Modèle de comportement communicatif conventionnel pour un agent en interaction avec des humains : approche par jeux de dialogue

Guillaume Dubuisson Duplessis

▶ **To cite this version:**

Guillaume Dubuisson Duplessis. Modèle de comportement communicatif conventionnel pour un agent en interaction avec des humains : approche par jeux de dialogue. Intelligence artificielle [cs.AI]. INSA de Rouen, 2014. Français. NNT : . tel-01017542v1

HAL Id: tel-01017542

https://theses.hal.science/tel-01017542v1

Submitted on 2 Jul 2014 (v1), last revised 8 Jul 2014 (v2)

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

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

THÈSE

Présentée à :
L'Institut National des Sciences Appliquées de Rouen

En vue de l'obtention du grade de :
Docteur en Informatique

Par :
Guillaume DUBUISSON DUPLESSIS

Intitulée :

Modèle de comportement communicatif conventionnel pour un agent en interaction avec des humains

Approche par jeux de dialogue

23 mai 2014

Devant le jury composé de :

Rapporteurs :

Frédéric Landragin	-	Chargé de Recherche CNRS, Laboratoire LATTICE
Nicolas Sabouret	-	Professeur, Université Paris-Sud

Examinateurs :

Catherine Pelachaud	-	Directrice de Recherche CNRS, TELECOM ParisTech
Jean-Christophe Routier	-	Professeur, Université Lille 1

Directeur :

Jean-Pierre Pécuchet	-	Professeur, INSA de Rouen

Encadrants scientifiques :

Nathalie Chaignaud	-	Maître de Conférences, INSA de Rouen
Jean-Philippe Kotowicz	-	Maître de Conférences, INSA de Rouen

Préambule

Résumé

Cette thèse a pour objectif l'amélioration des capacités communicatives des agents logiciels en interaction avec des humains. Dans ce but, nous proposons une méthodologie basée sur l'étude d'un corpus d'interactions Homme-Homme orientées vers la réalisation d'une tâche. Ce corpus est exploité en deux étapes essentielles : son enrichissement via une phase d'annotation et l'extraction de régularités. Nos travaux se concentrent sur les motifs d'interaction, c'est-à-dire les régularités observées au niveau de l'interaction dialogique. Nous étudions les structures de haut niveau spécifiables à partir de ces motifs dans le but de produire un modèle d'interaction pour un agent pouvant s'intégrer dans un processus délibératif. Pour modéliser de tels motifs dialogiques, nous proposons un cadre qui s'appuie sur les jeux de dialogue. Celui-ci se base sur les notions d'engagement social et de tableau de conversation qui permettent de s'abstraire des dispositions privées des agents prenant part à l'interaction. Nous illustrons la spécification de jeux de dialogue en appliquant l'ensemble des étapes de notre méthodologie à un corpus de dialogues orientés tâche. Les jeux spécifiés sont validés en montrant qu'ils décrivent de façon appropriée les motifs apparaissant dans le corpus de référence. Enfin, nous montrons l'intérêt interprétatif et génératif de notre modèle pour le fondement du comportement communicatif conventionnel d'un agent interagissant avec un humain. Nous implémentons ce modèle dans Dogma, un module exploitable par un agent afin de réguler son comportement communicatif dans un dialogue impliquant deux interlocuteurs.

Abstract

This research work aims at improving the communicative behaviour of software agents interacting with humans. To this purpose, we present a data-driven methodology based on the study of a task-oriented corpus consisting of Human-Human interactions. This corpus is processed in two main steps : its enrishment through an annotation phase and a regularity extraction phase. Our research work focuses on interaction patterns, that is to say regularities observed at the level of dialogical interaction. We study high-level structures that can be specified from these patterns. These structures form the basis of the interaction model for an agent that can be integrated into a deliberative process. We present a framework to specify dialogue games from these patterns based on the notion of social commitments and conversational gameboard. This framework is independent of the private states of interacting agents. We exemplify the specification of dialogue games by implementing all the steps of our methodology on a task-oriented corpus. The produced games are validated by showing that they appropriately describe the patterns appearing in a reference corpus. Eventually, we show that an agent can take advantage of our model to regulate its conventional communicative behaviour on both interpretative and generative levels. We implement this model into Dogma, a module that can be used by an agent to manage its communicative behaviour in a two-interlocutor dialogue.

Remerciements

Je tiens à remercier en premier lieu les rapporteurs de cette thèse, Frédéric Landragin et Nicolas Sabouret, pour l'intérêt porté à mes travaux et pour leurs remarques pertinentes et enrichissantes. Je remercie également Catherine Pelachaud pour m'avoir fait l'honneur de présider le jury et pour avoir éveillé ma curiosité sur la prise en compte de l'émotion dans la gestion du dialogue, ainsi que Jean-Christophe Routier pour avoir accepté d'examiner mon travail et pour avoir attiré mon regard sur les systèmes multiagent.

