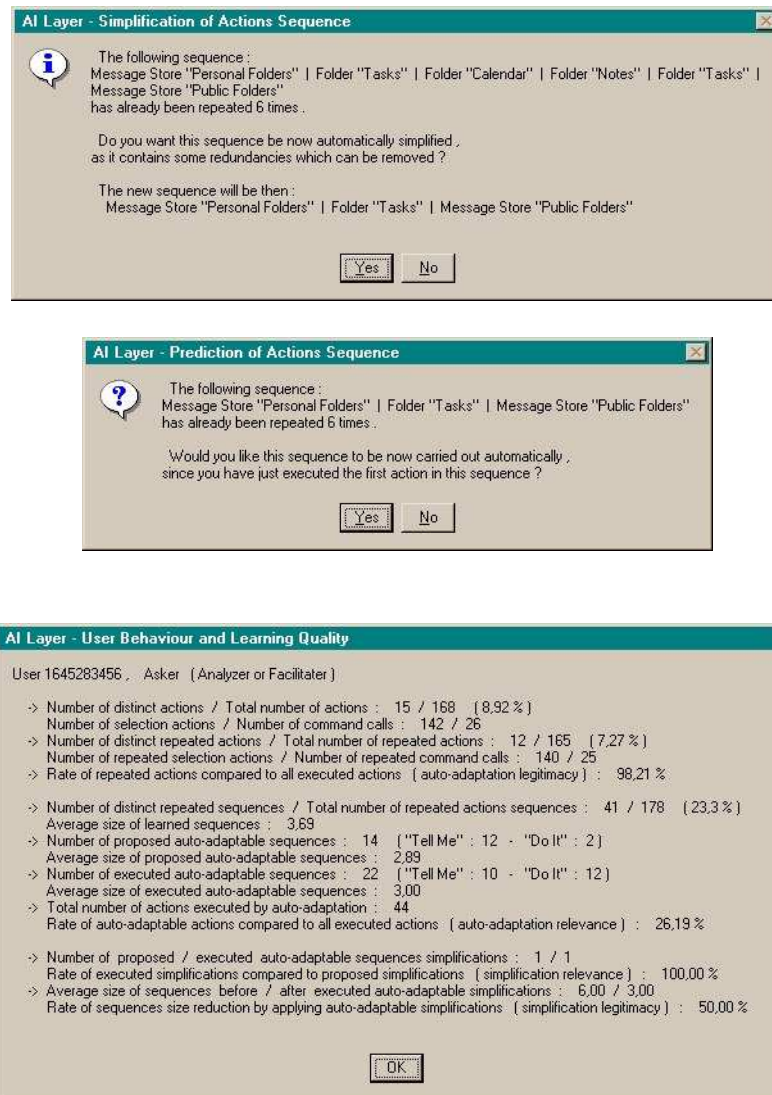


Figure A4.2.7.1 : *Résultats significatifs obtenus lors du test de la simplification auto-adaptable de séquences d'actions contenant au moins deux fois une même action*

Suite à notre récente approbation pour simplifier la dernière séquence courante, nous relevons, entre-autres, que nous sommes à présent catégorisés en qualité de "Demandant". Cela apparaît une nouvelle fois comme étant tout-à-fait "intelligent" puisque, non seulement nous venons de découvrir une nouvelle fonctionnalité, mais nous sommes aussi déjà suffisamment expérimentés pour appartenir à la classe "Analysant"...

De plus, en abordant maintenant, sur la page suivante, le second type de séquences d'actions pouvant admettre une simplification auto-adaptable, nous allons également constater que notre prototype ne va pas se contenter longtemps de cette catégorie intermédiaire de la méthode *Persona* [Julhiet, 93], et va en effet immédiatement chercher à affiner les caractéristiques de l'utilisateur concerné.

