

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

CONTRÔLE DES INTERACTIONS ORALES ENTRE HUMAIN ET MACHINE :
APPROCHE D'APPRENTISSAGE MACHINE

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
MINH-QUANG NGUYEN

SEPTEMBRE 2008

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

Cette thèse a été dirigée sous la direction de recherche de :

Monsieur Jean-Guy Meunier, directeur de thèse,

Professeur au Département de Philosophie de l'Université du Québec à Montréal

et

Monsieur Douglas O'Shaughnessy, codirecteur de thèse,

Professeur à l'Institut national de la recherche scientifique - Énergie, Matériaux et
Télécommunications.

Les membres du Jury sont constitués de :

Monsieur Jean-Guy Meunier, directeur de thèse, professeur au Département de philosophie de l'Université du Québec à Montréal,

Monsieur Douglas O'Shaughnessy, codirecteur de thèse, professeur à l'Institut national de la recherche scientifique – Énergie, Matériaux et Télécommunications,

Madame Lucie Ménard, membre interne, professeure au Département de linguistique de l'Université du Québec à Montréal,

Monsieur Hakim Lounis, membre interne, président du jury, professeur au Département d'informatique de l'Université du Québec à Montréal,

Monsieur Francis Rousseaux, membre externe, professeur au Département d'informatique de l'Université de Reims Champagne-Ardenne, France.

La machine peut réfléchir, penser et même parler.

Mais de quelle manière?

*Elle réfléchit machinalement de travers,
pense systématiquement de façon rationnelle et,
parle synthétiquement sans état d'âme ni émotions.*

@Minh-Quang Nguyen, 2008

REMERCIEMENTS

J'aimerais remercier sincèrement et chaleureusement les personnes suivantes :

Monsieur Ghislain Lévesque, ex-directeur du programme de doctorat en informatique cognitive.

Monsieur Mounir Boukadoum, directeur du programme de doctorat en informatique cognitive.

Monsieur Jean-Guy Meunier, directeur de thèse, professeur au Département de philosophie de l'Université du Québec à Montréal

Monsieur Douglas O'Shaughnessy, co-directeur de thèse, professeur à l'Institut national de la recherche scientifique – Énergie, Matériaux et Télécommunications.

Madame Henrietta Jonas-Cedergren, professeure retraitée au Département de linguistique de l'Université du Québec à Montréal. Elle était ma directrice de thèse au tout début de mon travail.

Les membres du jury : madame Lucie Ménard, professeure au Département de linguistique de l'Université du Québec à Montréal, monsieur Hakim Lounis, professeur au Département d'informatique de l'Université du Québec à Montréal et monsieur Francis Rousseaux, Professeur au Département d'informatique de l'Université de Reims Champagne-Ardenne, France. Merci d'avoir pris le temps de lire et de critiquer mes travaux de recherche. Vos commentaires ont été et sont toujours pertinents pour la suite de la carrière d'un chercheur scientifique.

La société Nuance Communications inc. dont monsieur Jean-Guy Dahan qui a pour rôle d'être responsable de recherche et de développement pour la partie de la reconnaissance de la parole. Nuance a mis à ma disposition, par entremise d'une

entente avec INRS-EMT, des matériels informatiques ainsi que des logiciels de dialogue nécessaires pour me permettre de tester ma théorie.

Monsieur Tho-Hau Nguyen, professeur au Département d'informatique de l'Université du Québec à Montréal. Vous avez toujours été à côté de moi pour me remonter le moral quand je me sentais déprimé et surchargé par l'énorme travail qu'il y a à faire.

Mon frère Philip H.P Nguyen, Ph.D, habite en Australie, qui a toujours su me supporter et me donner de très bons conseils pour mes travaux de recherche.

Madame Mariette Julien et madame Nathalie Langevin, professeures à l'École supérieure de mode de Montréal de l'Université du Québec à Montréal. Vous m'avez aidé à améliorer la présentation orale de thèse.

Madame Monique Martin (ESMM) a corrigé une partie de ma thèse. Sylvie, Arlette, Francine, Louise (ESMM) et tous mes ami(e)s qui m'ont encouragé dans cette aventure périlleuse.

Madame Lise Arsenault, assistante à la gestion de programmes d'études avancées, qui est toujours serviable et disponible pour m'aider à régler les problèmes administratifs.

Si j'ai réussi aujourd'hui à devenir Docteur, cela est dû à l'effort continu de mon directeur Jean-Guy Meunier. Il est pour moi un père spirituel, celui qui a su guider ses étudiants et leur donner de bons outils pour réussir.

Bien entendu, je n'aurai pas oublié mon co-directeur, Douglas O'Shaughnessy m'a supporté financièrement durant les deux dernières années doctorales et il a cru à tout ce que j'ai fait. Cela m'a beaucoup rassuré, surtout venant de la part d'une personne qui est mondialement connue dans son domaine.

Enfin, mes profonds remerciements et gratitude (sûrement pas les derniers) vont directement à mes parents, très fiers de leur fils, et à ma famille dont ma chérie Khanh, mes deux enfants Tuan-Anh Laurent et Anh-Khoa Julien qui sont mes inspirations et ma force de vie ; sans eux, tous ces efforts n'auraient aucun sens.

Laissons nos émotions à côté, regardons plutôt comment nous pouvons représenter nos connaissances aux machines de manière à ce que celles-ci puissent dialoguer avec nous comme si nous dialoguions. Et s'il est possible d'apprendre à la machine à imiter nos comportements dialogiques pour qu'elle puisse faire revivre nos sentiments avec nos êtres chers à travers le temps.

TABLE DES MATIERES

LISTE DES FIGURES	XVI
LISTE DES TABLEAUX	XX
LISTE DES ACRONYMES	XXII
RÉSUMÉ	XXIV
ABSTRACT	XXVI
CHAPITRE I	1
PRÉSENTATION DU SUJET DE THÈSE	1
I.1 INTRODUCTION.....	1
I.2 DEFINITIONS.....	2
I.2.1 <i>Système de dialogue oral humain-machine</i>	2
I.2.2 <i>La recherche en systèmes de dialogue</i>	3
I.2.3 <i>Stratégie optimale du dialogue</i>	4
I.2.4 <i>Apprendre une stratégie optimale à la machine</i>	4
I.3 PROBLÉMATIQUE.....	5
I.3.1 <i>Système de dialogue oral humain-machine</i>	7
I.3.2 <i>Sciences du langage</i>	7
I.3.3 <i>Contraintes cognitives</i>	8
I.4 HYPOTHÈSES DE RÉPONSES.....	8
I.5 OBJECTIFS DE RECHERCHE.....	11
I.5.1 <i>Objectif informatique</i>	12
I.5.2 <i>Objectif cognitif</i>	12
I.6 CONTEXTE DU PROJET DE RECHERCHE.....	13
I.7 CHEMINEMENT MÉTHODOLOGIQUE DE LA THÈSE.....	13
I.7.1 <i>Modèle triangulaire</i>	13

1.7.2	<i>Modèle spirale</i>	14
I.8	ORGANISATION DE LA THÈSE	15
CHAPITRE II		18
DIALOGUE ORAL HUMAIN-HUMAIN.....		18
II.1	INTRODUCTION.....	18
II.2	DÉFINITIONS.....	18
II.2.1	<i>Actes de langage</i>	18
II.2.2	<i>Discours et dialogue</i>	19
II.3	DIALOGUE HUMAIN-HUMAIN	20
II.4	UTILISATION DU LANGAGE	20
II.5	APPROCHES DE MODÉLISATION DU DIALOGUE	21
II.5.1	<i>Modèle structurel</i>	22
II.5.1.1	Modèle de Genève	22
II.5.1.2	Modèle de Luzzati.....	25
II.5.2	<i>Modèle orienté-plan</i>	26
II.5.2.1	Concept de planification	26
II.5.2.2	Modèle de Cohen et Perrault.....	27
II.5.2.3	Modèle de Litman	28
II.5.2.4	Modèle de Ramshaw	29
II.5.3	<i>Modèle logique</i>	30
II.5.3.1	Modèle mental.....	30
II.5.3.2	Modèle de jeux.....	31
II.5.3.3	Modèle de jeux associé à la psychologie sociale	31
II.6	CONCLUSION	32
CHAPITRE III		34
SYSTÈME DE DIALOGUE ORAL HUMAIN-MACHINE		34
III.1	INTRODUCTION.....	34
III.2	SYSTÈME DE DIALOGUE ORAL HUMAIN-MACHINE	34
III.3	ARCHITECTURE D'UN SDOHM.....	35

III.3.1	<i>Niveau acoustique</i>	35
III.3.1.1	Perception et reconnaissance du signal de la parole	35
III.3.1.2	Synthèse de la parole	37
III.3.2	<i>Niveau sémantique</i>	38
III.3.2.1	Compréhension de la parole	38
III.3.2.2	Gestion du dialogue	39
III.3.2.3	Génération de la parole	39
III.3.3	<i>Niveau pragmatique</i>	40
III.3.3.1	Connaissance du monde.....	40
III.4	CONCLUSION	40
	CHAPITRE IV	42
	GESTION DU DIALOGUE ORAL HUMAIN-MACHINE	42
IV.1	INTRODUCTION.....	42
IV.2	GESTION DU DIALOGUE HUMAIN-MACHINE	42
IV.2.1	<i>Modèles statiques</i>	43
IV.2.1.1	Modèle du langage	43
IV.2.1.2	Modèle de dialogue	43
IV.2.1.1	Modèle de la tâche.....	44
IV.2.2	<i>Modèles dynamiques</i>	44
IV.2.2.1	Contexte de la tâche.....	44
IV.2.2.1	Historisation du dialogue.....	45
IV.2.3	<i>Modèles statiques et dynamiques</i>	45
IV.2.3.1	Modèle de l'utilisateur.....	45
IV.2.3.2	Modèle du système	46
IV.2.3.3	Modèle de stratégie.....	46
IV.3	APPROCHE DE MODÉLISATION DE LA STRATÉGIE DU DIALOGUE ...	47
IV.3.1	<i>Rôle du gestionnaire de dialogue</i>	47
IV.3.1.1	Degrés d'initiative	48
IV.3.1.2	Stratégie de confirmation.....	49
IV.4	APPROCHE DE MODÉLISATION DES ACTES DE DIALOGUE	51

IV.4.1	<i>Les états sous forme de réseaux</i>	52
IV.4.2	<i>Les propriétés markoviennes</i>	52
IV.4.3	<i>Les cadres</i>	52
IV.4.4	<i>Les mises à jour de données</i>	53
IV.4.5	<i>L'agent collaboratif</i>	53
IV.5	CONCLUSION	53
CHAPITRE V		55
APPRENTISSAGE DE LA MACHINE		55
V.1	INTRODUCTION.....	55
V.2	DIALOGUE ET APPRENTISSAGE PAR RENFORCEMENT (AR)	56
V.3	APPRENDRE UNE STRATÉGIE OPTIMALE DU DIALOGUE	57
V.3.1	<i>Apprentissage supervisé</i>	57
V.3.2	<i>Apprentissage non supervisé</i>	57
V.3.3	<i>Apprentissage par renforcement (AR)</i>	57
V.4	PRINCIPE DE BASE D'AR	58
V.4.1	<i>Algorithme d'AR</i>	59
V.4.2	<i>Fonction récompense</i>	60
V.5	PROCESSUS DE DÉCISION DE MARKOV	63
V.5.1	<i>Propriété de Markov</i>	63
V.5.2	<i>Processus de décision de Markov dans le dialogue</i>	63
V.6	APPROCHE D'AR	66
V.7	AR AVEC UN UTILISATEUR SIMULÉ.....	67
V.7.1	<i>Architecture</i>	67
V.7.2	<i>Utilisateur simulé</i>	68
V.8	AR SANS UTILISATEUR SIMULÉ.....	70
V.8.1	<i>Exemple du système NjFun</i>	72
V.8.1.1	Les étapes de construction du système NJFun.....	72
V.8.1.1	Fonctionnement du gestionnaire du dialogue du NJFun.....	72
V.8.1.1	Fonction récompense	73
V.9	CONCLUSION	74

CHAPITRE VI.....	75
MODÉLISATION DU PROCESSUS D'APPRENTISSAGE POUR UNE STRATÉGIE OPTIMALE DU DIALOGUE	75
VI.1 INTRODUCTION.....	75
VI.2 CRITÈRES DE MESURE DE LA QUALITÉ DU DIALOGUE	76
<i>VI.2.1 Définitions.....</i>	<i>76</i>
VI.2.1.1 Dialogue réussi	76
VI.2.1.2 Dialogue non-réussi.....	77
<i>VI.2.2 Modélisation des critères de mesure.....</i>	<i>77</i>
VI.2.2.1 Mesure quantitative	77
VI.2.2.2 Mesure qualitative	78
VI.3 STRATÉGIE OPTIMALE DU DIALOGUE.....	83
<i>VI.3.1 Règles pour une stratégie optimale du dialogue.....</i>	<i>84</i>
VI.4 APPROCHE D'AR	85
<i>VI.4.1 Modèle d'AR avec un utilisateur simulé.....</i>	<i>86</i>
VI.4.1.1 Architecture d'AR	86
<i>VI.4.2 Fonctionnement de l'utilisateur simulé.....</i>	<i>90</i>
VI.4.2.1 But	90
VI.4.2.2 Rôle.....	90
VI.4.2.3 Degré d'initiative.....	91
VI.4.2.4 Stratégie de confirmation.....	91
<i>VI.4.3 Fonctionnement de l'agent apprenant</i>	<i>92</i>
VI.4.3.1 But	92
VI.4.3.2 Rôle.....	92
VI.4.3.3 Degré d'initiative.....	93
VI.4.3.4 Stratégie de confirmation.....	93
VI.5 CONCLUSION	93
CHAPITRE VII.....	95
MODÉLISATION INFORMATIQUE.....	95
VII.1 INTRODUCTION.....	95

VII.2	PROCESSUS DE DÉCISION DE MARKOV DANS LE DIALOGUE	95
VII.2.1	<i>États</i>	96
VII.2.2	<i>Actions</i>	98
VII.3	INTERACTION ENTRE L'UTILISATEUR SIMULÉ ET LA MACHINE.....	99
VII.3.1	<i>Agent machine</i>	100
VII.3.1.1	Algorithme d'apprentissage	101
VII.3.2	<i>Utilisateur simulé</i>	102
VII.3.2.1	Algorithme d'attribution d'une valeur de récompense....	104
VII.4	CONCLUSION.....	105
CHAPITRE VIII.....		107
EXPÉRIMENTATION ET RÉSULTATS		107
VIII.1	INTRODUCTION.....	107
VIII.2	TÂCHES DE RÉSERVATION.....	107
VIII.3	ANALYSE ORGANIQUE DES SOLUTIONS INFORMATIQUES	109
VIII.3.1	<i>Définitions</i>	109
VIII.3.2	<i>Organisation de système de fichiers</i>	110
VIII.3.2.1	Environnement de développement.....	110
VIII.3.2.2	Développement.....	110
VIII.3.2.3	Fichiers exécutables	110
VIII.3.2.4	Fichiers en entrées	110
VIII.3.2.5	Fichiers en sorties	115
VIII.4	PREMIÈRE EXPÉRIENCE	115
VIII.4.1	<i>But</i>	115
VIII.4.2	<i>Fonctionnement</i>	115
VIII.4.3	<i>Déroulement de l'expérience</i>	116
VIII.4.3.1	Comportement fixe	117
VIII.4.3.2	Comportement mixte	118
VIII.4.3.3	Résultat avec un utilisateur simulé coopéré.....	119
VIII.4.4	<i>Conclusion de la première expérience</i>	121
VIII.5	DEUXIÈME EXPÉRIENCE	122

<i>VIII.5.1</i>	<i>But</i>	122
<i>VIII.5.2</i>	<i>Fonctionnement</i>	122
<i>VIII.5.3</i>	<i>Déroulement de l'expérience</i>	123
VIII.5.3.1	Création des états de Markov.....	123
VIII.5.3.2	Apprentissage.....	123
VIII.5.3.3	Résultat d'un dialogue réussi après apprentissage.....	126
VIII.6	CONCLUSION	129
CHAPITRE IX.....		130
ÉVALUATIONS.....		130
IX.1	INTRODUCTION.....	130
IX.2	MÉTHODE D'ÉVALUATION	130
IX.2.1	<i>Satisfaction globale du dialogue</i>	131
IX.2.2	<i>Stratégie de confirmation</i>	132
IX.2.3	<i>Interprétation des énoncés</i>	132
IX.2.4	<i>Avant et après l'apprentissage</i>	132
IX.2.5	<i>Adaptation dans d'autres domaines</i>	133
IX.3	OBSTACLES POUR L'ÉVALUATION.....	133
IX.4	CONCLUSION	134
CHAPITRE X		135
CONTRIBUTION, DISCUSSION ET CONCLUSION.....		135
X.1	CONTRIBUTION.....	135
X.2	FUTUR TRAVAIL	137
X.3	DISCUSSION.....	137
X.3.1	<i>Informatique</i>	137
X.3.1.1	Annotation sémantique	138
X.3.1.2	Complexité de MPD	138
X.3.2	<i>Cognition</i>	139
X.3.2.1	L'intelligence machine versus l'intelligence humaine.....	139
X.3.2.2	Le corps, l'esprit et l'environnement	140

X.4	CONCLUSION	142
APPENDICE A		145
LES EXEMPLES DE CODE.....		145
A.1	CODE DU SIMULATEUR D'UTILISATEUR	145
A.2	CODE D'APPRENTISSAGE PAR RENFORCEMENT	145
A.3	ÉCRAN D'INTERFACE WINDOWS.....	145
A.4	EXEMPLES DE DIALOGUES CRÉÉS DANS LA 1IÈRE EXPÉRIENCE ...	147
BIBLIOGRAPHIE		151

LISTE DES FIGURES

Figure		page
Figure 1.1	Domaine de recherche de dialogue oral humain-machine.	3
Figure 1.2	Démarche expérimentale et processus de transition (extrait de l'exposé de Nicolle, 2001).	14
Figure 1.3	Cheminement méthodologique de la recherche.	14
Figure 1.4	Organisation de la thèse.	17
Figure 2.1	Unité de base de l'acte de dialogue.	23
Figure 2.2	Notion de planification système.	27
Figure 2.3	Définition d'une requête du modèle de Cohen & Perrault (extrait de Haller, 1999) [HAL99].	28
Figure 3.1	Vue générale des 5 modules d'un SDOHM (adaptée de Pietquin, 2004).	36
Figure 4.1	Connaissance requise pour un dialogue oral humain-machine (Selon Minker, 2000) [MIN00].	43
Figure 5.1	Principe d'apprentissage par renforcement.	58
Figure 5.2	Exemple d'un état markovien appliqué dans les actes de dialogue pour un domaine de réservation des chambres d'hôtel.	65
Figure 5.3	L'utilisateur simulé apprend des stratégies optimales du dialogue à la machine (adaptée de Schatzmann et al., 2006).	68
Figure 6.1	Répartition parfaite pour un dialogue réussi selon notre hypothèse.	81
Figure 6.2	Répartition idéale pour un dialogue réussi mais avec une possibilité de variation selon notre hypothèse.	83

Figure 6.3	Règles de stratégie optimale du dialogue.	85
Figure 6.4	Architecture générale d'approche d'apprentissage par renforcement à l'aide d'un utilisateur simulé.	87
Figure 6.5	Fonctionnement d'apprentissage par renforcement. L'agent apprenant est à l'intérieur de l'agent machine dans un processus d'apprentissage par renforcement.	89
Figure 7.1	Diagramme des séquences d'interaction du côté de l'agent machine.	101
Figure 7.2	Diagramme de séquences du côté de l'utilisateur simulé.	104
Figure 8.1	Nombre de tours de parole par exécution pour un dialogue sans aucun apprentissage avec un utilisateur simulé fixe (coopéré et non-coopéré).	117
Figure 8.2	Nombre de dialogues réussis totalement ou partiellement sans apprentissage.	119
Figure 8.3	Répartition de genre de questions posées par l'agent machine à l'utilisateur coopéré.	121
Figure 8.4	Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU selon le tableau 8.6.	124
Figure 8.5	Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU avec un utilisateur simulé coopéré.	125
Figure 8.6	Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU avec un utilisateur simulé non-coopéré.	125
Figure 8.7	Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU avec un utilisateur simulé avec un comportement mixte.	126
Figure 8.8	Répartition des types de question en 7 tours de parole.	129
Figure 10.1	Écran d'interface Windows : Simulation de dialogue entre un utilisateur simulé (Félix) et un agent machine (Julie) avant l'apprentissage.	146
Figure 10.2	Écran d'interface Windows : Simulation de dialogue entre un utilisateur simulé (Félix) et un agent machine (Julie) après 5000 exécutions d'apprentissage.	147

Figure 10.3 Répartition de genre de questions posées par l'agent machine à l'utilisateur non coopéré. 150

LISTE DES TABLEAUX

Tableau		page
Tableau 2.1	Structure hiérarchique de la conversation selon le modèle de Genève.	23
Tableau 2.2	Règles de réécriture de la structure hiérarchique (d'après Minker, 2000).	24
Tableau 4.1	Exemple d'une mauvaise stratégie de confirmation (explicite au lieu de répétitive) dans une situation où la qualité de reconnaissance est mauvaise.	50
Tableau 4.2	Exemple d'une mauvaise stratégie de confirmation (implicite au lieu d'explicite) dans une situation où la qualité de reconnaissance est moyenne.	51
Tableau 5.1	Exemple de dialogue généré par NJFun (extrait de [SIN02]). Le système utilise une stratégie <i>user-initiative</i> à l'ouverture du dialogue (S1) ensuite des confirmations explicites sont employées à chaque réponse de l'utilisateur (S2, S3).	73
Tableau 6.1	Exemple d'un dialogue réussi selon notre hypothèse.	82
Tableau 6.2	Exemple d'un dialogue réussi en tenant compte des facteurs humains et machines selon notre hypothèse.	84
Tableau 6.3	Exemple d'un cas où l'utilisateur simulé cherche à faire échouer le système. La stratégie appliquée dans ce cas consiste à 'obliger' l'utilisateur simulé d'être 'sérieux' après 3 tours de dialogue pour éviter les cas anodins.	92
Tableau 7.1	Les paramètres des états de dialogue et leurs valeurs.	97
Tableau 7.2	Exemple de l'algorithme d'apprentissage de l'agent apprenant qui met à jour la stratégie apprise.	102
Tableau 7.3	Exemple de l'algorithme d'attribution de valeur de récompense de l'utilisateur simulé.	105

Tableau 8.1	Exemple d'un dialogue entre deux personnes.	108
Tableau 8.2	Exemple du contenu du fichier de données Q-Learning.data.	111
Tableau 8.3	Exemple du contenu du fichier de question pour le domaine de réservation des chambres d'hôtel.	113
Tableau 8.4	Exemple du contenu du fichier de réponse pour le domaine de réservation des chambres d'hôtel.	114
Tableau 8.5	Dialogue presque réussi entre un utilisateur simulé coopéré et un agent apprenant.	121
Tableau 8.6	Résultats d'entraînement pour 9 épisodes.	124
Tableau 8.7	Exemple d'un dialogue réussi en 7 tours de parole après 5 épisodes d'apprentissage.	128
Tableau 10.1	Dialogue entre un utilisateur simulé non coopéré et un agent apprenant sans apprentissage.	150

RÉSUMÉ

Les techniques de reconnaissance et de synthèse vocale deviennent de plus en plus performantes et robustes, ce qui facilite la création des applications de dialogue oral humain-machine. Ce sont des applications à partir desquelles la machine dialogue oralement avec l'humain en imitant l'acte de communication des humains. Cependant, le dialogue entre l'humain et la machine manque encore de naturel, de souplesse et est parfois même agaçant. Alors que les recherches se sont orientées vers l'aspect acoustique et sémantique du signal de la parole, d'autres se sont positionnées au niveau de l'apprentissage de la machine. Cette technique consiste à lui apprendre une stratégie du dialogue. La stratégie optimale du dialogue permet à l'humain et à la machine de communiquer de manière efficace. La question qui se pose en informatique cognitive est de savoir comment une machine peut apprendre des séquences de comportements, en l'occurrence, des énoncés du langage naturel qui, lorsque confrontées à une séquence effective, doivent être aptes à interagir avec celui qui a ces comportements (humains)?

Plus généralement, l'un des domaines importants de recherche en informatique cognitive est celui de la représentation des connaissances qu'il faut construire pour permettre à la machine d'effectuer une tâche de type intelligence artificielle. Pour ce faire, il existe plusieurs approches. Quelle que soit celle retenue, il faut d'abord savoir quelle est sa **représentation**, ensuite, savoir comment **transférer** ces connaissances à la machine pour qu'elle puisse **apprendre** afin d'améliorer sa performance. Ce transfert relève de trois aspects fondamentaux de notre thèse : représentation des connaissances, acquisition des connaissances et recherche d'information.

Notre recherche ne vise cependant pas des réponses abstraites. Elle les explore directement dans un domaine spécifique où ces connaissances sont éminemment requises et doivent être apprises: le dialogue oral humain-machine qui se réalise dans un système de dialogue oral humain-machine. Dans ce système, nous nous concentrerons sur l'aspect stratégique du dialogue considéré comme étant important pour contrôler les interactions orales entre l'humain et la machine [ENG05], [HEN05], [LEV00], [PIE04], [SCH05], [SCH06], [SCH99]. Une stratégie optimale du dialogue permet à la machine de contrôler efficacement ces interactions orales. Nos hypothèses de solutions à appliquer dans le domaine du dialogue sont :

- 1- La connaissance que représente la stratégie du dialogue peut être vue comme un processus dynamique qui est composé de séquences d'actions. Il traduit un comportement stochastique, dynamique et coopératif avec lequel deux interlocuteurs s'engagent dans un dialogue. Ce processus peut être décrit comme le processus de décision de Markov (état, action, transition, récompense).

- 2- Pour transférer cette connaissance de l'humain à la machine, nous proposerons une approche d'apprentissage par renforcement avec l'aide d'un utilisateur simulé (Pietquin et Beaufort, 2005 ; J. Schatzmann et al., 2006). Cette approche permet à la machine d'acquérir les connaissances des stratégies optimales du dialogue.
- 3- Cette stratégie optimale est le résultat, non seulement de la formalisation des connaissances et de l'apprentissage, mais est aussi obtenue grâce à la recherche d'information qui se manifeste à travers un mécanisme de transition entre les états qu'offre le modèle de Markov.

Nous démontrerons la faisabilité de notre approche par la réalisation d'un prototype qui met en évidence le processus d'apprentissage proposé. Le dialogue issu de cet apprentissage sera simulé oralement par une interface Windows programmé en Visual Studio C++/.Net dans le laboratoire de R&D de Nuance Communications, Inc. Ainsi, nous démontrerons qu'un système avec une stratégie apprise, modélisée selon nos hypothèses de représentation et d'acquisition des connaissances sera meilleur qu'un système sans stratégie apprise grâce à cette approche.

Mots clés : Représentation des connaissances, acquisition des connaissances, recherche d'information, contrôle des interactions, intelligence artificielle, processus de décision de Markov, apprentissage machine par renforcement, stratégie du dialogue, système de dialogue oral humain-machine, utilisateur simulé.

ABSTRACT

Speech recognition and speech synthesis techniques have become increasingly efficient and robust, facilitating implementation of human-machine spoken dialog systems. In these applications, a machine *speaks* to a human by imitating human communication acts. However, human machine dialogs still lack naturalness and flexibility. One of the most important issues in this domain is the control and management of conversational interactions between the machine and the human. These interactions do not occur randomly, but rather follow precise rules of the communication acts. While some research is focused on the acoustic and semantic aspects of speech signals (what to say), other is directed towards dialog strategies (how to say) in order to control these interactions. A number of machine learning approaches for the design of such strategies have been proposed in literature. One recent promising technique is reinforcement learning (RL) involving a simulated user (SU). With RL, a machine could develop an optimal strategy from observation examples, provided that they are comprehensive. However, in the current state of the art, it is generally not possible to produce such a strategy by directly learning from corpora of dialog data (Schatzman et al., 2006), mainly due to their small sizes, insufficient to permit exploration of all possible states and actions pertinent to a dialog. In addition, it is not certain that an optimal strategy is present in those corpora even when they are of reasonable sizes. Hence the idea of creating a simulated user to assist learning. In our implementation, we model dialog acts on Markov properties (actions, states, and transitions). When these properties are satisfied, the resulting dialog strategy is called a Markov Decision Process (MDP) (Sutton and Barto, 1998).

The main feature of our architecture resides in a novel objective function that achieves optimal dialog strategy based on quality of conversation, rather than its "quantity" (or duration), similar to what is proposed in. This quality could be measured through the variation of illocution questions (such as direct, implicit, explicit and repetitive questions) that the machine poses to the human interlocutor. In this thesis, we will describe the MDP theory as used in dialog systems and summarize the RL technique. We will also detail our proposed RL architecture with a simulated user, including the parameterization of our objective function and the initialization of the MDP reward variables, all necessary for satisfactory unsupervised learning. And finally, we will document the results of an implementation in a laboratory of our method and conclude our proposal and suggest new directions for future research.

Keywords: Knowledge representation, knowledge acquisition, information retrieval, interaction control, artificial intelligence, Markov decision process, reinforcement learning, dialog strategy, spoken dialog system, simulated user.

CHAPITRE I

PRÉSENTATION DU SUJET DE THÈSE

I.1 Introduction

Les techniques de reconnaissance et de synthèse vocale deviennent de plus en plus performantes et robustes, ce qui facilite la création des applications de dialogue oral humain-machine. Cependant, le dialogue entre l'humain et la machine manque encore de naturel (i.e., en termes de suivi conversationnel en situation dialogique), de souplesse, et la machine réagit aussi de façon trop rigide. Alors que les recherches se sont orientées vers l'aspect acoustique et sémantique du signal de la parole, d'autres chercheurs comme Levin et al. (1998, 2000), Litman et al. (2000), Pietquin et Dutoit (2006), Schatzmann et al. (2005,2006) [LEV00], [PIE06], [SCH05], [SCH06] se sont positionnés au niveau du processus d'apprentissage des stratégies optimales du dialogue à la machine, considéré comme étant important pour contrôler efficacement ces interactions orales.

Ces approches présentent des solutions prometteuses mais elles font aussi face à des problèmes majeurs. D'abord, il n'existe pas de modèle de stratégie optimale qui soit adaptée à tous les domaines. Schatzmann et al. (2006) ont remarqué que, dans les corpus de dialogue humain-machine, on peut avoir des indications quant à la réussite ou à l'échec du dialogue, mais on ne peut pas savoir laquelle des actions exécutées par la machine est la plus efficace (ou la plus optimale) à un moment du dialogue. On ne peut pas savoir quelle est la stratégie optimale étant donné que l'on n'a pas d'exemples typiques de ce qu'est un dialogue réussi. De plus, développer des applications de dialogue nécessite énormément de corpus de données en parole et,

d'expertise en dialogue. Chaque application est souvent unique, il est donc difficile de concevoir une nouvelle application à partir de l'existence sans avoir tout à recommencer.

L'objectif de cette recherche est de proposer un modèle de stratégie optimale du dialogue et un modèle d'apprentissage machine. Le modèle d'apprentissage machine est basé sur une méthode d'apprentissage par renforcement à l'aide d'un utilisateur simulé, dans laquelle l'utilisateur simulé apprend à la machine des stratégies optimales du dialogue. Ces stratégies sont conçues pour être adaptées dans un système de dialogue de type questions-réponses s'orientant vers un but.

Dans ce chapitre, nous présenterons d'abord les aspects généraux du domaine de système du dialogue et les problématiques du domaine, ensuite, nous proposerons nos hypothèses de solution, suivi de l'objectif de la recherche. Enfin, nous décrirons notre méthode de travail pour une recherche applicative et expérimentale ; le plan général des chapitres de cette thèse sera aussi présenté.

I.2 Définitions

I.2.1 Système de dialogue oral humain-machine

Un Système de Dialogue Oral Humain-Machine (SDOHM)¹ est un système qui permet d'échanger des informations par voix orale entre un agent humain² et un agent machine³ de façon naturelle et souple. Par naturel et souple, on implique la capacité de l'agent machine à raisonner logiquement, à produire des réponses cohérentes en contexte dialogique par rapport aux demandes de l'agent humain et ceci à la manière des humains. Autrement dit, un système qui dialoguerait avec l'humain en imitant l'acte de communication des humains. Il existe des SDOHM de

¹ Certains auteurs emploient les termes *homme-machine*, *humain-ordinateur*, *personne-machine*, *personne-ordinateur*, *humain-machine* pour désigner la relation de communication et d'interaction entre un agent humain et un agent machine. Ainsi, dans notre contexte de recherche, les termes ci-dessous ont la même signification : système de dialogue oral humain-machine, système de dialogue oral, système de dialogue, système, agent machine ou simplement machine.

² Agent humain, utilisateur ou usager désignent la même entité.

³ Voir 1.

type questions-réponses (i.e., le système et l'utilisateur s'échangent des questions et réponses), et qui s'orientent vers un but (i.e., le système a un but à accomplir, tant que le but n'est pas atteint le dialogue continue) ou vers une tâche (i.e., le système a une tâche à exécuter selon les ordres de l'utilisateur).

I.2.2 La recherche en systèmes de dialogue

La recherche en SDOHM s'inscrit bien dans une perspective multidisciplinaire où l'intelligence artificielle (IA) (logique propositionnelle, planification système, apprentissage machine, etc.), l'informatique (conception et réalisation des logiciels vocaux), la psycholinguistique (représentation des connaissances, théorie de représentation du discours, l'acte de dialogue et de parole) et la physique (acoustique, traitement numérique du signal) entrecroisent leurs connaissances dans le but de produire une machine qui soit capable de dialoguer oralement avec des humains de la même façon que leurs dialogues. (figure. 1.1).

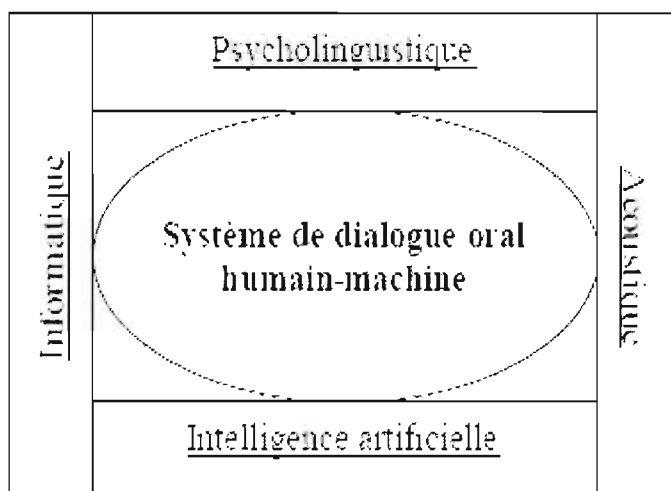


Figure 1.1 Domaine de recherche de dialogue oral humain-machine.

Tout système de dialogue oral humain-machine qui incorpore cette multidisciplinarité pourrait être considéré comme un système cognitif. En effet, la science cognitive définit un système cognitif comme étant un processus dynamique

d'apprentissage, d'adaptation et d'interaction avec un environnement (Meunier, 2005) [MEU05].

I.2.3 Stratégie optimale du dialogue

On a considéré, depuis longtemps, que la recherche d'une stratégie optimale du dialogue est un art plutôt qu'une science (Jones et Galliers, 1996) [SCH06]. Bien que le terme stratégie ait été employé par plusieurs auteurs, il n'existe pas de définition claire d'une stratégie efficace de dialogue (Schatzmann et al., 2006; Pietquin, 2005) [SCH06], [PIE05]. Une stratégie du dialogue, selon Austin, expliqué par Minker (2000) [MIN00], est un « *enchaînement des actes de langage dans le dialogue* ». Ces enchaînements se font à travers des actions, des exécutions et des décisions. Une **stratégie de dialogue** est dite **optimale**, si les actions exécutées satisfont le but recherché par l'agent humain. Il est donc nécessaire que les deux agents partagent le même but et que la recherche de stratégie optimale consiste à trouver les actions appropriées qui orientent le dialogue vers ce but. En d'autres mots, la stratégie optimale de dialogue est la manière de conduire un dialogue de façon souple et naturelle (au sens de suivi conversationnel) entre deux interlocuteurs. Le problème est de savoir quelle stratégie est appropriée par rapport à quelle situation dialogique. Par exemple, si la machine n'est pas sûre de « *comprendre* » la réponse de l'utilisateur, elle repose la question. Lorsque la réponse est ambiguë (e.g., situation liée au langage spontané : phrases agrammaticales, hésitations, anaphores, ellipses, etc.), elle ne doit pas répéter sans cesse la même question, sinon le dialogue serait agaçant. La machine doit '*savoir bien parler*'. En effet, c'est la rhétorique du discours que la machine doit apprendre à maîtriser. Cette rhétorique est en quelque sorte appelée *la stratégie du dialogue*. La stratégie est optimale si cela permet à une application de dialogue de mener un dialogue efficace, cohérent, fluide pour atteindre le but déterminé par l'application.

I.2.4 Apprendre une stratégie optimale à la machine

De manière générale, le terme apprentissage de la machine désigne une technique permettant à la machine d'acquérir de nouvelles connaissances et

d'optimiser sa performance. Apprendre signifie, pour la machine, améliorer la tâche d'exécution (Russell, 2003) [RUS03]. Il y a apprentissage si -et seulement si- la machine maîtrise mieux sa tâche d'exécution en fonction de ce qu'elle a apprise, lui permettant ainsi d'améliorer sa performance. De façon plus concrète, un SDOHM ayant un composant lui permettra d'apprendre une stratégie optimale de dialogue. Cela signifie que sa tâche d'exécution sera de trouver une meilleure façon de dialoguer pour conduire le dialogue vers l'objectif (but) qu'on lui a confié. S'il arrive à optimiser sa performance, il acquerra une stratégie optimale (i.e., nouvelles connaissances) et ainsi le système contrôlera mieux ses interactions orales avec l'humain. Pour réaliser cette performance, on implante une méthode d'apprentissage dans le système (i.e., apprentissage à partir des exemples d'observation, apprentissage supervisé/non supervisé ou apprentissage par renforcement avec ou sans modèle et/ou avec ou sans utilisateur simulé). Après une période d'apprentissage, le système acquerra des nouvelles capacités de contrôle et réajustera son comportement face à une situation nouvelle ou déjà apprise.

I.3 Problématique

Dans un dialogue oral entre deux personnes, l'aspect interactif est important, c'est-à-dire l'enchaînement du dialogue, la coordination des tours de parole et le but recherché par les deux interlocuteurs. Ils n'ont peut-être pas le même but mais ils ont certainement le même désir de se parler, sinon le dialogue n'aura pas lieu. Prenons l'exemple d'un client qui appelle un centre de réservation de chambres d'hôtel. Le préposé, après l'avoir écouté, doit réfléchir et essayer de comprendre et d'interpréter la demande du client pour lui donner les réponses qu'il attend. L'interaction entre ces interlocuteurs ne se fait pas par hasard, mais ces échanges se réalisent dans un ordre bien contrôlé selon des règles sociales de communication. De manière générale, cet échange peut être décrit comme une suite d'actions se succédant ainsi : à l'ouverture du dialogue, le client ou le préposé commence le dialogue par une formule de politesse (e.g., « *Bonjour, je vous appelle pour...* »), vient ensuite l'objet de l'échange où les deux interlocuteurs s'interrogent, négocient, partagent des idées (e.g., « *Avez-vous des chambres disponibles pour ce week-end? Je voudrais une*

chambre moins chère »). Enfin le dialogue se termine par une formule de politesse marquant la fermeture du dialogue (e.g., « *merci, au revoir* »). Le dialogue ne peut être réussi sans que le but du dialogue soit atteint (e.g., les deux interlocuteurs ne sont pas forcément d'accord sur une entente, mais leur échange s'est déroulé selon des règles sociales de communication).

L'exemple ci-dessus montre qu'il existe certaines règles régies dans un dialogue d'humain à humain. Ces règles de dialogue en science du langage montrent les principes de communication entre les humains comme les théories de jeux de langage de Wittgenstein (1953) [WIT53], théories des actes de langage d'Austin (1962) [AUS62] et de Searle (1969) [SEA69], théories sur les maximums conversationnels de Grice (1975, 1989) [GRI75], [GRI89], théories de la pertinence de Sperber et Wilson (1995) [SPE95], de l'attention, de but ou de l'intention de Grosz et Sidner (1986) [GRO86].

Lesquelles de ces théories peuvent être appliquées dans le dialogue humain-machine ? Si elles étaient applicables, comment les appliquer ? Par des méthodes d'apprentissage de machine ou méthode classique (i.e., traduire ces règles en des fonctions computables et les programmer dans la machine) ? Comment coordonner les tours de parole pour avoir un dialogue plus naturel ? Comment anticiper les questions des utilisateurs afin de lui fournir des réponses adéquates, c'est-à-dire deviner et reconnaître l'intention des utilisateurs ? Comment conduire correctement un dialogue ? Quelle est la stratégie de dialogue à employer pour maintenir la conversation afin d'atteindre le but ? Comment améliorer l'interaction de la machine lorsque la reconnaissance de la parole est médiocre ? Et dans cette condition, comment améliorer l'efficacité de la machine en agissant sur l'aspect stratégique du dialogue en tenant compte de la performance actuelle des systèmes de dialogue ? Tels sont les questionnements auxquels nous tentons d'apporter des solutions dans cette thèse.

I.3.1 Système de dialogue oral humain-machine

En général, un SDOHM est composé de cinq modules qui se succèdent séquentiellement : reconnaissance de la parole⁴, compréhension (interprétation) de la parole, gestion du dialogue, génération de la parole, et enfin production du signal acoustique (synthèse de la parole). Chaque module fait partie de l'objet de la recherche. Par exemple au niveau de la reconnaissance de la parole, la difficulté se trouve dans la sensibilité du traitement du signal acoustique face à l'environnement (e.g., bruit ambiant, bruit du canal de transmission et de microphone, problèmes d'interlocuteur et intra-locuteur⁵). Il faut des modèles acoustiques plus robustes et des modèles linguistiques plus pertinents (e.g., ambiguïtés, anaphores, ellipses, etc.). Pour le module de gestion du dialogue, le dialogue est encore rigide, répétitif, les tours de parole sont pauvres et les confirmations sont agaçantes (i.e., trop de questions explicites, implicites ou répétitives). Pour le module de compréhension de la parole, il y a encore trop de mots de vocabulaire hors contextes à interpréter, trop peu de modèles linguistiques qui soient capables d'interpréter la sémantique et la pragmatique des énoncés dans un large domaine.

I.3.2 Sciences du langage

Au niveau des sciences du langage, le dialogue est dynamique et imprévisible (e.g., changement d'idées en cours de dialogue). On peut produire, à partir d'un nombre défini de mots, un nombre infini de phrases qui donnent une infinité de significations. De plus, l'aspect spontané du langage oral contient des hésitations, des autocorrections, des anaphores et des ellipses. Cela crée des situations dialogiques imprévisibles pour un SDOHM dont il faudrait tenir compte. En fait, ce système doit, à partir des structures de surface (i.e., syntaxe, grammaire, contextuel,

⁴ Par abus du langage, nous utiliserons l'expression *reconnaissance de la parole* pour désigner une fonction de transformation des séquences de son en séquence de caractères réalisée par la machine. En effet, la machine ne reconnaît pas la parole, elle reconnaît seulement les sons de la parole. Mais, nous disons *reconnaissance de la parole* ou *reconnaissance du signal de la parole* pour alléger le texte.

⁵ Les problèmes d'interlocuteur sont liés aux différentes morphologies entre les personnes (e.g. grande taille, voix aiguë, voix grave, etc.). Les problèmes d'intra-locuteur sont dus au changement biologique de la personne (e.g. rhumatisme, stress, émotion forte, etc.).

etc.), trouver les structures profondes (i.e., sémantique, pragmatique) et donc deviner l'intention du locuteur.