Scénario A4.2.7.2 : *Descriptif d'un "alpha-test" concernant la simplification auto-adaptable des séquences d'actions contenant l'action | Command "Undo" |*

1. Exécuter | Folder "Inbox" | ; | Command "New Message" , avec bien entendu soi-même comme destinataire et par exemple "Alpha-Test" en sujet | . Fermer la fenêtre courante par | Command "Send" | .
2. Ouvrir | Folder "Sent Items" | , et noter d'emblée une intéressante propriété de *Microsoft Outlook* qui, en présence de messages dans le répertoire courant, sélectionne en effet automatiquement le premier de ceux-ci en partant du haut.
3. Faire alors | Command "Delete" | ; | Command "Undo" | .
4. Effectuer une seconde fois la nouvelle séquence précédente.
5. Sélectionner | Folder "Drafts" | .
6. Faire ensuite | Folder "Sent Items" | ; | Command "Delete" | ; | Command "Undo" | ; | Message Store "Public Folders" | .
7. Faire | Folder "Sent Items" | ; | Command "Delete" | ; | Command "Undo" | ; | Folder "Outbox" | .
8. Faire | Folder "Sent Items" | ; | Command "Delete" | ; | Command "Undo" | ; | Command "AI Layer - Repeated Actions Sequences" | .
9. Sélectionner finalement | Command "Delete" | .
10. Valider alors les informations de la nouvelle fenêtre apparue à l'écran, mais en y répondant cette fois-ci par la négative. Vérifier ensuite que la nouvelle proposition de prédiction auto-adaptable, légitimement mise en avant par le système au vu du nombre des manipulations effectuées ci-dessus, tient bien compte du précédent désir de ne pas, pour cette étape-ci, simplifier la séquence courante ; décliner ici, une fois n'est pas coutume, l'exécution automatique de cette dernière.
Remarquer aussi que, suite à un tel refus de la simplification proposée, notre prototype, en accord avec tous ses principes déjà jusqu'à présent validés, propose une nouvelle prédiction concernant toutes les actions de la séquence traitée. Dans le cas contraire d'une réponse positive déjà testée dans le scénario précédent, notre système aurait retiré de la séquence courante l'action | Command "Undo" | ainsi que celle qui l'a précède, à savoir ici | Command "Delete" | .
11. Observer, en définitive, que notre système demeure totalement opérationnel : sélectionner par exemple | Message Store "Personal Folders" | , et répondre encore une fois par l'affirmative aux deux nouvelles fenêtres successivement affichées.
Ne pas manquer non plus de relever ici que nous avons choisi, au travers de l'implémentation de notre "Couche IA", de toujours privilégier, s'il y a lieu bien entendu, la simplification par rapport à la prédiction ou encore l'anticipation, ceci afin de constamment permettre à nos utilisateurs d'apprendre à mieux maîtriser leurs manipulations respectives vis-à-vis de l'AUM.
12. Confirmer et conclure par l'analyse d'une ultime séquence représentée par | Folder "Drafts" | ; | Command "AI Layer - User Behaviour and Learning Quality" | . Retenir alors que nous sommes finalement passés de la catégorie intermédiaire de "Demandant" à celle de "Facilitant". Le système reste par conséquent très pertinent et perspicace puisque que nous venons effectivement de prendre connaissance, mais aussi d'accepter, une nouvelle simplification auto-adaptable.

D'ailleurs, à ce niveau précis de nos investigations et de tous nos récents scénarios, nous ne pouvons que convenir que l'utilisateur que nous avons représenté correspond bien à un individu apprenant et attentif, coopératif et conciliant...