I.3.3 Contraintes cognitives

La machine ne peut dialoguer avec l'humain à leur manière car elle n'a pas la même capacité intellectuelle. Par conséquent lorsqu'on développe une application en dialogue oral homme-machine, on doit penser à ces *contraintes cognitives*. Ce sont les capacités limitées de la machine à interpréter les messages sonores : perception et reconnaissance des mots, interprétation sémantique, génération des énoncés et reproduction du signal de la parole. La manière dont un système dialogue avec l'humain dépend grandement de ces contraintes. Plus il maîtrise ces contraintes, plus sa performance sera efficace.

I.4 Hypothèses de réponses

Compte tenu des problématiques connues du SDOHM, et surtout compte tenu des contraintes cognitives citées ci-dessus, il n'est pas possible, à l'heure actuelle, de réaliser un système de dialogue parfait et répondant à tous les domaines (Minker, 2000) [MIN00]. Cependant, en réduisant son champ d'action, en augmentant son vocabulaire, en définissant son but et en limitant le domaine d'application, on peut arriver à réaliser des systèmes de dialogue intéressants avec un comportement dialogique proche de celui des humains.

Par conséquent, nous devons d'abord examiner quatre aspects du problème de contrôle des interactions orales entre l'humain et la machine. D'abord, nous regarderons les théories de **dialogue humain-humain** pour comprendre les principes et les règles de ce contrôle. Comment un dialogue s'établit entre humains? Ensuite, nous regarderons la manière dont ces principes pourraient être appliqués dans un **dialogue humain-machine**, c'est-à-dire du côté de la machine, comment ces règles peuvent-elles être représentées et traitées? Ainsi, nous examinerons la possibilité d'apprendre ces règles à la machine pour qu'elle puisse produire une stratégie optimale afin de rendre le système de dialogue plus performant. En d'autres termes, comment peut-on faire **l'apprentissage** de ces règles à la machine ? Enfin, pour

mettre en œuvre une telle approche, il faut prévoir une **plate-forme informatique** flexible, adaptative et interopérable afin de rendre la machine exploitable, évolutive et performante, et d'assurer un dialogue souple et naturel.

Nous proposons donc quatre hypothèses de solution pour améliorer la performance du système de dialogue de type questions-réponses (i.e., le système pose des questions et l'utilisateur donne sa réponse) et qui orientent vers un but. Ces hypothèses nous amènent à élaborer un modèle de stratégie optimale du dialogue et un modèle d'apprentissage machine :

1-Concernant le dialogue humain-humain, nous prenons les théories des actes de langage et de dialogue d'Austin (1962) [AUS62], de Searle (1969) [SEA69], Grice (1975) [GRI75] et de Vanderveken (1991) [VAN91]. Nous employons les variations de l'acte illocutoire pour prévenir et empêcher les malentendus et les erreurs de compréhension. Nous avons choisi cinq variations de questions (force illocutoire) que la machine emploie dans son dialogue. Ces variations sont :

Question directe (i.e., *Quand viendras-tu me voir ?*)

Question implicite (i.e., *Avec qui viendras-tu me voir ce soir ?*)

Question explicite (i.e., *Viendras-tu me voir ce soir ?*)

Question répétitive due à un malentendu (i.e., *Je n'ai pas bien entendu, quand viendras-tu me voir ?*)

Question répétitive due à une incompréhension (i.e., *Excuse-moi, je ne comprends pas, répète ce que tu as dit.*)

2-La connaissance que représente la stratégie du dialogue peut être vue comme un processus dynamique composé de séquences d'actions. Il traduit un comportement stochastique, dynamique et coopératif avec lequel deux interlocuteurs s'engagent dans un dialogue. Ce processus peut être décrit par le processus de décision de Markov (état, action, transition, récompense). Puisque le problème de

recherche de stratégie optimale du dialogue peut être vu comme un problème d'optimisation (Biermann et Long, 1996; Levin et al., 2000; Walker et al., 2000; Pietquin, 2004; Cuayahuilt et Renals, 2006, Schatzmann et al., 2006) [BIE96], [LEV00], [WAL00], [PIE04], [CUA06], [SCH06], le choix du modèle du processus de décision de Markov (PDM) pour représenter les actes de dialogue comme des connaissances nous paraît pertinent, approprié et justifié.

3- Pour transférer cette connaissance de l'humain à la machine, nous proposerons une approche d'apprentissage par renforcement avec l'aide d'un utilisateur simulé (Pietquin et Beaufort, 2005 ; J. Schatzmann et al., 2006). Cette approche permet à la machine d'acquérir les connaissances des stratégies optimales du dialogue. Nous avons choisi l'approche apprentissage par renforcement avec un utilisateur simulé car elle est cadrée avec le processus de décision de Markov. Elle constitue une des meilleures façons d'apprendre dans un environnement incertain et stochastique. Les actions de l'utilisateur elles-mêmes sont dynamiques et imprévisibles, ce qui a un impact direct sur les actions de la machine qui peuvent être différentes à chaque fois qu'on revient dans un même état. Le choix de l'utilisateur simulé au lieu de l'utilisateur réel pour faire l'apprentissage par renforcement vient du fait que l'apprentissage par renforcement demande des échanges d'essais-erreurs pour trouver les stratégies optimales. Ces échanges peuvent être longs et prennent énormément de temps d'exécution ; l'emploi des utilisateurs réels n'est pas souhaitable pour ce type d'apprentissage. De plus, le fait d'utiliser un utilisateur simulé nous permet de mettre les nouvelles connaissances dans l'utilisateur simulé. Ensuite, par un processus d'apprentissage par renforcement, nous transférerons ces connaissances à l'agent machine sans avoir besoin de tout recommencer car le mécanisme d'apprentissage a déjà été dans la machine. Il suffit donc d'enrichir sa connaissance pour qu'il soit opérationnel dans

un autre domaine. L'algorithme d'apprentissage est basé sur l'algorithme Q-Learning⁶.

4-Concernant la plate-forme informatique, nous avons choisi un environnement Windows et Unix pour réaliser notre prototype. Les langages C++/.Net et Perl⁷ s'avèrent être les meilleurs langages de programmation compte tenu de leur richesse, de leur puissance et de leur performance en termes de manipulation des objets et de structures complexes (i.e., fichiers, tableaux multi-dimensions, variables, etc.).

I.5 Objectifs de recherche

Notre principal objet de recherche est de proposer un modèle de stratégie optimale du dialogue et un modèle d'apprentissage par renforcement à l'aide d'un utilisateur simulé pour apprendre à la machine à acquérir cette stratégie qui doit être formalisée et exprimée sous forme de règles, de principes et des critères de mesure (évaluation) afin de pouvoir l'appliquer dans le modèle d'apprentissage. Un sous objet de recherche s'impose alors : celui de définir en premier quels sont les règles, les principes et les critères de mesure pour une stratégie optimale.

Les stratégies du dialogue sont employées pour résoudre les problèmes rencontrés dans un dialogue spontané tels que la prévention, l'empêchement ou la réparation des malentendus, la détection, la correction et la clarification des erreurs. Parmi ces problèmes décrits ci-dessus, nous nous concentrerons sur la stratégie de prévention et d'empêchement des malentendus et des erreurs de compréhension durant le dialogue sans diminuer la qualité de celui-ci. Lorsque le modèle de stratégie optimale sera défini, on l'insérera dans la machine par l'apprentissage par renforcement. Pour parvenir à cet objectif, nous nous fixerons deux sous objectifs : informatique et cognitif.

⁶ L'algorithme Q-Learning permet de trouver l'action optimale en réalisant les expériences des essais-erreurs tout en poursuivant la stratégie en cours, ainsi l'agent peut avoir un comportement cohérent pendant l'apprentissage [PIE04].

⁷ Perl : www.perl.com

I.5.1 Objectif informatique

Nous développerons un prototype d'agent d'apprentissage (agent apprenant) qui soit capable de maîtriser une nouvelle stratégie de dialogue. Puis nous utiliserons le processus de décision de Markov pour représenter l'enchaînement des actes de dialogue. Le prototype développé permettra de démontrer qu'un système de dialogue qui possède une stratégie apprise sera meilleur. Ensuite, nous illustrerons la démonstration en créant deux versions du prototype : une sans apprentissage et une avec apprentissage. Cela permettra de constater la différence entre un système sans aucune stratégie apprise et un système avec une stratégie apprise.

I.5.2 Objectif cognitif

En se basant sur le dialogue entre les humains, et sur les théories d'apprentissage par renforcement, nous voulons reproduire certains comportements dialogiques des humains dans la machine, acquis par un processus cognitif d'apprentissage à travers un agent apprenant. La question qui se pose en informatique cognitive, dans cette perspective, est de savoir comment un système informatique (machine) peut apprendre des séquences de comportements. Ce sont des énoncés du langage naturel qui, lorsque confrontés à une séquence effective, doivent être aptes à interagir avec celui qui a ces comportements (humains).

En effet, l'un des domaines importants de recherche en informatique cognitive est celui de la représentation des connaissances qu'il faut construire pour permettre à la machine d'effectuer une tâche de type intelligence artificielle. Il existe plusieurs approches pour ce faire. Quelle que soit celle retenue, il faut d'abord savoir quelle est sa **représentation**, ensuite, il faut savoir comment **transférer** ces connaissances à la machine pour qu'elle puisse **apprendre** afin d'améliorer sa performance. Ce transfert relève de trois aspects fondamentaux de notre thèse : représentation des connaissances, acquisition des connaissances et recherche d'information.

I.6 Contexte du projet de recherche

Cette recherche est menée sous la direction du professeur Jean-Guy Meunier du département de philosophie de l'UQAM, du professeur Douglas O'Shaughnessy de l'INRS-EMT et en collaboration avec la société Nuance Communications Inc. La recherche est subventionnée par des bourses de l'INRS-EMT. La société Nuance Communications, Inc. met à notre disposition les matériels informatiques pour le développement.

I.7 Cheminement méthodologique de la thèse

Pour atteindre l'objectif de notre recherche, il faut avoir une bonne aptitude intellectuelle et une solide méthodologie de travail. Pour l'aptitude, nous avons choisi la méthode triangulaire (figure 1.2). Pour la méthodologie, nous avons adopté le modèle Spirale de Boehm (1988) [BOE88] (figure 1.3) pour organiser un plan directif de projet de recherche.

I.7.1 Modèle triangulaire

Ce modèle s'appuie sur un processus de transition d'états mentaux et une démarche cognitive pour une recherche expérimentale qui consiste à penser et agir selon une stratégie de réflexion en trois étapes : observer, analyser, modéliser.

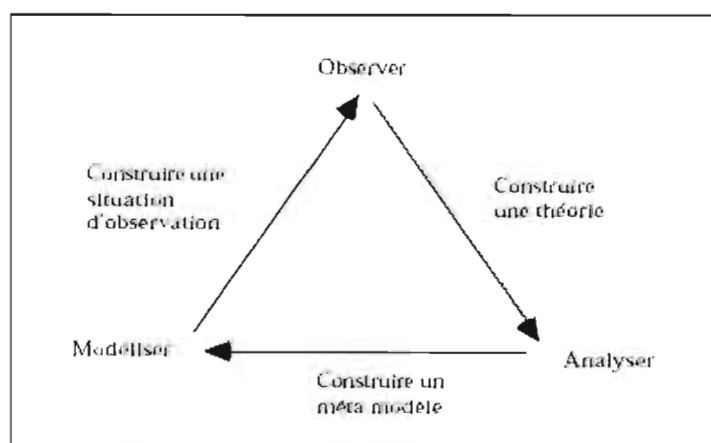


Figure 1.2 Démarche expérimentale et processus de transition (extrait de l'exposé de Nicolle, 2001)⁸.

I.7.2 Modèle spirale

Nous avons choisi un modèle permettant de planifier notre activité de recherche tout en répondant aux exigences du développement des projets informatiques. Le choix du modèle Spirale de Boehm (1988) [BOE88] s'avère judicieux. Ce modèle, en effet, nous offre la possibilité de décrire toutes les étapes clés du processus de conception et de réalisation d'un produit informatique (Fig. 1.3) (définition du problème, analyse de la faisabilité, conception, réalisation, test, évaluation, etc.).

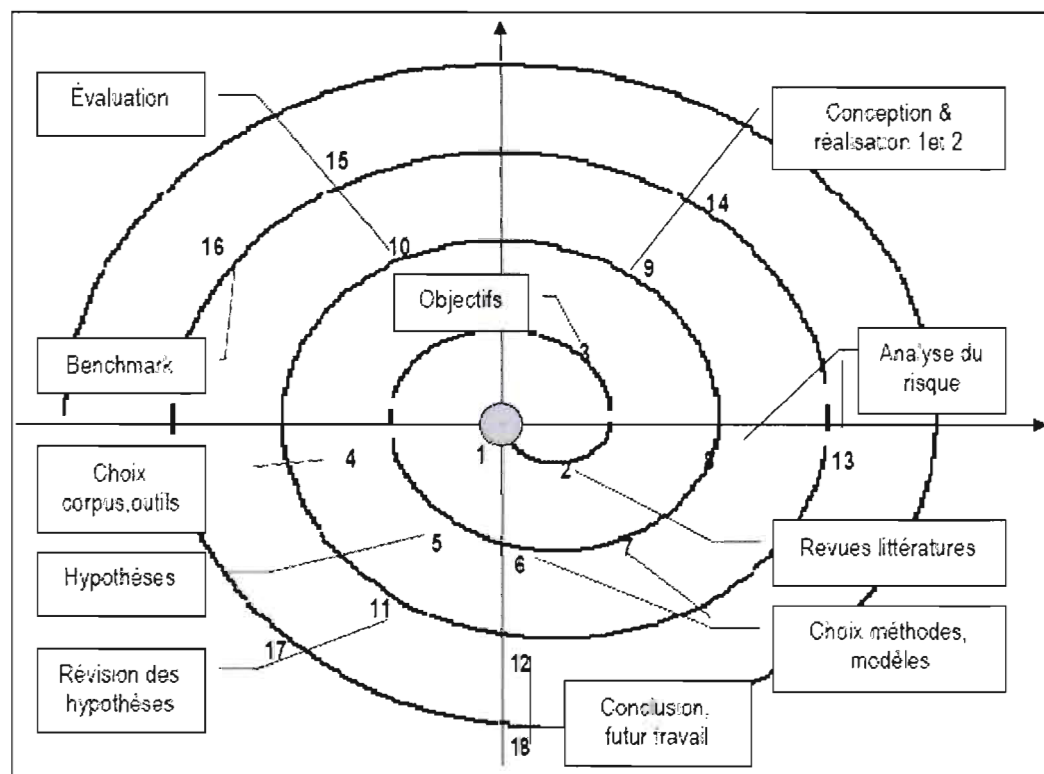


Figure 1.3 Cheminement méthodologique de la recherche.

⁸ Exposé d'Anne NICOLLE, professeur à l'Université de Caen – CNRS, 'Quels modèles de la langue et de l'activité langagière pour le dialogue personnes/machines ?', 2001, disponible sur Internet : <http://www.limsi.fr/Individu/gs/AtelierModeles/Reuniondecembre2001/ExposeAN.pdf>

La rédaction de la thèse se fait au fur et à mesure de l'avancement des travaux. Toutes les tâches en dessous de l'abscisse (ligne horizontale) font partie en général des recherches théoriques et celles au-dessus se réfèrent à la conception et à la réalisation du prototype.

Après avoir choisi le domaine de recherche (Système de dialogue oral humain-machine) (étape 1) et la revue de littérature (étape 2), nous sommes en mesure, dans l'étape 3, de fixer notre objectif de recherche. Dans l'étape 4, le but est de faire le tour des produits existants sur le marché. Cette étape nous permet de choisir le corpus et les outils informatiques ainsi que de faire quelques expériences de programmation afin d'appréhender le degré de difficulté du domaine. Les objectifs de recherche, maintenus par des hypothèses émises dans l'étape 5 en regardant les contraintes et avancements des travaux de recherches actuelles par d'autres chercheurs, seront révisés ou changés en fonction des résultats obtenus dans les étapes de développement. Les étapes 6 et 7 suivantes consistent à choisir les méthodes et modèles pour la résolution et l'évaluation du produit. Les étapes 8 et 13 permettent d'analyser la faisabilité du prototype, c'est-à-dire analyser les risques en estimant le temps de réalisation en fonction des outils et méthodes choisis et éventuellement, poser les contraintes en cas de non atteinte des objectifs ou dépassement du temps. Les étapes suivantes (étapes 9, 14) concernent la conception et la réalisation des produit 1 et produit 2. Après chaque réalisation, nous devons évaluer nos produits (étapes 10, 15). Cette évaluation nous permet de réviser nos hypothèses de départ (étapes 11, 17). L'étape 12 est la conclusion du premier travail. La conclusion finale et les réflexions sur les futures recherches (étape 18) se suivent en respectant la même démarche méthodologique que les étapes précédentes.

I.8 Organisation de la thèse

La thèse peut être lue dans un ordre tel que présenté dans la figure 1.4. Le chapitre I présente le sujet de thèse, les chapitres II, III, IV et V présentent l'état de l'art de la recherche. On peut lire indépendamment les chapitres II, III et V. Le chapitre VI est le cœur de notre thèse et les chapitres suivants sont la suite logique du chapitre VI.

Chapitre I : Présentation du sujet de la thèse.

Chapitre II : Dialogue oral humain-humain.

Chapitre III : Système de dialogue oral humain-machine.

Chapitre IV : Gestion du dialogue oral humain-machine.

Chapitre V : Apprentissage de la machine.

Chapitre VI : Modélisation du processus d'apprentissage pour une stratégie optimale du dialogue.

Chapitre VII : Modélisation informatique.

Chapitre VIII : Expérimentation et résultats.

Chapitre IX : Évaluation.

Chapitre X : Contribution, discussion et conclusion.

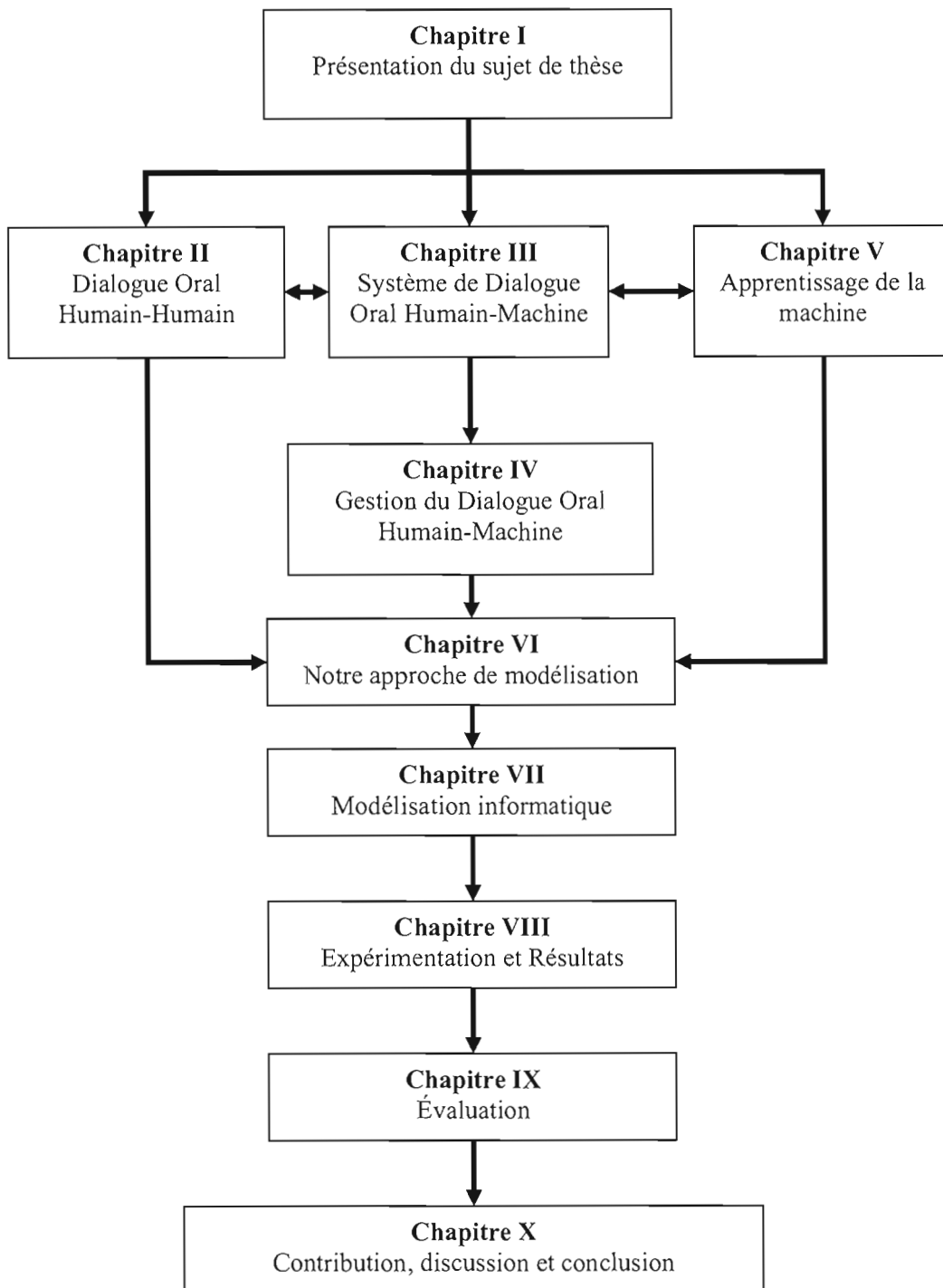


Figure 1.4 Organisation de la thèse.

CHAPITRE II

DIALOGUE ORAL HUMAIN-HUMAIN

II.1 Introduction

Ce chapitre a pour but de présenter les aspects cognitifs du dialogue. Nous commencerons par rappeler les définitions des termes employés dans la linguistique pour ensuite présenter les trois modèles de dialogue qui sont plus classiques, à savoir le modèle structurel (linguistique), le modèle orienté plan ou orienté-tâche (IA, planification et reconnaissance de plan) et le modèle logique (logique modale¹ pour la représentation de l'état mental, des attitudes mentales des interlocuteurs et le raisonnement induit par ces attitudes) (Minker, 2000) [MIN00].

II.2 Définitions

II.2.1 Actes de langage

Depuis les travaux en philosophie d'Austin (1962) [AUS62], de Searle (1969) [SEA69] et de Grice (1989) [GRI89], on distingue trois composantes lors d'un acte de langage:

- l'acte *locutoire* qui est défini comme l'action d'articulation et de combinaison des sons, un simple mécanisme biologique que nous possédons tous dès l'enfance et, que nous faisons bien naturellement. L'acte locutoire représente le mécanisme de production de la parole à l'aide d'une structure grammaticale acquise (syntactique, lexicale et sémantique).

¹ La logique modale est une logique qui exprime les concepts tels que possible, impossible et nécessaire. Elle peut être représentée par la logique propositionnelle ou la logique du premier ordre. Disponible sur : http://en.wikipedia.org/wiki/Modal_logic [consulté le 4 mai 2006].

- L'acte **illocutoire** concerne l'action de l'interlocuteur qui produit un énoncé avec une intention. Il est caractérisé par une force illocutoire (phrase affirmative, interrogative, déclarative, etc.). Selon Meunier (2006) [MEU06], « *il est certain qu'un ordinateur ne peut identifier le véritable contenu signifiant de la plus petite des conversations entre lui et un utilisateur sans mettre en jeu ces dimensions illocutoires* ». Il nous semble que la théorie des actes de langage est omniprésente dans les traitements de langue et aussi dans le dialogue humain-machine puisque c'est par les traitements syntaxiques et grammaticaux que la machine parvient avec succès ou non à deviner l'intention de l'autre.
- Enfin, l'acte **perlocutoire** concerne les effets que les actes illocutoires peuvent avoir sur l'interlocuteur (e.g., faire changer sa croyance, ses actions ou son jugement).

II.2.2 Discours et dialogue

Un discours est un acte de parole produit par un seul locuteur alors qu'un dialogue est un acte de communication, un acte d'échange entre deux agents humains. L'étude du dialogue oral ou dialogue² relève des théories issues de la linguistique, de la philosophie, de la communication et de la psycholinguistique. Parmi ces études théoriques, il faut citer l'analyse des actes de langage qui étudie l'acte de parole (Searle, 1976) et l'analyse de la conversation qui considère le phénomène du dialogue comme une interaction sociale (Levin et al., 1983) [LEV83]. Ces théories contribuent plus ou moins au développement d'un système de dialogue oral humain-machine. Par exemple, au point de vue psycholinguistique, l'analyse du dialogue se concentre sur les problèmes du contexte et de l'interprétation pragmatique (Levin et al., 1983) [LEV83], c'est-à-dire, la façon dont la connaissance du monde peut être intégrée, comprise et traduite par les deux interlocuteurs (Chomsky, 1969; Saussure, 1974; Searle, 1976) [CHO69], [SAU74], [SEA76]. Alors qu'au point de vue communicatif, le dialogue peut être vu comme une théorie de

² Dans ce document, le terme dialogue est sous-entendu dialogue oral.

jeux où chaque interlocuteur cherche à deviner le but de l'autre à travers son action (i.e., donner une réponse ou poser une question) (Wittgenstein, 1957) [WIT57].

II.3 Dialogue humain-humain

Le terme **dialogue humain-humain** désigne un acte de communication et d'interaction entre deux humains. Chaque acte produit un énoncé. La séquence des énoncés émis successivement par deux interlocuteurs et régis par certaines règles constitue un dialogue. Un dialogue n'est pas une séquence aléatoire d'énoncés, mais un échange qui se réalise en respectant certains principes de base où chaque agent, à tour de rôle, prend alternativement le rôle de locuteur et d'auditeur. L'échange pourrait être concrétisé sous forme de dialogue oral, gestuel (écrit) ou multimodal (à la fois parole et gestuel). Mais pour que la communication puisse se réaliser dans de bonnes conditions, encore faut-il que les deux interlocuteurs utilisent le même et juste langage.

II.4 Utilisation du langage

Il y a deux aspects que les interlocuteurs doivent maîtriser lors de l'utilisation du langage :

- avoir acquis la connaissance de la grammaire, de la syntaxe, de ses règles grammaticales, du lexique, de la sémantique et de la pragmatique du langage et,
- savoir les utiliser correctement à travers la langue.

Cette application de la grammaire permet une utilisation adéquate de la langue dans les différentes situations dialogiques. Une fois la grammaire maîtrisée, vient la compréhension des énoncés, c'est-à-dire la sémantique, le sens propre d'un énoncé dicté dans un contexte neutre, un contexte où il n'y a pas d'ambiguïté. Par exemple, la phrase « *cette maison est belle* » peut être traduite dans la plupart des langues du monde, les gens peuvent la comprendre sans ambiguïté. Mais un énoncé « *comment allez-vous ?* » ne peut être traduit mot-à-mot en anglais « *how are you going ?* » et

n'a pas la même signification que « *comment allez-vous ?* » en français. Le sens propre de cette proposition nous renvoie à une autre signification qui ne peut être comprise que par des gens qui partagent la même culture, le même consensus social.

Le deuxième aspect permet de réaliser des actes de dialogue cohérents ; c'est l'utilisation de ses règles à travers la langue qui fait véhiculer la pensée. La façon d'employer la grammaire peut modifier le sens de la phrase. Les comédiens et les humoristes jonglent avec les mots pour nous faire rire ou pleurer. Si la grammaire aide à former des phrases correctes dans une langue, la compréhension et l'interprétation, elles, demandent une connaissance plus élargie que le simple contexte du discours dans lequel le dialogue est établi. Cette connaissance relève de la connaissance du monde que l'on réfère à la pragmatique.

II.5 Approches de modélisation du dialogue

La modélisation du dialogue permet de formaliser les actes de dialogue. La modélisation consiste à trouver les règles ou les principes de base qui régissent le dialogue afin que l'on puisse les distinguer et les employer dans les différentes situations dialogiques.

Nous allons présenter trois approches de modélisation du dialogue ci-dessous. Elles sont issues des études en linguistique et en intelligence artificielle (IA). Elles sont basées sur les différents cas observés en situation dialogique (échange d'information, discussion, confirmation, répétition, etc.). Le modèle structurel est fondé sur les théories en linguistique, le modèle orienté plan est issu des travaux en IA avec les notions de plan, de reconnaissance de plan et de la planification, et le modèle logique vient de la logique modale³ (représentation des attitudes mentales et les raisonnements induits de ces attitudes). Il est à noter que les modèles structurels et orientés plan sont concentrés sur la tâche et se placent du côté de la machine (les

³ La logique modale est une logique qui exprime les concepts tels que possible, impossible, nécessaire. Elle peut être représentée par la logique propositionnelle ou la logique du premier ordre. Disponible sur : http://en.wikipedia.org/wiki/Modal_logic [consulté le 4 mai 2006].

actions que la machine pourrait être amenée à faire) alors que le modèle logique se place du côté humain (cherche à reproduire son comportement et/ou son état mental).

II.5.1 Modèle structurel

Dans le modèle structurel, on considère l'acte de langage comme l'unité de base du dialogue au même titre que l'entité invariante au sens du phonème en phonétique ou du morphème en morphologie. La modélisation structurelle du dialogue se base sur les travaux linguistiques en modélisation hiérarchique du dialogue (échanges, interventions) et sur les règles de réécriture de grammaire formelle. La structure du dialogue est présentée sous forme d'une structure hiérarchique dont les terminaux sont les actes de langage, les non-terminaux sont des sous-dialogues (Minker, 2000) [MIN00]. Nous allons voir deux modèles structurels ci-dessous, l'un est le modèle de Genève développé par Roulet [ROU81] et l'autre est celui de Luzzati (1995) [LUZ95].

II.5.1.1 Modèle de Genève

Le modèle de Genève développé par Roulet en 1981 [ROU81], [ROU82] a pour but de distinguer les fonctions illocutoires des fonctions interactives des énoncés. Ce modèle consiste à décomposer la conversation en structure hiérarchique et fonctionnelle. La composante hiérarchique est formée de trois constituants : un échange, une intervention et un acte de langage (figure 2.1). Ces constituants entretiennent entre eux des relations fonctionnelles. Ainsi, les constituants du rang **échange** sont reliés entre eux par des fonctions illocutoires, alors que les constituants du rang intervention sont reliés entre eux par des fonctions interactives (Minker, 2000) [MIN00].

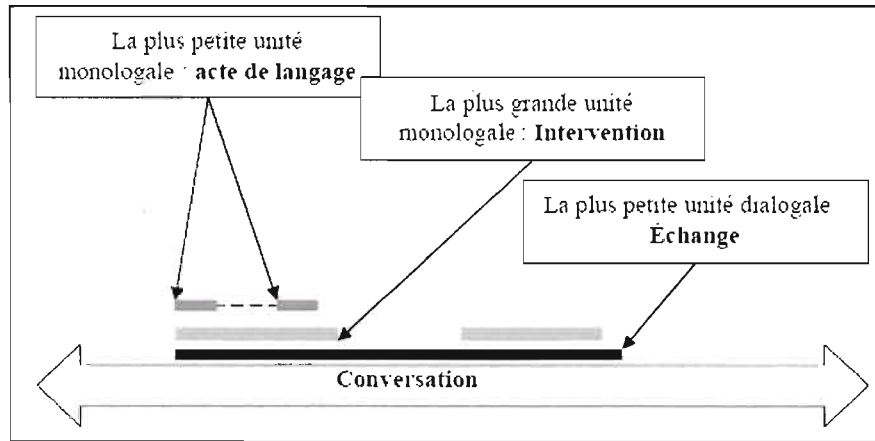


Figure 2.1 Unité de base de l'acte de dialogue.

Le modèle de Genève propose aussi une analyse dynamique de la conversation tout en imposant des contraintes conversationnelles telles que l'interaction (principes de nature sociale pour une bonne intervention), la structuration (contrainte structurelle pendant la conversation) et l'enchaînement (contrainte lors de l'interprétation et l'enchaînement de la conversation par exemple comment le système doit se comporter lors de l'ouverture, la fermeture et la poursuite de la conversation).

Conversation → Dialogue → Échange → [Intervention → Sous-dialogue] ²
Sous-dialogue → Sous-échange → [Intervention] ²

Tableau 2.1 Structure hiérarchique de la conversation selon le modèle de Genève.

Le modèle de Genève est appliqué dans le système d'interrogation de la base de données du personnel LOQUI (Ostler, 1989) [OST89], qui définit une structure hiérarchique de la conversation comme une séquence d'actes de langage (tableau 2.1). Une conversation (C) est composée des dialogues (D) eux-mêmes formés par au moins un ensemble d'échanges (E). L'acte de l'échange nécessite au minimum une intervention du locuteur et une intervention de l'auditeur (une paire d'intervention ou mouvement). Ce mouvement (M) est l'unité minimale de la conversation (l'acte de langage étant défini comme l'unité minimale). Chaque

mouvement peut être suivi ou non par plusieurs sous-dialogues (SD). Enfin, les sous-dialogues sont formés par des sous-échanges (SE) qui sont constitués d'une paire d'interventions (mouvement M). L'énoncé est donc soumis à des règles de réécriture [MOE89]. La règle de réécriture de cette structure hiérarchique est définie selon Minker (2000) (tableau 2.2) :

Conversation (C) = {D}+
Dialogue (D) = {E}+
Échange (E) = M {SD}* M {SD}*
Sous-Dialogue (SD) = {SE}+
Sous-Échange (SE) = M M
Mouvement ou Intervention (acte de langage) = M

Tableau 2.2 Règles de réécriture de la structure hiérarchique (d'après Minker, 2000).

Les accolades {} désignent un ensemble de symboles. Les + suivis après les {} signifient que les non-terminaux peuvent être répétés au moins une fois tandis que les * peuvent apparaître une infinité de fois ou jamais. Les terminaux (unités minimales) sont symbolisés par le caractère M (mouvement ou intervention). Après substitution des termes, la règle de réécriture de cette structure peut être présentée sous la forme suivante :

$$C = \{D\} + = \{E\} + = \{M \{SD\}^* M \{SD\}^*\}. \quad (1)$$

En substituant le terme SD par {SE} +, on obtient

$$C = \{M \{\{SE\} +\}^* M \{\{SE\} +\}^*\}. \quad (2)$$

En remplaçant {SE} + par M, on obtient

$$C = \{M \{M M\}^* M \{M M\}^*\}. \quad (3)$$

$$C = \{M M\}. \quad (4)$$

Si * est égal à zéro alors $\{M M\}^*$ n'apparaît pas dans la formule (3), on obtient donc l'acte de conversation sous sa forme minimale (4), qui représente une paire de mouvements dans un échange de dialogue. Donc, un acte de conversation est un acte de dialogue qui nécessite une intervention du locuteur et une intervention de l'auditeur (qui devient locuteur à son tour).

II.5.1.2 Modèle de Luzzati

Le modèle de Luzzati (1995) [LUZ95] distingue trois niveaux dans la conversation : un niveau principal, un niveau secondaire et un niveau incident. Le niveau principal représente la thématique des communications. Le niveau secondaire dépend du niveau principal. Il joue un rôle de complémentarité au niveau principal dans la limite où il peut ajouter ou modifier un élément de ce dernier. Le troisième niveau sert à clarifier un énoncé (demande de reformulation, de précision). Dans un dialogue dont l'enchaînement des actes de langage se fait par des liaisons entre les interactions, le modèle de Luzzati représente bien ces interactions. Luzzati modélise le dialogue sous deux axes : l'axe régissant (horizontal) qui correspond à une question principale et l'axe incident (vertical) correspondant aux questions-réponses incidentes. Le dialogue représenté par ces axes est un graphe sous forme d'escalier. L'application STANDIA⁴ (standard téléphonique intelligent-LIMSI-Communication Homme-Machine, 1992) qui gère à la fois le dialogue écrit et oral, est inspirée du modèle de Luzzati. Cette application prend en charge les fonctions élémentaires d'une standardiste téléphonique d'un laboratoire de recherche. Ces fonctions sont : l'aiguillage vers une personne demandée, l'information sur l'organisation du laboratoire ou des projets de recherche en cours et la gestion de messages vocaux et d'agendas individuels (Minker, 2000) [MIN00].

⁴ CASTAING M.F. STANDIA: an intelligent voice activated switchboard. Annual International Ergonomics Safety Conf. Denver, Colo, June 10-14, 1992. Paru dans : "Advances in Industrial Ergonomics & Safety IV". Taylor & Francis Ed., 1992.

II.5.2 Modèle orienté-plan

Le modèle orienté-plan ou orienté-tâche est celui qui reste le plus attaché aux applications du dialogue. Ce modèle est fait pour des applications bien définies. La tâche est vue comme une suite d'actions qui conduit à un ou plusieurs buts recherchés par l'agent humain. Selon Caelen (1994) [CAE94], une activité est une suite d'actes effectués par un utilisateur dans le cadre d'une tâche. La modélisation de la tâche consiste à modéliser les différentes connaissances d'une application donnée. La connaissance peut être représentée par des concepts, des objets, des relations et des règles d'inférence (déductive ou inductive) qui permettent d'engendrer de nouvelles connaissances. Mais, la modélisation de la tâche pourrait être aussi simplement une description des scénarios à exécuter (gestion des tâches) ou une description des objectifs à atteindre. Cette description est liée étroitement à la modélisation de l'intentionnalité des utilisateurs ou à leurs états mentaux. Cela indique que ce modèle utilise aussi une part du modèle logique (voir section modèle logique qui suit dans ce chapitre). Avant de présenter les modèles orienté-plan, il est nécessaire de comprendre le concept de base de la planification et comprendre comment ce concept est appliqué dans le dialogue. Nous allons voir le modèle de Cohen et Perrault (1979), ensuite celui de Litman (1984,1985) et de Ramshaw (1989) [COH79], [LIT84], [LIT85], [RAM89].

II.5.2.1 Concept de planification

Une planification est une tâche dans laquelle on s'exprime en termes de séquences d'actions qui sera acheminée vers un but. En IA, on utilise la logique propositionnelle et la logique du premier ordre pour représenter le problème de planification (M.Q. Nguyen et al., 2007) [NGU07a]. La résolution des problèmes par la planification requiert que l'environnement soit déterministe, discret (temps, actions, objets et effets), totalement observable et perceptible. Pour pouvoir construire un tel plan, il faut avoir une heuristique précise sinon il serait impossible de concevoir tous les plans. Ceci rend la planification limitée dans un SDOHM, car modéliser chaque phrase, chaque mot dans un dialogue revient à traduire une infinité de phrases qui ont les mêmes sens et expriment parfois les mêmes concepts.

Le modèle orienté-plan a été tiré de la théorie de la planification en IA et aussi de la théorie des actes de langage d'Austin. Dans ce modèle, on privilégie trois notions : le plan, la planification et la reconnaissance du plan. La notion de plan [HOC88] définie en psychologie cognitive est une représentation schématique de l'activité humaine. Un plan est une séquence d'actions qui consiste à transformer une situation initiale du monde vers une situation finale comme nous le montre la figure 2.2 où S_i représente la situation initiale, S_j la situation finale et $\{A_i\}$ un ensemble d'actions à exécuter.

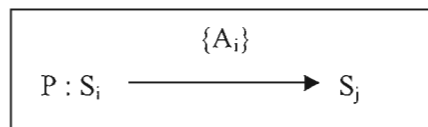


Figure 2.2 Notion de planification système.

La planification consiste à créer des actions pour atteindre le but (situation finale S_j). Les actions sont traduites à partir des énoncés émis par le planificateur à l'aide d'une base de référence. Enfin, la reconnaissance de ces plans est une opération déductive du but de l'agent humain à partir des actions créées par le planificateur. Pour reconnaître un plan, il faut que l'agent machine ait une connaissance du monde et des différents moyens d'atteindre le but. Cela signifie qu'à chaque instant dans le système, l'agent machine doit avoir un état mental actif dans lequel intervient la croyance, l'intention et la connaissance. Dans un modèle dit orienté-plan, on privilégie la recherche de l'intention des agents (but final), à l'acte de langage qui est traduit en tâches et sous-tâches. L'objectif est de mettre en œuvre une structure de plan qui représente les états initiaux et finaux ainsi que les actions associées à ces états. Cette structure permet de détecter l'intention de l'agent humain, de suivre les actions définies qui amèneront à la résolution du problème.

II.5.2.2 Modèle de Cohen et Perrault

Cohen et Perrault (1979) [COH79] ont initialement proposé un plan pour modéliser l'état mental des agents. En utilisant deux notions propositionnelles : la croyance et le vouloir pour représenter le changement d'état mental sous un

formalisme de langage STRIPS⁵ (Fikes/Nilsson, 1971). Cohen et Perrault ont représenté les actes de langage sous forme d'une séquence d'actions appelée schéma d'action. La figure 2.4 ci-dessous illustre une action de demande d'information : S veut que H fasse une action A. La pré-condition *Cando.Pr* (i.e., peut faire la proposition) est nécessaire pour que l'action *Request()* soit réalisée. S doit croire que H peut faire une action A. La pré-condition *want.Pr* exprime l'intention de S qui veut que H fasse l'action A. Le résultat est obtenu par l'effet H qui croit que S peut faire l'action A (Haller, 1999) [HAL99].

<u>Request(S,H,A)</u>	
Cando.Pr:	Believe(S, Cando(H,A))
Want.Pr:	Believe(S, Want(S, Request(S,H,A)))
Effect:	Believe(H, Believe(S, Want(S,A)))

Figure 2.3 Définition d'une requête du modèle de Cohen & Perrault (extrait de Haller, 1999) [HAL99].

II.5.2.3 Modèle de Litman

Le modèle de Litman [ALL79], [LIT84], [LIT85] est fondé sur trois catégories de plans : les plans du domaine (relatifs à l'application), les actes de langage (les aspects élémentaires de la communication) et les plans du discours qui consistent à mettre en évidence les phénomènes globaux du discours telle que la gestion des sous-dialogues de clarification (demande de clarification sur une question) [MIN00]. Les plans du discours sont des méta-plans utilisés pour créer d'autres plans afin de rendre le discours plus général et donc plus riche. La construction de méta-plans est un des points forts de ce modèle qui a été utilisé auparavant par Wilensky (1981, 1983) pour la compréhension des récits. Le modèle de Litman a été repris et amélioré par Nerzic (1993) pour traiter les erreurs d'attribution des intentions de l'agent humain par le système. Ces intentions sont

⁵ STRIPS : "Stanford Research Institute Problem Solver", un langage pour la représentation des conjonctions propositionnelles.

présentées par un ensemble de plans que le système essaie de valider par un apprentissage supervisé (catégoriser des données).

II.5.2.4 Modèle de Ramshaw

Ramshaw (1989) [RAM89a], [RAM89b] utilise des méta-plans pour identifier des formes manquantes provenant des erreurs de reconnaissance des mots. Ramshaw modélise les actions complexes du discours qui sont vues comme des processus de résolution des problèmes. Il propose quatre classes méta-plans de résolutions de problèmes : construction d'un plan, interrogation, évaluation et information du méta-plan.

Le travail de Ramshaw consiste à créer des méta-plans associés au contexte du discours. À partir de ces méta-plans, une instanciation de variables est générée (i.e., plan raffiné) et est disponible pour l'utilisateur. C'est grâce à cette instanciation que le modèle de Ramshaw est vu comme une sorte de générateur de plans. Ces plans représentent les actes de dialogue. Dans le cas où l'information venant des utilisateurs est manquante, le modèle utilise une recherche heuristique pour retrouver l'information. Ramshaw (1991) [RAM91] a modifié son modèle pour distinguer trois niveaux de domaine : un niveau d'exécution, un niveau d'exploration et un niveau de discours. Cette amélioration permet de séparer le plan du discours et le plan d'exécution (nous parlerons aussi de plan orienté-tâche). Quant au niveau d'exploration, ceci est simplement une tentative d'explorer une voie possible des actions à exécuter [BLA05].