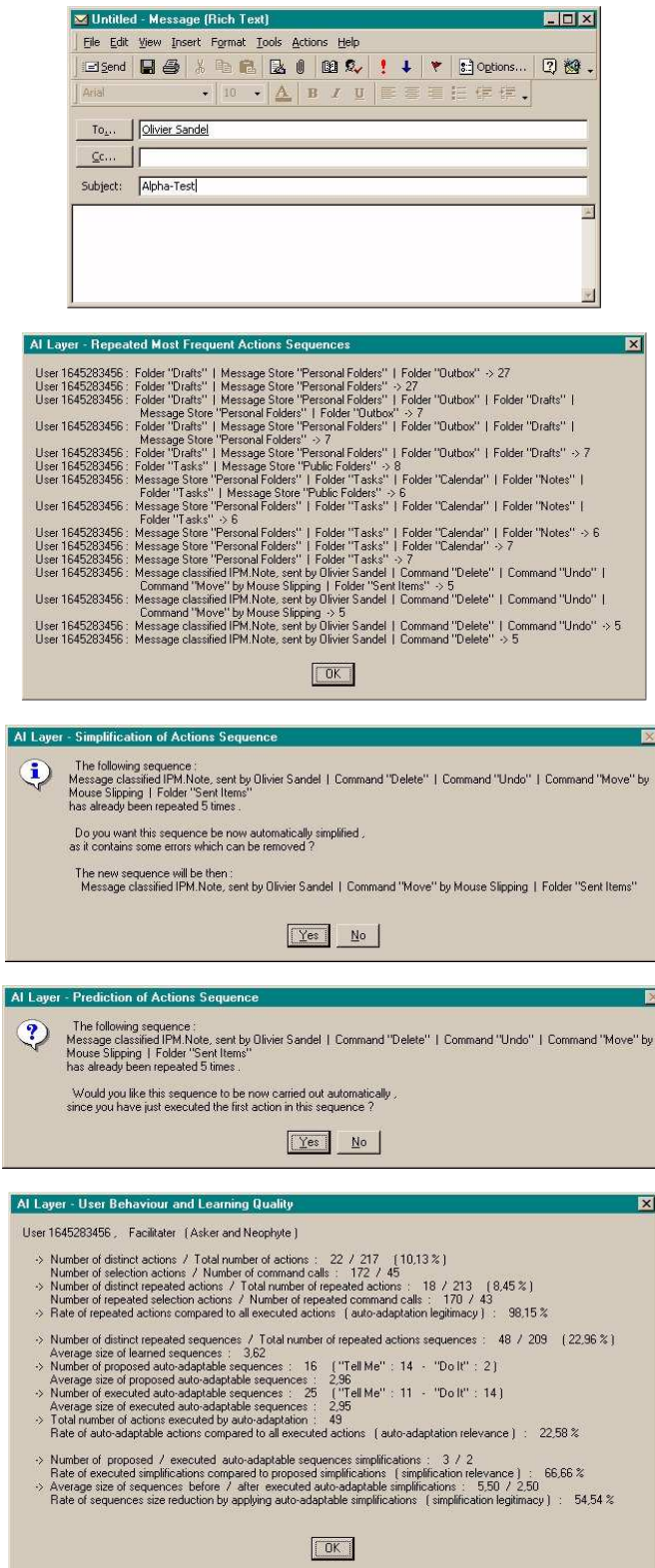


Figure A4.2.7.2 : Résultats significatifs obtenus lors du dernier test, à savoir celui de la simplification auto-adaptable de séquences d'actions contenant l'action | Command "Undo" |

Par suite, à l'image des différents procédés déjà établis plus haut dans ce mémoire, notre prototype réitérera sa proposition de simplification auto-adaptable après une nouvelle exécution de plusieurs autres séquences identiques à celle exposée au cours des points **2** et **3** de ce **Scénario A4.2.7.2**.

De plus, et toujours de manière à ne pas devenir trop intrusif pour nos utilisateurs, nous avons également pu constater, lors notamment du point **9** de ce même scénario, que nous avons opté, dans le cas d'une tête de séquence correspondant à un message, pour le déclenchement des mécanismes d'auto-adaptation au niveau, seulement, de la répétition de la seconde action de cette séquence : trop de messages possédant des caractéristiques similaires pourraient en effet, à terme, confondre et décontenancer le travail auto-adaptable courant de chaque utilisateur, pourtant alors déjà bien expérimenté.

En conclusion, nous pouvons retenir de ces ultimes "alpha-tests" que notre Assistant Individuel de Télécommunication, en plus de ses multiples autres fonctionnalités auto-adaptables validées jusqu'à maintenant, démontre finalement que toute séquence d'actions, qu'elle contiennent des redondances d'informations ou encore des actions non souhaitées, peut être simplifiée ou corrigée de manière auto-adaptable.

A4.2.8. Validation complète de la méthode d'analyse des comportements

De l'ensemble de nos nombreuses, mais exhaustives et nécessaires validations menées à terme au travers du présent rapport, il nous reste en définitive à démontrer que chacune des différentes catégories d'utilisateurs mises en avant par la méthode d'analyse des comportements *Persona* est tout-à-fait pertinente et donc accessible par le Modèle Auto-Adaptable de l'Utilisateur de notre "Couche IA".