Cependant, le modèle de Ramshaw a été l'objet de plusieurs critiques, d'abord, par (Carberry, Kazi, et Lambert, 1992 ; Elzser, 1995) [CAR92], [ELZ95] qui remarquent un blocage au niveau des résolutions de problèmes entre le niveau d'exploration et d'exécution. C'est-à-dire que si l'utilisateur commence à dialoguer avec le système sur des détails et qu'il l'informe de sa réelle intention, alors, une fois que le niveau d'exploration aura commencé, le système ne peut plus changer au niveau plan d'exécution sans avoir terminé avec le niveau d'exploration. D'où le

modèle de Ramshaw qui ne peut pas reconnaître les plans d'intention de l'utilisateur pendant le déroulement du dialogue.

II.5.3 Modèle logique

Un des champs de recherche dans l'étude des actes de langage est la modélisation des états mentaux des agents (humain ou machine) c'est-à-dire la modélisation des croyances et connaissances des agents. Les modèles logiques utilisent la logique modale pour représenter ces états mentaux, ils se rapprochent plus de la théorie mentale telles que l'intentionnalité et la rationalité. En fait, ces modèles cherchent à modéliser les attitudes mentales de l'interlocuteur et à formaliser les raisonnements de ces attitudes. Toutefois, il faut garder à l'esprit que le système opère sur la forme du langage et non pas sur la substance, c'est-à-dire sur la forme observable du langage (les signifiants) et non pas sur les substances qui amènent au signifié. Le point fort de ces modèles est qu'ils ont le mérite d'être méthodologiques et solides. Cependant, comme souligne Minker (2000), ces modèles restent trop théoriques au point de vue cognitif car ils tendent à modéliser l'intentionnalité des agents. Ils touchent aux aspects d'états mentaux des agents humains, il est difficile pour un système de gérer les situations complexes. En réalité, pour réaliser un système efficace, il serait plus ingénieux d'utiliser les modèles orienté-plan ou orienté-tâche cités précédemment. Nous présenterons ci-dessous deux modèles logiques : le modèle mental et le modèle de jeux.

II.5.3.1 Modèle mental

Bien que le modèle mental, développé par Pollack (1986, 1990) [POL86], [POL90] se classe dans la catégorie de modèle logique, il modélise les attitudes mentales à travers les plans organisationnels des intentions. Ce modèle a pour but de formuler les plans de connaissances de l'agent humain en termes de croyance individuelle. Plus tard, les travaux de Lochbaum et al. (1990) [LOC90] ont étendu la notion de plan individuel de Pollack en notion de plan partagé.

II.5.3.2 Modèle de jeux

La communication peut être vue comme une partie de jeu où chaque participant prend son tour de parole. L'approche de modélisation de la communication par des modèles de jeux consiste à voir la conversation comme un jeu de langage (Wittgenstein, 1957) [WIT57]. Selon Caelen (2005)⁶, « *les stratégies de dialogue sont des manières d'atteindre un but à travers le dialogue considéré comme une activité conjointe (Vernant, 1992)* ». Cette stratégie se manifeste sous forme d'un jeu de conversation dans lequel les objectifs des interlocuteurs pourraient être différents, mais ils ont quand même une motivation commune, celle de l'intention de communiquer pour partager leur but.

Les théories de jeux (Myerson, 1991) ont été appliquées dans le dialogue par plusieurs comme Levin et Moore en 1980, Mann en 1988 ou Lewin en 2000 avec le projet TRINDI [LEW00]. Ce que l'on retient dans ce modèle de jeux est la stratégie conjointe et coopérative. Dans une stratégie de jeu pur (jeu d'échecs par exemple), chaque participant essaie de gagner sa partie en maximisant ses gains tout en réduisant ses pertes, mais aussi de cacher ses intentions, c'est-à-dire ne pas laisser l'adversaire deviner sa stratégie auquel cas son plan d'attaque sera mis en danger. Dans un SDOHM où la coopération est importante, la stratégie coopérative doit être mise en évidence. Les agents ne doivent pas cacher leurs intentions, mais ils communiquent leur but à l'autre sauf s'il y a ambiguïté dans l'énoncé.

II.5.3.3 Modèle de jeux associé à la psychologie sociale

Les travaux d'Asai et al. (2005) [ASA05] ont pour but de rendre le modèle d'utilisateurs plus lisse grâce à une stratégie de dialogue plus performante à l'aide des théories de jeux et des théories cognitives en psychologie sociale (le comportement humain face à des situations dans laquelle l'être humain vit) et de maximiser les attentes mutuelles tout en réduisant les connaissances incertaines lorsque la perception est médiocre.

⁶ Rapport d'activités 2002-2005 de l'équipe GEOD (Groupe d'Étude sur l'Oral et le Dialogue). Disponible sur : http://www-clips.imag.fr/geod/doc/Rapport_Activites_2002-2005.pdf [consulté le 22 fev 2006].

Asai et al. utilisent le modèle collaboratif des utilisateurs qui est basé sur la stratégie de dialogue pour une interaction durable⁷.

Selon Asai et al., la principale caractéristique du modèle d'utilisateurs est de trouver une approche innovante pour stimuler le comportement humain. Mais, il semble que la nouvelle recherche oriente plus vers un modèle cognitif que vers un vrai modèle de jeux. Ce modèle cognitif est basé sur la perception et la génération (comment l'information entrante est traitée et comprise ; comment l'information sortie est créée). En effet, ces auteurs arguent que l'on s'attend à ce que psychologiquement, le dialogue gagne la satisfaction des utilisateurs par des machines qui permettent à ces utilisateurs d'induire des plans comportementaux liés aux comportements coopératifs sociaux.

II.6 Conclusion

Nous nous sommes placés dans le cadre des études théoriques en linguistique et en intelligence artificielle pour décrire l'aspect modélisation du dialogue. Nous avons présenté les trois modèles de dialogue qui, selon nous, représentent l'ensemble des travaux en dialogue depuis les cinquante dernières années. Ces trois modèles de dialogue sont les plus classiques et connus, à savoir le modèle structurel (linguistique), le modèle orienté plan ou orienté-tâche (IA, planification et reconnaissance de plan) et le modèle logique (logique modale pour la représentation de l'état mental, les attitudes mentales des interlocuteurs et le raisonnement induit par ces attitudes) [MIN00]. On distingue deux aspects majeurs dans ces modèles : les représentations sémantiques et pragmatiques. La représentation sémantique est la représentation du contenu de ce que les interlocuteurs expriment et la représentation pragmatique se réfère au mode d'action que l'interlocuteur est en train d'employer (prononcer une phrase déclarative, affirmative ou interrogative, etc.). En fait, modéliser un dialogue revient à modéliser les motivations, les comportements et les intentions des agents humains. Cela nous conduit à choisir un modèle mental. Mais, ce modèle est extrêmement difficile à mettre en œuvre car la pensée n'est pas un

⁷ Traduit de l'anglais pour *long-lasting dialogue strategy interaction*.

concept que l'on peut facilement transformer en objet rationnel pour qu'un système informatique soit capable de le 'comprendre'. Il lui faut d'autres modèles plus rationnels comme le modèle logique ou modèle orienté-plan par exemple. Nous proposons donc de combiner le modèle de jeux (classé dans la catégorie de modèle mental) et celui de planification (modèle orienté plan) pour modéliser le comportement des agents humains à condition que ce comportement soit coopératif et que le but de la machine soit bien déterminé. Nous pouvons définir un but pour la machine qui lui permettra de '*savoir où elle s'en va*'. Nous parlons alors d'un système orienté but. Dans ce système la machine doit apprendre à se maîtriser pour parvenir à son but.

CHAPITRE III

SYSTÈME DE DIALOGUE ORAL HUMAIN-MACHINE

III.1 Introduction

Comprendre l'architecture générale d'un système de dialogue oral humain-machine (SDOHM) et savoir comment fonctionne ce système permettra par la suite d'intégrer une stratégie optimale du dialogue dans ce dernier afin de l'aider à améliorer sa performance. Ce chapitre se consacrera essentiellement à la description de l'architecture générale d'un SDOHM. Elle est découpée en trois niveaux : un niveau acoustique, un niveau sémantique et un niveau pragmatique. Chacun de ces niveaux contient un ou plusieurs modules spécifiques permettant de traiter le signal sonore de son entrée jusqu'à sa sortie en tant que message sonore que nous interpréterons comme étant la parole.

III.2 Système de dialogue oral humain-machine

Rappelons qu'un SDOHM est un système qui permet l'échange d'informations par voix orale entre un agent humain et un agent machine de façon naturelle et souple (chapitre I). Le terme agent humain ou simplement humain désigne une personne physique en pleine possession de ses capacités cognitives (perception, adaptation, interaction, évolution, etc.). Un agent machine ou machine est une entité à qui l'on attribue une capacité de simulation de certaines de nos capacités cognitives (i.e., la perception du signal de la parole, le raisonnement, interaction avec l'environnement, etc.). L'agent machine représente une entité autonome qui interagit avec l'agent humain. Par définition, un SDOHM est un agent machine qui réalise un ensemble de tâches, telles que la reconnaissance et l'interprétation de l'énoncé oral, la gestion du dialogue, la génération et la synthèse

du signal de la parole. Ces tâches particulières ont pour but de satisfaire la demande de l'agent humain sinon un tel dialogue n'aurait pas d'intérêt. Le terme dialogue oral humain-machine sous-entend aussi un dialogue finalisé, c'est-à-dire que l'agent machine en cherchant à satisfaire la demande de l'agent humain, doit deviner son intention (but) à travers l'expression orale. Il doit guider et contrôler le dialogue pour s'assurer du bon déroulement de l'échange.

III.3 Architecture d'un SDOHM

Un SDOHM peut être découpé en cinq modules qui forment une chaîne de traitement et qui fonctionnent en séquentiel : un module de reconnaissance du signal de la parole, un module de compréhension de la parole, un module de gestion du dialogue, un module de génération de la parole, et enfin un module de production du signal de la parole. Pietquin [PIE04] a distingué trois niveaux de traitement dans un système de dialogue (figure 3.1) : un niveau acoustique, un niveau sémantique et un niveau pragmatique. Nous détaillerons chacun des niveaux de traitement ci-dessous.

III.3.1 Niveau acoustique

III.3.1.1 Perception et reconnaissance du signal de la parole

Le niveau acoustique est le niveau le plus bas dans la chaîne de la communication parlée. Le processus de reconnaissance de la parole traite le signal de parole comme une suite de séquences de sons qui subit des transformations particulières (série de Fourier, normalisation, découpage, etc.) avant d'être transcrite sous forme textuelle, pour enfin être perçue comme un message linguistique et éventuellement compréhensible.

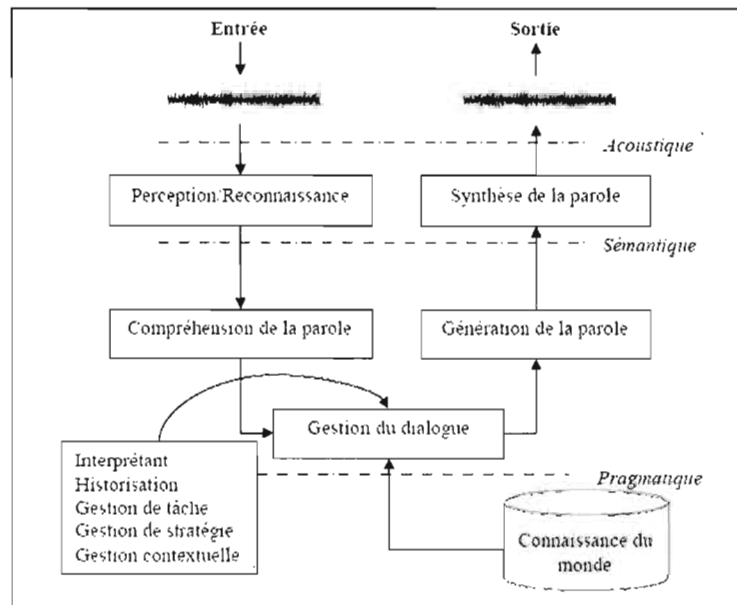


Figure 3.1 Vue générale des 5 modules d'un SDOHM (adaptée de Pietquin, 2004).

Pour O'Shaughnessy (2000) [OSH00], ces processus particuliers font partie de la perception de la parole dans laquelle on peut distinguer plusieurs étapes d'analyses : auditive, phonétique, phonologique, lexicale (mots), syntaxique et sémantique. Pour McQueen et Culter (1997) [MCQ97], les processus de perception et de reconnaissance ne peuvent être traités comme des processus indépendants et séquentiels, car ils se réalisent en parallèle et s'activent mutuellement de façon dynamique selon la qualité de l'information entrante. Cependant, quelques étapes doivent être réalisées en parallèle ou en simultané avec rétroaction. Par exemple, les étapes de traitements auditif, phonétique et lexical se réalisent simultanément. Lorsque l'information de bas niveau (phonétique acoustique) est manquante, le processus de reconnaissance de mots (lexical) utilise la connaissance de haut niveau (phrasique ou contextuel) pour corriger ce manque d'information (réalisation en parallèle) et ainsi permettre au processus de bas niveau de retarder la décision phonétique jusqu'à la réception complète de l'information. L'exemple typique de ce processus simultané se manifeste lorsque le signal acoustique représentant un mot, contient du bruit ou lorsque ce mot est en cours de se produire. L'agent humain est

capable de percevoir ce mot avant que le son du mot soit entièrement sorti de l'appareil phonatoire. Cela a pu se réaliser grâce au processus simultané qui a lieu en même temps que l'analyse acoustique et linguistique.

Les étapes de la perception évoquées ci-dessus nous donnent des indications quant à la nature des informations extraites du signal sonore afin de les percevoir comme de la parole. La question qui se pose est de savoir quand et comment ces informations peuvent être extraites et intégrées dans la connaissance [MCQ97]. Selon McQueen et Culter (1997), l'un des aspects importants, sinon le plus important dans la perception de la parole est la reconnaissance des mots. En effet, les auteurs renvoient le problème de la perception de la parole au problème du lexique mental (dictionnaire mental), qui est le point essentiel dans tous les modèles de compréhension de la parole et qui se situe au niveau sémantique. Donc, le module de reconnaissance se situe à la limite entre le niveau acoustique et sémantique. Au point de vue de la réalisation informatique, il est du ressort du concepteur de décider quelle est la façon la mieux adaptée pour intégrer les modèles acoustiques et linguistiques dans un système de dialogue. Mais, quoiqu'il en soit, la perception et la reconnaissance emploient presque les mêmes processus de traitement du signal et ceux-ci ne sont pas tous situés au niveau acoustique comme le soulignent O'Shaughnessy, (2000) [OSH00], McQueen et Culter, (1997) [MCQ97].

III.3.1.2 Synthèse de la parole

La synthèse de la parole fait partie du niveau acoustique, et consiste à transformer les séquences de caractères en sons digitalisés de la parole. La synthèse de la parole pourrait très bien se placer au niveau sémantique, car certains traitements comme les recherches d'homophones et allophones font appel aux modèles linguistiques. La synthèse de la parole est la dernière phase de traitement dans un système de dialogue. Elle reproduit directement une suite sonore à partir du texte généré par le module de génération de la parole (plus loin dans ce chapitre). Il y a plusieurs techniques pour reproduire un son de la parole. Nous ne détaillons pas ces techniques dans cette partie, mais citerons seulement les faits. Pour de plus amples informations sur la synthèse de la parole, les lecteurs peuvent consulter la

référence de [OSH00]. Ces techniques sont : la synthèse par prédiction linéaire, la synthèse par concaténation (phonème, diphone), la synthèse par règle, etc. Bien qu'il y ait toujours des recherches et des découvertes à faire dans ce domaine, la synthèse de la parole est la partie la mieux maîtrisée dans l'ensemble du traitement du signal acoustique de parole (reconnaissance, compréhension et génération de la parole). Ceci est dû en partie à l'implication moins importante de la sémantique dans le processus de conversion du texte au son de la parole.

III.3.2 Niveau sémantique

III.3.2.1 Compréhension de la parole

À ce niveau, le traitement de la compréhension des langues naturelles cherche à extraire le sens des séquences de caractères issues de la reconnaissance de la parole. En dialogue oral spontané entre les interlocuteurs, la sémantique dépend du contexte dans lequel le dialogue est engagé, par exemple, la phrase « *le chat prend la souris et la traîne vers la fenêtre* » décrit un comportement instinctif de l'animal (on peut sous-entendre le verbe *prendre* comme *manger*), par contre la phrase « *je prends la souris et la traîne vers la fenêtre X* » décrit une situation où le mot « *souris* » désigne un outil de pointage que l'on utilise avec le clavier d'ordinateur. Ce « *souris* » n'a pas la même signification que l'animal « *souris* ». Ainsi, l'extraction de la sémantique d'une suite de caractères se fait en analysant la syntaxe, la sémantique et l'interprétation contextuelle.

Le module de la reconnaissance se trouve en amont du module de la compréhension. Ces modules fonctionnent de façon séquentielle, ce qui veut dire que l'extraction sémantique dépend beaucoup de la performance du module de reconnaissance. À l'issue du module de compréhension, une représentation non linguistique de l'énoncé est créée sous forme de schéma structurel. Cette représentation peut être vue comme un transfert d'information de l'agent humain vers l'agent machine (Pietquin, 2004) [PIE04]. Cette représentation est appelée « concept ». Par exemple, en supposant que pour réserver une chambre d'hôtel, le système informatique dédié à cette tâche a besoin de connaître seulement le nombre

de chambres et la date de réservation, alors la phrase « *je voudrais réserver une chambre pour ce vendredi* » pourrait être conceptualisée sous une forme propositionnelle réduite telle que : <Concept> *Réservation (Nbc : 1 ; Dte : vendredi)* où *Nbc* est le nombre de chambres et *Dte* contient la date de réservation. Le but d'une telle description est de représenter les phrases par des concepts, des objets et des règles à travers lesquels le système interagit avec le monde extérieur. Le niveau sémantique se réfère à la théorie de l'usage de la langue. Les interlocuteurs essaient de reconnaître (et faire reconnaître) leur but et leur croyance dans l'objectif de parvenir à satisfaire leur demande. Cela laisse présager une coopération et un partage de connaissance entre deux interlocuteurs.

III.3.2.2 Gestion du dialogue

Le module de gestion du dialogue se situe entre le module de compréhension et le module de la génération de la parole. Contrôler les interactions orales entre l'agent humain et l'agent machine signifie gérer les échanges entre ces deux agents. C'est un rôle d'interprétation des entrées et des sorties des signaux acoustiques car sans une gestion appropriée de ces informations, le système ne peut donner de réponses cohérentes. Cette gestion est considérée comme le cœur du système de dialogue. Pour bien remplir son rôle, le module de gestion du dialogue doit aussi tenir compte de l'historique du dialogue, de la gestion des tâches, de la gestion des stratégies (e.g. enchaînement et synchronisation des tours de parole) et de la gestion des contextuelles. Tous ces aspects seront expliqués plus en détails dans le chapitre suivant sur la stratégie du dialogue.

III.3.2.3 Génération de la parole

Le module de génération de la parole (GL) est situé en aval du module de synthèse et en amont du module de gestion du dialogue. Le processus de GL consiste à transformer les concepts (formés par le module de compréhension) en une forme textuelle compréhensible dans un langage humain. Pietquin (2004) distingue trois niveaux de GL dans ce processus qui va de la signification du mot jusqu'à la forme du mot : niveau sémantique, niveau syntactique et niveau lexical [PIE04].

III.3.3 Niveau pragmatique

III.3.3.1 Connaissance du monde

Par rapport au système informatique, l'information provenant de l'environnement extérieur est souvent représentée dans une base de données dans laquelle on essaie de rassembler, autant que possible, les éléments associés aux connaissances du monde. La base de connaissance représente toute la connaissance que l'agent machine est supposé posséder. Plus la base de connaissance est grande mieux ce sera pour l'agent machine afin qu'il puisse élever son niveau de connaissance et augmenter la probabilité de reconnaître le but et la croyance de l'agent humain.

III.4 Conclusion

Nous avons décrit l'architecture générale d'un SDOHM. Nous avons vu différents modules qui participent à la réalisation d'un système de dialogue robuste et performant. Chaque module joue un rôle important dans la chaîne de traitement des entrées et sorties du signal acoustique. Il est important de noter que dans ce processus, à aucun moment, nous ne considérons que notre système possède un état mental comme le nôtre. Bien que ce système soit capable d'*interpréter* les sons et de produire de la '*parole*', il est impossible qu'il soit capable de générer d'autres connaissances sans que nous ne lui ayons préalablement incorporé la connaissance de base dans son noyau (sa structure). Cette remarque a pour but de rappeler que parmi les trois niveaux de fonctionnement du système de dialogue, le niveau pragmatique est le niveau le plus difficile à maîtriser, car il peut être assimilé au niveau mental de notre cerveau dans lequel il est difficile d'accéder, alors que le niveau acoustique est le plus facile à saisir par le système. En effet, toute information sonore (signal sonore) est présente et peut être perçue et analysée par le module de reconnaissance de la parole qui est le premier module à traiter l'information sonore. Sa robustesse a alors un impact direct sur les autres modules surtout sur le module de compréhension et de gestion du dialogue.

Nous allons voir dans le chapitre qui suit quelle est la fonction exacte du module de gestion du dialogue et comment la stratégie optimale du dialogue peut être gérée dans ce module. Comme le module de gestion du dialogue est le cœur du SDOHM dans lequel on définit certains principes et règles de base du dialogue, cela permettra au système de produire une sortie acoustique cohérente par rapport à l'énoncé à l'entrée du système. Il est donc important de comprendre comment le module de gestion du dialogue fonctionne à l'intérieur du système de dialogue humain-machine et comment nous pourrions améliorer l'ensemble du SDOHM grâce à la performance de ce module.

CHAPITRE IV

GESTION DU DIALOGUE ORAL HUMAIN-MACHINE

IV.1 Introduction

La gestion du dialogue est le cœur du système dialogique, puisque sans elle, un système de dialogue ressemblerait à une personne qui peut entendre et parler mais ne sait pas répondre correctement (e.g. la personne répond aléatoirement quand on lui parle). Le gestionnaire du dialogue est un agent qui s'occupe de la gestion du dialogue, c'est-à-dire de la manière dont le système gère les entrées et les sorties acoustiques. Dans ce chapitre, nous allons voir quels sont les modèles de gestion du dialogue (section IV.2), ensuite, nous discuterons des approches de modélisation de la stratégie (section IV.3) du dialogue ainsi que des actes du dialogue (section IV.4). Cette modélisation permettra au gestionnaire du dialogue de 'faire correctement' son travail.

IV.2 Gestion du dialogue humain-machine

Pour bien gérer les échanges entre les agents, il faut savoir quelles sont les tâches à gérer et comment les gérer efficacement. C'est le rôle du gestionnaire de dialogue dans le module de gestion du dialogue. Avoir un bon gestionnaire de dialogue garantira l'efficacité et la performance du SDOHM. Minker (2000) a distingué deux catégories de connaissances requises pour bien gérer l'acte de dialogue : les connaissances statiques et dynamiques. Les connaissances statiques sont celles du langage, du dialogue, de la tâche, d'une partie de l'utilisateur et du système. Les connaissances dynamiques concernent l'historique du dialogue, le contexte de la tâche, la stratégie du dialogue (peut être modélisée comme statique ou

dynamique), le contexte du système lui-même (une partie dynamique et une partie statique), et aussi de l'utilisateur qui pourrait être modélisé comme statique ou dynamique.

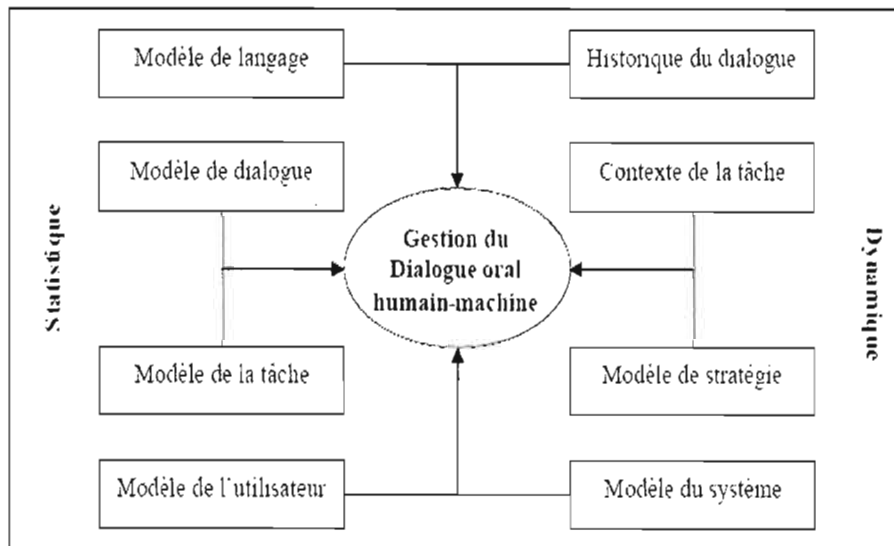


Figure 4.1 Connaissance requise pour un dialogue oral humain-machine (Selon Minker, 2000) [MIN00].

IV.2.1 Modèles statiques

IV.2.1.1 Modèle du langage

Le modèle du langage inclut les composantes lexicales, syntaxiques et sémantiques [MIN00]. Dans ce modèle (fig. 4.1), on utilise la connaissance linguistique pour guider l'acte de langage. Le modèle du langage tend de modéliser le dialogue sans être attaché au contexte du ceci. Il est donc plus global que le modèle de dialogue qui doit rester relativement lié au contexte du dialogue.

IV.2.1.2 Modèle de dialogue

Le modèle de dialogue se réfère à une connaissance plus large du domaine (la connaissance du monde). Ce modèle permet au système d'avoir le contrôle afin d'appeler le bon module dans les moments appropriés puisqu'il a le contrôle contextuel du dialogue. Ce modèle permet aussi de maintenir la croyance entre deux

agents. Cette croyance permet aux interlocuteurs d'atteindre leurs buts. On peut résumer le rôle du modèle de dialogue en lui attribuant quatre grandes fonctionnalités : interpréter le contexte des énoncés, anticiper et/ou prédire les intentions des utilisateurs, lever les ambiguïtés et corriger les erreurs d'interprétation.

IV.2.1.1 Modèle de la tâche

Le modèle de la tâche représente la connaissance liée à une application donnée comme les objets, les relations, les concepts, les règles d'inférence et la description des actions à exécuter (Minker, 2000) [MIN00]. En IA, en ce qui concerne la représentation de connaissance, la modélisation de la tâche se réfère à la preuve de théorème¹ ou la construction d'un plan de résolution de problème² [PIE04]. Le modèle de la tâche reçoit, en entrée, un but issu du module de compréhension, donc, il doit développer un plan d'action pour résoudre ce but. Ce modèle reste donc très près de l'application et de l'utilisateur. Les systèmes qui utilisent ce modèle se veulent plus réalistes, ils ne reposent pas sur les théories linguistiques, ni sur des aspects fondamentaux de l'interaction mais, ils se concentrent sur l'accomplissement de la tâche.

IV.2.2 Modèles dynamiques

Le dialogue est dynamique et imprévisible. Pour qu'un SDOHM puisse suivre la conversation, il faut lui donner la capacité de mémoriser ses actions et celles de l'utilisateur (i.e., ce qu'il a dit à l'utilisateur et ce qu'il a entendu et compris de sa part). Ce sont les contextes et historiques du dialogue qu'un SDOHM doit pouvoir maîtriser.

IV.2.2.1 Contexte de la tâche

Conduire un acte de dialogue oral suppose que le système doit comprendre les demandes venant de la part de l'utilisateur et doit générer les actions adéquates (i.e.,

¹ En anglais se dit Theorem proving.

² En anglais se dit Problem-solving.

répondre à la demande ou demander une information supplémentaire, enregistrer une demande de réservation, etc.). Donc le but est de garder les actions émises par l'utilisateur dans un contexte qui se rattache à la tâche du domaine pour pouvoir par la suite, générer des réponses cohérentes.

IV.2.2.1 Historisation du dialogue

Le dialogue entre l'agent humain et l'agent machine ne peut être mené à bien sans qu'il y ait mémorisation de ce qui a été dit, c'est-à-dire garder une historique du dialogue permettant, entre autre, de résoudre les ambiguïtés résultantes des anaphores (répétition) ou des ellipses (raccourcies). Par exemple le système ALICE³ est capable de garder une historisation sur les questions et les noms des personnes avec qui il a discuté.

IV.2.3 Modèles statiques et dynamiques

Il y a des modèles qui sont à la fois dynamiques et statiques dépendant de la tâche et du but de l'application spécifique. Par exemple, le modèle de l'utilisateur peut être défini comme statique si on veut plus privilégier le côté machine que le côté humain. Ce dernier devient dynamique si on décide de tenir compte des changements produits pendant le dialogue. Dans ce cas, le modèle de l'utilisateur est plus complexe et il doit représenter le comportement dialogique de la plupart des utilisateurs. Il en est de même pour le modèle du système et le modèle de stratégie qui deviennent dynamiques si on veut rendre disponible immédiatement les nouvelles informations créées ou modifiées au cours du dialogue.

IV.2.3.1 Modèle de l'utilisateur

Ce modèle est particulièrement utile lorsqu'on veut modéliser l'intention de l'utilisateur. Bien qu'il soit considéré comme un modèle statique, il pourrait devenir dynamique, car en langage spontané, on sait que l'état mental de l'utilisateur peut changer au cours du dialogue puisque l'acte de dialogue, lui-même, a pour but de

³ Agent conversationnel ALICE. Disponible sur : <http://www.alicebot.org/> [consulté le 10 avr 2006].

faire changer l'état mental des interlocuteurs. Ce modèle doit être particulièrement adapté à chaque situation de dialogue, il doit à la fois représenter fidèlement l'intention des utilisateurs et rester suffisamment souple pour permettre une meilleure adaptation au changement d'intention au cours du dialogue.

IV.2.3.2 Modèle du système

Le modèle du système est souvent statique car, dans les applications orientées-tâche, ce modèle représente le comportement du système. Ce comportement est conçu et défini au début de l'application et ne changera pas tant qu'il n'y aura pas de modifications. Mais ce modèle peut devenir dynamique lorsqu'on utilise l'approche apprentissage par renforcement⁴ pour modéliser son comportement face aux nouvelles connaissances.

L'apprentissage par renforcement permet au système d'ajuster sa réaction devant une situation incertaine comme par exemple une nouvelle connaissance que le système n'a jamais rencontrée auparavant ou une question posée par l'utilisateur dont le système a su extraire le sens mais la réponse à cette question n'a pas été prévue (i.e., on parle aussi de vocabulaire hors du contexte⁵).

IV.2.3.3 Modèle de stratégie

Existe-t-il vraiment un modèle de stratégie '*standard*' pour les différentes situations dialogiques ? Le modèle de stratégie est le cœur du système, car la performance du système dépend de la façon dont il gère ses actes de dialogue. Notre thèse consiste justement à trouver une stratégie optimale de dialogue par apprentissage par renforcement avec un utilisateur simulé⁶. Mais, en réalité, il n'y a pas de modèle de stratégie pour tous les types de systèmes (système à question-réponse, système à conversation ouvert, système de tutorial, système de recherche d'information, etc.), il y a des stratégies adaptées à chaque type de système. Bien souvent, on choisit une stratégie du dialogue en fonction du but du système. Donc,

⁴ Voir chapitre V pour l'explication sur l'apprentissage par renforcement.

⁵ *Out of vocabulary (OOV)* en anglais.

⁶ Voir chapitres V, VI, VII.

tout comme le modèle du système, le modèle de stratégie peut être à la fois dynamique et statique. Par exemple, on peut le définir localement dynamique en agissant sur le degré d'initiative du système (i.e., qui commence le dialogue et qui prend le contrôle pendant l'acte de dialogue) et sur la stratégie de confirmation (i.e., comment confirmer une réponse donnée par l'utilisateur sans tomber dans un dialogue répétitif, monotone et ennuyant). Nous allons voir par la suite comment on peut modéliser une stratégie du dialogue selon ces deux aspects : degré d'initiative et stratégie de confirmation. Cette modélisation permettra de l'appliquer dans un système de dialogue, on peut donc apprendre un dialogue selon ce modèle de stratégie à la machine.

IV.3 Approche de modélisation de la stratégie du dialogue

Dans un système de dialogue, c'est le gestionnaire du dialogue qui décide de la stratégie qui doit être employée pour répondre adéquatement au message en entrée (chapitre III, figure 3.1). Le gestionnaire du dialogue doit adopter une stratégie pour répondre aux signaux en entrée. Rappelons que le gestionnaire du dialogue est celui qui s'occupe de la gestion du dialogue, il a un rôle à jouer dans un système de dialogue et son rôle fait partie de la stratégie du dialogue.

IV.3.1 Rôle du gestionnaire de dialogue

Le rôle du gestionnaire du dialogue dans un système de dialogue n'est pas clairement défini par la communauté scientifique. En effet, Schatzman et al., (2006) [SCH06] distinguent au moins 11 tâches pour le gestionnaire du dialogue :

⁷The task of the Dialogue Manager (DM) is to control the flow of the dialogue (1) and to decide which actions to take (2) in response to user input. The DM has to interpret (3) the incoming semantic representation of the user input in the context of the dialogue, resolve ellipsis (4) and anaphora (5), evaluate the relevance (6) and

⁷ La tâche du gestionnaire du dialogue est de contrôler le suivi du dialogue et décider quelles actions à prendre en réponse de la demande de l'utilisateur. Le gestionnaire du dialogue doit interpréter la sémantique de la phrase produite par l'utilisateur en contexte du dialogue, résoudre des problèmes d'ellipses, d'anaphores, d'évaluer la pertinence et la consistance des demandes de l'utilisateur, d'identifier et réparer les erreurs de reconnaissance et de compréhension du module de la parole, de rechercher d'informations à partir des bases de données périphériques (externes) et de formuler des réponses appropriées.

completeness (7) of user requests, identify (8) and recover (9) from recognition and understanding errors, retrieve information (10) from peripheral databases and format appropriate (11) system replies.

Le gestionnaire du dialogue doit contrôler les tours de parole, décider quelle action est à prendre, interpréter la représentation sémantique en entrée pour résoudre des problèmes d'ellipses et anaphores, évaluer la pertinence et la perfection des demandes d'utilisateur, identifier et récupérer les erreurs issues des modules de reconnaissance et de compréhension, rechercher l'information des bases de données périphériques et y répondre dans un format approprié. Pour accomplir toutes ces tâches, il faut une *stratégie adéquate de dialogue* par rapport au domaine d'application. L'aspect stratégique du dialogue est considéré par plusieurs auteurs comme un des aspects primordiaux qui garantissent le succès d'un système de dialogue [PIE04], [ALL01], [MIN00], [LIT84], [LEV81]. Sans une gestion appropriée du dialogue, il n'y aurait pas de réponse cohérente, fluide et naturelle. La stratégie de dialogue est souvent définie en fonction du comportement des utilisateurs (agent humain), de la connaissance *a priori* et de la complexité de l'application [BAL99].

IV.3.1.1 Degrés d'initiative

Le degré d'initiative est une manière de contrôler l'interaction entre l'agent humain et l'agent machine (système). Il y a trois manières, pour l'agent machine, de prendre l'initiative dans un dialogue : soit le système prend le contrôle total du dialogue (*système-initiative*), soit le système laisse l'utilisateur prendre l'initiative (*utilisateur-initiative*), soit le système partage le contrôle avec l'utilisateur (*initiative-mixte*) (Chu-Carroll, système MIMIC, 2000) [CHU00].

Dans le cas où le système prend le contrôle total du dialogue, il est plus facile d'orienter l'utilisateur de manière à ce qu'il reste dans le contexte du dialogue et qu'il réponde directement à la question. Dans ce cas, la charge du gestionnaire de dialogue est moins lourde, puisque le système contrôle les entrées et les sorties des données. Donc, le contexte du dialogue sera mieux maîtrisé ; par exemple, le

système sait qu'il est en train de travailler sur une date donc il s'organise pour entreprendre les actions appropriées par rapport à ce but.

Quant à la stratégie utilisateur-initiative, le système laisse le choix à l'utilisateur de transmettre sa demande et le système essaie d'interpréter ses messages. Cette stratégie laisse plus de '*liberté d'expressions*' à l'utilisateur, quoique ce dernier ne peut dire n'importe quoi au système.

La troisième stratégie est le résultat du mélange des deux stratégies précédentes. L'initiative du dialogue peut être prise, soit par l'utilisateur, soit par le système. Par exemple, au début d'un dialogue, le système peut le commencer en adressant, à l'utilisateur, une phrase d'invitation « *comment puis-je vous aider ?* » Ensuite, le système laisse l'utilisateur parler et s'il y a un doute, le système reprend le contrôle. Ce mode de fonctionnement semble plus naturel et plus souple que les deux précédents modes, car il laisse vraiment place à la communication mutuelle (écouter/parler), mais ce mode est aussi plus difficile à gérer pour le système car il faut tenir compte de plusieurs paramètres (taux d'erreurs de reconnaissance, synchroniser les tours de parole, détection de parole croisée⁸).

IV.3.1.2 Stratégie de confirmation

Selon le résultat du module de reconnaissance, la confirmation d'une réponse d'utilisateur peut être explicite (e.g. « *Avez-vous dit deux personnes ?* »), implicite (e.g. *combien de nuits resterez-vous à deux ?*), direct (e.g. « *Quand voulez-vous venir ?* ») ou répétitive (e.g. « *Pouvez vous répéter la date s.v.p. ?* »).

Choisir une stratégie de confirmation en fonction du résultat de la reconnaissance est une façon élégante et intelligente de contrôler le déroulement du dialogue, mais il faut aussi tenir compte du comportement des utilisateurs et de leurs intentions. Une stratégie efficace répondant le plus possible à des situations dialogiques et satisfaisant le plus grand nombre d'utilisateurs, n'est certainement pas

⁸ Les sons de la parole entrent dans le système mais ne font pas partie du dialogue. Par exemple l'utilisateur parle à une autre personne pendant que son micro est allumé.

évidente à modéliser. Dans certains cas, les utilisateurs préfèrent une réponse explicite, car cela leur donne l'impression que le système les comprend, mais pour d'autres, cette stratégie de confirmation peut être irritante et ennuyeuse, car la conversation est longue, répétitive et moins naturelle. La solution idéale serait de modéliser une stratégie dynamique qui adapte son comportement en fonction des spécificités de chaque domaine. Mais cela rend le modèle trop spécifique au domaine choisi et la généralisation ne serait pas automatique.

Une autre stratégie de confirmation a été employée par Scheffer et Young (2002) dans une application de recherche de films. La technique consiste à déterminer un point de choix à partir duquel le système décide de l'action à choisir. Cette technique est efficace lorsque le système a choisi la bonne réponse de l'utilisateur, sinon il faut 'réparer' le dialogue et cela risque de le rendre plus difficile.

<i>A</i>	<i>Quand voulez-vous réserver?</i>
<i>U</i>	<i>Lundi</i>
<i>A</i>	[machine a compris vendredi au lieu de lundi] <i>Voulez-vous réserver pour vendredi?</i>
<i>U</i>	<i>Non, je n'ai jamais dit ça, j'ai dit lundi</i>

Tableau 4.1 Exemple d'une mauvaise stratégie de confirmation (explicite au lieu de répétitive) dans une situation où la qualité de reconnaissance est mauvaise.

Le tableau 4.1 montre un exemple de la stratégie de dialogue qui prend le risque d'enchaîner le dialogue avec une question explicite. L'agent machine (A) demande la date de réservation « *Quand voulez-vous réserver?* », l'utilisateur (U) a répondu lundi, mais (A) a compris vendredi, alors (A) veut confirmer la réponse de (U) en lui posant une question de confirmation explicite « *Voulez-vous réserver pour vendredi?* ». Cette question de confirmation peut irriter (U) car ce n'est pas sa réponse. Le dialogue risque d'être plus difficile à gérer par la suite. Donc pour ne pas tomber dans ce cas, il serait préférable de ne pas employer de questions explicites quand on n'est pas sûr de la réponse pour ne pas induire les utilisateurs dans l'erreur et dans la confusion.

Un autre cas particulier que nous avons décelé avec cette approche de stratégie est l'emploi de la question implicite au lieu d'explicite quand la qualité de reconnaissance est moyenne (tableau 4.2). Lorsque la qualité de reconnaissance est moyenne (i.e., le système n'est pas sûr de la réponse), la stratégie suggère quand même une question implicite pour enchaîner le dialogue tout en vérifiant l'information entrée par l'utilisateur. Dans ce cas, au lieu de confirmer sur le champ *date* (i.e., la date de réservation doit être confirmée en premier avec une question explicite), la machine a posé une question implicite « *Et pour combien de personnes voulez-vous réserver pour ce vendredi?* », en prenant la chance que l'utilisateur ait donné *vendredi* comme réponse. Cette fois-ci encore, l'utilisateur a dû 'corriger' l'erreur de la machine « *Non, j'ai dit lundi, pas vendredi et pour 3 personnes* ». La dernière réponse de l'utilisateur est plus difficile à interpréter pour la machine.

<i>A</i>	<i>Quand voulez-vous réserver?</i>
<i>U</i>	<i>Lundi</i>
<i>A</i>	[machine a compris vendredi] <i>Et pour combien de personnes voulez-vous réserver pour ce vendredi?</i>
<i>U</i>	<i>Non, j'ai dit lundi, pas vendredi et pour 3 personnes.</i>

Tableau 4.2 Exemple d'une mauvaise stratégie de confirmation (implicite au lieu d'explicite) dans une situation où la qualité de reconnaissance est moyenne.

IV.4 Approche de modélisation des actes de dialogue

Comment les actes de dialogues sont représentés par le système? Comment le dialogue peut-il être modélisé pour être utilisé dans un SDOHM? Nous avons, ci-dessous, l'état de l'art sur ces questions (extraire de récents travaux de Cuayahuitl et al. 2005 [CUA05] sur les approches de modélisation des actes de dialogue). Cette modélisation représente le passage de connaissances entre l'agent humain et l'agent machine. Autrement dit, la formalisation des actes du dialogue permet à l'agent machine (système) d'inférer les connaissances, d'induire des raisonnements logiques pour que le système puisse se comporter (dialoguer) de façon souple et naturelle avec l'agent humain.

IV.4.1 Les états sous forme de réseaux

La structure du dialogue est représentée sous forme de réseau où les nœuds représentent des questions et les transitions entre les nœuds représentent des actes de dialogue (Mc Tear, 1998, Pietquin, 2004) [PIE04]. Cette approche est connue sous le nom de réseau Bayésien, employé par Pietquin (2004) dans le cadre de l'apprentissage par renforcement dans un environnement virtuel.

IV.4.2 Les propriétés markoviennes

La structure du dialogue est basée sur les propriétés de Markov. Les actions du système de dialogue représentent les possibilités de réponses et de questions que le système peut fournir à l'utilisateur. Les états représentent les connaissances du système (ressources internes et externes qui entrent en relation avec le système). La transition entre les états détermine quelle action doit être choisie pour aller à l'état suivant. Une stratégie optimale de dialogue consiste à choisir le bon couplage entre action et état. Cette approche a été utilisée dans plusieurs travaux de (Singh et al., 2002 ; Scheffler et Young, 2002 ; Pietquin, 2004, 2006) [SIN02], [CSH02], [PIE04], [PIE06].

IV.4.3 Les cadres

Cette structure est la plus simple à gérer en termes de tâches à accomplir, car le dialogue est défini comme un cadre contenant l'ensemble des attributs variables – valeurs. Le dialogue est guidé par les questions/réponses du système. Les réponses de l'utilisateur sont les valeurs des variables. Cette approche permet d'utiliser la stratégie utilisateur-initiative qui laisse l'initiative aux utilisateurs de donner l'information, alors que le système associe les différentes réponses données aux différentes variables-valeurs (Goddeau et al., 1996 ; Chu-Carroll, 1999 ; Pieraccini et al., 2001) [GOD96], [CHU99], [PIER01].

IV.4.4 Les mises à jour de données

Cette architecture est un dérivé de l'architecture basée sur les cadres et les domaines les plus complexes dans la construction de stratégie (Bohus et al. 2003) [BOH03].

IV.4.5 L'agent collaboratif

Cette approche est utilisée dans COLLAGEN (Rich et al., 2001) [RIC01], elle modélise le comportement collaboratif des agents (humains) sur des tâches devant être partagées entre deux agents humains. La gestion de dialogue de ce système est guidée par la croyance, la tâche et le but de l'agent humain (Rich et al. 1998, Asai et al., 2005; Chu-Carroll et al., 1998,2000) [RIC98], [ASA05], [CHU00b], [CHU98].

IV.5 Conclusion

Nous avons vu les différentes tâches du gestionnaire de dialogue. Le gestionnaire de dialogue a besoin d'une stratégie pour gérer toutes ces tâches. Quel que soit le type des applications en dialogue, il y a deux aspects à considérer dans une stratégie de dialogue : degré d'initiative et stratégie de confirmation.

Le degré d'initiative est un paramètre permettant au système de choisir le mode d'interaction avec l'utilisateur (système-initiative, utilisateur-initiative ou mixte-initiative). La stratégie de confirmation (implicite ou explicite) représente, en quelques sortes, la rhétorique (l'art de bien parler) du système de dialogue. Puisque la stratégie de confirmation peut être choisie en fonction des autres facteurs du système comme le taux de reconnaissance ou la difficulté de la tâche, on peut intégrer ces facteurs dans un processus de prise de décision.

La modélisation des actes de dialogue représente le passage de connaissances entre l'agent humain et l'agent machine. Elle permet au système d'inférer les connaissances, d'induire les raisonnements afin de se comporter (dialoguer) de façon souple et naturelle avec l'agent humain. La modélisation des actes de dialogue est une formalisation des enchainements des actes de dialogue dans un SDOHM. Nous

allons voir par la suite comment utiliser ces modèles pour les intégrer dans un SDOHM par une méthode d'apprentissage.

CHAPITRE V

APPRENTISSAGE DE LA MACHINE

V.1 Introduction

Nous avons vu dans les chapitres précédents, différents modèles de dialogue humain humain et humain machine. Pour le dialogue humain-machine, nous avons vu, en particulier les approches de modélisation de la stratégie du dialogue. Posséder une stratégie du dialogue représente, pour la machine, une connaissance. La question qui se pose dans ce chapitre est de savoir comment transférer cette connaissance de l'homme vers la machine?

Nous présenterons l'état de l'art sur l'apprentissage de la machine. Nous introduirons brièvement les notions de l'apprentissage supervisé, non supervisé pour ensuite présenter plus en détail l'approche d'apprentissage par renforcement pour trouver la stratégie optimale du dialogue. L'apprentissage par renforcement qui représente les propriétés de Markov s'appelle Processus de Décision de Markov (PDM). Les propriétés de Markov s'expriment en états, actions et fonctions de transition. Dans le cas du dialogue, ce sont les actes de dialogue qui doivent être formalisés selon les propriétés de Markov pour pouvoir faire l'apprentissage par renforcement. Il y a deux façons de faire l'apprentissage par renforcement : une avec un utilisateur simulé et une autre sans utilisateur simulé.

Nous décrirons ces approches et nous illustrerons l'approche sans utilisateur simulé dans un exemple d'application du système NJFun (Singh et al., 2002). Ce système, parmi tant d'autres, est une illustration parfaite de l'utilisation de l'apprentissage par renforcement associé avec le processus de décision de Markov.

V.2 Dialogue et apprentissage par renforcement (AR)

Le problème avec cette méthode d'apprentissage est qu'il faut réaliser énormément d'essais erreurs de dialogue avant de trouver la stratégie optimale. De plus, le dialogue est dynamique et il n'existe pas de corpus de dialogue typique pour faire apprendre à la machine une stratégie optimale du dialogue, alors les recherches (Levin et al. (1998, 2000) [LEV98], [LEV00], Litman et al. (2000) [LIT00], Walker (2000) [WAT00], Scheffler et Young (2001,2002) [SCH00], [SCH02], Pietquin et Renals, (2002) [PIE02], Pietquin et Dutoit (2006) [PIE06], Schatzmann et al. (2005,2006) [SCH05], [SCH06]) se sont orientées vers l'utilisation d'un utilisateur simulé pour mettre en œuvre cette technique.

L'apprentissage par renforcement (AR) est un champ de recherche en intelligence artificielle. L'approche d'AR a été utilisée pour résoudre les problèmes de contrôle, de recherches opérationnelles, de jeux, etc. (Sutton et al., 1998; Singh et al., 2002) [SUT98], [SIN02]. Cette approche a récemment été utilisée pour gérer les interactions orales humain-machine. Lorsqu'on parle d'AR, on évoque les notions de la cognition, de la méthode d'apprentissage et de la représentation de l'objet que l'on désire faire apprendre à la machine.

Apprendre à une machine à imiter le comportement cognitif humain consiste avant tout à comprendre le processus d'apprentissage humain (i.e., étudier les capacités d'acquisition de la connaissance, du raisonnement et de la généralisation). Rappelons qu'apprendre signifie aussi, pour la machine, améliorer sa tâche d'exécution (Russell, 1995). Il y a apprentissage si et seulement si la machine maîtrise mieux sa tâche d'exécution en fonction de ce qu'elle a appris (chapitre I, section 2). Le choix du mode de représentation de connaissance a aussi un impact direct sur la mise en œuvre du processus d'apprentissage. Par conséquent, il est important de connaître l'objet que l'on veut faire apprendre à la machine avant de lui fournir une méthode d'apprentissage.

V.3 Apprendre une stratégie optimale du dialogue

Nous détaillerons ci-dessous les trois approches les plus connues de l'apprentissage machine pour trouver une stratégie optimale de dialogue.

V.3.1 Apprentissage supervisé

L'apprentissage supervisé consiste à entraîner la machine pour lui faire apprendre des nouvelles connaissances à partir des prototypes d'exemples d'observation. Ces prototypes doivent représenter des stratégies optimales du dialogue. Or il est difficile d'obtenir des exemples typiques d'un dialogue réussi qui est souvent basé sur la satisfaction de l'utilisateur. Mais la satisfaction de l'utilisateur est un facteur subjectif, donc difficile à mesurer. Cependant, l'apprentissage supervisé peut servir de départ pour apprendre une stratégie plus courante avec un minimum de corpus de données, mais certainement pas celles qui sont optimales. Par conséquent, l'apprentissage supervisé n'est pas approprié pour résoudre des problèmes d'optimisation des stratégies de dialogue (Walker, 1993). Levin et al., (2000) [LEV00], en travaillant sur le modèle stochastique pour l'apprentissage de la stratégie optimale, ont souligné que le problème avec l'apprentissage supervisé se trouve dans la nature dynamique du corpus de données. Une fois que le système est entraîné avec un type de corpus, il n'existe pas de moyen pour évaluer l'apprentissage.

V.3.2 Apprentissage non supervisé

L'apprentissage non-supervisé ne reçoit pas de prototypes d'exemples. La machine doit apprendre à détecter les corrélations entre les exemples de données pour se construire des groupes non connus à l'avance. L'approche d'apprentissage non supervisé pourrait convenir pour la recherche d'une stratégie optimale du dialogue mais il faut réaliser plusieurs essais-erreurs avant d'en trouver une qui soit optimale. C'est le principe même de l'apprentissage par renforcement.

V.3.3 Apprentissage par renforcement (AR)

La principale technique d'apprentissage par renforcement consiste à réaliser des interactions entre la machine et son environnement. Cette technique se situe entre les techniques d'apprentissage supervisées et non supervisées, mais, au lieu d'avoir des prototypes d'exemples en entrée, la machine reçoit des indications approximatives de la réponse. La machine apprend par des essais-erreurs en interaction avec l'environnement pour trouver le classement adéquat des données. Par exemple, un joueur de tennis sait qu'il gagne un point s'il a fait un bon coup et en perd un sur un coup fautif, mais personne ne lui dit comment il faut faire pour gagner un point. C'est à lui d'ajuster son coup pour parvenir à gagner le match grâce au pointage.

V.4 Principe de base d'AR

L'apprentissage par renforcement est un processus de prise de décisions qui permet à l'agent apprenant d'interagir avec son environnement (Sutton et Barto, 1989 ; Puttman, 1994). Tout ce qui est en dehors de l'agent est son environnement. On peut décrire le processus d'apprentissage par renforcement comme un processus séquentiel dans lequel un agent interagit avec son environnement en un temps discret $T = t_0, t_1, t_2, \text{etc.}$ (figure 5.1) (Pietquin, 2004).

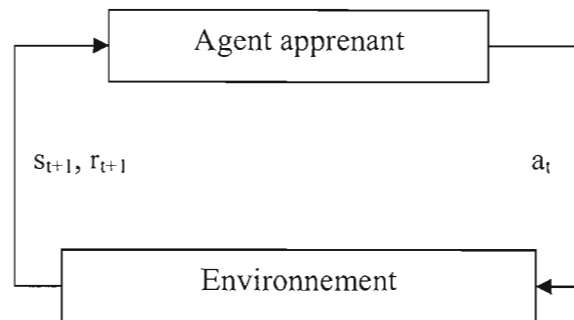


Figure 5.1 Principe d'apprentissage par renforcement.

À chaque tour d'interaction (figure 5.1), l'agent reçoit quelques observations (valeur de récompense) $R = r_1, r_2, r_3, \text{etc.}$ de son environnement (état ou state) en accord avec son action $A = a_1, a_2, a_3, \text{etc.}$ Au tour suivant, l'agent reçoit une valeur de récompense de son action et ainsi son état interne change de S_t à S_{t+1} . À chaque tour, l'agent essaie, à partir de son état S_t , d'associer ('mapping') l'état actuel avec

l'action choisie. C'est une probabilité pour faire la meilleure action possible. Ce choix est une stratégie¹ employée par l'agent apprenant.

V.4.1 Algorithme d'AR

Il n'existe pas d'algorithme particulier pour tous les problèmes d'AR, mais il y a un ensemble de problèmes d'apprentissage par renforcement. Chaque algorithme permettant de résoudre un problème d'AR est appelé un algorithme d'apprentissage par renforcement (AAR). Cet algorithme définit une stratégie de mise à jour des séquences d'actions de l'agent en se basant sur les résultats de l'expérience (Pietquin, 2004) [PIE04]. Il y a trois principaux algorithmes d'apprentissage par renforcement : l'algorithme de programmation dynamique², l'algorithme de Monte Carlo et l'algorithme de différence temporelle³ [SUT98].

La programmation dynamique est à la base de toutes les méthodes d'apprentissage par renforcement, elle est efficace et est la plus simple à implanter. Elle sert de valeur-itération et de description complète pour tous les états (connaissance complète de l'environnement) pour trouver une stratégie optimale du dialogue. Cependant, cette méthode est très coûteuse en termes d'opérations computationnelles. Par contre, l'algorithme Monte Carlo, un peu plus complexe et basé sur l'expérience, privilégie trois paramètres : les états, les actions et les valeurs de retour d'observation (fonction récompense). La combinaison de ces deux algorithmes de programmation dynamique et de Monte Carlo donne le troisième algorithme appelé l'algorithme de différence temporelle. Dans cette méthode, les deux algorithmes les plus connus sont Q-Learning et Sarsa (Watkin; 1989, 1992) [WAT89], [WAT92]. L'algorithme Q-Learning permet de trouver l'action optimale en réalisant les expériences des essais-erreurs tout en poursuivant la stratégie en cours. Ainsi l'agent peut avoir un comportement cohérent pendant l'apprentissage [PIE04]. La différence entre Q-Learning et Sarsa se situe dans le mécanisme d'apprentissage par renforcement. L'algorithme Q-Learning acquière un

¹ On emploie aussi le terme *politique* pour dire *stratégie*.

² Traduit d'anglais de *Dynamic Programming*.

³ Traduit d'anglais de *Temporal Difference*.

comportement optimal mais prend des risques (e.g., il peut recevoir des valeurs de récompense négatives, c'est-à-dire, la machine apprend des mauvaises actions) tandis que l'algorithme Sarsa ne peut pas obtenir de comportements optimaux mais obtient des comportements sans risques (Takadama et H. Fujita , 2005) [TAK05].

Il existe d'autres algorithmes qui combinent ces différents algorithmes de programmation dynamique et de différence temporelle comme l'algorithme 'Eligibility Traces' qui est basé sur un mécanisme d'affectation temporaire de valeur⁴ [SUT98], [KAE96]. Le problème avec l'apprentissage par renforcement est d'affecter la bonne valeur à la fonction récompense pendant le déroulement du dialogue, mais la valeur ne peut être connue qu'à la fin du dialogue, alors l'astuce est de donner une valeur arbitraire en fonction de l'état actuel du dialogue et de la réajuster à la fin du dialogue.

V.4.2 Fonction récompense

L'apprentissage par renforcement est basé sur la valeur du retour d'observation représentée par une fonction mathématique appelée fonction récompense. Le but de l'agent est donc de maximiser la quantité totale des valeurs de récompense après une longue exécution par essais-erreurs [PIE04]. La fonction récompense R représente la somme de toutes les valeurs du retour d'observation à chaque instant t (e.q. 1)

$$R_t = r_{t+1} + r_{t+2} + \dots + r_T \quad (\text{eq. 1})$$

Cette valeur de récompense R_t suppose que les interactions entre l'agent et son environnement se terminent en un temps T (temps total des t_0, t_1, t_2, \dots). Pietquin explique que, par exemple, une tâche qui se termine en temps T est une *tâche épisodique* et un *épisode* correspond à l'exécution de cette tâche à l'instance T . La valeur R_t est appelée la valeur de *retour non escompté*⁵. Si les interactions (des tours de parole) ne se terminent pas, la tâche est continue et la valeur de R_t ne sera jamais

⁴ Terme traduit d'anglais *temporal credit assignment*

⁵ Terme traduit d'anglais de *non-discounted return*.

obtenue, elle est donc infinie ou non déterminée. Dans ce cas, on ne peut pas utiliser la valeur de retour \mathbf{R} comme fonction récompense pour apprendre une stratégie optimale. C'est pour cette raison que le concept de *retour escompté*⁶ est introduit :

$$\mathbf{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (\text{eq.2})$$

Où $\gamma = [0,1]$ est le taux d'escompte. Cette valeur détermine la présence de la valeur de la fonction récompense. Si $\gamma = 0$, l'agent est seulement concerné par la valeur de récompense $\mathbf{R}_t = r_{t+1}$ (valeur immédiate). Si γ approche de 1, la R sera plus grande car il faut tenir compte des autres valeurs ultérieures (i.e., r_{t+2} , r_{t+3} , etc.) Si l'ensemble des valeurs de r_{t+k} est borné alors la valeur de \mathbf{R}_t sera finie ou déterminée.

La plupart des algorithmes d'AR sont basés sur l'estimation des deux fonctions de valeur suivantes pour une stratégie π donnée : la fonction État-valeur $\mathbf{V}^\pi(\mathbf{s})$ et la fonction Action-valeur $\mathbf{Q}^\pi(\mathbf{s}, \mathbf{a})$. La fonction $\pi(\mathbf{s}, \mathbf{a})$ définit la probabilité d'exécution d'une action \mathbf{a} à l'état \mathbf{s} . Si la stratégie est déterministe alors $\pi(\mathbf{s}, \mathbf{a}) = 1$ (i.e., l'action est toujours exécutée quand on est dans l'état \mathbf{s}).

Ainsi, une stratégie optimale π^* est obtenue par le résultat de la fonction \mathbf{R}_t à l'instant \mathbf{t} et à l'action \mathbf{a} . Le but de l'agent d'apprentissage est d'acquérir une stratégie optimale à la fin du processus d'apprentissage.

La valeur de $\mathbf{V}^\pi(\mathbf{s})$ de l'état \mathbf{s} sous la stratégie π est donnée par une estimation de valeur de retour attendue à partir de l'état \mathbf{s} en suivant la stratégie π . La valeur de $\mathbf{Q}^\pi(\mathbf{s}, \mathbf{a})$ de l'exécution de l'action \mathbf{a} dans l'état \mathbf{s} sous la stratégie π est définie comme une estimation de la valeur de retour attendue lors de l'exécution de l'action \mathbf{a} à l'état \mathbf{s} en suivant la stratégie π . Une stratégie optimale π^* est une stratégie résultante en une valeur maximale de $\mathbf{V}^*(\mathbf{s})$ pour tous les états pour lesquels l'agent suivra cette stratégie optimale par sélection dans chaque état une action qui a une valeur maximale de \mathbf{Q}^* .

⁶ *Discounted return* en anglais.

La tâche d'apprentissage, ici, consiste à optimiser l'interaction entre l'humain et la machine. Donc, son but est de trouver une stratégie qui maximise la valeur de \mathbf{R} . Cette valeur peut être calculée récursivement à l'aide de la fonction valeur-état $V^\pi(\mathbf{s})$ de la stratégie π et de la fonction état-action ou Q-learning $Q^\pi(\mathbf{s}, \mathbf{a})$ de la stratégie π . Les fonctions d'optimisation associées sont $V^*(\mathbf{s})$ et $Q^*(\mathbf{s}, \mathbf{a})$ telles que :

$$V^*(s) = \max_{\pi} V^\pi(s) \quad (\text{eq. 3})$$

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) \quad (\text{eq. 4})$$

Le processus d'apprentissage par renforcement est basé sur le calcul de ces valeurs en faisant des cycles d'essais-erreurs. Ainsi, chaque état et chaque paire action-état peuvent être visités et exécutés une infinité de fois pour estimer la valeur de \mathbf{V} et de \mathbf{Q} . Ce processus est appelé l'*exploration* des états. Une fois l'apprentissage fait, l'agent doit être capable, dans un environnement réel, de choisir une action (prendre une décision) en fonction de ce qu'il a pris ultérieurement. Ce choix ne devrait pas être trop écarté de ce qu'il a appris. C'est le processus d'*exploitation*. Et le problème avec l'apprentissage par renforcement est aussi l'écart entre l'*exploitation* et l'*exploration* des différentes possibilités de stratégies. L'agent apprenant doit *exploiter* les stratégies apprises, afin d'obtenir la valeur de récompense, mais il doit aussi *explorer* des nouvelles stratégies afin de découvrir les meilleures actions possibles [PIE04], [CHU00a], [CHU00b], [SUT98], [KAE96]. Pour être efficace dans son apprentissage, l'agent apprenant doit choisir entre essayer différentes actions et distinguer celles qui semblent être les plus efficaces. Les méthodes de base pour cela sont *ϵ -greedy* et *softmax*. La méthode *ϵ -greedy* consiste à choisir la meilleure action en fonction de la probabilité $(1-\epsilon)$ sinon choisir une action aléatoire. La méthode *softmax*, quant à elle, est basée sur la méthode de Boltzmann qui se fie à sa *Q-valeur* pour choisir une action de manière à ne pas exécuter trop souvent une mauvaise action.

V.5 Processus de Décision de Markov

V.5.1 Propriété de Markov

Un problème d'apprentissage par renforcement qui satisfait les propriétés de Markov est un **Processus de Décision de Markov (PDM)**. Ce processus est décrit en termes d'états, d'actions, de transitions et de récompenses. La décision prise à l'instant courant ne dépend que de l'état et de l'action de l'instant précédent.

Lorsque l'état du monde (objet, concept, etc.) peut être décrit de manière compacte, formelle et dans un ordre de séquences successives interceptant toute (ou presque toute) l'information, alors on dit que cet état représente des propriétés markoviennes ou simplement un état de Markov. Les propriétés de Markov sont formellement décrites dans l'équation mathématique suivante :

$$P(S_{t+1}, r_{t+1} | S_t, a_t, r_t, S_{t-1}, a_{t-1}, \dots, r_1, S_0, a_0) = P(S_{t+1}, a_{t+1} | S_t, a_t) \quad (\text{eq. 5})$$

La probabilité de passer à l'état s_{t+1} et de recevoir une valeur de récompense r_{t+1} dépend de l'action a_t , de r_t et de l'état s_t , qui lui-même dépend de l'état à l'instance $t-1$, ainsi de suite jusqu'à l'état initial s_0 et l'action initiale a_0 . L'idée principale de propriété de Markov réside dans les paramètres de décision qui influencent le choix des actions. Ce choix est déterminé uniquement en fonction de l'état précédent. En effet, l'état et la valeur de récompense à l'instant $t+1$ ne dépend pas seulement de l'action et de l'état à l'instant t (l'instant présent) (eq. 5). Autrement dit, seul l'état et l'action précédents sont suffisants pour prédire l'état courant, nul n'a besoin de connaître plus de détails sur les états et actions antérieurs.

V.5.2 Processus de décision de Markov dans le dialogue

Dans le cas de la gestion du dialogue, les états représentent l'historique et le contexte du discours, les actions sont les actes de dialogue, les transitions entre les états sont les décisions quant aux choix de l'action à exécuter pour passer d'un état à un autre, et les récompenses sont les valeurs de retours donnés par l'environnement.

Cette valeur permet à l'agent machine d'améliorer sa tâche d'exécution donc d'augmenter sa performance.

Les récents travaux de recherches suggèrent que nous utilisons le modèle du processus de décision de Markov comme algorithme pour les séquences d'action dialogique et comme base pour la modélisation des problèmes d'optimisation de stratégie du dialogue (Biermann et Long, 1996; Levin et al., 2000; Walker et al., 2000; Pietquin, 2004; Cuayahuilt et Renals, 2006, Schatzmann et al., 2006). Un PDM est défini par 4-tuplets $\{S, A, T, R\}$ où S est un ensemble d'espaces états, A est un ensemble d'actions, T est la distribution de probabilité de transition au-delà de l'espace d'état et R est la distribution des récompenses attendues. Ainsi, un PDM appliqué dans le dialogue est défini tel que :

. L'ensemble des états $S = \{s_1, s_2, \dots, s_n\}$. Ces états représentent les énoncés possibles du domaine concerné. Un état à l'instance t est s_t ou s , à l'instance $t+1$ est s_{t+1} ou s' . Dans le domaine de réservation de chambres d'hôtel par exemple, un état du dialogue peut être défini comme un concept tel que, par exemple $s_1 = (\text{date} : \text{inconnue}, \text{ch} : 0)$ et $s_2 = (\text{date} : 12\text{-fev}, \text{ch} : 0)$ où *date* désigne la date de réservation et *ch* contient le nombre de chambres à réserver.

. Un ensemble d'actions $A = \{a_1, a_2, \dots, a_m\}$. Ces actions désignent les actes de dialogue. Une action exécutée à l'instance t est nommée a_t ou a , à l'instance $t+1$ est a_{t+1} ou a' . Par exemple $a_1 = (\text{Pour quelle date voulez-vous réserver ?})$, l'action a_1 fait progresser le dialogue de l'état s_1 (*date* est inconnue) à l'état s_2 (*date* est connue).

. Une fonction de transition, $T: S \times A \rightarrow S$. Lorsqu'une action est exécutée, elle fait progresser le dialogue. La propriété importante du PDM est la dépendance entre les actions et les états telle que définie par la formule $P(s_{t+1}, r_{t+1} | s_t, a_t)$ où P est la probabilité de la transition de l'état s_t à s_{t+1} en fonction uniquement de l'action et état courant. À chaque fois qu'une action est prise, une valeur de récompense est attribuée à cette action dans cet état.

. Une fonction récompense $R(s_t)$ est la valeur de récompense reçue à l'état s (éq.1) telle que définie dans la fonction récompense de l'AR.

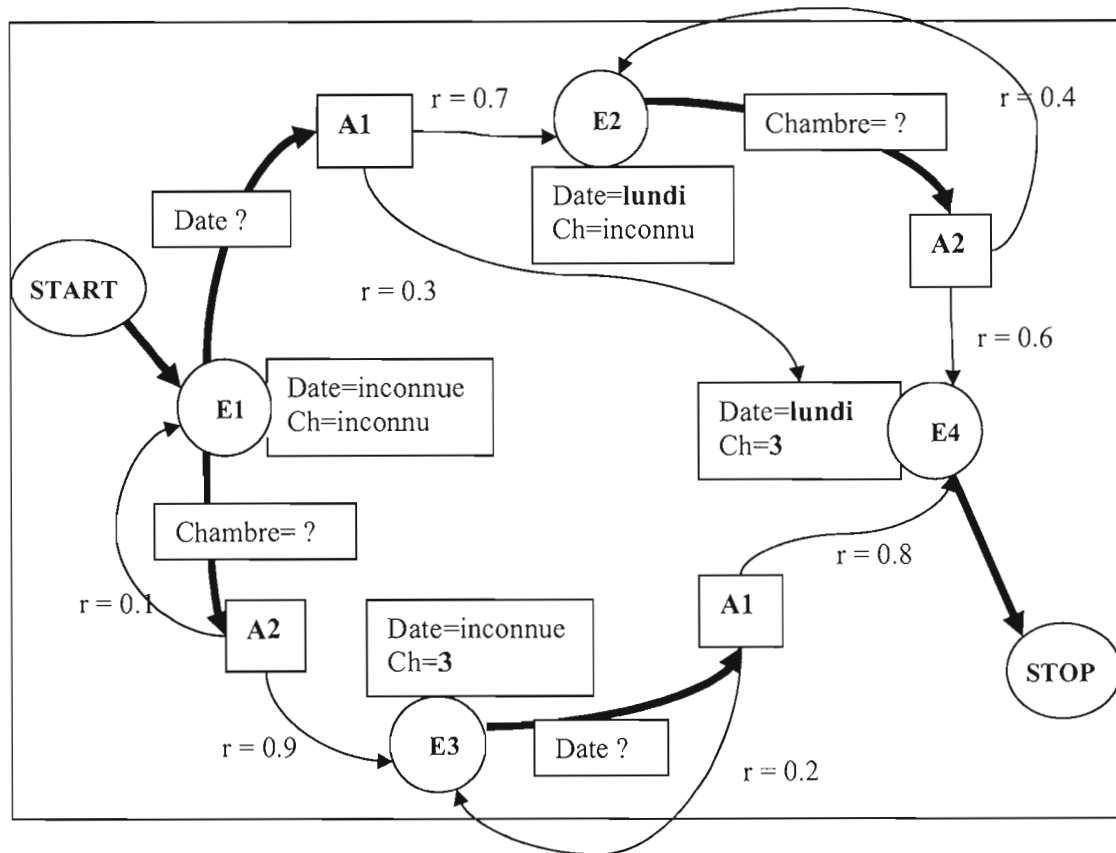


Figure 5.2 Exemple d'un état markovien appliqué dans les actes de dialogue pour un domaine de réservation des chambres d'hôtel.

La figure 5.2 montre un exemple de représentation des séquences d'actes de dialogue selon les propriétés markoviennes. Supposons que, pour réserver une chambre d'hôtel, nous avons seulement besoin de connaître la date de réservation (Date=inconnue) et le nombre de chambres à louer (Ch=inconnu). Nous avons au moins (il y aura certainement plus d'états en réalité) six états markoviens (représenté par des cercles) et deux actions (les carrés). Chaque action, dans un état, a une valeur de réponse associée (i.e., $r=0.9$). Au début, on se trouve à l'état E1 (Date=inconnue, Ch=inconnu) qui représente l'état où la date et le nombre de chambres sont inconnus. L'état E1 a deux actions possibles A1 (Date=?) et A2 (Chambre=?). L'action A1, exécutée par l'agent apprenant, consiste à poser la question « quand

voulez-vous réserver? » par exemple. La réponse à cette question (par l'agent apprenant ou l'utilisateur), « *lundi* », par exemple, permet de passer de l'état E1 à l'état E2 (Date=lundi, Ch=inconnu) avec une valeur de récompense $r = 0.7$. Dans certains cas, quoique la probabilité est faible ($r=0.3$), l'utilisateur peut donner toute information en une seule action, par exemple « *je voudrais réserver pour lundi et nous avons besoin de trois chambres* », alors, dans ce cas, l'agent apprenant peut passer directement à l'état E4 où la date et le nombre de chambres à louer sont connus (i.e., Date=lundi, Chambre=3) et ainsi le processus se termine à l'état Stop.

V.6 Approche d'AR

Les expériences de conception des systèmes de dialogue avec des connaissances préalables accélèrent la tâche d'apprentissage de la machine (Singh et al., 2002; Pietquin, 2004), par exemple, pour apprendre une stratégie optimale du dialogue, nous pouvons nous demander quel type d'apprentissage serait convenable dans ce domaine, supervisé, non-supervisé, renforcement ou hybride.

L'approche d'apprentissage par renforcement offre plus de possibilités de solutions en recherche de stratégie optimale du dialogue que les deux autres approches (supervisée et non supervisée).

L'approche d'apprentissage par renforcement demande une interaction directe entre l'agent apprenant et l'utilisateur car ce dernier doit donner son avis (récompense ou punition) sur la tâche réalisée par l'agent apprenant. Cette interaction peut être faite entre un utilisateur réel et l'agent apprenant. Cependant, il faut interagir des millions de fois pour faire assimiler correctement à l'agent apprenant une stratégie optimale. Cela demande beaucoup de temps et de ressources. Pour résoudre ce problème, il existe deux solutions : soit on crée un utilisateur qui simule l'utilisateur réel pour interagir avec l'agent apprenant, soit on réduit le domaine d'application à un niveau raisonnable pour que l'interaction entre l'utilisateur réel et l'agent apprenant ne soit pas une tâche fastidieuse.

La première approche exige que la réponse de l'utilisateur simulé soit aussi proche et réaliste que celle de l'utilisateur réel. La seconde approche demande des compétences de l'expert en dialogue pour déterminer, au préalable, tous les états et les actions possibles dans un domaine étudié et l'expert en dialogue doit savoir comment réduire l'espace d'état du domaine à un niveau raisonnable.

Pour éclaircir ces approches, nous proposerons, dans ce qui suit, un état de l'art sur l'utilisateur simulé et son implantation dans la méthode d'apprentissage par renforcement. Ensuite, nous illustrerons la seconde approche en présentant le système NJFun (Singh et al., 2002) qui utilise l'approche sans utilisateur simulé.

V.7 AR avec un utilisateur simulé

V.7.1 Architecture

L'architecture générale de l'apprentissage avec un utilisateur simulé est illustrée dans la figure 5.3. D'abord, il faut apprendre à l'utilisateur simulé des stratégies optimales. Cet apprentissage se fait par la technique d'apprentissage supervisé en utilisant des exemples d'observation à partir de données réelles provenant du dialogue humain-machine ou par la conception manuelle des réponses. Ensuite, l'utilisateur simulé interagit en direct avec le système qui apprend à réajuster sa stratégie en fonction des réponses et des remarques de l'utilisateur simulé. Cette forme d'entraînement permet de générer un grand nombre de dialogues et de créer des scénarios de stratégie qui ne se présentent pas dans le dialogue humain-machine. Donc, c'est un avantage énorme pour l'exploration des différentes possibilités stratégiques.

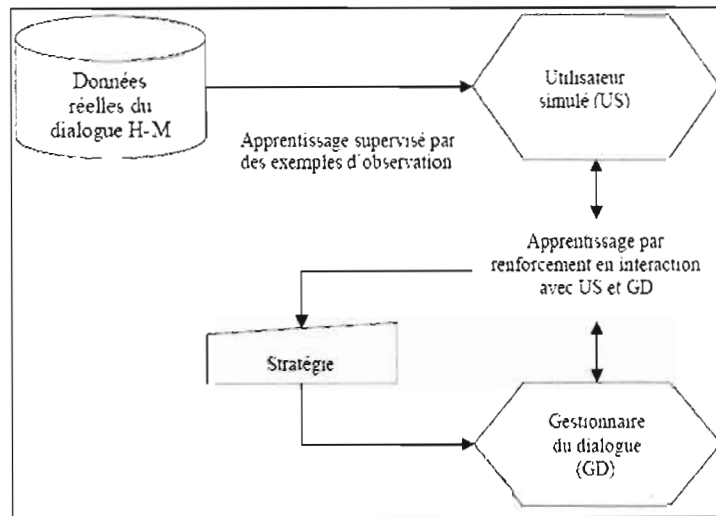


Figure 5.3 L'utilisateur simulé apprend des stratégies optimales du dialogue à la machine (adaptée de Schatzmann et al., 2006).

V.7.2 Utilisateur simulé

La meilleure façon de savoir si un comportement dialogique humain-machine est proche de celui des humains consiste à implémenter une stratégie de dialogue dans ce système pour ensuite le tester avec des utilisateurs réels et obtenir leurs commentaires. Ce processus est continu jusqu'à ce qu'on trouve un dialogue satisfaisant donc une stratégie optimale du dialogue. On arrête le processus de test et on implémente cette stratégie dans un vrai système pour son exploitation (mettre en opération). Cette façon de procéder est la base de l'apprentissage par renforcement. Cependant, faire apprendre la machine avec des utilisateurs réels est très couteux compte tenu de la quantité des essais-erreurs à réaliser. On a donc décidé de créer un **utilisateur simulé** qui remplace l'humain dans cette tâche. Le rôle de l'utilisateur simulé est : (1) reproduire l'intention de l'humain à travers l'expression orale, (2) entraîner la machine à acquérir une stratégie optimale en lui retournant des indices sur son action (i.e., correct, incorrect), (3) générer de grande quantité de scénarios de dialogue afin de permettre à la machine d'explorer toutes les stratégies possibles du dialogue là où les experts en dialogue ne voient pas tout. Par exemple, le projet

TALK⁷ (2006), utilise l'approche AR avec un utilisateur simulé. L'idée d'avoir un utilisateur simulé de dialogue pour identifier les stratégies de dialogue a été proposée par plusieurs chercheurs :

Levin et al. (1998) [LEV98] ont prouvé qu'un système de dialogue peut converger vers une stratégie raisonnable, même si on l'apprend avec une connaissance vierge au départ. Levin, Pieraccini et Eckert (2000) [LEV00] ont fait des simulations au niveau de l'intention plutôt qu'au niveau acoustique. Scheffler et Young (2000) ont trouvé que l'utilisateur simulé peut aider à trouver des stratégies de dialogue que les experts et développeurs ne voient pas. Litman et al. (2000) [LIT00] ont conclu que l'approche d'apprentissage par renforcement (AR) permet, en recherchant dans un espace d'états, d'optimiser empiriquement une stratégie de dialogue de façon beaucoup plus efficace (en recherchant dans un espace d'états) qu'en employant les méthodes traditionnelles (concevoir et tester empiriquement plusieurs versions de stratégies avant d'en choisir une pour l'implanter dans le système). En 2001, les travaux de Lin et Lee [LIN01] ont démontré que le simulateur est très utile dans l'analyse et la conception des systèmes de dialogue. En 2002, Levin et al. [LEV02] ont continué leurs travaux dans cette direction et ont aussi démontré qu'on peut apprendre un comportement relativement complexe d'utilisateurs en choisissant quelques critères simples. Par exemple, choisir adéquatement une représentation de l'espace d'états et un juste modèle d'utilisateur simulé qui soit adapté au domaine d'application. Leurs études sont comparables aux recherches heuristiques. Pietquin et Renals, (2002) [PIE02], dans leurs essais avec le réseau bayésien, ont démontré que, après plusieurs tests de simulation, le système reste stationnaire et semble converger vers une stratégie optimale. Chung (2004) [CHU04] a démontré que le simulateur faciliterait le développement des nouvelles applications des systèmes de dialogue, car il y a des millions de dialogues artificiels disponibles à développer et à tester. Les travaux de Schatzmann et al. (2005,2006) [SCH05], [SCH06] ont avancé que la recherche future dans le domaine du dialogue devrait inclure l'apprentissage machine basé sur l'utilisateur simulé. Ils ont démontré

⁷ Projet TALK. Disponible sur : (<http://www.talk-project.org/>) [consulté le 1 oct 2006].

que le choix des modèles d'utilisateurs a un impact significatif sur la stratégie apprise, ce qui veut dire que si on développe des modèles d'utilisateurs simulés plus proches des utilisateurs réels, cela rendra la simulation et l'environnement simulé plus réaliste. Cependant, en 2006, Pietquin et Dutoit [PIE06] ont affirmé que la simulation de dialogue et l'AR ne modélise pas entièrement la conception du dialogue, mais une première stratégie acceptable pourrait être obtenue avec la méthode bayésienne.

V.8 AR sans utilisateur simulé

On peut implanter une approche d'apprentissage par renforcement sans employer l'utilisateur simulé. Dans ce cas, il faut déterminer tous les cas possibles du dialogue, ensuite faire des essais (entraîner l'agent machine) avec l'utilisateur réel et avoir leurs commentaires (récompense ou punition) sur le comportement de l'agent machine (la stratégie utilisée par l'agent machine). Le système NjFun a été conçu autour de ces idées. Le système NjFun, construit en 2002 [SIN02], est un système qui informe les utilisateurs sur les différentes activités d'attraction au New Jersey. Ce système fonctionne en temps réel et est basé sur la plate-forme du système de dialogue de Levin, Pieraccini, Eckert, Fabrizio et Narayanan qui a été construite en 1999 [SIN02]. NjFun est composé de cinq modules : reconnaissance, compréhension, gestion du dialogue, accès aux bases de données et synthèse de la parole.

Le module de reconnaissance et compréhension utilise l'engin de reconnaissance Watson basé sur un modèle stochastique de langue et un modèle de compréhension. L'engin de reconnaissance Watson est entraîné à partir d'exemples de dialogues entre utilisateurs.

Le module de gestion du dialogue utilise l'approche d'AR pour entraîner le système à trouver une stratégie optimale. Ce module utilise le langage DMD⁸ pour

⁸ DMD (*Dialogue Manager Developer*) est un langage utilisé pour développer les actions de dialogue aussi bien que de construire les stratégies de dialogue.

gérer le dialogue. La génération du texte est incorporée dans la réponse générée par le module de gestion du dialogue.

Le module d'accès aux bases de données permet de rechercher les informations depuis la page Web *nj.online* qui contient les informations relatives aux activités proposées par la ville telles que les visites aux zoos, aux parcs d'attractions, aux musées, etc.

Enfin, le module de synthèse de la parole utilise la méthode de concaténation des diphonies pour transformer la réponse issue du module de gestion du dialogue en signal acoustique.

Ce système a su montrer que les stratégies optimales de dialogue peuvent être apprises par des techniques simples en apprentissage par renforcement (Monte Carlo, PDM) dont les tâches sont bien décrites (42 états, 2 actions possibles sur chaque état). Mais, surtout, selon les auteurs, Singh et al. (2002) [SIN02], ces résultats ont démontré que l'approche d'apprentissage par renforcement permet d'exploiter de manière empirique un plus large espace états (situations dialogiques) que les approches classiques. Néanmoins, plusieurs défis restent à résoudre afin de rendre le système plus robuste en termes de stratégie dialogique, comme par exemple, de trouver des méthodes pour réduire l'espace d'états à une taille gérable. Dans le cas du système NJFun, la méthode pour réduire l'espace d'états réside dans le choix des cas spécifiques. Les actions de dialogue sont représentées par des opérations vectorielles de 14 variables. Les états de dialogue se composent de 7 variables. Après l'analyse préliminaire, les concepteurs du système ne retiennent que 42 choix de stratégie de dialogue (e.g., on ne commence pas le début du dialogue par un état 'good bye') et pour chaque stratégie de dialogue, il n'y a seulement deux actions possibles, ce qui donne 2^{42} possibilités de stratégie de réponse. Ce choix est justifié par le fait qu'il travaille avec un modèle déterminé et un corpus minimal de données pour générer une stratégie optimale. Nous allons décrire ci-dessous la technique de conception du système NJFun.

V.8.1 Exemple du système NjFun

V.8.1.1 Les étapes de construction du système NJFun

Singh et al. (2002) [SIN02] proposent cinq étapes de construction d'un système de dialogue qui optimise son comportement par apprentissage à partir des données de dialogue humain-machine.

- 1- Choisir une fonction de calcul de coût approprié. La fonction Monte Carlo a été choisie pour ce calcul. Ensuite, il faut choisir une représentation appropriée des états de dialogue (les cas) et construire une stratégie de dialogue qui associe chaque état à un ensemble d'actions raisonnables. Les auteurs font remarquer que, dans beaucoup de cas, il n'y a seulement qu'une action possible,
- 2- construire un système initial d'apprentissage basé sur ces états. À partir de ce système initial, on peut explorer l'ensemble des données, en dépit de cette exploration, le système devrait encore créer les actions élémentaires souhaitées,
- 3- utiliser les données du dialogue apprises pour construire un modèle empirique de l'espace d'états selon les propriétés de processus de décision de Markov (PDM). La transition de ce PDM modélisera les réactions et les remarques des utilisateurs pour les actions variées du système,
- 4- exécuter l'algorithme de recherche de stratégie optimale du dialogue en accord avec le modèle MDP et enfin,
- 5- réimplanter le système de dialogue en utilisant la nouvelle stratégie optimale apprise.

V.8.1.1 Fonctionnement du gestionnaire du dialogue du NJFun

Ce système utilise la méthode attributs-valeurs⁹ pour obtenir les données venant de l'utilisateur et gérer les séquences de dialogue. Dans cette méthode, le système se réduit à utiliser trois attributs : activité, location et temps (i.e., *morning*, *afternoon*, *night*). Au début du dialogue, NJFun demande à l'utilisateur d'entrer une

⁹ Termes traduit en anglais *form-filling* ou *slot-value*.

valeur (initialisée par défaut à l'attribut activité). Si l'utilisateur dit une phrase dans laquelle NJFun 'reconnaît' les mots employés et qu'il arrive à attribuer correctement ces valeurs (mots) aux attributs courants alors il passera à la question suivante, sinon, il emploiera une action appropriée pour éclaircir la question (i.e., poser une question explicite ou répéter la question selon les taux d'erreurs retournés par le module de reconnaissance). À la fin de chaque scénario de dialogue, l'utilisateur est invité à donner son opinion sur le déroulement du dialogue ou aller sur le site Internet du NJFun et répondre au sondage. Ces réponses constituent les paramètres de la fonction de calcul de coût ou fonction récompense qui permet au gestionnaire de dialogue de réajuster les actions du dialogue et de choisir les actions optimales pour le prochain dialogue. C'est le point clé dans la méthode d'apprentissage par renforcement de ce système.

<p>S1: Welcome to NJFun. How may I help you? U1: I'd like to find um winetasting in Lambertville in the morning. <i>(ASR: I'd like to find out wineries the in the Lambertville in the morning.)</i> S2: Did you say you are interested in Lambertville? U2: Yes. S3: Did you say you want to go in the morning? U3: Yes. S4: I found a winery near Lambertville that is open in the morning. It is the Poor Richard's Winery in Lambertville. Thank you for using the system. Please give me feedback by saying 'good', 'so-so', or 'bad'. U4: Good. S5: Goodbye and have a nice day!</p>
--

Tableau 5.1 Exemple de dialogue généré par NJFun (extrait de [SIN02]). Le système utilise une stratégie *user-initiative* à l'ouverture du dialogue (S1) ensuite des confirmations explicites sont employées à chaque réponse de l'utilisateur (S2, S3).

V.8.1.1 Fonction récompense

Pour calculer la fonction récompense, le système emploie la technique *binary completion*, une simple technique qui prend la valeur 1 si le dialogue est satisfaisant

et -1 dans le cas contraire. Cette technique permet aussi d'évaluer la performance du système en termes du succès de la tâche, de la qualité et de l'efficacité du dialogue.

V.9 Conclusion

Nous avons vu l'architecture d'apprentissage par renforcement avec un utilisateur simulé pour apprendre à la machine à acquérir une stratégie optimale du dialogue. L'utilisation de la technique d'apprentissage par renforcement a plusieurs buts : elle permet de gérer les situations incertaines, de contrôler l'historique du dialogue et de permettre à la machine de rester dans le contexte du dialogue (contexte du discours). Avec un utilisateur simulé, la machine peut exploiter toutes les possibilités de stratégie du dialogue même celles qui ne sont pas vues par les experts en dialogue. Nous avons vu les propriétés de Markov. L'apprentissage par renforcement qui satisfait les propriétés de Markov est appelé processus de décision de Markov.