A ce titre, et sans oublier celles que nous avons d'ailleurs déjà exposées lors de nos différentes études précédentes, une nouvelle exploitation de notre logiciel auto-adaptable, nettement plus avancée bien entendu que jusqu'à présent, nous permet alors de prendre connaissance de l'ensemble des catégorisations possibles pour nos utilisateurs respectifs. Nous en proposons par conséquent, sur la page suivante, un complet récapitulatif (**Figure A4.2.8**), toujours grâce à notre fonctionnalité |Command "AI Layer - User Behaviour and Learning Quality" |.

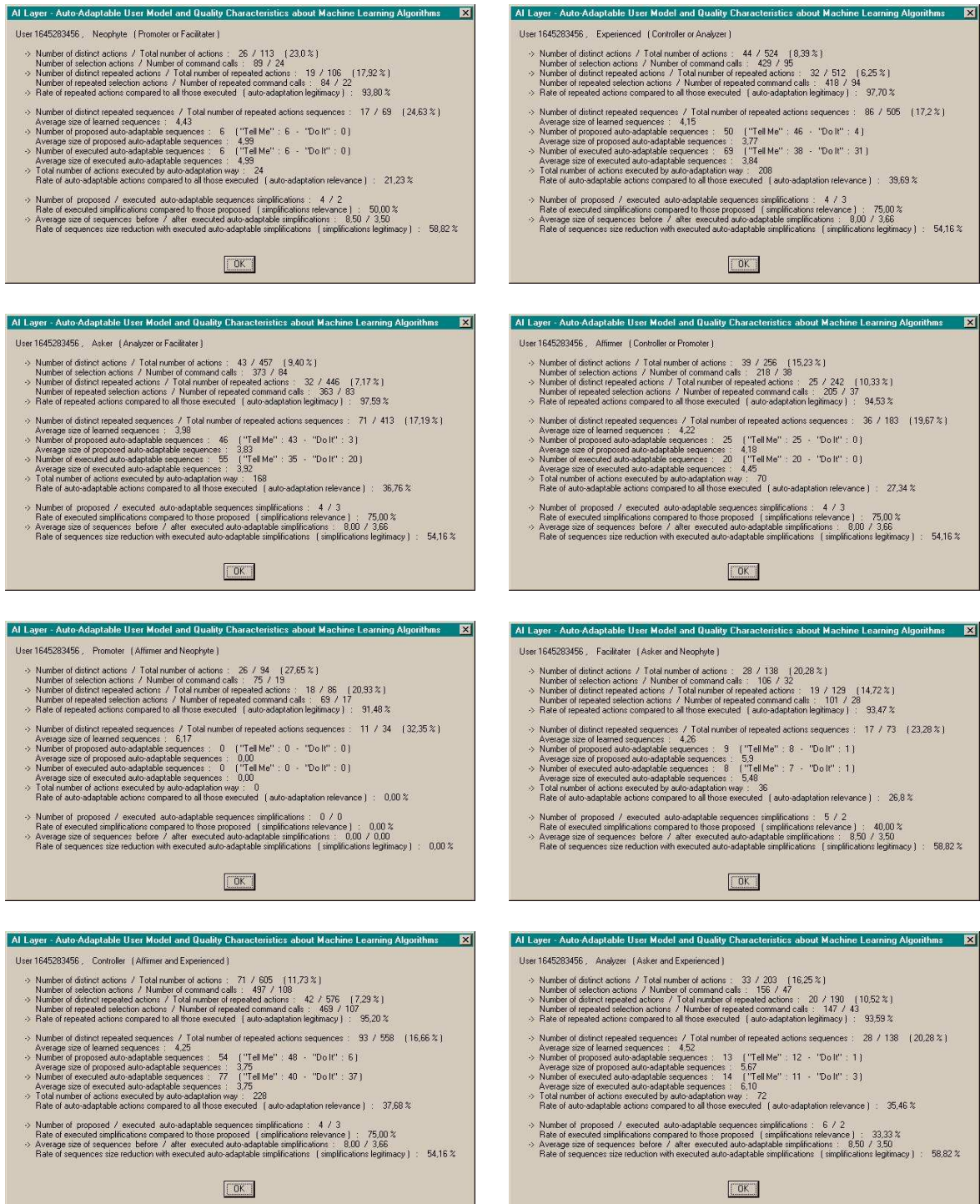


Figure A4.2.8 : Copies d'écrans démontrant finalement que toutes les catégories d'utilisateurs mises en avant par la méthode "Persona" sont réellement accessibles par notre prototype d'Interface Intelligente Auto-Adaptable