CHAPITRE VI

MODÉLISATION DU PROCESSUS D'APPRENTISSAGE POUR UNE STRATÉGIE OPTIMALE DU DIALOGUE

VI.1 Introduction

Nous proposerons, dans ce chapitre, un modèle d'apprentissage pour améliorer la performance du système de dialogue. C'est un modèle d'apprentissage par renforcement avec un utilisateur simulé appliqué dans un système de dialogue qui emploie les principes de système-initiative (chapitre IV.3, voir section sur le degré d'initiative). Les échanges sont de nature questions-réponses (la machine pose des questions et l'utilisateur répond) et s'orientent vers un but. Notre approche est suffisamment générale pour pouvoir être appliquée dans n'importe quel domaine pourvu qu'elle respecte ces principes.

Le modèle d'apprentissage par renforcement à l'aide d'un utilisateur simulé que nous proposons permet à la machine d'acquiescer une stratégie optimale de dialogue. Un système avec une stratégie optimale de dialogue apprise sera meilleur. Rappelons qu'une stratégie optimale désigne une façon de communiquer entre deux interlocuteurs et elle permet aussi de contrôler les interactions entre la machine et l'humain. Cette façon de communiquer doit refléter le plus possible le comportement dialogique de la plupart des utilisateurs.

Nous avons vu dans l'état de l'art que le dialogue entre humains a une certaine structure et régularité et que cette dernière peut être modélisée (Chap. II) par différents modèles de gestion du dialogue (Chap. IV). Nous pouvons apprendre à la machine les règles de contrôle de ces interactions (Chap. V). Ces règles de contrôle

établissent une stratégie optimale du dialogue qui conduit à un dialogue réussi, sinon elle conduit à un dialogue non réussi ou échec.

Nous décrirons, d'abord, dans ce chapitre, les critères de mesure quantitative et qualitative pour déterminer si un dialogue est réussi ou non (section VI.2), ce sont des critères pour mesurer la qualité du dialogue. Ensuite, nous proposerons des règles pour déterminer une stratégie optimale à partir de ces critères (section VI.3). Enfin, nous vous exposerons un modèle d'apprentissage permettant à la machine d'apprendre cette stratégie (section VI.4).

VI.2 Critères de mesure de la qualité du dialogue

Nous proposons que la qualité du dialogue humain-machine peut être mesurée en se basant sur le nombre de tours de parole, sur le nombre d'informations non complétées (mesure quantitative) et sur les rapports entre les questions de type direct, explicite, implicite et répétitif (mesure qualitative). Ces questions doivent être réparties de manière équitable dans un dialogue. Pour pouvoir les mesurer correctement, il faut s'entendre sur la définition de ce qu'est un dialogue réussi et ce qui ne l'est pas.

VI.2.1 Définitions

VI.2.1.1 Dialogue réussi

Un dialogue est considéré comme réussi si le but de dialogue est atteint, c'est-à-dire, si le nombre de tours de parole (i.e., une question suivie d'une réponse) est minimal par rapport aux informations demandées. De plus, le système ne doit pas employer les mêmes énoncés (e.g., répéter toujours les mêmes questions).

Dans un système de type question-réponse où le système pose des questions et l'utilisateur donne sa réponse, le but du système est considéré comme atteint si toutes les demandes sont confirmées par l'utilisateur. Par exemple, au cours d'un dialogue, l'utilisateur peut répondre spontanément aux questions (e.g., hésitations, phrases agrammaticales, changements d'avis, etc.). Cependant la machine ne doit pas répéter toujours la même question, elle doit savoir varier les types de questions

de manière à ne pas rendre le dialogue monotone, ennuyant et agaçant. Au cas où l'utilisateur sort du contexte du dialogue, la machine doit aussi savoir trouver une *stratégie* pour ramener l'utilisateur dans le contexte du dialogue afin d'atteindre son but.

VI.2.1.2 Dialogue non-réussi

Un dialogue non réussi ou échec est un dialogue dont le but n'est pas atteint, dont la stratégie appliquée ne répond pas aux critères de mesure quantitative (e.g., les tours de parole sont trop longs) et qualitative (e.g., il y a trop de questions répétitives, etc.).

VI.2.2 Modélisation des critères de mesure

Lors de l'apprentissage par renforcement, il faut déterminer à l'avance quels sont des paramètres qui déterminent une action appropriée associée à un état donné. Ces paramètres sont exprimés selon des critères de mesure quantitative et qualitative.

VI.2.2.1 Mesure quantitative

Nous proposons que cette mesure quantitative soit basée **seulement** sur le nombre de tours de parole et le nombre d'informations non complétées. Le nombre de tours de parole doit être minimal et ne doit pas dépasser un seuil maximal acceptable par rapport au nombre d'informations demandées. Car la mesure quantitative sert de critère d'éliminatoire dans le dialogue ; elle permet d'éliminer les dialogues non pertinents ou échecs. Par exemple, si les deux agents (utilisateur et machine) veulent échanger seulement deux informations (e.g., date et heure du rendez-vous), à supposer que l'utilisateur soit coopératif (i.e., répond clairement et correctement aux questions), si la machine doit poser plus de vingt questions à l'utilisateur pour obtenir la confirmation des deux informations alors on peut conclure que la stratégie employée dans ce dialogue est pauvre puisqu'elle elle génère un dialogue non réussi. Donc, le nombre de tours de parole par rapport au nombre d'informations complétées est suffisant pour éliminer un dialogue non réussi.

Nous définissons donc une équation pour une mesure quantitative du dialogue de la manière suivante :

$$1.6 * N_s \leq N_{dt} \leq 3.0 * N_s \quad (1)$$

N_{dt} est le nombre total des tours de parole dans un dialogue et N_s est le nombre d'informations demandées dont le système essaie d'avoir la confirmation auprès de l'utilisateur. Par exemple, un système avec quatre informations, le nombre de tours de parole doit être compris entre 6 tours (i.e., $1.6*4 = 6.4$) et 12 tours (i.e., $3.0*4$). Puisqu'on considère que la qualité du module de reconnaissance dans un SDOHM est moyennement acceptable (sans être performante), alors, il faut au moins que le système pose des questions de vérification à l'utilisateur pour avoir la confirmation des quatre informations demandées. Dans ce cas, moins de 6 tours n'est généralement pas recommandé car le système doit poser deux questions de confirmation en plus quatre questions pour les quatre informations demandées. En dessous de ce seuil, le système ne sera pas être sûr de '*comprendre*'. Plus de 12 tours de parole témoigne d'une mauvaise qualité de dialogue (e.g., le taux de reconnaissance est trop mauvais ou le système pose des questions répétitives, redondantes, ce qui agace l'utilisateur).

VI.2.2.2 Mesure qualitative

Lorsque cette mesure est appliquée en apprentissage par renforcement, on la place au niveau de la fonction objective ou fonction récompense. La fonction objective permet donc de mesurer la performance et la qualité du dialogue. Par exemple, Levin et al. (2000, 2002) définissent une fonction objective $C = \sum C_i$ où C représente la somme des mesures de performances telles que le nombre d'interactions, d'erreurs et d'attributs non complétés, etc. Puisque nous avons une stratégie de dialogue basée sur la qualité d'échanges et non pas uniquement sur le nombre de tours de parole (M.Q. Nguyen et al., 2007) [NGU07b], alors nous choisissons les paramètres d'évaluation suivants pour la fonction objective

si $(N_{inc}) = 0$ alors

$$C_i = W_{dir}(N_{dir}) + W_{imp}(N_{imp}) + W_{exp}(N_{exp}) + W_{rep}(N_{rep})$$

sinon $C_i = 0$

La variable N_{inc} représente le nombre d'attributs incomplets (i.e., les questions sans réponses). Après un dialogue, si N_{inc} est égal à zéro alors on peut calculer C_i (i.e., mesure qualitative) ; autrement dit, il n'est pas intéressant de le calculer car le dialogue est considéré comme un échec (l'agent machine n'a pas pu obtenir la confirmation d'une ou plusieurs informations). C_i représente le coût total des paramètres de récompense. Les variables W_{dir} , W_{imp} , W_{exp} , W_{rep} représentent les poids qui déterminent le rapport de force parmi ces coûts. Les variables N_{dir} , N_{imp} , N_{exp} , N_{rep} représentent respectivement le nombre de questions directes, implicites, explicites, répétitives (d'ASR¹ et d'NLU²) dans un dialogue. Les rapports N_{imp}/N_{exp} , N_{imp}/N_{rep} , N_{exp}/N_{rep} donnent une mesure d'estimation de la performance du système. Ce sont ces rapports qui nous permettent de dire qu'un dialogue est réussi ou non, donc une mesure qualitative.

Cette mesure qualitative est calculée à partir du nombre de questions-réponses directes, explicites, implicites et répétitives. Nous avons choisi cinq variations de questions que la machine emploie dans son dialogue. Ces variations proviennent de la force illocutoire tirée des théories sur les actes de langage et de dialogue humain-humain d'Austin (1962), de Searle (1969), de Grice (1975) et de Vanderveken (1991) pour les proposer dans le dialogue humain-machine. Ces variations sont :

- Question directe (e.g., « *Quand viendras-tu me voir ?* »)
- Question implicite (e.g., « *Avec qui viendras-tu me voir ce soir ?* »)
- Question explicite (e.g., « *Viendras-tu me voir ce soir ?* »)

¹ En anglais, ASR = Automatic speech recognition.

² En anglais, NLU = Natural language understanding.

- Question répétitive due à un malentendu à cause du bruit (e.g., « *Je n'ai pas entendu, répète ce que tu dis.* »)

- Question répétitive due à une incompréhension (e.g., « *Je ne comprends pas, quand viendras-tu me voir ?* »)

La question directe est utilisée par la machine lorsque l'information est demandée pour la première fois. La question implicite permet d'enchaîner le dialogue tout en vérifiant la réponse qui vient d'être donnée. La question explicite sert pour confirmer immédiatement une réponse avant de passer à la question suivante. La question répétitive due à un malentendu (répétition due à ASR, figure 6.1) permet à la machine de faire répéter l'utilisateur. Cette question est employée lorsque le taux de la reconnaissance est dégradé (i.e., le premier module de reconnaissance de la parole dans un système de dialogue, voir chapitre III). La question répétitive due à une incompréhension (répétition d'NLU) indique à l'utilisateur que ses réponses sont peut-être contradictoires ou alors que la machine n'a pas pu interpréter la sémantique. Les deux questions répétitives apportent une variation et une souplesse dans le dialogue.

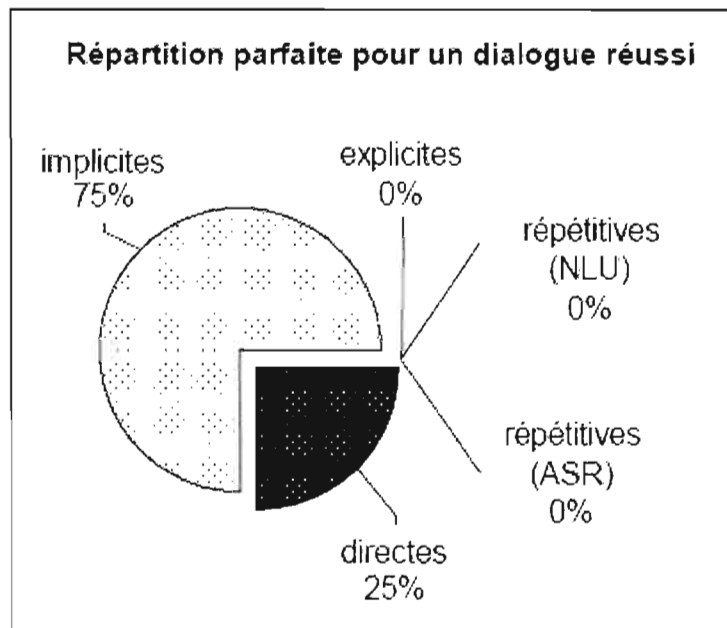


Figure 6.1 Répartition parfaite pour un dialogue réussi selon notre hypothèse.

La mesure qualitative consiste à calculer le pourcentage du nombre de questions. En analysant ces pourcentages, nous pouvons distinguer entre un dialogue réussi et un dialogue non réussi. Une première hypothèse de répartition parfaite pour un dialogue réussi au quelle on pourrait s'attendre de la part d'une machine est présentée dans la figure 6.1.

Dans cette hypothèse, on suppose qu'un dialogue réussi peut ne pas avoir de répétition (aucune question répétitive) ni de question explicite, mais il peut avoir 75% de questions implicites et 25% de questions directes. Un exemple de dialogue réussi selon cette hypothèse est montré dans le tableau 6.1 où nous avons trois questions sur quatre qui sont des questions implicites (75%) et une seule question directe (25%). Nous supposons que les deux interlocuteurs se sont parfaitement « compris ». *A* représente la machine et *U* l'utilisateur. Le but de ce dialogue consiste pour *A* à poser des questions à *U* pour obtenir des informations. Par exemple pour une réservation de chambres d'hôtel, *A* n'a seulement besoin que trois informations (e.g., la date, le nombre de personnes et le type de chambre). Lorsque ces trois informations sont confirmées, le dialogue s'arrête.

1	A	<i>[question directe] Quand voulez-vous réserver ?</i>
	U	<i>je voudrais réserver pour ce weekend</i>
2	A	<i>[question implicite] Pour combien de personnes réservez-vous ce weekend ?</i>
	U	<i>nous sommes deux personnes.</i>
3	A	<i>[question implicite] Quel type de chambre voulez-vous pour deux ?</i>
	U	<i>je voudrais une suite royale</i>
4	A	<i>[question implicite] Pouvez-vous me confirmer que vous voulez réserver une suite royale pour deux ce weekend ?</i>
	U	<i>oui</i>
	A	<i>Félicitations ! Vous avez réservé une suite royale pour deux ce weekend. Votre numéro de réservation est 1914. Merci et à bientôt.</i>
<hr/>		
<i>Total de tours de parole : 4</i>		

Tableau 6.1 Exemple d'un dialogue réussi selon notre hypothèse.

Mais cette répartition 'parfaite' n'est pas réalisable en tout temps car nous devons tenir compte des facteurs humains et machines qui ne sont pas toujours contrôlables dans un dialogue humain-machine (e.g., l'utilisateur change d'avis, la machine a mal interprété le message ou il y a trop de bruits d'ambiance ce qui rend la reconnaissance mauvaise, etc.). Donc, nous proposons une deuxième hypothèse de répartition idéale pour un dialogue réussi dans laquelle on tient compte des facteurs humains et machines. Ainsi, la figure 6.2 propose qu'un dialogue réussi pourrait avoir entre 40 et 60% de questions implicites, 20 et 40 % de questions directes, et laissant le reste 20% réparti entre les questions explicites (10%) et répétitives (5% pour ASR et 5% pour NLU), mais il peut ne pas du tout avoir de questions explicites ou répétitives.

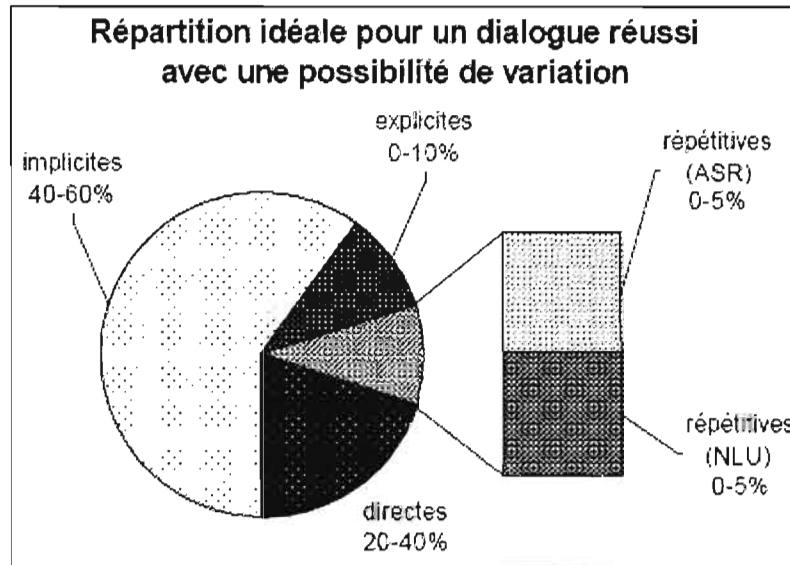


Figure 6.2 Répartition idéale pour un dialogue réussi mais avec une possibilité de variation selon notre hypothèse.

Le tableau 6.2 montre un exemple de dialogue réussi en tenant compte de notre hypothèse. Dans cet exemple, les questions directes (il y en a une seule) et implicites (il y en a trois) représentent respectivement 20% et 60% par rapport au nombre total de tours de parole, tandis que les questions explicites (0%) et répétitives représentent 20% du dialogue. Cette hypothèse montre que nous pouvons nous baser sur les rapports de variations des forces illocutoires pour définir un modèle de dialogue réussi. Tout le travail consiste à trouver une stratégie de dialogue qui conduit au plus près de ce résultat.

VI.3 Stratégie optimale du dialogue

Dans un système de dialogue de type questions-réponses au cours duquel la machine pose des questions et l'utilisateur donne sa réponse, si la machine applique une stratégie qui répond aux critères de mesures proposées, alors cette stratégie sera optimale et elle produira un dialogue réussi. Cependant, sachant que la machine n'a pas le même niveau de connaissance que les humains, ni leur comportement social (i.e., émotion, politesse, culture sociale, etc.), il nous est donc difficile de demander à une machine d'agir comme nous. Néanmoins, nous pouvons lui fournir des règles et

des principes de base pour lui permettre de bien communiquer avec nous. Autrement dit, nous pouvons lui fournir des règles pour qu'elle puisse avoir une stratégie optimale de dialogue avec l'humain.

1	A	[question directe] <i>Quand voulez-vous réserver ?</i>
	U	<i>je voudrais réserver pour ce weekend</i>
2	A	[question implicite] <i>Pour combien de personnes réservez-vous ce weekend ?</i>
	U	<i>on est deux pour ce weekend</i>
3	A	[question implicite] <i>Quel type de chambre voulez-vous pour deux ?</i>
	U	<i>je voudrais une suite royale</i>
4	A	[question répétitive] <i>excusez moi, je n'ai pas compris. Quel type de chambre s.v.p?</i>
	U	<i>une suite</i>
5	A	[question implicite] <i>Voulez-vous réserver une suite pour deux ce weekend ?</i>
	U	<i>oui</i>
	A	<i>Vous avez réservé une suite royale pour deux ce weekend. Votre numéro de réservation est 1914. Merci et à bientôt.</i>
<hr/> <i>Total de tours de parole : 5</i>		

Tableau 6.2 Exemple d'un dialogue réussi en tenant compte des facteurs humains et machines selon notre hypothèse.

VI.3.1 Règles pour une stratégie optimale du dialogue

Les règles pour déterminer une stratégie optimale du dialogue sont basées sur la qualité de la reconnaissance (ASR) et de compréhension de la parole (NLU). La qualité de reconnaissance représente la robustesse du modèle acoustique. La qualité de compréhension représente la capacité d'interprétation du modèle linguistique. Dans un système de dialogue, une stratégie de dialogue est définie en fonction du degré d'initiative et de la stratégie de confirmation (chapitre IV, section 3) ; elle dépend du taux de reconnaissance et de compréhension du système ainsi que du but de l'application finale. Nous proposons cinq valeurs uniques issues des modèles acoustiques et linguistiques sur lesquels se construisent les règles pour déterminer

une stratégie optimale du dialogue (figure 6.3). Ces valeurs allant d'échec (x=0) à excellente (x=4) représentent les taux d'erreurs de la reconnaissance et de la compréhension. Chaque valeur correspond à une action à exécuter par l'agent machine (i.e., question directe, implicite, etc.). Nous n'avons besoin que cinq valeurs pour déterminer les actions à prendre. En effet, le module d'ASR retourne trois valeurs et le module d'NLU interprète la sémantique à partir de ces 3 valeurs, donc, soit il le comprend, soit il ne le comprend pas. La combinaison donne lieu à 5 valeurs issues selon une arborescente hiérarchique illustrée dans la figure 6.3.

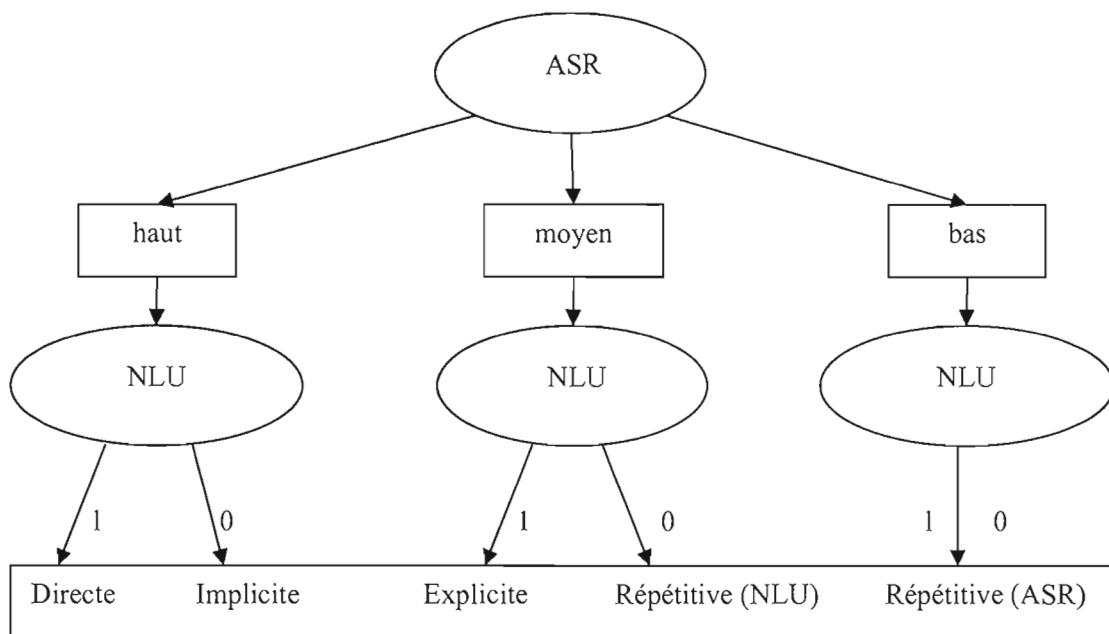


Figure 6.3 Règles de stratégie optimale du dialogue.

VI.4 Approche d'AR

Nous avons défini ce qu'est un dialogue réussi ainsi que des règles pour déterminer une stratégie optimale. Ces règles permettront à la machine de contrôler les interactions orales et de mener à un dialogue réussi. Nous allons expliquer, par la suite, comment ces règles peuvent être modélisées et incorporées dans notre modèle d'apprentissage. En effet, lorsque l'objet a été déterminé (stratégie optimale du dialogue) et que l'on sait quelles sont les règles pour déterminer cet objet alors on veut les insérer dans la machine pour qu'elle puisse trouver l'objet. Pour ce faire,

nous proposons d'utiliser l'approche d'apprentissage par renforcement avec l'aide d'un utilisateur simulé. L'utilisateur simulé détient ces règles. Il apprend à la machine à acquérir l'objet (stratégie optimale) par une technique dite apprentissage par renforcement dont nous allons proposer une modélisation ci-dessous. Un système de dialogue avec une stratégie apprise sera meilleur qu'un système sans stratégie apprise.

VI.4.1 Modèle d'AR avec un utilisateur simulé

Nous choisissons l'approche d'apprentissage par renforcement avec un utilisateur simulé car elle constitue une des meilleures façons d'apprendre dans un environnement incertain et stochastique (chapitre V). L'utilisateur simulé peut reproduire le comportement dialogique des humains de façon dynamique et aléatoire (i.e., imprévisible). En plus, le choix de l'utilisateur simulé vient du fait que l'apprentissage par renforcement demande des échanges d'essais-erreurs pour trouver les stratégies optimales. Ces échanges peuvent être longs et prennent énormément de temps d'exécution. L'emploi des utilisateurs réels n'est pas souhaitable pour ce type d'apprentissage. De plus, l'utilisateur simulé peut être modifié facilement pour ajouter des nouvelles connaissances (e.g., étendre l'application vers d'autres domaines).

Nous allons décrire ci-dessous l'architecture générale de notre approche d'apprentissage, ensuite nous détaillerons le fonctionnement de chaque composant dans cette architecture générale.

VI.4.1.1 Architecture d'AR

Selon les récentes recherches menées depuis ces dix dernières années, la recherche sur la stratégie optimale du dialogue peut être vu comme un problème d'optimisation (Biermann et Long, 1996; Levin et al., 2000; Walker et al., 2000; Pietquin, 2004; Cuayahuilt et Renals, 2006, Schatzmann et al., 2006) (chapitre V). Ce problème peut être résolu en prenant le modèle du processus de décision de Markov (PDM) pour représenter les actes de dialogue. Rappelons que l'apprentissage par renforcement (AR) qui satisfait les propriétés de Markov

s'appelle un processus de décision de Markov (chapitre V) et ce processus permet de résoudre les problèmes d'optimisation. Résoudre des problèmes d'optimisation à l'aide de PDM revient à trouver des stratégies optimales de dialogue.

Notre approche est inspirée de l'architecture décrite dans Levin et al., 2000; Schatzmann et al. 2006. Dans cette architecture (chapitre V, figure 5.2), nous avons deux agents : le professeur³ (l'utilisateur simulé joue le rôle d'enseignant) et l'apprenti (la machine est l'élève).

L'apprentissage d'une stratégie optimale du dialogue se fait à l'aide d'un utilisateur simulé (agent professeur) lequel peut être construit soit manuellement, soit à partir des corpus de données (chapitre V) en se connectant directement à l'agent machine, l'apprentissage peut se réaliser à travers des essais-erreurs. À la fin de l'apprentissage, l'agent machine peut *exploiter* cette stratégie pour rendre le système de dialogue plus souple et naturel donc plus performant et robuste.

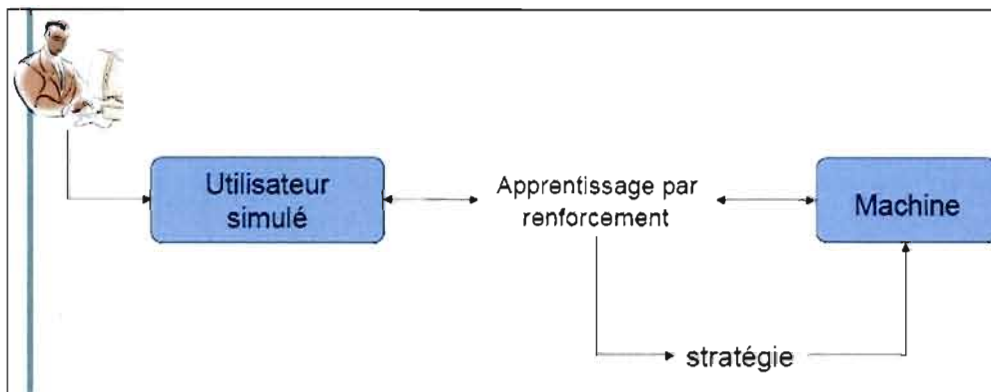


Figure 6.4 Architecture générale d'approche d'apprentissage par renforcement à l'aide d'un utilisateur simulé.

Dans notre modèle d'apprentissage (figure 6.4), nous avons choisi de construire manuellement le comportement dialogique de l'utilisateur simulé (e.g., quoi répondre et comment répondre à une question). Ce choix est motivé par le fait que la construction des corpus de données servant comme prototypes

³ Dans la suite, nous emploierons le terme agent apprenneur ou apprenneur pour désigner l'utilisateur simulé.

d'apprentissage supervisé (chapitre V) pour développer l'utilisateur simulé risque d'être plus longue et plus complexe que de développer manuellement son comportement (i.e., trouver les exemples de dialogue et faire des annotations sémantiques).

Lorsque nous parlons de l'agent machine⁴, nous désignons le système de dialogue au complet (chapitre III). Ce système ne possède actuellement pas de fonction d'apprentissage. Nous allons créer la fonction d'apprentissage dans ce système telle que présentée dans la figure 6.5. L'agent apprenant est en charge de cette fonction et se place au niveau du module de gestion du dialogue lequel est à l'intérieur de l'agent machine (chapitre III). Autrement dit, l'agent apprenant fait partie d'une sous tâche de l'agent machine. L'agent apprenant exécute la fonction d'apprentissage à l'intérieur de l'agent machine. Il reçoit en entrée les résultats des modules et en amont, la reconnaissance et la compréhension de la parole. Comme l'apprentissage de la stratégie optimale dépend de ces deux modules, nous les simulerons à l'aide d'une fonction aléatoire. En fonction des résultats issus de ces modules, l'agent apprenant construira sa stratégie de dialogue. Nous devons aussi traduire le comportement du dialogue de l'utilisateur réel en le simulant à travers cette fonction aléatoire mise dans l'utilisateur simulé. Ces simulations (figure 6.5) permettent plus tard de réaliser un prototype de ce modèle.

Il est à noter que dans cette approche d'apprentissage, les échanges entre l'utilisateur et la machine se réalisent à travers une technique de remplissage des formes ou des attributs-valeurs⁵. Les messages transmis entre l'agent apprenant et apprenant sont de nature textuelle. Le processus d'apprentissage par renforcement avec un utilisateur simulé dans ce modèle fonctionne de façon suivante :

À l'instant t , l'agent machine reçoit une valeur de récompense r_t en réponse à une action a_t exécutée à l'état s_t . L'agent machine transmet cette valeur ainsi que les

⁴ L'agent machine, la machine ou le système désigne le système de dialogue oral humain-machine.

⁵ Termes traduit en anglais *form-filling* ou *slot-value*.

valeurs simulées issues des modules de reconnaissance et de compréhension à l'agent apprenant.

L'agent apprenant met à jour sa table Q-learning (table permettant de sauvegarder les stratégies optimales avec les valeurs de récompense associées aux actions et états) (chapitre V). Ensuite, l'agent apprenant choisit une action (i.e., une question) prédéterminée dans la table Q-Learning, exécute l'action et attend une valeur de retour de l'agent apprenant (valeur de récompense) sur cette action à l'instant $t+1$. Cette valeur de retour indique à l'agent apprenant la réussite ou non de son action exécutée à l'instant t . C'est en faisant ces essais-erreurs que l'apprentissage par renforcement se poursuit jusqu'à ce que le but soit atteint. L'agent apprenant apprend à améliorer son comportement dialogique grâce à l'apprenant (utilisateur simulé).

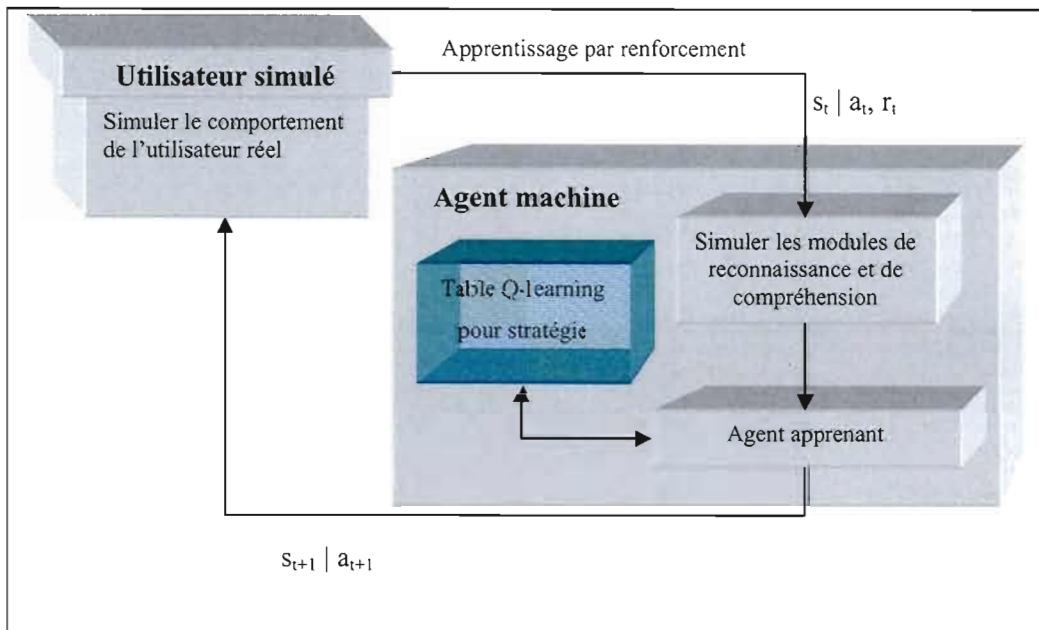


Figure 6.5 Fonctionnement d'apprentissage par renforcement. L'agent apprenant est à l'intérieur de l'agent machine dans un processus d'apprentissage par renforcement.

L'apprenant, quant à lui, simule la compréhension du message venant de l'agent apprenant. Cette simulation permet à l'agent apprenant de reproduire les comportements dialogiques de l'utilisateur réel. Le mécanisme d'apprentissage par

renforcement ne peut pas fonctionner correctement et parfaitement sans déterminer clairement en termes de but, de rôle, de degré d'initiative et de stratégie de confirmation le rôle que joue chaque agent. C'est-à-dire, comment l'un doit se comporter pour apprendre à l'autre et comment l'autre doit apprendre de ses erreurs pour mieux perfectionner son action. Autrement dit, comment l'un transfère sa connaissance à l'autre et comment l'autre assimile cette connaissance.

VI.4.2 Fonctionnement de l'utilisateur simulé

VI.4.2.1 But

L'utilisateur simulé a pour but de remplacer l'utilisateur réel pour assister l'agent apprenant dans son apprentissage. Pour atteindre ce but, l'utilisateur simulé doit avoir un rôle bien déterminé dans cette architecture.

VI.4.2.2 Rôle

Nous avons identifié trois rôles de l'utilisateur simulé (voir chapitre V).

- (1) Assister l'agent apprenant dans l'apprentissage d'une stratégie optimale en lui retournant des indices sur chaque action exécutée par l'agent apprenant. Si l'action est correcte, l'utilisateur simulé retourne une valeur positive (e.g., +1) sinon il retourne une valeur négative (e.g., l'agent apprenant répète successivement plusieurs fois la même question).
- (2) Générer de grande quantité de scénarios de dialogue afin de permettre à la machine d'explorer toutes les stratégies possibles du dialogue. L'utilisateur simulé doit posséder un large vocabulaire pour être capable de générer les diverses réponses.
- (3) Reproduire l'intention de l'humain à travers l'expression orale. L'utilisateur simulé reproduit autant que possible le comportement dialogique de l'humain. Ce comportement dialogique doit être modélisé en termes de degré d'initiative et de stratégie de confirmation pour

répondre au but et au rôle auquel on l'assigne. Par conséquent, le modèle d'utilisateur simulé doit inclure une simulation d'un comportement stochastique, c'est-à-dire, soit un comportement coopéré ou non coopéré (voir la section stratégie de confirmation plus loin dans ce chapitre).

VI.4.2.3 Degré d'initiative

Pour faire l'apprentissage dans un dialogue de type question-réponse, nous avons choisi le mode de fonctionnement de type système-initiative pour dialoguer. Dans ce mode, l'utilisateur simulé ne prend pas d'initiative pour contrôler le dialogue, il ne fait que répondre à la question posée par l'agent apprenant.

VI.4.2.4 Stratégie de confirmation

La stratégie de confirmation employée par l'utilisateur simulé a pour but de mettre à l'épreuve le mécanisme d'apprentissage de l'agent apprenant. Dans le rôle d'enseignant, l'utilisateur simulé se comporte selon deux profils : coopéré et non coopéré.

- (1) **Profil coopéré.** Un profil coopéré indique à l'utilisateur simulé d'adopter un comportement coopéré (i.e., répondre correctement à la question).
- (2) **Profil non coopéré.** Un profil non coopéré génère des réponses incohérentes (i.e., chercher à faire échouer le système, changer d'avis, donner des réponses contradictoires).

Les deux profils ont pour but d'entraîner l'agent apprenant à faire ressortir des cas d'exceptions et de lui permettre d'apprendre les diverses stratégies optimales du dialogue. Cependant, lorsque l'utilisateur simulé adopte un comportement non coopéré, il se peut que l'utilisateur simulé change constamment d'avis, ce qui risque de rendre le dialogue trop long et inutile pour l'apprentissage. Pour éviter ces cas irréalistes, une solution simple consiste à arrêter le dialogue après n tours de

dialogue (i.e., si les informations ne sont pas confirmées après vingt tours de parole, on arrête le dialogue).

Une autre solution consiste à imposer à l'utilisateur simulé d'arrêter le comportement non coopéré après trois ou quatre changements d'avis successifs. Le tableau 6.3 montre un exemple où l'agent apprenant *A* dialogue avec l'utilisateur simulé *U* lequel se comporte de manière non coopérée. L'utilisateur simulé a été « forcé » de changer de comportement après trois tours de parole.

<i>A</i>	<i>Quand voulez-vous réserver?</i>
<i>U</i>	<i>je ne sais pas</i>
<i>A</i>	<i>Répéter s.v.p. Quand voulez-vous réserver?</i>
<i>U</i>	<i>je ne sais pas peut-être vendredi</i>
<i>A</i>	<i>Voulez-vous réserver pour vendredi?</i>
<i>U</i>	<i>Non</i>
<i>A</i>	<i>Voulez-vous réserver pour vendredi?</i>
<i>U</i>	<i>oui</i>

Tableau 6.3 Exemple d'un cas où l'utilisateur simulé cherche à faire échouer le système. La stratégie appliquée dans ce cas consiste à 'obliger' l'utilisateur simulé d'être 'sérieux' après 3 tours de dialogue pour éviter les cas anodins.

VI.4.3 Fonctionnement de l'agent apprenant

VI.4.3.1 But

Le but de l'agent apprenant est d'acquérir une stratégie d'après un certain temps d'apprentissage.

VI.4.3.2 Rôle

Le rôle de l'agent apprenant est de fournir une stratégie optimale à l'agent machine pour qu'il puisse contrôler son interaction avec l'utilisateur simulé. Comme pour l'utilisateur simulé, le comportement de l'agent apprenant se manifeste à travers son interaction avec l'utilisateur simulé, et est contrôlé par le degré d'initiative et la stratégie de confirmation.

VI.4.3.3 Degré d'initiative

Comme nous avons choisi un mode système-initiative pour faire l'apprentissage, l'agent machine (l'agent apprenant est à l'intérieur de l'agent machine) commence le dialogue par une question directe, l'utilisateur simulé donne sa réponse accompagnée d'une valeur de récompense et le dialogue continue jusqu'à ce qu'il n'y aura plus de questions à poser et que le but soit atteint.

VI.4.3.4 Stratégie de confirmation

La stratégie de confirmation de l'agent apprenant est guidée par la valeur de récompense retournée par l'utilisateur simulé (voir section sur le rôle de l'utilisateur simulé dans ce chapitre). Ces stratégies sont inscrites dans la table Q-Learning (i.e., action-état et récompense) (figure 6.5). Pour décider quelle stratégie employée, il suffit pour l'agent apprenant d'aller lire la table Q-Learning. À chaque exécution, la table Q-learning sera mise à jour avec une valeur de récompense indiquée par l'utilisateur simulé.

VI.5 Conclusion

Nous avons proposé un modèle d'apprentissage par renforcement des stratégies optimales pour réaliser des applications de dialogue oral humain-machine. Cette stratégie est basée sur nos critères de mesure de la qualité du dialogue. Notre stratégie de dialogue est prudente et progressive car elle emploie les actions (i.e., questions posées par la machine) en fonction de la performance et de la robustesse des modules de reconnaissance et d'interprétation de la parole. Cette performance est simulée par cinq valeurs discrètes. Cette stratégie empêche les malentendus et prévient les erreurs durant le dialogue. Puisque l'environnement dans lequel un système de dialogue oral fonctionne peut être différent au même titre que le comportement des utilisateurs est imprévisible, un système de dialogue doit constamment apprendre à s'adapter avec son nouvel environnement. C'est pour cela que nous avons proposé notre modèle d'apprentissage par renforcement pour améliorer la performance du système de dialogue. L'intérêt de l'apprentissage est de permettre une plus grande adaptation du modèle dans d'autres domaines. Nous

allons décrire dans le chapitre suivant la réalisation informatique de ce modèle ainsi que les expériences réalisées avec notre approche. Nous allons démontrer que notre modèle fonctionne et qu'il est computable.

CHAPITRE VII

MODÉLISATION INFORMATIQUE

VII.1 Introduction

Le chapitre VI a proposé un modèle d'apprentissage pour un système de dialogue. Nous allons montrer dans ce chapitre que l'approche proposée est computable. D'abord, nous traduirons les actes de dialogue dans le processus de décision de Markov (PDM). Ensuite, nous proposerons un algorithme d'interactions entre deux agents pour que l'un puisse transférer la connaissance à l'autre. Notre explication sur le modèle du PDM prend exemple dans le domaine de réservation des chambres d'hôtel, mais l'approche n'est pas spécifiquement conçue pour ce domaine, elle peut être appliquée dans n'importe quel domaine construit sur la base d'un système de type questions-réponses et orienté-but.

VII.2 Processus de décision de Markov dans le dialogue

Utiliser le modèle de Processus de Décision de Markov (PDM) pour formaliser les actes de dialogue demande les assumptions suivantes [PIE04]:

- la transition entre les états se fait par un processus stochastique probabiliste et l'état actuel ne dépend uniquement que de l'état et de l'action précédente.
- Il y a un coût associé aux actions du dialogue pour chaque état de dialogue et ce coût est distribué selon la formule $P_c(C_t | S_t, A_t)$, c'est-à-dire que le coût est une probabilité en fonction de l'action A dans l'état S à l'instant t.

Pour utiliser le modèle PDM dans le dialogue, nous devons spécifier un domaine d'application. Nous allons l'appliquer dans le domaine de réservation des

chambres d'hôtel. Le but n'est pas de construire un système complet de dialogue oral humain-machine pour la réservation des chambres d'hôtel mais de montrer que le PDM peut être appliqué pour modéliser les actes de dialogue. Bien que nous empruntons un domaine spécifique pour décrire les actes et actions de PDM, nous allons construire les états et les actions de la manière la plus abstraite possible afin de pouvoir les adapter à n'importe quel domaine.

VII.2.1 États

Notre système est défini comme étant orienté vers un but, donc l'enchaînement d'un état à un autre est conçu comme une sorte d'automate déterministe¹. Lorsque le but est atteint, on dit que son état est terminal ; autrement, son état est non terminal.

Les états de notre système de dialogue sont censés être représentatifs de toutes les connaissances du système (les ressources internes et externes qui interagissent avec le système) [PIE04]. Dans notre cas d'étude, les ressources internes sont les entrées-sorties de l'agent apprenant (qui est à l'intérieur de la machine ou l'agent machine) et la ressource externe est l'utilisateur simulé. Les états du système incluent trois attributs : date de réservation, le nombre de personnes pour qui les chambres sont réservées (adulte ou enfant) et le type de chambre (vue sur mer, fumeur, non fumeur, chambre double, suite, suite royale, etc.). Pour chaque attribut, nous associons quatre variables pour distinguer les possibilités d'états. Ces variables sont :

1- La valeur de chaque attribut (non connue, connue, confirmée),

2-La qualité de reconnaissance et de compréhension issue de la simulation (définie par la stratégie de confirmation du modèle de l'utilisateur simulé, tableau 6.3),

3- Le nombre de répétitions d'une question (0 à n),

¹ Finite State Machine. Disponible sur : http://en.wikipedia.org/wiki/Finite_state_machine [consulté le 15 jan 2004]

4- Type de question posée par l'agent apprenant (question implicite, explicite, répétitive ASR, répétitive NLU et directe).

Pour contrôler toutes les actions du dialogue, notre système de dialogue représente les états du dialogue sous forme de vecteurs opérationnels. Ces vecteurs sont des valeurs numériques portant le statut de chaque variable. On peut voir ces valeurs numériques comme une sorte de 'photo' prise pendant une séquence de dialogue.

VARIABLES D'ÉTAT	VALEURS	EXPLICATIONS
Valeur (V_0)	0,1,2	Indique la valeur du premier attribut (e.g., date) sur lequel le système travaille (0= inconnue, 1 = connue, 2 = confirmée).
Valeur (V_1)	0,1,2	Indique la valeur du deuxième attribut sur lequel le système travaille (e.g., nombre de personnes).
Valeur (V_2)	0,1,2	Indique la valeur du troisième attribut sur lequel le système travaille (e.g., type de chambre).
Qualité de ASR/NLU (Q)	0, 1, 2,3,4	Cette valeur permet au système de choisir une action (selon le tableau 6.2).
Répétition ASR (R_{asr})	0, 1	0 = aucune répétition, 1= une fois
Répétition NLU (R_{nlu})	0, 1	0 = aucune répétition, 1= une fois
Actions (Ac)	0, 1, 2, 3,4	0 = question directe, 1 = question explicite, 2 = question implicite, 3 = question répétitive ASR, 4 = question répétitive NLU

Tableau 7.1 Les paramètres des états de dialogue et leurs valeurs.

Le tableau 7.1 montre les différentes valeurs pouvant être affectées à un état. Si on veut que la machine ait plus d'informations à obtenir de l'utilisateur, on peut ajouter d'autres attributs V_n dans les variables d'états (par exemple, V_3 indique le choix d'une voiture de location et V_4 indique le type de la voiture). Un simple calcul montre que pour cinq actions avec trois attributs ou variables (i.e., date de réservation, nombre de personnes et type de chambre), on obtient $5^3 = 125$ possibilités d'états. De plus, les possibilités de combinaison entre les valeurs pour produire une stratégie optimale sont au maximum $(125)^3 = 1953125$. Cela devient vite inimaginable pour un système qui ne peut gérer ces innombrables possibilités. Il

faut optimiser manuellement les stratégies « évidentes » pour faciliter l'apprentissage. Par exemple, au début du dialogue, il ne peut pas y avoir une action de vérification implicite, ni de confirmation totale ou de fermeture de dialogue à moins que le taux de reconnaissance soit très mauvais, mais même dans cette condition, le système peut prendre l'initiative de mettre fin à un dialogue. Par conséquent, il faut définir rigoureusement les variables d'états et choisir les actions appropriées au début de la conception des états de Markov pour essayer de réduire l'espace d'états de dialogue. Le système évitera ainsi d'introduire inutilement les cas anodins dans l'apprentissage. Cependant, notre algorithme adapté de Q-learning permet de créer au fur et à mesure que le dialogue progresse entre les deux agents. Nous n'avons pas besoin de travailler sur les cas anodins, mais nous allons utiliser l'utilisateur simulé pour faire le travail à notre place. Cela illustre l'intérêt de l'utilisation de l'utilisateur simulé.

VII.2.2 Actions

Les actions du système de dialogue représentent les possibilités des réponses et des questions que le système peut fournir à l'utilisateur. Nous définissons sept types d'actions du système :

- 1- Ouverture de dialogue ou invité. (i.e. "*Bonjour, bienvenue à notre système. Que puisse-je pour vous ?*").
- 2- Question directe. Cette action correspond à la stratégie système-initiative. L'utilisateur répond à une question spécifique pour remplir une variable, par exemple, la question « *Quand voulez-vous venir ?* » est pour connaître la date de départ.
- 3- Clarifier une information par une vérification
 - a. Implicite. Cette action est appelée lorsque la reconnaissance est haute (e.g., « *Quel type de chambre voulez-vous pour ce samedi ?* »)

- b. Explicite. Cette action est appelée lorsque la reconnaissance est moyenne (e.g., « *Avez-vous dit samedi ?* »)
 - c. Répétitive. Cette action est appelée lorsque la reconnaissance est dégradée (e.g., « *Désolé, je n'ai pas bien **entendu**. Pouvez-vous répéter s.v.p. ?* »)
 - d. Répétitive. Cette action est appelée lorsque l'interprétation est imprécise (e.g., « *Désolé, je n'ai pas **compris**. Pouvez-vous répéter s.v.p. ?* »)
- 4- Confirmation totale. Cette action est appelée lorsque toutes les variables sont remplies (i.e., « *Vous avez demandé pour 3 personnes pour lundi dans 1 suite royale ?* »)
- 5- Fermeture du dialogue (i.e., « *Votre réservation est confirmée. Merci et au revoir.* »)

La première action (ouverture du dialogue) et la dernière action (fermeture du dialogue) sont des actions fixes, c'est-à-dire que quelle que soit la réponse de l'utilisateur, ces actions sont toujours exécutées au début et à la fin du dialogue. Par conséquent, ces actions ne sont pas prises en compte dans les paramètres du processus de décision de Markov. Cela diminue encore les cas irréalistes (i.e., l'agent machine ne choisira pas une action d'ouverture du dialogue au milieu de la conversation). Finalement, nous n'avons seulement que cinq actions à considérer pour le PDM (i.e., question directe, implicite, explicite, répétitive ASR et NLU).

VII.3 Interaction entre l'utilisateur simulé et la machine

L'interaction entre l'utilisateur simulé et l'agent machine doit être réalisée de façon synchronisée pour que le processus d'apprentissage puisse se dérouler correctement afin d'obtenir des résultats satisfaisants. Elle doit aussi reproduire le comportement dialogique des humains. Pour modéliser cette interaction en termes

informatiques, nous allons la présenter sous forme de diagrammes de séquences au niveau de la machine et de l'utilisateur simulé.

VII.3.1 Agent machine

Le diagramme de séquences d'interactions illustré dans la figure 7.1 indique que l'agent machine commence le dialogue. L'agent machine appelle l'agent apprenant en initialisant le premier état de Markov avec une action aléatoire associée à cet état. Cette action n'est pas optimale (puisqu'elle est choisie de façon aléatoire au départ), l'agent apprenant va chercher le *bon* code d'action associé à cet état dans le fichier Q-Learning (fonction *Chercher_action*). Le fichier Q-Learning contient tous les états de Markov et les codes d'actions associées à ces états. Lorsque la fonction *Chercher_action* retourne un code d'action, l'agent apprenant va chercher le message textuel correspondant à ce code d'action (i.e., si le code d'action est 0, le message correspondant est une question directe, par exemple « *Quand voulez-vous réserver une chambre ?* »). Ce texte sera transmis à l'agent apprenant par l'utilisateur simulé (fonction *Envoie_1^{ière}_question*). C'est une action qui est exécutée par la machine (i.e., la machine *parle* à l'utilisateur). Lorsque l'utilisateur simulé répond à cette question (point 2), la première tâche à faire pour l'agent machine est de mettre à jour le fichier Q-Learning. Ce fichier contient toutes les actions qui viennent d'être exécutées par l'agent apprenant (fonction *Mise_à_jour_fichier_Q-Learning*). Ensuite, l'agent machine simule la qualité de reconnaissance du module de reconnaissance et de compréhension de la parole (fonction *Simuler_ASR_NLU*); cette simulation retourne une valeur. Cette valeur constitue un nouvel état de Markov. L'agent machine appelle à nouveau l'agent apprenant avec un nouvel état de Markov associé avec une nouvelle action (fonction *Apprenant*). L'agent apprenant va chercher le *bon* code d'action (fonction *Chercher_action*), puis chercher le message textuel (fonction *Chercher_question*) et exécute l'action (fonction *Envoie_question*). Et les échanges se répètent jusqu'à ce qu'il n'y aura plus de questions à poser (le but est atteint).

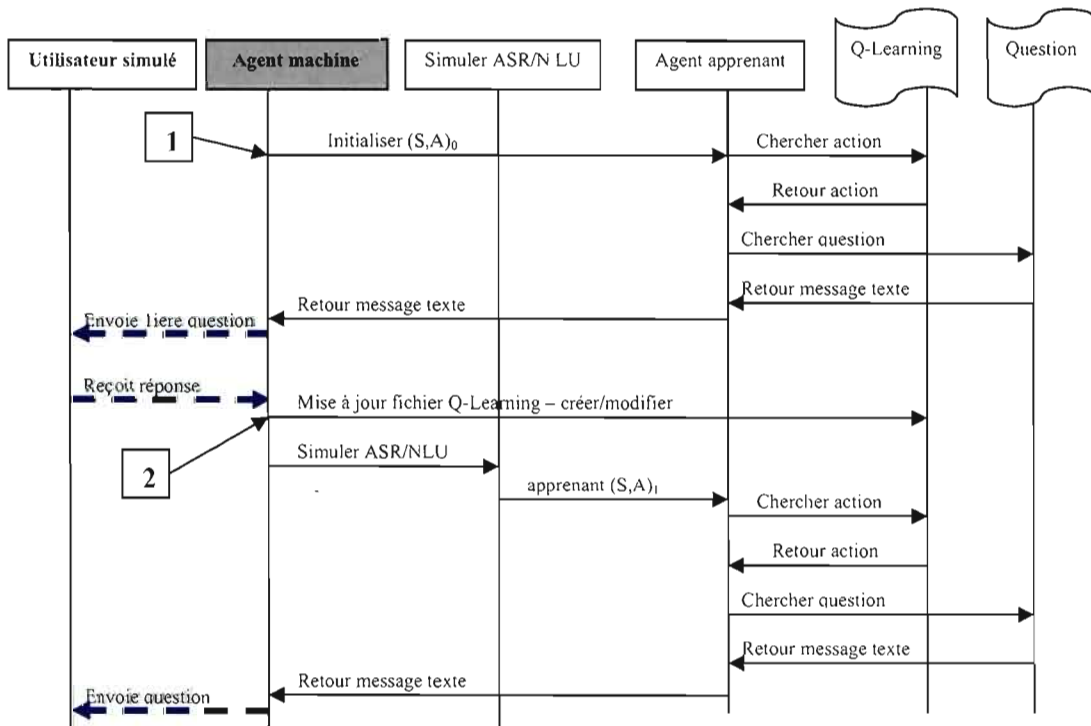


Figure 7.1 Diagramme des séquences d'interaction du côté de l'agent machine.

VII.3.1.1 Algorithme d'apprentissage

La tâche la plus importante pour l'agent apprenant est la mise à jour du fichier Q-Learning, lequel peut être vu comme la mémoire ou la connaissance de l'agent apprenant. C'est grâce à ce fichier que l'agent apprenant sait comment dialoguer avec l'utilisateur simulé. Pendant le dialogue, l'agent machine peut acquérir de nouvelles stratégies de dialogue (grâce à l'agent apprenant) et il n'est pas nécessaire d'attendre la fin de dialogue pour mettre à jour le fichier Q-Learning (fichier qui contient la connaissance en stratégie du dialogue). C'est l'avantage de la méthode Q-Learning par rapport à la méthode Sarsa [TAK05] (voir chapitre V.4.1). Le tableau 7.2 montre l'algorithme d'apprentissage du côté de l'agent apprenant. Cet algorithme n'est autre que la recherche et mise à jour d'une suite de caractères uniques dans un fichier. Cette suite de caractères représente des états-actions de Markov (voir les

définitions informatiques du fichier `q_learning.dat` au chapitre VIII.4). Nous avons illustré une partie de l'algorithme en prenant la syntaxe du langage Perl².

```

$flag == 0;
open (inf, "<q_learning.data") || die "erreur ouverture en lecture : $!";
while (<inf>)
{
    $sline_lu = $_;
    # recherche l'état de Markov dans le fichier Q-Learning et extraire
    l'action recommandée par l'utilisateur simulé si trouvé
    if ( $sline_lu eq $etat_courant) {
        print "Action choisie";
        $flag=1;
        Break;
    }
}
if ($flag == 0) {
    Mise_a_jour_fichier_Q_learning ($nouveau_etat, $action);
    print "nouvel état et action, ajouter dans Q-Learning";
}
close(inf);

```

Tableau 7.2 Exemple de l'algorithme d'apprentissage de l'agent apprenant qui met à jour la stratégie apprise.

Lorsque l'agent apprenant a choisi une action a dans un état s (mémorisé par la variable `$etat_courant`), cet état sera comparé avec chaque état déjà mémorisé dans le fichier `q_learning.dat`. Le script parcourt le fichier `q_learning.dat` du début à la fin jusqu'à ce qu'il trouve cet état. S'il ne le trouve pas, il l'ajoutera à la fin du fichier `q_learning.dat` avec une action choisie par hasard. Le coup suivant il se fera corrigé par l'utilisateur simulé.

VII.3.2 Utilisateur simulé

Nous avons volontairement conçu et placé les règles de stratégie optimale du dialogue au niveau de l'utilisateur simulé et non pas au niveau de l'agent apprenant ni agent machine, car le processus d'apprentissage reste le même pour l'agent apprenant quel que soit le domaine, seules les règles qui déterminent les stratégies du dialogue peuvent être différentes d'un domaine à l'autre. Donc, en mettant les règles

² Internet : <http://www.perl.com>

de stratégies du côté de l'utilisateur simulé, l'agent machine sera plus indépendant par rapport au domaine d'application. Si on veut modifier des règles de stratégie, on modifiera l'utilisateur simulé sans toucher à l'agent machine.

Du côté de l'utilisateur simulé (figure 7.2), lorsqu'il reçoit une question de l'agent machine, sa première tâche consiste à vérifier si l'action qui vient d'être exécutée par l'agent machine est correcte ou non (fonction *Vérifier_stratégie*). C'est l'utilisateur simulé qui détient les règles de stratégie optimale. Si cette action est correcte, un code de récompense +1 est retourné à l'utilisateur simulé, sinon -1 sera retourné. Ce code est renvoyé à l'agent machine qui s'en sert pour corriger son action. C'est le processus d'apprentissage par renforcement qui est mise en œuvre à cette étape. L'agent apprenneur simule le comportement dialogique des utilisateurs réels en choisissant un profil coopéré ou non coopéré (chapitre VI, section sur la stratégie de confirmation de l'utilisateur simulé) (fonction *Choisir_un_profil*) avant de répondre à l'agent machine (fonction *Chercher_reponse* qui renvoie un message textuel par lequel l'utilisateur simulé exprimera sa réponse). La réponse peut être cohérente ou non selon le profil qui a été choisi. Et le dialogue continu jusqu'à ce que l'agent machine ait atteint son but, c'est-à-dire, qu'il ait obtenu toutes les réponses à ses questions.

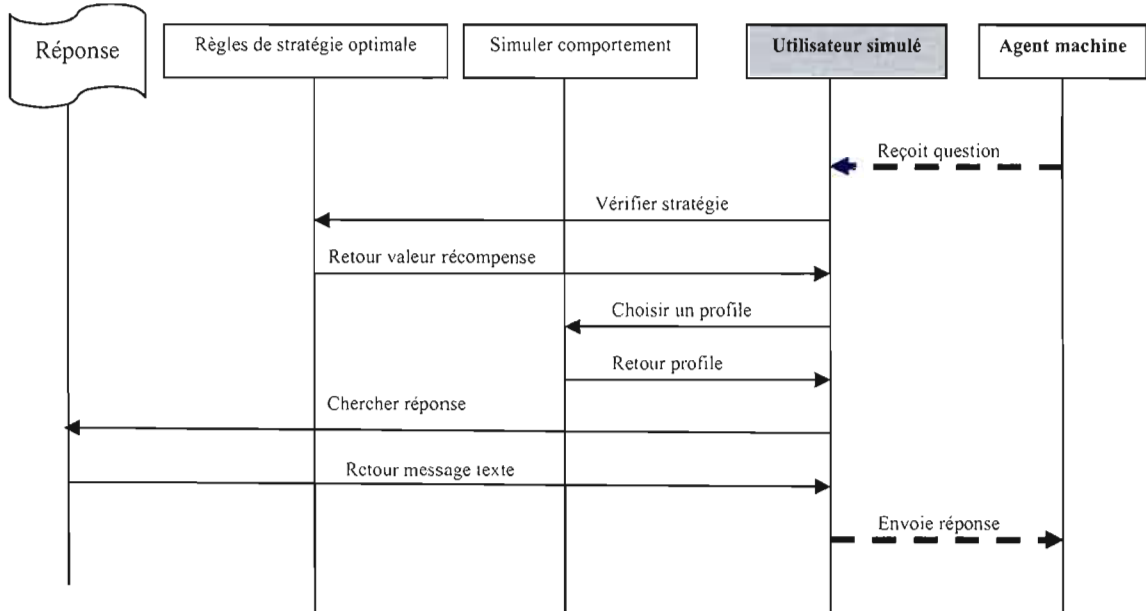


Figure 7.2 Diagramme de séquences du côté de l'utilisateur simulé.

VII.3.2.1 Algorithme d'attribution d'une valeur de récompense

Nous montrons ci-dessous (tableau 7.3) un aperçu de l'algorithme d'attribution d'une valeur de récompense, suite à l'action de l'agent apprenant. La valeur de récompense retournée par l'utilisateur simulé permettra à l'agent apprenant de corriger son action et de choisir une autre action jusqu'à ce qu'elle soit meilleure ou optimale. Cet algorithme traduit les règles de stratégie optimale du dialogue présentée dans le chapitre VI, figure 6.3. Si le module de simulation (chapitre VI, figure 6.5) renvoie une valeur excellente alors la meilleure action proposée sera une action de type question directe ou lorsqu'une action est exécutée plus de deux fois successivement, la question explicite sera un meilleur choix et ainsi de suite.


```

if ( $qualite == EXCELLENTE ) { $best_action = DIRECTE;}
if ( $qualite == TRESBIEN ) { $best_action = IMPLICITE;}
if ( $qualite == BIEN ) { $best_action = EXPLICITE;}
if ( $qualite == PASSABLE ) {
    $best_action = REPETE_NLU;
    if ($rep_asr >2) {
        # trop de repetition, on change de type de question
        $best_action = EXPLICITE;
    }
}
if ($best_action != $action_agt_apprenant )
{
    print "Mauvaise action\n";
    $recompense= -1;
} else {
    print "Bonne action\n";
    $recompense= 1;
}
return ($recompense);

```

Tableau 7.3 Exemple de l'algorithme d'attribution de valeur de récompense de l'utilisateur simulé.

VII.4 Conclusion

Nous avons, dans ce chapitre, montré comment appliquer les propriétés de Markov (les actions et les états) dans le dialogue. Ces propriétés sont indispensables pour mettre en œuvre notre modèle d'apprentissage par renforcement. Nous avons aussi proposé un algorithme d'interaction entre deux agents pour permettre un apprentissage efficace dans lequel l'utilisateur simulé est important car il détient les règles de stratégie optimale et on peut le modifier sans perturber le fonctionnement normal du système de dialogue. L'apprentissage avec un utilisateur simulé peut être fait à n'importe quel moment pendant l'exploitation de l'application ; il suffit de connecter l'utilisateur simulé à l'agent machine pour en faire l'apprentissage.

Remarquons que l'on peut déplacer l'algorithme de recherche de stratégie optimale de l'utilisateur simulé vers l'agent machine. En faisant ainsi, le problème d'apprentissage n'aura plus de raisons d'exister. Et de cette façon, on n'aura pas besoin d'utilisateur simulé. Le tout se réalise avec un modèle traditionnel du

développement d'un système de dialogue. On déploie tous les efforts pour réaliser un système de dialogue qui répond à un besoin spécifique dans un domaine spécifique. L'agent apprenant, dans ce cas, exécutera la stratégie optimale à coup sûr pour chaque état rencontré. Mais, procéder ainsi revient à travailler avec une approche traditionnelle, c'est-à-dire l'expert du dialogue est primordial et il n'y a pas de possibilité d'expansion dans d'autres domaines sans tout avoir à recommencer. Avec notre approche, l'agent apprenant sera développé une seule fois et les nouvelles règles de stratégies seront prises en charge par l'utilisateur simulé qui n'a pas besoin d'être actif en production, ce qui a un avantage énorme pour les applications qui devraient fonctionner sans interruption.

CHAPITRE VIII

EXPÉRIMENTATION ET RÉSULTATS

VIII.1 Introduction

Dans ce chapitre, nous décrivons l'expérience et les résultats obtenus avec notre prototype d'apprentissage développé selon l'approche proposée. Le but de ces expériences est de démontrer la faisabilité de l'approche (apprendre une stratégie optimale du dialogue à la machine par une technique de renforcement avec l'aide d'un utilisateur simulé). L'expérience consiste à réaliser un prototype d'apprentissage appliqué dans le domaine de la réservation des chambres d'hôtel.

Ce chapitre est organisé en deux parties. La première partie (sections VIII.2 et VIII.3) décrit d'abord les tâches du domaine de la réservation des chambres d'hôtel, ensuite, par une analyse organique des solutions informatiques dont nous décrivons l'environnement de développement, l'emplacement des fichiers de données et les fichiers exécutables ainsi que leur signification. La deuxième partie (section VIII.4) expliquera le déroulement de l'expérience et l'analyse des résultats. Nous avons réalisé deux prototypes : un sans apprentissage et un avec apprentissage.

VIII.2 Tâches de réservation.

De façon générale, les informations qu'un client peut fournir à un représentant (préposé) de l'hôtel lorsqu'il veut louer une chambre d'hôtel sont : la date de réservation, la date de départ ou le nombre de jours restés, le nombre de personnes, le type de chambre à réserver (i.e., chambre double, suite, suite royale, etc.), le prix des chambres, etc. Le client et le préposé s'engagent dans un dialogue de type

question-réponse. Le dialogue cessera lorsque les deux interlocuteurs seront d'accord sur une entente (leur but est atteint) ou lorsque l'un d'entre eux décidera d'abandonner le dialogue sans arriver à une entente. L'exemple du tableau 8.1 montre un dialogue entre un préposé d'hôtel *P* et un client *C*. Dans ce dialogue très simple, les deux personnes arrivent à s'échanger des informations sans ambiguïté. On peut dire que ce dialogue est réussi. Bien entendu, les informations échangées dans ce dialogue ne sont pas complètes pour le domaine de la réservation. Le préposé doit vérifier si les chambres sont disponibles au prix recherché par le client, et ensuite, il demandera certainement le nom du client, son numéro de téléphone, un dépôt, etc. Mais, l'exemple est suffisant pour illustrer le type d'interaction qui pourrait s'établir entre l'humain et la machine. Dans ce type d'interaction, on parle de type question-réponse ou de demande d'information.

<i>P</i>	<i>Bonjour, que puis-je faire pour vous?</i>
<i>U</i>	<i>Je voudrais réserver une chambre pour ce weekend.</i>
<i>P</i>	<i>Combien de personnes êtes-vous?</i>
<i>U</i>	<i>Nous sommes 3 adultes.</i>
<i>P</i>	<i>Quel type de chambre voulez-vous?</i>
<i>U</i>	<i>Nous aimerons une chambre double.</i>
<i>P</i>	<i>D'accord, pouvez-vous me confirmer que vous voulez une</i>
<i>U</i>	<i>chambre pour 3 personnes ce weekend?</i>
<i>P</i>	<i>Oui, s.v.p.</i>

 [le dialogue continu avec d'autres informations]

<i>U</i>	<i>Votre réservation est 234. Merci et à bientôt.</i>

Tableau 8.1 Exemple d'un dialogue entre deux personnes.

Nous n'avons pas l'intention de reproduire dans cette expérience la totalité des scénarios de dialogue portant sur le domaine de la réservation, mais nous prenons la partie du scénario du dialogue portant sur ces trois informations : la date de réservation (i.e., les jours de la semaine : lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche et weekend), le nombre de personnes (i.e., un chiffre quelconque) et le type de chambre (i.e., suite, suite royale, chambre double). Nous ne vérifions

pas les règles d'affaires dans ce prototype, par exemple, le client peut réserver une chambre double pour trois cent personnes. Bien entendu, ces informations sont trop peu nombreuses pour couvrir tous les cas pratiques dans ce domaine, mais elles sont suffisantes pour illustrer l'approche d'apprentissage par renforcement.

VIII.3 Analyse organique des solutions informatiques

VIII.3.1 Définitions

Il nous est important de préciser le sens des termes employés pour la suite de ce chapitre. Ces définitions sont plus proches de l'informatique que de la linguistique.

Un tour de parole constitue une question de l'agent machine et une réponse de l'utilisateur simulé.

Une séquence de dialogue ou **une session de dialogue** ou simplement **dialogue** est une suite de plusieurs tours de parole. Une session de dialogue est le résultat d'une exécution du programme.

Un programme est un code exécutable que nous avons écrit en langage Perl et C++/.Net. Notre programme met en œuvre les théories proposées (prototype).

Un prototype est un modèle théorique qui sera validé par un programme. Un prototype n'est pas un *exécutable*. C'est le programme lui-même qui l'est.

Une exécution est l'action d'exécuter un programme. Le résultat d'une exécution réside en un fichier qui contient une séquence de dialogue (une session de dialogue). Dans ce fichier, nous avons toutes les données pour analyser le comportement dialogique de l'agent machine et pour mesurer la qualité du dialogue (mesure quantitative et qualitative).

Un épisode constitue plusieurs exécutions successives du même programme. Dans notre cas, un épisode équivaut à 1000 exécutions. L'apprentissage consiste à exécuter un ou plusieurs épisodes.

VIII.3.2 Organisation de système de fichiers

VIII.3.2.1 Environnement de développement

Nous avons développé notre prototype sous Windows/XP et sous Unix. L'environnement Windows sert à présenter le prototype et lancer les exécutables (i.e., fenêtre windows, écrans d'affichage). L'environnement Unix permet de calculer les statistiques, de produire les données pour les graphiques et tableaux (i.e., les commandes shell/Unix sont particulièrement efficaces et robustes pour produire des résultats analytiques). Nous avons utilisé les langages Perl pour la programmation et C++/Visual Studio pour réaliser les interfaces. Ces langages ont été choisis pour leur efficacité, leur puissance et leur facilité en termes de manipulation des objets (tableaux, fichiers, variables et structures complexes).

VIII.3.2.2 Développement

Nous avons trois catégories de fichiers : fichiers exécutables, fichiers en entrées, fichiers en sorties ou fichiers résultats.

VIII.3.2.3 Fichiers exécutables

Les fichiers exécutables `dialog_simule.bat` et `dialog_simule1000.bat` permettent de lancer le programme. Le premier exécute le programme une fois alors que le second l'exécute 1000 fois, ce qui correspond à un épisode d'apprentissage. Ces fichiers (.bat) appellent les programmes Perl `agt_main.pl`, `agt_apprenant.pm` et `agt_appreneur.pm`.

Les fichiers exécutables sont groupés dans une fenêtre (figures 10.1 et 10.2 dans l'annexe) que nous appelons écran d'interface de simulation. Cet écran permet de lancer les programmes et de visualiser le résultat d'exécution.

VIII.3.2.4 Fichiers en entrées

Ce sont des fichiers de données en entrée du programme. Le programme ne peut pas s'exécuter correctement sans ces fichiers. Nous avons cinq types de fichiers :

- 1) ***Q-Learning.data***. Ce fichier est indispensable pour l'apprentissage ; il est considéré comme la mémoire de l'agent apprenant puisque ce fichier contient tous les états, les actions et les valeurs de récompense (tableau 8.2). Les états de Markov sont traduits en termes informatiques de la manière suivante (chapitre VII, section 2. PDM dans le dialogue) :

- $E = (date, nbp, typc, qualité, rep_asr, rep_nlu, action, récompense)$ où les champs *date*, *nbp* et *typc* peuvent recevoir 3 valeurs (inconnue=0, connue=1 et confirmée=2).

date	nbp	typc	qualité	rep asr	rep nlu	action	recomp
0	0	0	0	0	0	0	-1
2	1	0	1	0	0	2	1

Tableau 8.2 Exemple du contenu du fichier de données Q-Learning.data.

Le champ *qualité* peut avoir 5 valeurs (excellente=4, très bien=3, bien=2, passable=1, échec=0) et les 2 derniers champs *rep_asr* et *rep_nlu* peuvent avoir 2 valeurs (question non répétée=0, question répétée une fois=1). Basé sur les variables définies dans le tableau 7.1, un état possible parmi les états dans le processus de décision de Markov peut être représenté sous forme vectorielle comme ceci : $S_i = (V_0, V_1, V_2, Q, R_{asr}, R_{nlu}, Ac)$, par exemple, le fichier Q-learning.data peut contenir des états de Markov comme indique le tableau 8.2.

- La première ligne du tableau 8.2 indique l'état initial du dialogue. La date de réservation, le nombre de personnes et le type de chambre ne sont pas connus (*typc*=0), la qualité de reconnaissance est très mauvaise (*qualité*=0). Il n'y a pas eu de questions répétitives lors de cet état (*rep_asr* et *rep_nlu* sont à 0). L'action définie dans cet état est une action portant sur une question directe (*action*=0). La dernière colonne indique que la

valeur de récompense a été initialisée à -1. Cela signifie que soit l'agent apprenant n'ai pas passé par cet état, soit l'utilisateur simulé a retourné une valeur négative suite à l'action exécutée par l'agent apprenant. Cette action n'était pas optimale. Au prochain passage par cet état, l'agent apprenant devra choisir une autre action. La deuxième ligne signifie que le dialogue est dans un état où le champ *date* a été confirmé (*date*=2), le champ *nombre de personnes* a été donné mais est non confirmé (*nbp*=1), le champ *type de chambre* n'est pas connu (*typc*=0), la qualité de reconnaissance est passable (*qualilté*=3) et il n'y a pas eu de questions répétitives à cause de la mauvaise qualité de reconnaissance et de compréhension (*rep_asr*=0, *rap_nlu*=0). L'avant dernière colonne indique qu'une question explicite (*action*=2) sera exécutée quand le dialogue se trouvera dans cet état car la valeur de récompense est à 1.

- 2) ***DialogDataHotel_question.txt***. Ce fichier contient les questions relatives au domaine de réservation des chambres d'hôtel. Ce fichier ainsi que les 3 suivants servent comme annotation sémantique. L'agent machine utilise ce fichier pour poser des questions à l'utilisateur simulé. Le tableau 8.3 montre un exemple du contenu du fichier *DialogDataHotel_question.txt*. La première colonne indique le champ *date* (*date* a la valeur 0, *nombre de personnes*, la valeur 1 et *type de chambre*, la valeur 2). La deuxième colonne indique le type de l'action, par exemple, 0 signifie que la question « *Quand voulez-vous réserver?* » est une question directe. Les variables <X1>, <X2> et <X3> représentent la date, le nombre de personnes et le type de chambre à réserver. Ces variables permettent de simuler les réponses aléatoires de l'utilisateur simulé (e.g., « *Pour combien de personnes réservez-vous ce <lundi>?* » ou « *Quel type de chambres voulez-vous pour <2> personnes?* »).

champ	type	questions
0	0	Quand voulez-vous réserver ?
0	1	Pour combien de personnes réservez-vous ce <X1>?
0	2	Avez-vous demandé pour ce <X2>?
0	3	s.v.p. je n'ai pas bien entendu, quand voulez-vous réserver ?
0	4	s.v.p. je n'ai pas bien compris, quand voulez-vous réserver ?
2	1	Quel type de chambre voulez-vous pour <X2> personnes ?
2	2	Voulez-vous réserver une chambre <X3>?

Tableau 8.3 Exemple du contenu du fichier de question pour le domaine de réservation des chambres d'hôtel.

- 3) *DialogDataHotel_Reponse-1.txt*. Ce fichier contient les réponses relatives au domaine de réservation des chambres d'hôtel. L'agent appreneur (utilisateur simulé), qui adopte un comportement coopéré, accède à ce fichier. De la même manière, le fichier *DialogDataHotel_Reponse-1.txt* contient les réponses de l'utilisateur simulé (tableau 8.4). Nous avons séparé les fichiers de questions et réponses dans le but de ne pas surcharger les informations dans un même fichier. L'utilisateur simulé a trois possibilités de réponses, par exemple, à la question « *Quand voulez-vous réserver?* » (champ=0, type=0), l'utilisateur simulé peut répondre « *je voudrais réserver pour ce <lundi>* » (réponse R1) ou simplement « *ce <lundi>* » (réponse R3) en remplaçant X1 par lundi. Une fonction aléatoire se chargera de choisir une réponse (R1, R2 ou R3) pour l'utilisateur simulé.

champ	type	R1	R2	R3
0	0	je voudrais réserver pour ce <X1>	Ce <X1> me convient très bien	ce <X1>
0	1	oui, on est <X2> pour ce <X1>	non, on est <X2> pour ce <X1>	rien
0	2	non, je veux pour <X1>	oui	non
0	3	ok, je veux réserver pour ce <X1>	j'ai dit pour ce <X1>	pour ce <X1>
0	4	Il n'y a rien à comprendre, je veux réserver pour ce <X1>	j'ai dit pour ce <X1>	pour ce <X1>

Tableau 8.4 Exemple du contenu du fichier de réponse pour le domaine de réservation des chambres d'hôtel.

- 4) *DialogDataHotel_Reponse-2.txt*. Ce fichier contient les réponses relatives au domaine de réservation des chambres d'hôtel. L'agent appreneur (utilisateur simulé), qui adopte un comportement, non-coopéré accède à ce fichier.
- 5) *Reponses_specifique.txt*. Ce fichier contient des termes spécifiques au domaine de la réservation des chambres d'hôtel. Ce fichier sert à simuler les réponses aléatoires de l'utilisateur simulé (i.e., lundi, mardi, chambre double, suite royale, etc.)

Les fichiers de données (2 à 5) sont définis de manière à ce que l'on puisse séparer clairement les codes du programme Perl et les données propres au domaine de la réservation. En faisant ainsi, nous assurons une meilleure adaptation à d'autres domaines sans avoir besoin de réécrire tous les codes. Il suffit de modifier les quatre fichiers .textes pour créer un nouveau domaine. Par conséquent, il nous sera plus facile d'ajouter ou de modifier l'application en cours et, ou de changer les domaines d'application. Par exemple, une application permet d'entrer une adresse civique (i.e., ville, numéro, rue) au système de navigation pour qu'il puisse trouver le chemin à partir d'un point local. Pour cela, il suffit de changer le contenu des fichiers d'entrées (section VIII.4.1.2) et de garder la même stratégie d'apprentissage (i.e., le même fichier q-learning.dat).

VIII.3.2.5 Fichiers en sorties

Le résultat de l'exécution du programme `dialog_simule.bat` est écrit dans le fichier `dialog_output.html` lequel sera affiché automatiquement à l'écran (figures 10.1 et 10.2 en annexe). Le programme `dialog_simule1000.bat` permet d'exécuter un épisode donc 1000 exécutions successives. Le résultat de cette commande est un ensemble de 1000 fichiers `f1.txt`, `f2.txt`, ..., `f1000.txt` créés sous le répertoire `<résultat>`. Chaque fichier contient la séquence de dialogue et les valeurs statistiques sur ce dialogue (e.g., nombre de tours de parole, nombre d'informations incomplètes, etc.).

VIII.4 Première expérience

VIII.4.1 But

La première expérience vise à démontrer que le PDM est computable. On construit un prototype qui met en place un environnement d'apprentissage (i.e., créer des appels de fonctions sans le cœur de la fonction), créer des fonctions de base telles que simuler le comportement de l'utilisateur simulé en lui attribuant un comportement coopéré et non coopéré (chapitre VI sur la stratégie de confirmation de l'utilisateur simulé), simuler la qualité de reconnaissance par des valeurs aléatoires (5 valeurs de 0 à 4), manipuler le fichier `Q-Learning.data` (mise à jour les actions, ajouter un nouveau état de Markov, etc.), synchroniser les tours de parole, contrôler les états du dialogue pour savoir s'il faut arrêter ou continuer le dialogue. En effet, toutes les fonctions d'enveloppe permettant de faire tourner correctement le mécanisme d'interaction entre deux agents sont développées dans ce prototype.

La première version est une version sans apprentissage. Cela permet de voir la progression entre avant (première expérience) et après l'apprentissage (deuxième expérience).

VIII.4.2 Fonctionnement

Le prototype est construit selon le principe système-initiative basé sur les questions-réponses. Autrement dit, l'agent machine commence le dialogue en posant

l'une des trois questions soit sur la date de réservation, soit sur le nombre de personnes ou sur le type de chambres à réserver. Lorsque ces trois informations seront confirmées par l'utilisateur simulé, le programme s'arrêtera, sinon, il continuerait à poser infiniment des questions. Puisqu'il n'y a pas de fonction d'apprentissage dans cette version, les critères de mesures quantitatifs et qualitatifs ne seront pas rencontrés alors le programme risquerait de ne jamais s'arrêter. Pour éviter ce cas nuisible, nous avons imposé une limite arbitraire au nombre de tours de parole soit quarante tours de parole. Ce nombre est grand pour un dialogue avec seulement trois informations à confirmer. Cependant, cela permet à l'agent machine, dans une seule session de dialogue, d'explorer un maximum de possibilités d'actions (e.g., poser des questions directes ou explicites) afin d'obtenir la confirmation de réponses de l'utilisateur.

VIII.4.3 Déroulement de l'expérience

Nous avons exécuté le programme lors d'un épisode (i.e., 1000 exécutions) sans avoir activé la fonction d'apprentissage et en simulant un comportement totalement coopéré de l'utilisateur simulé (i.e., il donne des réponses cohérentes) puis un autre épisode avec un comportement totalement non-coopéré (i.e., il cherche des réponses contradictoires). Rappelons que notre premier critère quantitatif (chapitre VI, section 2.1) sert pour éliminer des dialogues non réussis (i.e., le nombre de tours de parole doit être minimal par rapport au nombre d'informations demandées). Nous avons implanté ce comportement au niveau de l'utilisateur simulé que nous appellerons : comportement fixe et comportement mixte (aléatoire). Tout au long de cette première expérience, le simulateur d'erreurs d'ASR/NLU génère aléatoirement un chiffre entre 0 et 4 représentant le taux d'erreurs d'ASR/NLU. Le chiffre zéro correspond à une qualité de reconnaissance et de compréhension excellente alors que le chiffre quatre signifie une très mauvaise qualité.

VIII.4.3.1 Comportement fixe

Un comportement fixe consiste à adopter une attitude totalement coopérée ou totalement non coopérée tout au long du dialogue de la part de l'utilisateur simulé. Avec ce comportement fixe, nous avons analysé le rapport entre le nombre de tours de parole et le nombre d'exécutions sous forme de courbes graphiques. Nous avons remarqué que les courbes sont en forme de zigzag, une irrégularité dans les résultats. Cela est expliqué par le fait qu'il n'y a pas de fonction d'apprentissage, donc les actions de l'agent machine sont totalement aléatoires. Par conséquent, le résultat obtenu avec 1000 exécutions n'est pertinent seulement que dans les 120 premières exécutions. La figure 8.1 montre que lorsque l'utilisateur simulé est coopéré (ligne solide), il y a plus de dialogues dont le nombre de tours de parole oscillent entre 10 et 25 tours que lorsque l'utilisateur simulé est non coopéré (ligne pointillé). De plus, même sans aucune stratégie apprise, l'agent machine a réussi à générer certains dialogues proche du comportement dialogique de l'utilisateur réel sans, toutes fois, atteindre les critères d'un dialogue réussi.

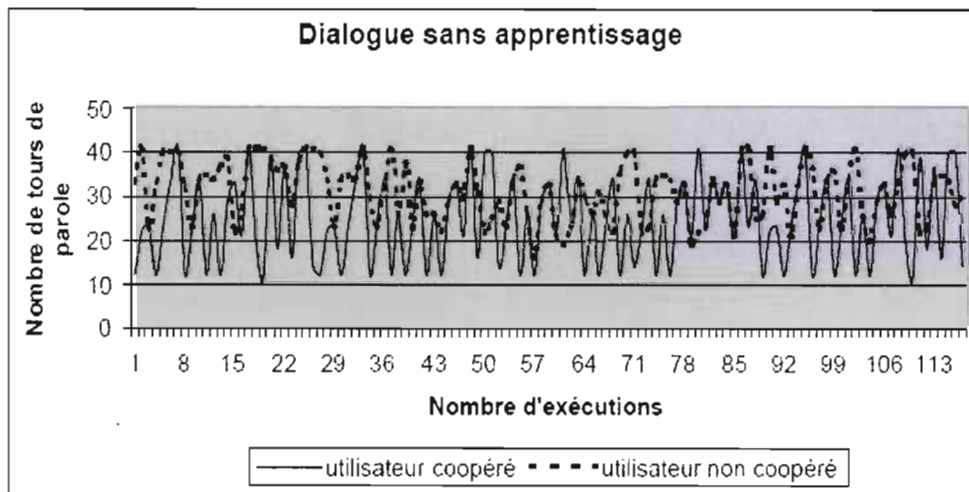


Figure 8.1 Nombre de tours de parole par exécution pour un dialogue sans aucun apprentissage avec un utilisateur simulé fixe (coopéré et non-coopéré).

VIII.4.3.2 Comportement mixte

Un comportement mixte ou aléatoire consiste, pour l'utilisateur simulé, à adopter une attitude tantôt coopéré, tantôt non coopéré tout au long du dialogue. Nous avons lancé 1000 exécutions (un épisode) et relevé le nombre de dialogues qui satisfont seulement le critère quantitatif, ce dialogue est presque '*réussi*'. Pour des raisons de commodité, nous avons présenté le résultat dans la figure 8.2 en l'appelant dialogue réussi pour dire que ce dialogue a réussi à obtenir la confirmation de toutes les informations obligatoires en respectant le seuil maximal de tours de parole. Dans ce cas, nous avons autorisé jusqu'à 22 tours de parole au lieu de 9 tours comme indique le critère de mesure (selon l'équation $1.6 * N_s \leq N_{dt} \leq 3.0 * N_s$ décrit dans la section VI.2). Ce chiffre est voulu, car en faisant ainsi, on permet aux deux agents de dialoguer plus et donc d'explorer au maximum l'espace d'états. Nous avons voulu vérifier si cela était possible pour un agent machine, ayant une stratégie aléatoire du dialogue (les questions que l'agent machine pose à l'utilisateur simulé sont choisies par hasard), de dialoguer correctement avec l'utilisateur simulé. Et dans ces conditions, combien y a-t-il de '*réussites*' partielles avant l'apprentissage ? Nous avons constaté qu'il y a 33% de dialogues partiellement '*réussis*' sur 1000 exécutions (tous les champs sont confirmés). Les dialogues non réussis représentent près de 67% (les dialogues dont il reste 1, 2 ou 3 champs qui n'ont pas pu être confirmés avant 22 tours de parole). L'expérience montre que, sans une stratégie optimale de dialogue, la machine peut, quand même réussir à avoir un comportement dialogique presque optimal, bien que ce comportement optimal ne soit pas garanti en tout temps et cela même dans des situations où la machine a déjà *exploré* ces situations.

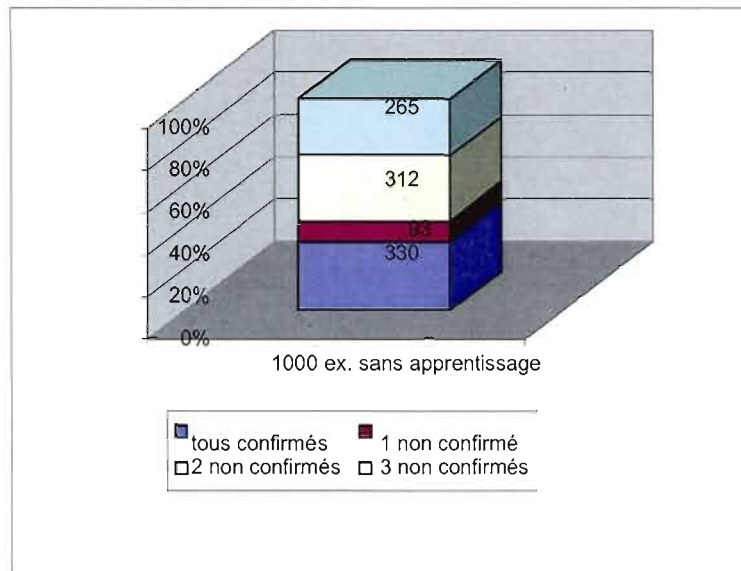


Figure 8.2 Nombre de dialogues réussis totalement ou partiellement sans apprentissage.