RÉSUMÉ

La présente Thèse en Intelligence Artificielle est basée sur un tout premier concept nommé "Assistant Conversationnel", lequel est abordé par le fait qu'un utilisateur quelconque souhaitant joindre un interlocuteur doit pouvoir toujours voir sa requête se réaliser, quelque soit le moyen choisi par le terminal et sans que l'utilisateur ait à choisir lui-même le moyen adéquat. Le problème principal à résoudre se définit en conséquence par la modélisation et la conception d'une interface intelligente, personnalisée et praticable par tous les publics sur les terminaux de communication spécialisables accédant à *Internet*.

Nous avons alors créé et développé de nouveaux procédés "intelligents", chargés entre-autres de rendre les opérations courantes moins fastidieuses sur des applications de messagerie électronique. Pour cela, nous avons établi un état de l'art des divers terminaux de communication disponibles sur le marché mondial, ainsi qu'une classification taxinomique poussée et une nouvelle hiérarchisation efficace des interfaces intelligentes. Puis, nous avons construit un "Modèle Auto-Adaptable d'Utilisateur" sur un apprentissage comportemental, incrémental et évolutif, finalement implémenté en quatre phases :

1. Analyse, représentation et classification des différentes actions réalisées par l'utilisateur.
2. Proposition régulière à l'utilisateur d'effectuer de manière auto-adaptable les actions apprises.
3. Exécution auto-adaptable de ces différentes séquences d'actions répétitives.
4. Simplification auto-adaptable d'éventuelles "erreurs" ou incohérences de manipulation.

Ainsi, cette contribution innovante à l'interfaçage homme-machine avancé a permis d'aboutir à la validation complète d'un logiciel intelligent de communication, tout-à-fait autonome et portable, capable d'assister chaque utilisateur, néophyte comme très expérimenté, dans ses divers travaux quotidiens, et ce de manière conviviale et toujours respectueuse de ses préférences.

MOTS-CLÉS Interface Intelligente, Terminal de Communication, Système Multi-Agents, Modèle Utilisateur, Apprentissage Automatique, Séquencement d'Actions, Prédiction, Anticipation, Simplification, Auto-Adaptation, Spécification Formelle, LOTOS, Messagerie Electronique, MAPI, Validation.

TITLE **Intelligent Interface Model for Communication Terminals**

ABSTRACT

This PhD Thesis in Artificial Intelligence is first based on the concept of "Conversational Assistant", which is introduced by the fact that each user who wishes to join an interlocutor should always see his request to be executed, whatever mean the terminal chooses and without the user should have to find himself the appropriate mean. The main problem to be solved here is consequently defined by the modelisation and the conception of an intelligent interface, personalised but practicable by all publics using communication specializable *Internet* machines.

We have then created and developed new "intelligent" prototypes, which especially must achieve current operations less boring on any applications of electronic messaging. For that, we have established a state of the art about all various communication terminals available on the worldwide market, as well as an elaborate taxonomic classification and a new effective hierarchical system about intelligent interfaces. Afterwards, we have built an "Auto-Adaptable User Model" on a behavioural learning, incremental and evolutionary, at last set in four stages :

1. Analysis, representation and classification of the different actions realized by the user.
2. Regular proposal to the user with the aim of execute the learned actions by auto-adaptation way.
3. Auto-adaptable execution of whole these repetitive action sequences.
4. Auto-adaptable simplification of any eventual "errors" or handling incoherences.

So, this innovating contribution to the advanced human-machine interfaces has allowed to validate of an intelligent communication software, quite autonomous and portable, able to assist each user, neophyte as well as very experimented, in its all daily works, and that in user-friendly way and always with respect to its preferences.

KEYWORDS Intelligent Interface, Communication Terminal, Multi-Agents System, User Model, Machine Learning, Hierarchical Sequence of Actions, Prediction, Anticipation, Simplification, Auto-Adaptation, Formal Specification, LOTOS, Electronic Messaging, MAPI, Validation.

DISCIPLINE Sciences

SPÉCIALITÉ Informatique