VIII.4.3.3 Résultat avec un utilisateur simulé coopéré

Nous montrerons, ci-dessous, un dialogue presque réussi en seulement 7 tours de parole. En effet, l'agent machine (A) a réussi à obtenir la confirmation de toutes les valeurs données par l'utilisateur simulé (U). Cette réussite est attribuée à la fonction aléatoire du module simulateur de qualité de reconnaissance (le tableau 8.5 indique quatre reconnaissances de bonne qualité (excellente et très bien) contre trois de mauvaise qualité (passable et échec)), ainsi l'action choisie par hasard par l'agent apprenant, correspond à ce que l'utilisateur simulé attendait. La figure 8.3 montre une situation 'chanceuse' où l'agent machine n'a pas de connaissance en stratégie optimale du dialogue, mais il a réussi quand même à avoir un comportement presque optimal (selon notre hypothèse proposée, chapitre VI, figure 6.2). L'agent machine a employé 57% de questions implicites (nous avons proposé entre 40 et 60%) et seulement 14% de questions directes (nous avons proposé entre 20 et 40%) et aucune question explicite (nous avons proposé entre 0 et 10%). Cependant, en examinant de près ce résultat, on s'aperçoit que l'agent machine aurait dû choisir une meilleure action au troisième tour au lieu d'employer le même genre de questions

que lors du tour précédent (i.e., « *Pour combien de personnes réservez-vous ce weekend ?* »).

Recherche de stratégie optimale de dialogue par apprentissage par renforcement
Version : 1.4 (sans d'algorithme d'apprentissage - dialogue sans aucune stratégie optimale, utilisateur coopéré)
Auteur : @Minh-Quang Nguyen, 2003-2007, Doctorat en informatique cognitive, UQAM/INRS/Nuance Communications Inc.

1-(A): Quand voulez-vous réserver?

(U): je voudrais réserver pour ce lundi

2-(A): Pour combien de personnes réservez-vous ce lundi?

(U): on est 2 pour ce weekend

3-(A): Pour combien de personnes réservez-vous ce weekend?

(U): on est 3 pour ce mercredi

4-(A): s.v.p. je n'ai pas bien compris, pour combien de personnes voulez-vous réserver?

(U): il n'y a rien à comprendre, on est 1

5-(A): Quel type de chambre voulez-vous pour 1 personne?

(U): je voudrais une suite royale

6-(A): s.v.p. je n'ai pas bien compris, quel type de chambre voulez-vous réserver?

(U): il n'y a rien à comprendre, je veux une suite

7-(A): Voulez-vous prendre une suite pour 1 personne pour ce mercredi?

(U): oui

(A): Vous avez réservé une (suite) pour (1) personne(s) ce (mercredi)

(A): Votre numéro de réservation est 306. Merci et à bientôt.

Total de tours de parole : 7

Nombre de champs non complétés (DATE,NBP,TYPC)(2 2 2) : 0

Nombre de questions posées :

directes : 1

implicites : 4

explicites :

répétitives (asr) :

répétitives (nlu) : 2

Nombre de Qualité de ASR et NLU:

EXCELLENTE : 3

TRÈS BIEN	: 1
BIEN	:
PASSABLE	: 2
ECHEC	: 1

Tableau 8.5 Dialogue presque réussi entre un utilisateur simulé coopéré et un agent apprenant.

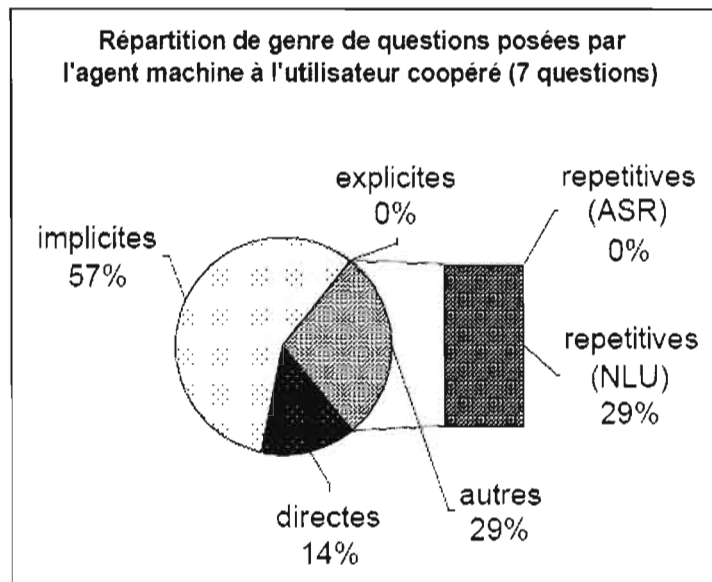


Figure 8.3 Répartition de genre de questions posées par l'agent machine à l'utilisateur coopéré.

VIII.4.4 Conclusion de la première expérience

À l'issue de cette première expérience, nous avons mis en place notre modèle informatique et avons mis en place l'environnement d'interaction dialogique entre les deux agents. Cet environnement permet, par la suite, d'activer la fonction d'apprentissage. L'approche proposée (i.e., un utilisateur simulé, processus de décision de Markov) est computable.

Les interactions entre deux agents fonctionnent même sans aucun algorithme d'apprentissage car nous avons constaté que même sans aucune connaissance en matière de stratégie de dialogue, l'agent machine a obtenu 33% de dialogue

partiellement réussi tout en employant une technique aléatoire. Il suffit maintenant d'introduire notre modèle d'apprentissage dans ce prototype. L'agent machine, après un certain temps d'apprentissage, aura un comportement dialogique moins aléatoire, plus stable et donc tendra vers un dialogue réussi. C'est ce que nous allons démontrer dans la deuxième expérience.

VIII.5 Deuxième expérience

VIII.5.1 But

Dans cette deuxième expérience, nous allons démontrer qu'un système de dialogue avec une stratégie apprise sera meilleure et, en particulier, avec notre architecture d'apprentissage par renforcement à l'aide d'un utilisateur simulé qui est capable de simuler le comportement dialogique de l'humain. Lorsque nous parlons de dialogue réussi dans cette deuxième expérience, cela signifie que ce dialogue répond aux critères de mesure d'un dialogue de qualité.

VIII.5.2 Fonctionnement

Pour mettre en place, correctement, notre deuxième version du prototype d'apprentissage et pour s'assurer que ce nouveau prototype créera suffisamment d'états de Markov, nous avons généré un ensemble d'états de Markov avant d'en faire l'apprentissage. Pendant l'apprentissage, l'utilisateur simulé est libre de choisir son comportement car nous voulons reproduire le phénomène dynamique et stochastique du comportement dialogique entre les humains. Nous ne l'avons pas forcé à adopter un comportement particulier (comportement fixe ou mixte) tout au long des épisodes car cela risquerait de biaiser les résultats. Un comportement trop typique de l'utilisateur simulé, tout au long d'un dialogue contribuera à la réussite du dialogue mais ne représentera pas un comportement commun à la plupart des dialogues entre humains. Néanmoins, pour bien faire l'apprentissage, dans la phase d'entraînement, nous avons exécuté le programme avec un utilisateur non coopéré, puis avec un utilisateur coopéré. Ensuite, pour la phase de test, nous avons attribué un comportement mixte (tantôt coopéré, tantôt non) à l'utilisateur simulé. Cela permet aussi d'obtenir des données différentes pour la phase d'entraînement et pour

la phase de test. Parce que le comportement de l'utilisateur change, il générera des réponses différentes.

VIII.5.3 Déroulement de l'expérience

Pour bien faire l'apprentissage et pour s'assurer que cette deuxième expérience apporte des résultats significatifs, nous avons procédé de la manière suivante :

- 1- Création des états de Markov,
- 2- Pour la phase d'entraînement ou phase d'apprentissage, on exécute le programme avec un utilisateur simulé non-coopéré puis avec un utilisateur simulé coopéré. Et pour chaque cas, on entraîne le programme sur au moins 9 épisodes, car nous avons constaté qu'après 9 épisodes, l'apprentissage se stabilise. Enfin, répéter l'opération en simulant le taux d'erreurs d'ASR/NLU de 0 à 4 niveaux (de très mauvais à excellent),
- 3- Pour la phase de test. On répète la même opération qu'en (2) mais seulement pour un utilisateur simulé ayant adopté un comportement mixte, c'est-à-dire tantôt coopéré, tantôt non coopéré.

VIII.5.3.1 Création des états de Markov

À l'aide d'une fonction aléatoire qui génère les couples états-actions uniques de Markov tels que définis $S_i = (V_0, V_1, V_2, Q, R_{asr}, R_{nlu}, Ac)$ (section VIII.4.2.1), nous avons pu créer, avant de commencer l'apprentissage, 705 états-actions uniques contre 269 états-actions dans la première version. Ce nombre est suffisamment grand pour permettre à l'agent machine *d'exploiter* les différentes situations dialogiques que peuvent se produire au cours de la conversation entre lui et l'utilisateur simulé.

VIII.5.3.2 Apprentissage

Le tableau 8.6 montre les résultats d'apprentissage après 9 épisodes. Nous avons constaté que lorsque le taux d'erreurs d'ASR/NLU est supérieur à 60%, il est pratiquement impossible d'avoir un dialogue réussi. Cependant, lorsque le taux

d'erreur se situe entre 40 et 60%, nous pouvons quand même réussir à obtenir 60 dialogues réussis. Il y a, donc, un net changement entre 40-60%, et 60% et plus. La figure 8.4 illustre le tableau 8.6. Elle témoigne de ce changement avec une pente très raide. Nous avons remarqué, aussi, que lorsque l'utilisateur est non-coopéré, il est difficile pour un système, d'avoir un dialogue réussi car dans la plupart des cas, c'est la mesure quantitative qui fait échouer le dialogue (e.g., le nombre de tours est trop long dû au caractère non coopéré de l'utilisateur).

Taux d'erreurs d'ASR/NLU	Dialogues réussis	
	Utilisateur simulé non	Utilisateur simulé coopéré
0-20%	61	345
20-40%	10	187
40-60%	4	60
60-80%	0	3
80-100%	0	2

Tableau 8.6 Résultats d'entraînement pour 9 épisodes.

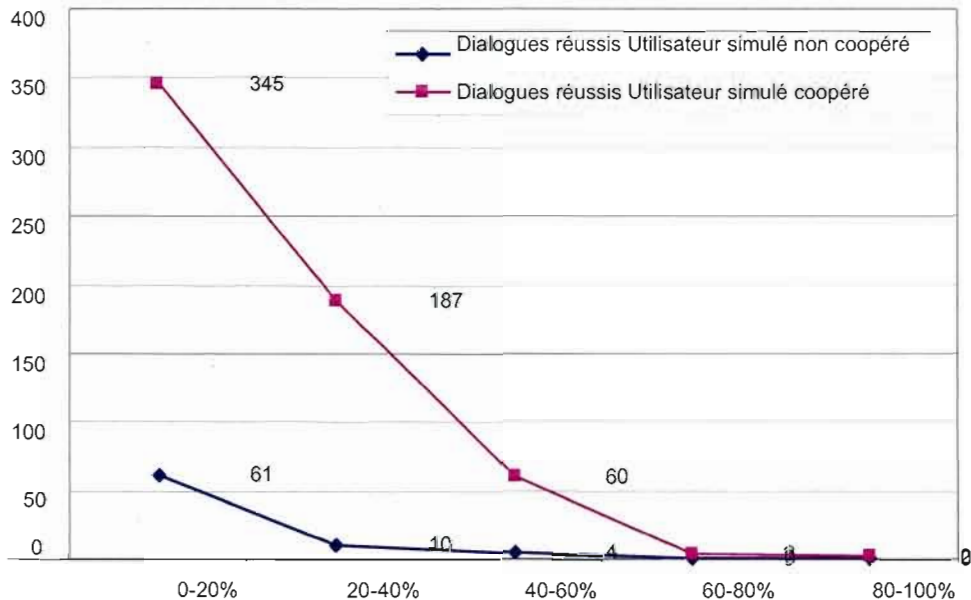


Figure 8.4 Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU selon le tableau 8.6.

La figure 8.5 montre le résultat après 9 épisodes d'entraînement avec un utilisateur simulé coopéré selon différents niveaux d'erreurs d'ASR/NLU. La figure

8.6 montre le résultat mais, cette fois-ci, avec un utilisateur simulé non-coopéré. On constate qu'il y a seulement que 61 dialogues réussis.

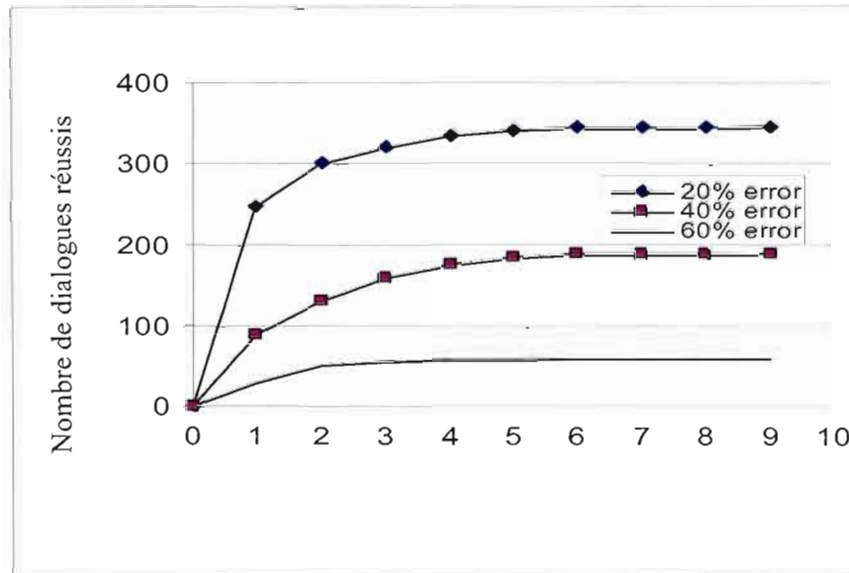


Figure 8.5 Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU avec un utilisateur simulé coopéré.

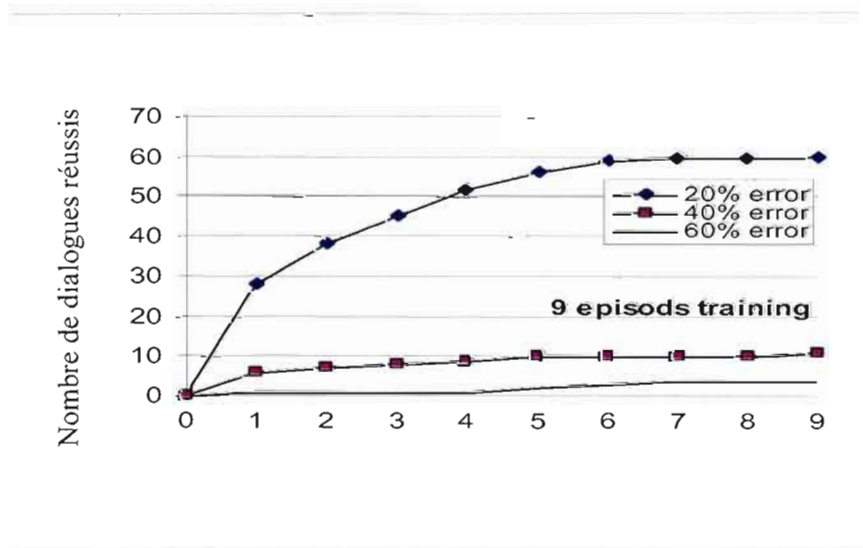


Figure 8.6 Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU avec un utilisateur simulé non-coopéré.

Enfin pour la phase de test, la figure 8.7 nous montre la courbe après neuf épisodes, il y a seulement 193 dialogues réussis. Cela peut être expliqué par le fait que l'utilisateur a un comportement mixte et ce comportement est simulé par une fonction aléatoire (cette fonction génère un nombre soit 0, soit 1 à chaque tour de parole, voir diagramme de séquences au chapitre VII, figure 7.2). De plus, ce résultat est inférieur au résultat obtenu lors de la phase d'entraînement avec un utilisateur coopéré (345 dialogues réussis avec un taux d'erreurs d'ASR/NLU de 0-20%), mais supérieur au résultat obtenu pour la même phase avec un utilisateur non-coopéré (61 dialogues réussis). Cela indique qu'il y a une certaine cohérence dans nos résultats. Le comportement mixte est un comportement 'moins' coopéré que le comportement coopéré, donc logiquement, il produit moins de dialogues réussis.

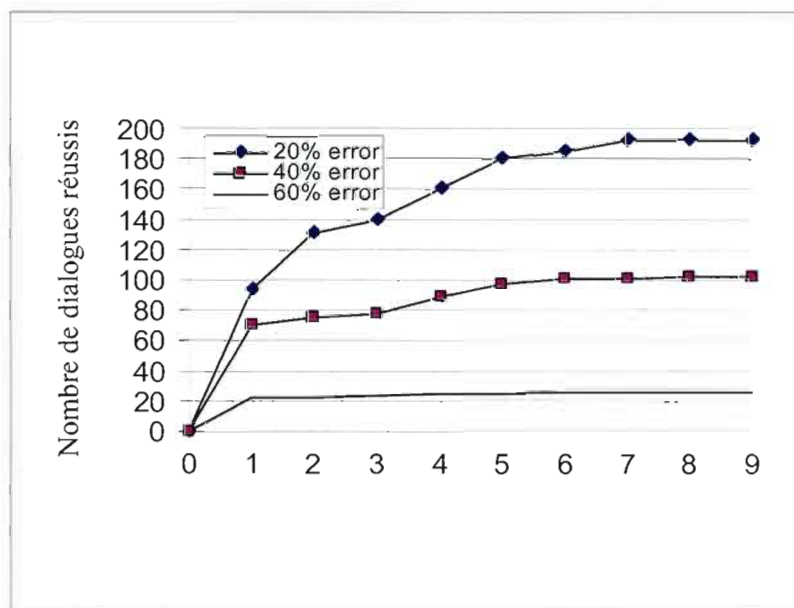


Figure 8.7 Nombre de dialogues réussis en fonction du taux d'erreurs d'ASR/NLU avec un utilisateur simulé avec un comportement mixte.

VIII.5.3.3 Résultat d'un dialogue réussi après apprentissage

La figure 8.8 montre la répartition du nombre de questions pour 7 tours de parole. Dans ce dialogue, les questions implicites représentent 42%, les questions explicites 29%, les questions directes 29% et aucune question répétitive. Notre

hypothèse suggère entre 40-60% pour les questions implicites, 10% pour les questions explicites et 20-40% pour les questions directes et 10% pour des questions répétitives (chapitre VI, figure 6.2) quelle que soit la qualité de reconnaissance. La différence entre ce résultat et celui de notre hypothèse sur un dialogue réussi se trouve dans le nombre de questions explicites et répétitives (29% au lieu de 20%, figure 6.2). Cette différence de 9% peut être expliquée en examinant de près les enchainements de dialogue généré (tableau 8.7). Nous avons remarqué que le module qui simule le taux de reconnaissance de la parole et d'interprétation nous retourne des valeurs bonnes ou excellentes. Notre stratégie a opté pour un choix prudent parce que la stratégie employée a pour but de prévenir et d'empêcher les malentendus et les erreurs de compréhension au cours de la conversation (notre objectif de recherche, chapitre I). Lorsque le taux de reconnaissance n'est pas excellent, il faut mieux ne pas prendre le risque de trop poser de questions implicites pour ne pas induire les utilisateurs en erreur et ne pas créer des confusions chez eux. Cela risquerait de diminuer leur satisfaction et d'augmenter le nombre d'échecs de dialogue.

Recherche de stratégie optimale de dialogue par apprentissage par renforcement
Version : 3.0 (ajout d'algorithme d'apprentissage pour trouver la stratégie optimale du dialogue)

Auteur : @Minh-Quang Nguyen, 2003-2007, Doctorat en informatique cognitive, UQAM/INRS/Nuance Communications Inc.

Simulation d'un dialogue entre humain-machine dans un domaine de réservation des chambres d'hôtel

La machine possède un agent apprenant lui permettant d'apprendre une stratégie optimale du dialogue.

(A): *Quand voulez-vous réserver?*

(U): *je voudrais réserver pour ce jeudi*

(A): *Avez-vous demandé pour ce jeudi?*

(U): *oui*

(A): *Pour combien de personnes réservez-vous ce jeudi?*

(U): *non, on est 3 pour ce dimanche*

<p>(A): <i>Voulez-vous réserver pour 3 personnes?</i> (U): <i>oui</i></p> <p>(A): <i>Quel type de chambre voulez-vous pour 3 personnes?</i> (U): <i>si possible je voudrais une chambre normale</i></p> <p>(A): <i>Quel type de chambre voulez-vous réserver?</i> (U): <i>une suite</i></p> <p>(A): <i>Voulez-vous réserver une suite pour 3 personnes pour ce dimanche?</i> (U): <i>oui</i></p> <p>(A): <i>Vous avez réservé une (suite) pour (3) personne(s) ce (dimanche)</i> (A): <i>Votre numéro de réservation est 1057. Merci et à bientôt.</i></p> <hr/> <p>Total de tours de parole : 7 Nombre de champs non complétés (DATE,NBP,TYPC)(2 2 2) : 0 Nombre de questions posées : directes : 2 implicites : 3 explicites : 2 répétitives (asr) : répétitives (nlu) :</p> <p>Qualité de ASR et NLU: EXCELLENTE: 5 TRESBIEN : 1 BIEN : 1 PASSABLE : ECHEC :</p>
--

Tableau 8.7 Exemple d'un dialogue réussi en 7 tours de parole après 5 épisodes d'apprentissage.

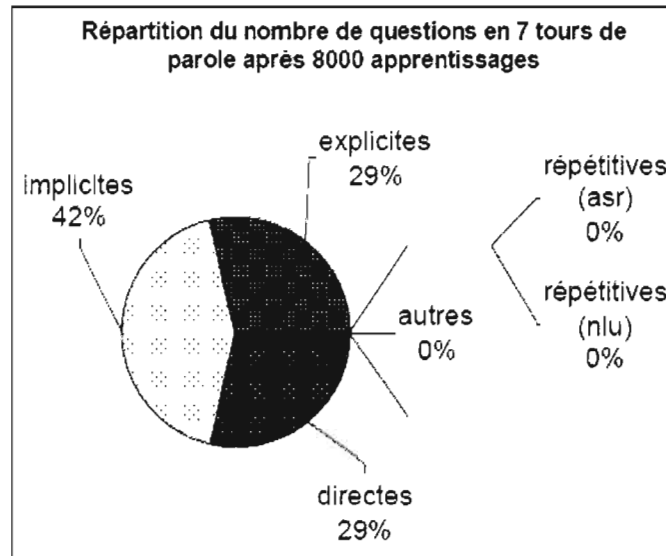


Figure 8.8 Répartition des types de question en 7 tours de parole.

VIII.6 Conclusion

Nous avons mis en place l'environnement d'apprentissage par renforcement avec un utilisateur simulé et nous avons réalisé deux prototypes d'apprentissage : le premier prototype est sans apprentissage, le second possède un agent apprenant qui est capable de corriger lui-même ses erreurs grâce à l'utilisateur simulé. Nous avons démontré que notre approche fonctionne bien et a donné des résultats cohérents selon nos hypothèses.

Les expériences ont démontré que, sans avoir de connaissance en matière de stratégie dialogique, la machine ne peut pas tenir un dialogue cohérent. Et cette stratégie a été transférée à la machine grâce aux méthodes d'apprentissage par renforcement que nous avons proposé. La façon dont nous avons, informatiquement, implanté notre approche (section VIII.4 Organisation de système de fichiers) permet de créer facilement d'autre application dans d'autres domaines.

CHAPITRE IX

ÉVALUATIONS

IX.1 Introduction

Dans ce chapitre, nous présenterons une méthode d'évaluation pour les applications de dialogue, la méthode PARADISE. Bien que cette méthode soit adaptée pour évaluer notre prototype, il nous fallait plus de temps et de ressources pour bien mettre en place cette évaluation. Nous décrirons, néanmoins, les étapes et les scénarios ainsi que les principaux critères pour l'évaluation de notre prototype.

IX.2 Méthode d'évaluation

Walker et al. (1997) [WAL97] ont proposé la méthode d'évaluation PARADISE qui a largement été acceptée par la communauté scientifique pour évaluer les systèmes de dialogue humain-machine. Cette méthode a été appliquée dans l'évaluation du système INSPIRE [MÖL04] (*smart home système*). Cependant, elle a fait l'objet de plusieurs critiques par un grand nombre de chercheurs, notamment Williams et Young (2004) [WIL04]¹. Le point nébuleux de cette méthode réside dans la difficulté de définir de manière objective et constante, le niveau de satisfaction des usagers quant à la tâche à accomplir. Donc, le lien entre la satisfaction des usagers et le temps de complétion d'une tâche n'a pas été mis en évidence. Même si les usagers ne manifestent pas leur mécontentement, il ne serait pas acceptable qu'un système prenne plus que le temps nécessaire pour répondre à la demande des usagers. De plus, il n'est pas démontré, dans le modèle de Walker,

¹Young Researchers' Roundtable on Spoken Dialog Systems. Disponible sur : <http://www.cs.cmu.edu/~dod/YRR/proceedings.pdf> [consulté le 17 avril 2006]

que le succès de la tâche dépend de la stratégie de gestion des erreurs [SKA03], c'est-à-dire que la stratégie de gestion des erreurs ne soit pas considérée comme pertinente et primordiale pour accomplir la tâche dans un système du dialogue humain-machine. Cela est totalement inacceptable dans cette méthode d'évaluation, car on sait qu'une meilleure gestion des erreurs permettrait d'augmenter la chance de compléter la tâche et donc de donner satisfaction aux usagers. Cependant, il est vrai qu'il est difficile d'évaluer la satisfaction des usagers, donc difficile d'évaluer un système de dialogue, étant donné que le dialogue est subjectif et souvent réalisé dans des contextes particuliers (domaine complexe, comportement des usagers, tâches complexes, etc.). Nous avons relevé néanmoins les critères importants pour évaluer les résultats de la recherche.

IX.2.1 Satisfaction globale du dialogue

La pertinence de la stratégie employée dans des situations différentes se mesure par la satisfaction des agents humains (AH) ainsi que par le type de questions posées par le système (question précise et désambiguïsée, ne pas répéter la même question deux fois, etc.). La rhétorique du dialogue (l'art de bien parler) joue aussi un rôle non négligeable dans le succès d'un système de dialogue. Une stratégie est dite optimale lorsqu'elle permet à l'AH d'atteindre son but (l'AH confirmera à la fin du dialogue si le système a compris son but ou non). Nous distinguons au moins quatre situations dialogiques entre humains dans lesquelles nous pouvons évaluer notre prototype :

Situation a : l'utilisateur ne connaît pas le système, mais coopère avec le système (il suit les directives du système et fait des contre-indications). Cela correspond au comportement coopéré de l'utilisateur simulé lors du dialogue.

Situation b : l'utilisateur ne connaît pas le système et ne coopère pas (il donne des réponses erronées, hors contexte, change de buts, revient sur la confirmation, etc.), il cherche à faire échouer le système. Cette situation correspond à un utilisateur simulé dont le comportement est totalement non coopéré.

Situation c : l'utilisateur connaît le système mais son comportement est plus ou moins coopéré. Cependant, il ne cherche pas systématiquement à faire échouer le dialogue comme dans la situation b, mais il n'aide pas non plus à construire un dialogue réussi. Par exemple, il change d'avis de temps à autre et donne plusieurs réponses en même temps. Cette situation correspond à un utilisateur simulé dont le comportement est mixte (tantôt coopéré tantôt non).

Situation d : l'utilisateur connaît le système et il coopère totalement sans contrainte (dialogue idéal). Cette situation correspond à un comportement fixe et totalement coopéré de l'utilisateur simulé.

IX.2.2 Stratégie de confirmation

La gestion des ambiguïtés est effectuée par des vérifications explicites et/ou implicites. Par exemple, notre stratégie de confirmation est basée sur le taux de reconnaissance et d'interprétation de la parole (chapitre VI) pour poser des questions de vérification (implicite, explicite, répétitive). Dans ce cas, le rôle de l'évaluateur est de relever la pertinence des choix des questions du système. Les questions explicites doivent être posées de façon convenable pour ne pas alourdir le système. Mais comme nous avons simulé les modules de compréhension et d'interprétation de la parole dans notre prototype, alors ce niveau d'évaluation ne cadre pas vraiment dans notre démarche. Cependant, cette évaluation sera utile dans un vrai système.

IX.2.3 Interprétation des énoncés

L'interprétation du type d'acte de langage : forme illocutoire, anaphore (répétition sémantique) et ellipse (les références à un ou plusieurs niveaux historiques). Cette mesure est particulièrement nécessaire dans les situations **b** et **c** où l'utilisateur cherche à faire échouer le système.

IX.2.4 Avant et après l'apprentissage

En réalisant deux versions du prototype d'apprentissage, nous avons pu constater qu'il y a une nette amélioration dans le dialogue entre les deux agents

avant et après l'apprentissage. Cette progression reste stable après 9 épisodes d'apprentissage.

IX.2.5 Adaptation dans d'autres domaines

Pour évaluer les degrés de difficulté quant à la réalisation d'une nouvelle application à partir d'une application existante développée selon notre approche, il faut procéder à des mesures et comparaisons suivantes :

1. Le temps de développement d'une application avec et sans notre approche.
2. Le degré de difficulté en modifiant une application développée avec notre approche par rapport à une application sans notre approche.
3. Le degré de difficulté en adaptant notre application dans d'autres domaines par rapport à une application sans notre approche.
4. La performance entre les deux applications avec et sans notre approche.
5. Le degré de difficulté quant à l'ajout d'une nouvelle langue dans une application avec notre approche et une application sans notre approche (e.g., française, anglaise, chinoise, espagnole, etc.).

IX.3 Obstacles pour l'évaluation

La meilleure façon pour évaluer notre approche est de l'implanter dans un domaine plus complexe et de tester la performance du système en termes de dialogue réussi après un certain temps d'apprentissage. Pour cela, il faut mettre en place une structure de développement plus complexe (équipe de développement, équipe de test, équipe d'évaluation) ; cela demande plus de ressources financières et de temps.

IX.4 Conclusion

Nous avons proposé la méthode d'évaluation PARADISE pour tester la performance de notre prototype. Par cette évaluation, nous pouvons vérifier nos hypothèses et notre méthodologie concernant l'apprentissage de la stratégie de dialogue humain-machine. La stratégie apprise est-elle optimale ? Et enfin, concernant la plate-forme informatique, nous n'avons pas de mesures spécifiques pour vérifier si les programmes développés dans cette plate-forme sont flexibles et portables dans d'autres environnements de développement. Cela est attribuable à la difficulté de mise en place d'une telle mesure pour une plate-forme informatique. Cependant, nous pouvons facilement tester la flexibilité de la plate-forme en observant comment les programmes sont structurés (code source libre, programmation modulaire, création des bibliothèques de fonction, d'objets, la facilité de modifier un code, etc.).

CHAPITRE X

CONTRIBUTION, DISCUSSION ET CONCLUSION

X.1 Contribution

Avec nos hypothèses de solutions pour un meilleur système de dialogue, nous avons contribué, dans cette thèse, à éclaircir plusieurs aspects de recherche dans ce domaine. D'abord, pour la méthodologie de recherche elle-même, nous avons proposé un cadre de recherches selon une méthode triangulaire et un cheminement spiral (chapitre I sur le cheminement méthodologique de la recherche). Nous avons proposé, ensuite, une approche d'apprentissage par renforcement à l'aide d'un utilisateur simulé. Nous avons démontré que notre approche fonctionne en créant un prototype d'apprentissage. Un système avec une stratégie apprise sera meilleur (expérience 1 et 2). Nous avons mis en place une stratégie simple mais efficace pour les applications de dialogue orienté-but et questions-réponses. Cette stratégie basée sur la qualité moyenne de reconnaissance et d'interprétation, a permis de rendre le système de dialogue plus robuste et plus performant.

Lors de notre recherche, nous avons constamment remis en question nos hypothèses et notre réflexion, à savoir : (1) l'architecture d'apprentissage à l'aide d'un utilisateur simulé est-elle pertinente pour la construction de nouvelle application en dialogue ? (2) Le processus de décision de Markov donne-t-il des résultats plus valables que d'autres méthodes ? (3) La stratégie choisie (degrés d'initiatives et stratégie de confirmation) permet-elle de réaliser des dialogues souples et naturels ?

Les réponses à ces questions justifient la contribution de la thèse. D'abord, concernant la question sur l'utilisateur simulé et l'apprentissage par renforcement,

(1) il se trouve que cette méthode permet de faire apprendre par renforcement en 'online' avec l'aide de l'utilisateur simulé ce qui est impraticable avec un utilisateur réel en raison du coût et du temps de traitement. Nous avons pu constater qu'après 5 périodes d'apprentissage (5000 exécutions), une solution optimale est trouvée.

Quant au processus de décision de Markov, (2) nous avons constaté qu'il ne peut pas modéliser tous les cas complexes du domaine. D'autres voies de solutions comme le processus de décision partielle de Markov sont proposées par [CUA05], mais cela ne résout pas toute la complexité du domaine. Notre approche consiste à dire qu'un système de dialogue a ses limites et qu'il nous faut trouver une manière de faire dialoguer l'humain et la machine en tenant compte de la performance actuelle des systèmes de reconnaissance et de compréhension de la parole. Cette manière se trouve dans le (3) de la stratégie de dialogue. Nous avons démontré que :

1. Pour réaliser un dialogue satisfaisant, il suffit d'adopter une stratégie de dialogue basée sur cinq types de questions : directe, implicite, explicite et répétitive d'ASR et d'NLU (mal entendu et mal compris). Il n'est pas nécessaire de chercher à tout prix à rendre le module de reconnaissance parfaitement robuste et performant, ni de chercher à résoudre absolument le problème d'interprétation sémantique de la parole. Il suffit de demander intelligemment à l'utilisateur de confirmer sa réponse. Et cela se trouve dans la technique de communication, c'est ce que nous appelons, dans cette thèse, une stratégie optimale de dialogue.
2. Les actes de dialogue peuvent être modélisés selon le processus de décision de Markov et cette modélisation permettra de mieux contrôler les interactions (orales ou non) entre l'humain et la machine.
3. Un système de dialogue peut apprendre une stratégie optimale du dialogue et enfin,
4. Un système de dialogue avec une stratégie apprise sera meilleur.

X.2 Futur travail

Dans le but de comparer notre stratégie avec celle des autres, il serait intéressant de développer une troisième version du prototype qui consiste à adopter une stratégie plus *risquée*. Par exemple, lorsque la qualité de reconnaissance est moyenne, l'agent machine devrait poser plus de questions implicites qu'explicites ou répétitives, ou alors opter pour une stratégie plus *prudente*. L'agent machine peut forcer l'utilisateur à répondre par oui ou non. Cela pourrait conduire à un dialogue plus long mais, au moins, cette stratégie pourrait pallier aux très mauvaises performances de reconnaissance.

Nous pouvons développer un système avec cinq variables et plus, et regrouper les questions de confirmations par groupe de trois. Par exemple, outre les informations requises pour une réservation de chambres d'hôtel, le système a besoin de savoir si le client veut louer une voiture ou non, quel type de voiture, s'il désire des places d'attraction ou les noms des restaurants, etc. Dans ce dialogue, si on attend la fin du dialogue pour poser les questions de confirmation, le système risque de construire une phrase très longue (e.g., « *vous avez réservé une chambre de type 'suite royale' pour 2 du 23 au 24, avec une location de voiture de type compact et vous avez acheté 3 billets pour le spectacle Mickey Mouse ainsi qu'une réservation au restaurant Les trois mousquetaires. Pouvez-vous nous confirmer le tout par oui ou non S.V.P.* »). Cela n'est pas acceptable, donc il faut confirmer avec le client une partie de ses réponses avant de continuer à lui poser d'autres questions.

X.3 Discussion

X.3.1 Informatique

Comme cela a été dit dans la section 'objectif de recherche', la recherche entreprise, dans ce projet, vise à contribuer à l'amélioration de la performance du système de dialogue et à la réduction du temps de développement de nouvelles applications. Réduire le temps veut aussi dire amélioration de la méthode d'apprentissage par renforcement. Les problèmes rencontrés sont plutôt de l'ordre de la représentation des connaissances que de l'ordre de la réalisation informatique,

d'abord au niveau de l'annotation sémantique, ensuite au niveau de la formalisation des actes de dialogue pour le MPD.

X.3.1.1 Annotation sémantique

Il existe des transcriptions de données pour le dialogue dans divers domaines d'application (service, renseignement, interrogation des bases de données, etc.), mais ces données ne sont pas tout sémantiquement annotées, ce qui rend difficile le travail d'apprentissage avec les corpus de données. Il faut annoter manuellement ces données ce qui rend le processus d'annotation long à finir. D'après notre expérience, il est plus pratique de construire l'utilisateur simulé avec une méthode empirique plutôt qu'avec une méthode stochastique. C'est-à-dire que l'utilisateur simulé possède des règles de construction de stratégies optimales de dialogue. Ces règles sont programmées à partir des analyses fonctionnelles et non à partir de l'apprentissage supervisé avec des données typiques. Par la suite, lorsque deux agents se mettent à *'dialoguer'*, nous employons la méthode stochastique pour apprendre à l'agent machine à acquérir les stratégies optimales.

X.3.1.2 Complexité de MPD

Les états et les actions de MDP s'accroîtront de manière exponentielle en fonction de la complexité de l'application. Si on a cinq attributs (e.g., date de départ, date d'arrivée, nombre de chambres, nombre de personnes, type de chambre) et six actions à gérer pour chaque attribut, alors le nombre d'états-actions devient vite ingérable. Il faut, donc, trouver une façon de réduire ce nombre. La solution que nous avons trouvée pour réduire ce nombre est le travail de l'expert en dialogue. Nous avons choisi, au préalable, les questions implicites, explicites ou directes qui doivent être posées par l'agent machine à l'utilisateur. Notre algorithme permet de créer des états pour le processus de décision de Markov au fur et à mesure que le dialogue évolue entre les deux agents. Le nombre d'états est proportionnel à la complexité de l'application et du domaine (e.g., des millions de combinaisons états-actions). Cela devient problématique lorsque nous devons travailler dans un domaine plus complexe. Par conséquent, il faut à un moment donné, trouver une façon de gérer

efficacement les fichiers de grande taille en termes de temps d'accès et de mise à jour des données. Il faut probablement recourir aux méthodes des systèmes de base de données relationnelles pour gérer ce fichier. Mais, pour un domaine restreint (quelques milliers d'états-actions), il n'est pas nécessaire de l'employer.

X.3.2 Cognition

Un de nos objectifs cognitifs dans cette recherche, est d'appliquer le dialogue humain-humain dans le dialogue humain-machine et d'appliquer le processus d'apprentissage d'humain à la machine en se basant sur la méthode d'apprentissage par renforcement. Si ce processus est transférable d'humain à la machine, alors on peut se demander si la machine pourrait penser ou parler un jour ? Nous nous retrouvons face à un dilemme entre l'intelligence de l'être et celle de la machine.

X.3.2.1 L'intelligence machine versus l'intelligence humaine

Nous sommes convaincus que l'intelligence de la machine ne parviendra jamais à égaler celle de l'homme. En effet, elle ne peut pas s'élever au même niveau que la nôtre sans que nous ne mettions nos propres intelligences dans ces machines. Et même, en supposant que nous pouvons le faire, combien faudrait-il de quantités des connaissances à la machine pour qu'elle puisse égaler la nôtre ? Sans parler de la qualité de celles-ci. Et mêmes si les machines aujourd'hui s'exécutent des millions de fois plus rapidement que nous et que leur capacité d'apprentissage est infailible et infatigable, elles ne pourront cependant jamais parvenir à posséder notre intelligence, car elles auront toujours besoin du cerveau d'un homme (homoncule) pour leur indiquer le bon choix. Cela nous amène, logiquement, à déduire qu'elles ne peuvent exécuter que des tâches monotones, procédurales. Elles peuvent inférer d'autres connaissances à condition que nous les ayons soigneusement préparé et que nous ayons introduit des règles et des exemples de raisonnement dans leurs mémoires. Donc, l'intelligence de la machine est encore très loin de notre intelligence.

Nous pensons que la machine ne pourra jamais penser ni parler comme nous. En effet, nous soutenons l'idée de la théorie de Searle (1969, 1980) concernant

l'intelligence de la machine, qu'il place dans la catégorie de l'intelligence artificielle faible. Pourquoi, à travers une simple expérience de programmation d'un prototype d'apprentissage automatique, peut-on défendre l'idée de Searle à propos de l'intelligence de la machine ? Searle a affirmé que les ordinateurs n'ont pas d'état mental (l'exemple de la Chambre Chinoise), ils ne peuvent pas penser, ni avoir de conscience car ils ne font que donner des réponses à partir de règles que nous leur avons montrées (Searle, 1969, 1980) [SEA69], [SEA80]. Par analogie, les systèmes de dialogue oral humain-machine peuvent générer des signaux acoustiques en sortie, cohérents par rapport aux signaux acoustiques en entrée à condition que nous fournissions des règles explicites. Ces règles sont gérées par le gestionnaire du dialogue. Bien que le gestionnaire du dialogue soit capable de 'trouver' la bonne réponse, il ne saura pas véritablement ce que cela signifie. La question est de savoir si ce système est capable de générer lui-même ce '*rulebook*'¹ et d'ajouter d'autres règles dans ce '*rulebook*'. La réponse est non, évidemment, car pour pouvoir enrichir ce livre, il lui faudrait une règle d'enrichissement qui, doit, encore une fois, être créée par nous, donc une sorte d'homuncule à l'intérieur d'un autre homuncule.

Par conséquent, même si le processus d'apprentissage d'humain est transférable d'humain à la machine, alors la machine ne pourrait jamais acquérir les quantités de connaissances que nous possédons, donc elle ne pourra jamais parler, ni penser.

X.3.2.2 Le corps, l'esprit et l'environnement

Si on considère l'aspect d'apprentissage de la machine du SDOHM comme une sorte d'imitation de notre processus d'acquisition de connaissance, alors de quelle manière peut-on construire un tel système artificiel? Et quelle partie de ce processus pourrait être modélisée? L'esprit, le corps, l'environnement ou les trois en même temps? Nous ne prétendons pas donner une réponse à l'issue de cette recherche, mais nous sommes convaincus qu'aucun de ces trois éléments doit être privilégié plus que d'autres, puisque selon Descartes, le corps et l'esprit,

¹ Site Internet : <http://www.psych.utoronto.ca/~reingold/courses/ai/cache/searle.html>. Consulté 1-oct-06.

constamment en dual, bien qu'indissociable, ne font qu'un. Cet ensemble de dualité ne peut pas exister s'il ne reste pas dans un environnement qui lui permet d'évoluer. Autrement dit, la science cognitive moderne introduit le troisième élément que Descartes exclut de sa pensée ; celui de l'environnement qui fait partie de l'homme. Nous ne pouvons pas évoluer sans intégrer cette partie dans notre processus d'acquisition des connaissances. On peut se demander en quoi ce 3^{ème} élément nous aide à comprendre la capacité cognitive de la machine vis-à-vis la pensée et la parole? Le problème est de savoir comment intégrer ce troisième élément dans le système de dialogue. Les chercheurs n'ont-ils pas attribué ce troisième élément au niveau du pragmatisme ? Le niveau qui sépare notre capacité de pensée, de parole de celle d'un perroquet.

Nous voulons, dans cette recherche, essayer d'apporter une piste de réflexion quant à la conception des systèmes de dialogue oral humain-machine de demain. Ces systèmes, doivent-ils être construits sur la base des théories de la communication entre les humains ? Ou existe-t-il d'autres façons de communiquer entre l'humain et la machine qui soient différentes de celles des humains? Devons-nous nous concentrer sur l'aspect surfacique du système, c'est-à-dire sur la façon de communiquer, donc sur la stratégie de dialogue ? Où devons-nous élaborer des plans d'actions cohérents dans un domaine restreint plutôt que de chercher à résoudre les grandes questions comme le transfert de la capacité cognitive humaine vers la machine? (Rouillard, 1999, 2000) [ROU99], [ROU00].

Toutes ces questions ouvertes ne trouveront certainement pas de réponses immédiatement, mais sachant que nos machines actuelles sont très limitées par rapport à notre capacité cognitive (perception, apprentissage et adaptation dans l'environnement, etc.), nous devons, donc, mettre plus d'efforts sur les méthodes qui permettraient à la machine d'apprendre plus vite et plus efficacement plutôt que de chercher à tout prix à implanter des théories cognitives dans la machine.

Selon nous, pour qu'un SDOHM soit performant et efficace, il faut qu'il acquière des connaissances du monde (pragmatisme) et ce, même dans un domaine

très restreint. Cela nous oblige à reconsidérer la place du système de dialogue dans notre société. Est-ce que nous voulons construire un système qui nous aide à accomplir nos tâches ou nous voulons qu'il travaille à notre place ? La réponse à cette question se trouve dans le but de chaque application de dialogue que nous voulons construire. Et dès le début de notre thèse, nous avons expliqué que notre modèle est fait pour être appliqué dans un système de type orienté-but. Chaque système à construire doit avoir un but précis, sans cela, il nous serait difficile de construire un système performant et efficace.

X.4 Conclusion

Selon nous, un système de dialogue oral humain-machine est considéré comme efficace, robuste et performant lorsqu'il répond à ces critères :

- 1- Le système doit être capable de maintenir un échange naturel et souple lors du dialogue.
- 2- À la fin de la conversation ou du dialogue, les deux parties arrivent à un consensus (consentement) ou un compromis général qui satisfera les deux parties.
- 3- L'abandon du dialogue n'est pas dû au système (mauvaise réception du signal de la parole, temps de réponse trop long, mauvaise interprétation, erronée, incompréhension, mauvaise génération de la parole, réponse du système inappropriée).
- 4- Il n'est pas nécessaire d'avoir la même capacité cognitive que l'homme pour pouvoir entretenir un dialogue avec lui (notre hypothèse de départ).
- 5- Approfondir l'apprentissage machine, c'est la seule façon pour un système d'élever sa connaissance et sa performance.
- 6- Choisir une plate-forme informatique flexible et adaptée pour pouvoir gérer les changements et les évolutions de programmes.

La puissance des ordinateurs n'est pas un facteur déterminant pour un système de dialogue humain-machine robuste, ni le module de reconnaissance et de synthèse de la parole. Ces modules ne font que traduire les représentations symboliques des structures de surface de la parole en une structure de surface du texte et vis-versa. Quant au module d'interprétation de la parole, il est important de bien *comprendre* les énoncés en entrée du système, mais cela n'est pas la fonction majeure du SDOHM. Concernant la gestion dialogique par l'emploi d'une stratégie optimale, elle contribue partiellement à formuler une réponse intelligente, elle guide et oriente le dialogue dans un sens ou dans l'autre sans se perdre dans le dialogue. Le module de génération de la parole s'occupe, à partir des règles grammaticales et de la stratégie, des réponses prises en aval par la gestion de stratégie dialogique et de construire des réponses sous forme de propositions (phrases, énoncés). Autrement dit, le module de génération construit une structure de surface supposée représenter la connaissance – le message parlé. Mais, la génération de la parole est en quelque sorte, commanditée par la gestion de la stratégie dialogique. Il n'y a pas de réponse intelligente sans une gestion intelligente. Donc, la puissance des ordinateurs ne permet pas seulement que de rendre le dialogue plus agréable à entendre (e.g., voix plus naturelle, moins hachée, temps de réponse plus rapide), mais le cœur du problème reste toujours de savoir comment générer une sortie vocale cohérente avec le contexte du discours, donc, comment générer une connaissance pertinente dans une situation donnée? Emettons l'hypothèse que l'outil adéquat pour générer la connaissance se trouve en partie dans la stratégie du dialogue, car sans cette stratégie, le module de génération ne peut donner une réponse intelligente. Par exemple, une personne possède toutes ses capacités physiques et mentales, mais ne sait pas communiquer, elle ne peut pas établir un dialogue cohérent (on peut même dire dialogue naturel) avec les autres. Tout comme un SDOHM, sans une stratégie de dialogue bien contrôlé, le dialogue ne peut pas se dérouler de façon souple et naturelle.

Nous parlons de la façon dont un système de dialogue est construit en regard de la science cognitive. Ce système, doit-il ressembler à notre comportement de

communication entre humains ? Ou alors, en tenant compte des limites d'intelligence de la machine et de sa performance en calcul, devons-nous nous inspirer de cette théorie (de la communication humain-humain) pour trouver une autre façon d'établir une communication entre l'humain et la machine? (l'oiseau versus l'avion, la marche versus la voiture, etc.). Tel est notre point de réflexion pour de futures directions de recherche.

Pour le moment, on peut, avec des méthodes d'apprentissage machine inspirées de la science cognitive, transférer nos connaissances vers la machine. On imagine qu'avec la quantité de connaissances que nous enseignons à la machine et avec les moyens d'acquisition de connaissance plus rapides (par la parole) et plus robustes, la machine de demain pourra nous aider efficacement lors de notre propre apprentissage afin de mieux nous adapter à l'environnement.

APPENDICE A

LES EXEMPLES DE CODE

A.1 Code du simulateur d'utilisateur

Référence sur demande (www.nuance.com)

A.2 Code d'apprentissage par renforcement

Référence sur demande (www.nuance.com)

A.3 Écran d'interface windows

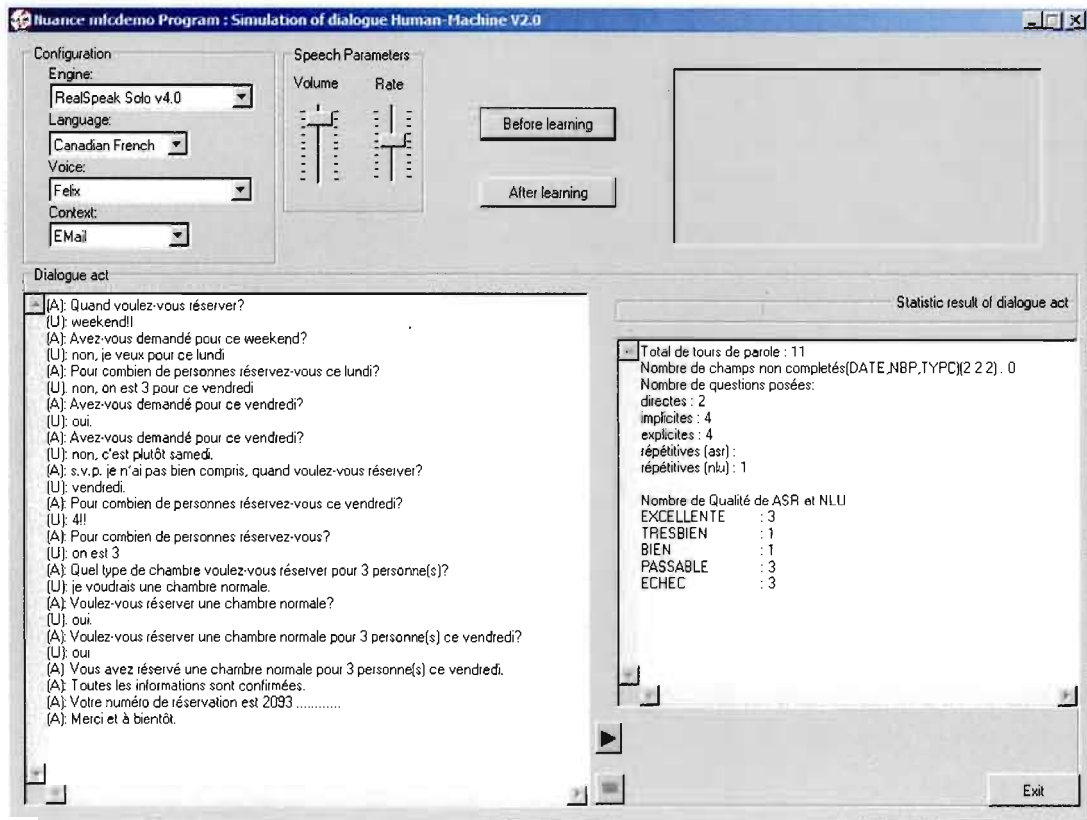


Figure 10.1 Écran d'interface Windows : Simulation de dialogue entre un utilisateur simulé (Félix) et un agent machine (Julie) avant l'apprentissage.

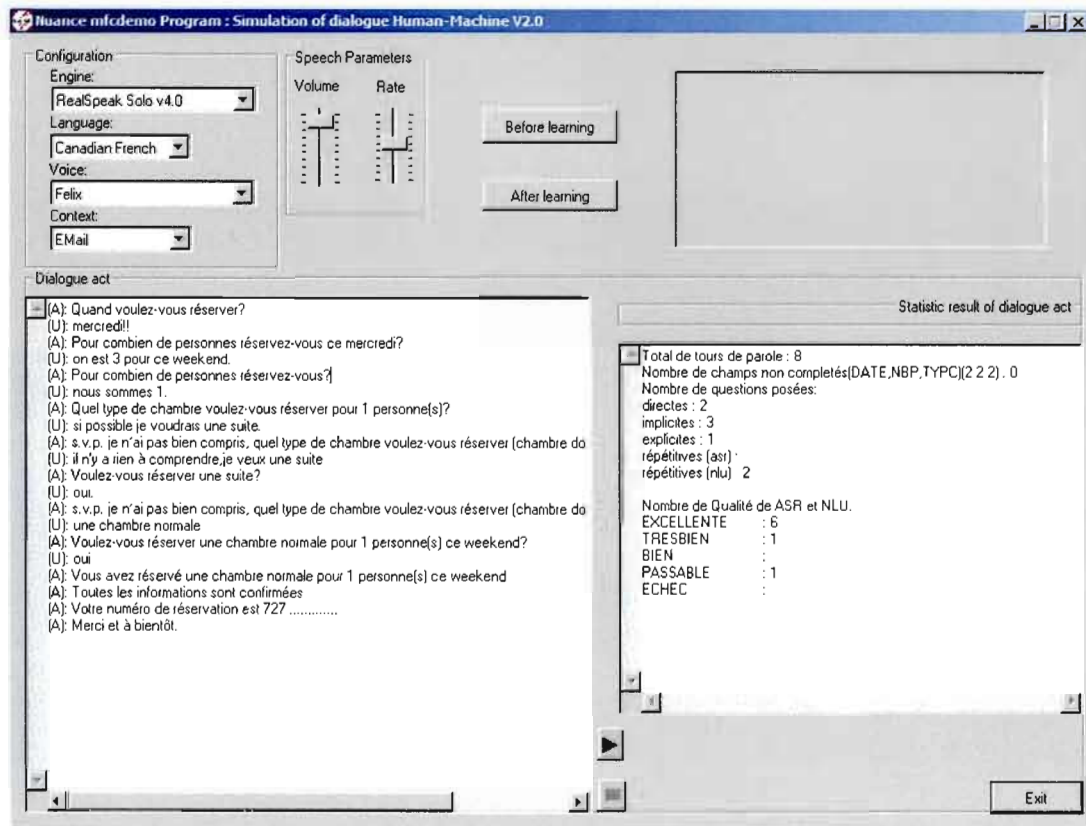


Figure 10.2 Écran d'interface Windows : Simulation de dialogue entre un utilisateur simulé (Félix) et un agent machine (Julie) après 5000 exécutions d'apprentissage.

A.4 Exemples de dialogues créés dans la 1ière expérience

Sans apprentissage et avec un utilisateur simulé non-coopéré

Lorsque l'utilisateur simulé adopte un comportement non coopéré, il cherche à faire échouer le système en donnant des réponses incorrectes (*Quand voulez-vous réserver ? non*) ou hésitantes (*Quand voulez-vous réserver ? euhm...je ne sais pas*). Le tableau 10.1 donne un exemple d'une session de dialogue réussi avec un utilisateur non-coopéré. On remarque, dans ce dialogue que la première question que l'agent machine a posé est « *s.v.p. je n'ai pas bien entendu. Quand voulez-vous réserver ?* » Ensuite, il a demandé « *Avez-vous demandé pour ce ?* ». L'agent machine n'a pas de fonction apprentissage, donc il choisit au hasard une question.

Malgré le fait que les réponses de l'utilisateur sont incohérentes et que le dialogue est long, l'agent machine, quand même, a réussi à obtenir la confirmation de tous les champs en 21 tours de parole. Ce résultat nous sert à analyser le comportement dialogue humain-machine et ne peut être considéré comme une stratégie optimale du dialogue car il est inconcevable de constater qu'il faut 21 tours de parole pour obtenir seulement trois informations.

Recherche de stratégie optimale de dialogue par apprentissage par renforcement
Version : 1.4 (sans d'algorithme d'apprentissage - dialogue sans aucune stratégie optimale, utilisateur non coopéré)
Auteur : @Minh-Quang Nguyen, 2003-2007, Doctorat en informatique cognitive, UQAM/INRS/Nuance Communications Inc.

État initial du Processus de Décision de Markov (pdm)

[simuler la qualité d'ASR/NLU=(2) => action choisie=(3)]

(A): s.v.p. je n'ai pas bien entendu, quand voulez-vous réserver?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(3) => action choisie=(2)]

(A): Avez-vous demandé pour ce ?

(U): oui

[simuler la qualité d'ASR/NLU=(0) => action choisie=(4)]

(A): s.v.p. je n'ai pas bien compris, quand voulez-vous réserver?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(2) => action choisie=(2)]

(A): Avez-vous demandé pour ce ?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(3) => action choisie=(4)]

(A): s.v.p. je n'ai pas bien compris, quand voulez-vous réserver?

(U): je voudrais réserver pour ce weekend

[simuler la qualité d'ASR/NLU=(0) => action choisie=(1)]

(A): Pour combien de personnes réservez-vous ce weekend?

(U): non, on est 3 pour ce jeudi

[simuler la qualité d'ASR/NLU=(2) => action choisie=(1)]

(A): Pour combien de personnes réservez-vous ce jeudi?

(U): non, on est 1 pour ce mardi

[simuler la qualité d'ASR/NLU=(0) => action choisie=(0)]

(A): Quand voulez-vous réserver?

(U): je voudrais réserver pour ce mardi

[simuler la qualité d'ASR/NLU=(4) => action choisie=(0)]

(A): Quand voulez-vous réserver?

(U): je voudrais réserver pour ce mardi

[simuler la qualité d'ASR/NLU=(3) => action choisie=(4)]

(A): s.v.p. je n'ai pas bien compris, pour combien de personnes voulez-vous

réserver?

(U): on est 3

[simuler la qualité d'ASR/NLU=(1) => action choisie=(0)]

(A): Pour combien de personnes réservez-vous?

(U): oui

[simuler la qualité d'ASR/NLU=(2) => action choisie=(1)]

(A): Quel type de chambre voulez-vous pour 3 personnes?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(3) => action choisie=(1)]

(A): Quel type de chambre voulez-vous pour 3 personnes?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(1) => action choisie=(1)]

(A): Quel type de chambre voulez-vous pour 3 personnes?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(4) => action choisie=(1)]

(A): Quel type de chambre voulez-vous pour 3 personnes?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(0) => action choisie=(0)]

(A): Quel type de chambre voulez-vous réserver?

(U): oui

[simuler la qualité d'ASR/NLU=(4) => action choisie=(0)]

(A): Quel type de chambre voulez-vous réserver?

(U): oui

[simuler la qualité d'ASR/NLU=(0) => action choisie=(4)]

(A): s.v.p. je n'ai pas bien compris, quel type de chambre voulez-vous réserver?

(U): non,je voudrais une suite

[simuler la qualité d'ASR/NLU=(3) => action choisie=(2)]

(A): Voulez-vous réserver une suite?

(U): oui

[simuler la qualité d'ASR/NLU=(2) => action choisie=(2)]

(A): Voulez-vous réserver une suite?

(U): euhm...je ne sais pas

[simuler la qualité d'ASR/NLU=(4) => action choisie=(1)]

(A): Voulez-vous prendre la suite pour 3 personnes pour ce mardi?

(U): euhm...je ne sais pas

(A): Vous avez réservé une (suite) pour (3) personne(s) ce (mardi)

(A): Votre numéro de réservation est 32. Merci et à bientôt.

Total de tours de parole : 21

Nombre de champs non complétés (DATE,NBP,TYPC)(2 2 2) : 0

Nombre de questions posées :

directes : 5

implicites : 7

explicites : 4

répétitives (asr) : 1

répétitives (nlu) : 4

Nombre de Qualité de ASR et NLU:	
EXCELLENTE	: 4
TRÈS BIEN	: 5
BIEN	: 5
PASSABLE	: 2
ECHEC	: 5

Tableau 10.1 Dialogue entre un utilisateur simulé non coopéré et un agent apprenant sans apprentissage.

On remarque que dans cette répartition (figure 10.1), l'agent machine a posé moins de questions implicites que dans le cas où l'utilisateur est coopéré. Cela s'explique par le fait que l'utilisateur simulé change constamment d'avis et que la qualité de reconnaissance est moins performante.

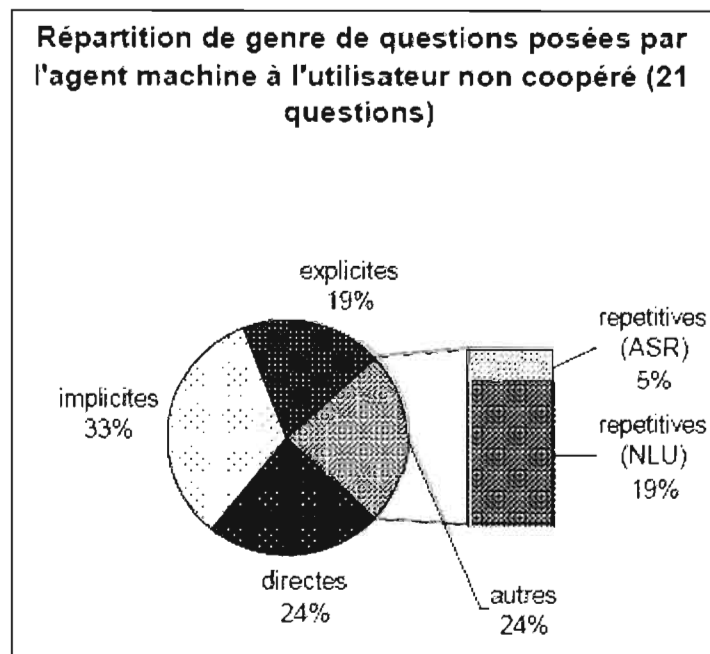


Figure 10.3 Répartition de genre de questions posées par l'agent machine à l'utilisateur non coopéré.

BIBLIOGRAPHIE

- [ABE99] A. Abella and A. Gorin. 1999. Construct Algebra: Analytical Dialog Management. In *Proc. ACL*, Washington D.C.
- [ALL00] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent. 2000. An Architecture for a Generic Dialogue Shell. *Natural Language Engineering*. Vol. 6. Cambridge University Press.
- [ALL01] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent. 2001. Towards conversational human-computer interaction. *AI Magazine*, 22(4):27-38.
- [ALL76] J. Allwood. 1976. *Linguistic Communication as Action and Cooperation*. Gothenburg Monographs in Linguistics 2. Göteborg University, Department of Linguistics, Sweden.
- [ALL79] J. Allen. 1979. A plan-based approach to speech act recognition. Technical Report 131/79, University of Toronto. PhD thesis.
- [ALL96] J. Allen, B.W. Miller, E.K. Ringger and T. Sikorski. 1996. A Robust System for Natural Spoken Dialogue. *Proc. of ACL '96*.
- [AMA93] R. Amalberti, C. Valot. 1993. Le Magicien d'Oz. CERMA, journée du PRC Rhône-Alpes.
- [ANT01] J.Y. Antoine, J.Goulian. 2001. Word order variations and spoken man-machine dialogue in French : a corpus analysis on the ATIS domain. *Corpus Linguistics'2001*, UCREL, Lancaster.
- [ANT01b] J.Y. Antoine, J.Goulian. 2001. Étude des phénomènes d'extraction en français parlé sur deux corpus de dialogue oral finalisé. *TAL*, 42:1-20.
- [ARA97] M. Araki, T.Watanabe and S. Doshita. 1997. Evaluating dialogue strategies for recovering from misunderstandings. In *Proc. IJCAI Workshop on Collaboration Cooperation and Conflict in Dialogue Systems*, pages 13–18.
- [ASA05] H. Asai, T. Koshizen, M. Watanabe, H. Tsujino and K. Aihara. 2005. Cognitive User Modeling Computed by a Proposed Dialogue Strategy Based on an Inductive Game Theory. *Studies in Computational Intelligence (SCI)*, 7, 325-351. Springer-Verlag Berlin Heidelberg.
- [AUS62] J.L. Austin. 1962. *How to Do Things with Words*. Harvard University Press, Cambridge, Massachusetts.
- [BAL99] B. Balentine, D.P. Morgan. 1999. *How to build a Speech Recognition Application*, Enterprise Intergration Group., CA.
- [BIL91] E. Bilange. 1991. An approach to oral dialogue modelling. In *Proc. of Second Venaco Workshop on the structure of multimodal dialogue*, Acquafredda di Maratea, pp. 1-12.
- [BLA05] N. Blaylock and J. Allen. 2005. Generating artificial corpora for plan recognition. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, *User Modeling 2005*, in *Lecture Notes in Artificial Intelligence*. Springer, Edinburgh, 3538:179–188
- [BOE88] B. Boehm. 1988. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5).
- [BOH03] D. Bohus and A. Rudnicky. 2003. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. In *Proc. of Eurospeech*, Geneva, Switerland, pp. 597-600.
- [CAR83] S.K. Card, T.P. Moran and A. Newell. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1983.
- [CHA92] D. J. Chalmers. 1992. Subsymbolic Computation and the Chinese Room. In *The Symbolic and Connectionist Paradigms: closing the gap*, ed: John Dinsmore, Lawrence Erlbaum Associates, pp. 25-48.
- [CHO56] N. Chomsky. 1956. Three Models for Description of Languages. *IRE. Transaction on Information Theory*, pp. 113-124.
- [CHO65] N. Chomsky. 1965. *Aspect of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965.

- [CHU00a] J. Chu-Carroll. 2000. MIMIC: An Adaptive Mixed Initiative Spoken Dialogue System for Information Queries. *ANLP*, pp. 97-104
- [CHU00b] J. Chu-Carroll and S. Carberry. 2000. Conflict resolution in collaborative planning dialogs. *Int. J. Hum.-Comput. Stud.* 53(6): 969-1015.
- [CHU04] G. Chung. 2004. Developing a Flexible Spoken Dialog System Using Simulation. In *ACL*, pp. 63-70, Barcelona, Spain.
- [CHU98] J. Chu-Carroll and M.K Brown. 1998. An evidential model for tracking initiative in collaborative dialogue interactions. *User Modeling and User-Adapted Interaction*, 8(3-4):215-253.
- [CHU99] J. Chu-Carroll. 1999. Form-Based Reasoning for Mixed-Initiative Dialogue Management in Information-Query Systems. In *Proc. of Eurospeech*, Budapest, Hungary, pp. 1519-1522.
- [COH86] P.R. Cohen and C.R. Perrault. 1986. Elements of a Plan-based Theory of Speech Acts. Readings in NL Processing. 423-439.
- [CUA05] H. Cuayahuitl, S. Renals, O. Lemon and H. Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. in *Proc. of IEEE ASRU Workshop*, Mexico.
- [CUA06] H. Cuayahuitl, S. Renals, O. Lemon and H. Shimodaira. 2006. Reinforcement Learning of Dialogue Strategies With Hierarchical Abstract Machines. In *Proc. of IEEE/ACL Workshop on Spoken Language Technology*. Palm Beach, Aruba.
- [ECK97] W. Eckert, E. Levin and R. Pieraccini. 1997. User Modeling for Spoken Dialogue System Evaluation. in *Proc. of IEEE ASRU Workshop*, Santa Barbara, Cal., USA.
- [ELZ95] S. Elzer. 1995. The role of user preferences and problem-solving knowledge in plan recognition for expert consultation systems. In *Working Notes of the IJCAI-95 Workshop on The Next Generation of Plan Recognition Systems*, pages 37-41, Montreal, Canada.
- [ENG05] M. English and P. Heeman. 2005. Learning mixed initiative dialog strategies by using reinforcement learning on both conversants. In *Proc. of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 1011-1018, Vancouver, Canada.
- [FAB04a] G. Di Fabbrizio and C. Lewis. 2004. Florence: a Dialogue Manager Framework for Spoken Dialogue Systems. In *Proc. of the 8th International Conference on Spoken Language Processing (ICSLP 2004)*, Jeju Island, Korea.
- [FAB04b] G. Di Fabbrizio and C. Lewis. 2004. An XPath-based Discourse Analysis Module for Spoken Dialogue Systems. *The 9th International World Wide Web Conference*, New York.
- [FIL68] Ch. J. Fillmore. 1968. *The case for case*. Universals in Linguistic Theory, pages 1-90.
- [FOU04] Y. Fouquet. 2004. *Modélisation des attentes en dialogue oral*. Thèse de doctorat d'informatique, Université Grenoble I.
- [GAI93] B.R. Gaines and M.L.G. Shaw. 1993. Eliciting Knowledge and Transferring It Effectively to a Knowledge-Based System. *IEEE Trans. Knowl. Data Eng.* 5(1): 4-14.
- [GEO05] K. Georgila, J. Henderson and O. Lemon. 2005. Learning user simulations for information state update dialogue systems. In *Eurospeech*, Lisbon, Portugal.
- [GOD00] D. Goddeau and J. Pineau. 2000. Fast Reinforcement Learning of Dialog Strategies. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP2000)*.
- [GOL05] C. Golanski, J. Caelen. 2005. *Test d'utilisabilité de l'agent conversationnel « Angela »*, WACA, Grenoble, France.
- [GRI75] H.P. Grice. 1975. *Logic and conversation*. New York: Academic Press Edition, 3:41-58.
- [GRI89] H.P. Grice. 1989. *Study in the way of words*. Cambridge. Harvard UP.
- [GRO86] B.J. Grosz and C.L. Sidner. 1986. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175-204.
- [HAL99] S. Haller. 1999. An introduction to interactive discourse processing from the perspective of plan recognition and text planning. *Artificial Intelligence Review*, 13:259-311.
- [HEN05] J. Henderson, O. Lemon, K. Georgila. 2005. Hybrid reinforcement/supervised learning for dialog policies from communicator data. In *Proc. of IJCAI on KRPS*, Edinburgh, Scotland.

- [HOC04] M. Rayner and B.A. Hockey. 2004. Side Effect Free Dialogue Management in a Voice Enabled Procedure Browser. In *Proc. of INTERSPEECH*, Jeju Island, South Korea.
- [HOC05] B.A. Hockey and M. Rayner. Proceedings of the AAAI Workshop on Spoken Language Understanding, Pittsburgh, PA. 2005.
- [HOC88] J.M. Hoc. 1988. *Cognitive Psychology of Planning*. Academic Press London.
- [JAR05] I. Jars. 2005. Contribution des Sciences Sociales dans le domaine de l'Intelligence Artificielle Distribuée : ALONE, un modèle hybride d'agent apprenant. Thèse de Doctorat, Université Claude Bernard - Lyon.
- [JAR05a] I. Jars, N. Kabachi et M. Lamure. 2005. Proposal for a Vygotsky's theory based approach for learning in MAS. *The AAAI-04 Workshop on Agent Organizations: Theory and Practice*, San Jose, California.
- [JOH96] B.E. John and D.E. Kieras. 1996. Using GOMS for user interface design and evaluation: Which technique?. *ACM Transactions on Computer-Human Interaction*, 3, pp. 287-319.
- [KAE96] L.P. Kaelbling, M.L. Littman and A.W. Moore. 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237-285.
- [LAR00] S. Larsson and D. Traum. 2000. Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. In *Natural Language Engineering*, 1:1-17.
- [LAV92] N. Lavrac and I. Mozetic. 1992. Second generation knowledge acquisition methods and their application to medicine. In Keravnou, E. (Ed.) *Deep Models for Medical Knowledge Engineering*, pages 177-198. Elsevier.
- [LEV00] E. Levin, R. Pieraccini, W. Eckert. 2000. A Stochastic Model of Human Machine Interaction for Learning Dialog Strategies. In *Proc. of the IEEE ICASSP*, Istanbul, Turkey, pp. 1883-1886.
- [LEV81] S. Levinson. 1981. Some pre-observations on the modelling of dialogue. *Discourse Processes*, 4(1):93-116.
- [LEV83] S. Levinson. 1983. *Pragmatics*. Cambridge University Press.
- [LEV97] E. Levin, R. Pieraccini and W. Eckert. 1997. A Stochastic Model of Computer-Human Interaction for Learning Dialog Strategies. In *Proc. of Eurospeech*, Rhodes, Greece, pp. 1883-1886.
- [LEV97b] E. Levin, R. Pieraccini and W. Eckert. 1997. Learning Dialogue Strategies within the Markov Decision Process Framework. In *Proc. of ASRU'97*, Santa Barbara, California.
- [LEW00] I. Lewin. 2000. A Formal Model of Conversational Game Theory. In *Gotalog: Proc. 4th Workshop on Semantics & Pragmatics of Dialogue*. Gothenburg, Sweden.
- [LIN01] B. Lin and L. Lee. 2001. Computer-Aided Analysis and Design for Spoken Dialogue Systems Based on Quantitative Simulations. In *Proc. of the IEEE Transactions on Speech and Audio Processing*, 9(5):534-548.
- [LIT00] D.J. Litman, S. Pan. 2000. Predicting and Adapting to Poor Speech Recognition in a Spoken Dialogue System. In *Proc. of 7th National Conference on AI (AAAI-2000)*, Austin, Texas.
- [LIT84] D.J. Litman, J. Allen. 1984. A Plan Recognition Model for Clarification Subdialogues. In *Proc. of Coling84*, pp. 302-311.
- [LIT85] D.J. Litman. 1985. Plan Recognition and Discourse Analysis : an Integrated Approach for Understanding Dialogues. Thèse de l'Université de Rochester.
- [LOP03] R. Lopez-Cozar, A. De la Torre, J.C. Segura and A.J. Rubio. 2003. Assessment of Dialogue Systems by Means of a New Simulation Technique. In *Speech Communication*, 40:387-407.
- [LUZ95] D. Luzzati. 1995. *Le dialogue verbal homme-machine*, Masson, Paris.
- [MCQ97] A.M. McQueen, A. Cutler. 1997. Cognitive Processes in Speech Perception. In Hardcastle. W. J. and Laver. J. (eds). *Handbook of Phonetic Sciences*. Cambridge/Oxford. Blackwell Publishers. pp. 566-585.
- [MEU05] J.G. Meunier. 2005. Module 1: La cognition. Note du cours de Doctorat en Informatique Cognitif, Département de Philosophie, UQAM. [page web]. www.unites.uqam.ca/philo/cours/DIC81004/textes/notes_cours1-06.pdf; [consulté le 10-10-2005].
- [MEU06] J.G. Meunier. 2006. Introduire à la philosophie du langage. Partie 1 : L'action langagière. Note du cours de philosophie PHI-1004. Département de Philosophie, UQAM.

- [MIN00] W. Minker and S. Bennacef. 2000. *Parole et dialogue homme-machine*. Paris, Edition d'Eyrolles, 212p.
- [MIN02] W. Minker. 2002. Overview on recent activities in speech understanding and dialogue systems evaluation. In *ICSLP-2002*, pp. 337-340.
- [MIN97] W. Minker, S. Bennacef and J.L. Gauvain. 1997. A Stochastic Case Frame for Natural Language Understanding. *Proc. of ICSLP'97*.
- [MIT97] T. Mitchel. 1997. *Machine Learning*. Mc Graw Hill.
- [MOE89] J. Moeschler. 1989. *Modélisation du Dialogue : Représentation de l'Inférence Argumentative*. Hermès, Paris.
- [MÖL04] S. Möller, J. Krebber, A. Raake and al. 2004. INSPIRE: Evaluation of a Smart-Home System for Infotainment Management and Device Control. In *Proc. 4th Int. Conf. on Language Resources and Evaluation (LREC 2004)*, P-Lisbon, 5:1603-1606.
- [MYE91] R. B. Myerson. 1991. *Game Theory: Analysis of Conflict*. Harvard University Press.
- [MYE98] B.A. Myers. 1998. A Brief History of Human Computer Interaction Technology. *ACM Interactions*. Vol. 5, no. 2, March, 1998. pp. 44-54.
- [NÉE99] F. Néel, and W. Minker. 1999. *Computational Models of Speech Pattern Processing*. Springer Verlag, Berlin/Heidelberg, Germany.
- [NER93] P. Nerzic. 1993. Erreurs et échecs dans le dialogue oral homme-machine, détection et réparation. Thèse de l'Université de Rennes I.
- [NGU04] H. Nguyen et J. Caelen. 2004. *Multi-session Management in Spoken Dialogue System*. C. Lemaître, C.A. Reyes, J.A. Gonzalez éditeurs, *Advances in Artificial Intelligence*, IBERAMIA 2004, Springer ed.
- [NGU05] H. Nguyen. 2005. Dialogue homme-machine : modélisation de multisession. Thèse de doctorat. Université Joseph Fourier - Grenoble I.
- [NGU06] H.P. Nguyen, C. Dan. 2006. A Basic Mathematical Framework for Conceptual Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 18(2):261-271.
- [NGU07a] M.Q. Nguyen, Philip H.P. Nguyen, X.F. Zhao, T.H Nguyen, J.G. Meunier, D. O'Shaughnessy. 2007. A Proposed AI Planning Approach for Staff Management in a Service Center. In *Proc. of the Fifth International Conference on Industrial Automation*. ETS, Montreal, Canada.
- [NGU07b] M.Q. Nguyen, Philip H.P. Nguyen, T.H Nguyen, J.G. Meunier, D. O'Shaughnessy. 2007. Apprentissage par renforcement à l'aide d'un utilisateur simulé pour optimiser automatiquement les stratégies de dialogue. Cinquième Conférence Internationale sur l'Automatisation Industrielle. ETS, Montréal, Canada.
- [OSH00] D. O'Shaughnessy. 2000. *Speech Communication: Human and Machine*. Addison-Wesley Publishing Co., Reading, MA, 1987; 2nd edition, 536 pages, IEEE Press, 568 pages.
- [OST89] N.D.M. Ostler. 1989. LOQUI: How Flexible can a formal prototype be?. In *M.M.Taylor, F. Neel & D.G. Bouwhuis (Eds.), The Structure of Multimodal Dialogue*, North-Holland:Netherlands.
- [PIE02] O. Pietquin and S. Renals. 2002. ASR System Modeling for Automatic Evaluation and Optimization of Dialogue Systems. In *Proc. of the IEEE ICASSP*, Orlando, USA, pp. 46-49.
- [PIE04] O. Pietquin. 2004. A Framework for Unsupervised Learning of Dialogue Strategies. Presses Universitaires de Louvain, *SIMILAR Collection*, ISBN 2-930344-63-6.
- [PIE05] O. Pietquin and R. Beaufort. 2005. Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning. *Proc. of Interspeech/Eurospeech*, Lisbon, Portugal.
- [PIE06] O. Pietquin and T. Dutoit. 2006. A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589-599.
- [PIER01] R. Pieraccini, S. Caskey, K. Dayanidhi, B. Carpenter, B. and M. Phillips. 2001. ETUDE, A Recursive Dialogue Manager With Embedded User Interface Patterns. In *Proc. of IEEE ASRU Workshop*, Madonna di Campiglio, Italy.
- [PUT94] M.L. Puterman. 1994. *Markov Decision Processes*. Wiley.
- [PRI90] P. Price. 1990. Evaluation of spoken language systems: The ATIS domain. In *Proc. 1990*

- DARPA Speech and Natural Language Workshop*, Morgan Kaufmann.
- [RAM89a] L.A. Ramshaw. 1989. A metaplan model for problem-solving discourse. In *Proc. of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 35–42, Manchester, England.
- [RAM89b] L.A. Ramshaw. 1989. Pragmatic Knowledge for Resolving Ill-Formedness. PhD thesis, University of Delaware, Newark, Delaware.
- [RAM91] L.A. Ramshaw. 1991. A three-level model for plan exploration. In *Proc. of the 29th ACL*, pp. 39–46, Berkeley, CA.
- [RAY05] M. Rayner, B. Hockey, J.M. Renders, N. Chatzichrisafis and K. Farrell. 2005. Spoken Language Processing in the Clarissa Procedure Browser. *International Computer Science Institute*, California.
- [REB02] J. Rebecca. 2002. How can a dialogue system compensate for speech recognition deficiencies?. Department of Linguistics, Goteborgs University. Available at : http://www.speech.kth.se/~rolf/gslt_papers/RebeccaJonson.pdf
- [RIC01] C. Rich, C.L. Sidner and N.B. Lesh. 2001. COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. *Artificial Intelligence Magazine*, 22(4):15-25.
- [RIC98] C. Rich and C.L. Sidner. 1998. COLLAGEN:A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3–4):315–350.
- [ROL05] C. Rolland. 2005. L'ingénierie des méthodes : une visite guidée. e-TI, Vol.01, <http://www.revue-eti.net/document.php?id=726>.
- [ROU00] J. Rouillard. 2000. Hyperdialogue sur Internet – Le système HALPIN. thèse de l'Université Joseph Fourier. CLIPS-IMAG.
- [ROU01] E. Roulet, L. Fillietaz, A. Grobet. 2001. Un modèle et un instrument d'analyse de l'organisation du discours. Berne, Peter Lang.
- [ROU81] E. Roulet. 1981. Echanges, interventions et actes de langage dans la structure de la conversation. *Etudes de linguistique appliquée*, 44:5-39.
- [RUB98] J. Rumbaugh, I. Jacobson, G. Booch. 1998. *The Unified Modeling Language Reference Manual*. Addison-Wesley.
- [RUS03] S. Russell, P. Norvig. 2003. *Artificial Intelligence: A modern approach*, Pearson Education, 2003, 2nd edition.
- [SCH01] K. Scheffler, S. Young. 2001. Corpus-Based Simulation for Automatic Strategy Learning and Evaluation. In *Workshop on Adaptation in Dialogue Systems (NAACL)*, Pittsburgh, Pennsylvania, USA.
- [SCH02] K. Scheffler, S. Young. 2002. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proc. Human Language Technology*, San Diego, 12-18.
- [SCH05] J. Schatzmann, K. Geogila, S. Young. 2005. Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In *Proc. of Workshop on Discourse and Dialogue*, Lisbon, Portugal.
- [SCH06] J. Schatzmann, K. Weilhammer, M. Stuttle, S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review*.
- [SCH99] K. Scheffler and S. Young. 1999. *Simulation of Human-Machine Dialogues* Cambridge, U.K.: Engineering Dept., Cambridge University, Tech. Rep. CUED/F-INFENG/TR 355, 1999.
- [SEA69] J.R. Searle. 1969. *Speech Acts : An Essay in the Philosophy of Language*. Cambridge University Press.
- [SEA80] J.R. Searle. 1980. Minds, brains and programs. *Behavioral and Brain Sciences*, 3:417-424.
- [SEA85] J.R. Searle and D. Vanderveken. 1985. *Foundations of Illocutionary Logic*. Cambridge : Cambridge University Press.
- [SIN02] S. Singh, D. Litman, M. Kearns and M. Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. *Journal of Artificial Intelligence Research*, 16:105–133.
- [SIN99] S. Singh, M. Kearns, D. Litman and M. Walker. 1999. Reinforcement Learning for

- Spoken Dialogue Systems. *Proc. of NIPS'99*, Denver, USA.
- [SOU00] B. Souvignier, A.Kellner, B. Rueber, H. Schramm and F. Seide. 2000. The thoughtful elephant - strategies for spoken dialog systems. In *IEEE Transactions on Speech and Audio Processing*, 8(1):51-62.
- [SPE95] D. Sperber, D. Wilson. 1995. *Relevance Communication and Cognition*. Second edition. Oxford.
- [STA06a] The Stanford University, 'Stanford Encyclopedia of Philosophy: SHRDLU', [page web]. <http://hci.stanford.edu/~winograd/shrdlu/>; [consulté 29-2-2006].
- [STA06b] The Stanford University, 'Stanford Encyclopedia of Philosophy.', [page web]. <http://plato.stanford.edu/entries/chinese-room/>; [consulté 29-2-2006]
- [STE99] A. Stent, J. Dowding, J.M. Gawron, E.O. Bratt and R. Moore. 1999. The CommandTalk spoken dialogue system. In *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 183-190.
- [SUT98] R. Sutton and A. Barto. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.
- [TAK05] k. Takadama and H. Fujita. 2005. Toward Guidelines for Modeling Learning Agents in Multiagent-Based Simulation: Implications from Q-Learning and Sarsa Agents. In *Multi-Agent and Multi-Agent-Based Simulation*, Springer Berlin / Heidelberg, ISBN 978-3-540-25262-7, 3415:159-172.
- [TAL99] M. Tallis, J. Kim, J. and Y. Gil. 1999. User studies of knowledge acquisition tools: Methodology and lessons learned. In *Proc. of KAW*.
- [THO04] C. Thompson, M. Goker, P. Langley. 2004. A Personalized System for Conversational Recommendations. In *Journal of Artificial Intelligence Research*, 21:1-36.
- [TUR50] A.M. Turing. 1950. Computing Machinery and Intelligence. *Mind* 49: 433-460. Site Internet <http://www.turing.org.uk/turing/> consulté mai-2005.
- [VAN91] D. Vanderveken. 1991. Meaning and Speech Acts. *Formal Semantics of Success and Satisfaction*, vol. 2, Cambridge: CUP.
- [VIL99] L. Villasenor-Pineda. 1999. Contribution à l'apprentissage dans le dialogue homme-machine. Thèse de doctorat, Université Joseph-Fourier - Grenoble I.
- [WAL00] M.A. Walker. 2000. An Application of Reinforcement Learning to Dialogue Strategy Selection in a Spoken Dialogue System for Email. In *Journal of Artificial Intelligence Research*, 12:387-416.
- [WAL97] M.A. Walker, D.J. Litman, C.A. Kamm and A. Abella. 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pp. 271-280, Madrid, Spain.
- [WAL98] M. Walker, J. Fromer and S. Narayanan. 1998. Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email. In *Proc. of ACL/COLING 98*.
- [WAT89] C. Watkins. 1989. Learning from delayed rewards. PhD Thesis, University of Cambridge, England.
- [WAT92] C. Watkins and P. Dayan. 1992. Technical note: Q-learning. *Machine Learning*, 8:279-292.
- [WIL04] J. D. Williams and S. Young. 2004. Characterizing Task-Oriented Dialog using a simulated ASR Channel. *Proc. of the ICSLP*.
- [WIN72] T. Winograd. 1972. *Understanding Natural Language*. Edinburgh, Edinburgh University Press.
- [WIT53] L. Wittgenstein. 1953. *Philosophical investigations*. (Anscombe, G.E.M., trans.). Oxford: Basil Blackwell.
- [YEH05] H. Ye, S. Young. 2005. Improving Speech Recognition Performance of Beginners in Spoken Conversational Interaction for Language Learning. *Proc. of Interspeech05*, Lisbon, Portugal.
- [ZHA07] X. Zhao, D. O'shaughnessy, M.Q. Nguyen. 2007. A Processing Method for Pitch Smoothing Based on Autocorrelation and Cepstral F0 Detection Approaches. *Proc. of ISSSE 2007*, pp.59-63, Montréal, Canada.
- [ZUE00] V.W. Zue, J.R. Glass. 2000. Conversational Interfaces: Advances and Challenges. In *Proc. of the IEEE. Special Issue on Spoken Language Processing*, 88(8).