Je tiens à remercier mon directeur de thèse, Jean-Pierre Pécuchet, pour la confiance qu'il m'a toujours accordée et pour m'avoir soutenu à chaque sollicitation.

Un grand merci à Nathalie Chaignaud et Jean-Philippe Kotowicz, mes directeurs scientifiques qui m'ont suivi depuis mes débuts dans le département Génie Mathématique de l'INSA jusqu'à l'aboutissement de ce travail. Merci d'avoir toujours été là pour moi quand j'en avais besoin, et de m'avoir laissé une grande liberté dans mes travaux de recherche. Je tiens à souligner leur travail exemplaire de relecture et de correction aussi bien pour le manuscrit que pour la soutenance.

Je souhaite tout particulièrement associer Alexandre Pauchet à mon encadrement scientifique. Merci pour ton dynamisme, ta curiosité, ton regard critique toujours pertinent, les discussions diverses et variées, les relectures, les corrections, et j'en passe ! Tu as permis de faire avancer et aboutir ces travaux.

Merci à Philippe Mathieu de m'avoir accueilli au sein de l'équipe SMAC du Laboratoire d'Informatique Fondamentale de Lille (LIFL) en tant qu'ATER pendant cette quatrième année très enrichissante. J'en profite pour remercier tous les membres de cette équipe de recherche d'une qualité et d'un dynamisme exemplaire. Un grand merci à Maxime Morge pour son intérêt pour mon travail, pour ses remarques pertinentes et pour le concert d'Oldelaf !

Merci à Brigitte, Jean-François et Sandra pour leur gentillesse, leur compétence et l'aide qu'ils m'ont accordée tout au long de ce périple au LITIS.

Merci à Nicolas Delestre, Nicolas Malandain, Romain Hérault et Laurent Vercouter pour les discussions enrichissantes à propos de l'enseignement et de la recherche.

Merci à l'équipe ASI du midi avec qui j'ai passé des moments conviviaux, très plaisants et remplis de trolls : Alain, Gilles, Nico D., Nico M., Pierrick, Romain, Elsa, ...

Merci aux doctorants croisés tout au long de la thèse pour leur sympathie : Émilien, Amandine, Florian, Louise, Amnir, Yadu, Damien, et les lillois Lisa et Fabien.

Merci à mes amis dont la présence compte beaucoup pour moi : Clément, Aurélie C., Nico, Philippine, Arnaud, Emma, Zach, Diana, Polo, Cocaux, Vincent, Claire, Carlo, André, Julien, Aurélie B., Rémi F., Ovidiu, Alina, Joseph, ... Merci aux points de suspension qui représentent les amis que j'ai oublié de citer et qui me pardonneront.

Merci à ma famille pour son soutien permanent : maman, papa, Aurélien, Justine, Romain, et Paul ! Merci enfin à Virginie avec qui je partage ma vie et qui m'a soutenu dans cet accomplissement.

Table des matières

Introduction 1

I Positionnement 9

1 Fondements pour la modélisation du dialogue 11
1.1 Généralités sur le dialogue . . . 11
1.1.1 Le dialogue est une activité conjointe et opportuniste . . . 11
1.1.2 Gestion de la tâche sous-jacente et gestion de la communication . . . 14
1.2 Actes de langage et actes de dialogue . . . 16
1.2.1 Actes de langage . . . 17
1.2.2 Des actes de langage aux actes de dialogue . . . 19
1.3 Synthèse . . . 27

2 Modélisation du dialogue : intention versus convention 29
2.1 Approches intentionnelles . . . 29
2.1.1 Fondements des approches intentionnelles . . . 30
2.1.2 Approches par planification . . . 32
2.1.3 Principales mises en œuvre . . . 36
2.1.4 Avantages et limites des approches intentionnelles . . . 38
2.2 Approches conventionnelles . . . 39
2.2.1 Fondements des approches conventionnelles . . . 39
2.2.2 Modèles fondés sur la structuration . . . 42
2.2.3 Modèles fondés sur les questions en discussion . . . 46
2.2.4 Avantages et limites des approches conventionnelles . . . 52
2.3 Discussion . . . 54

3 Vers des modèles mixtes à base de jeux de dialogue 57
3.1 Fondements des jeux de dialogue . . . 57
3.1.1 Les jeux de dialogue comme scripts partagés . . . 58
3.1.2 Les jeux de dialogue comme recettes partagées . . . 60
3.1.3 Les jeux de dialogue comme projet conjoint . . . 62
3.1.4 Synthèse sur la métaphore des jeux de dialogue . . . 64
3.2 Les jeux de dialogue pour l'interaction Homme-Machine . . . 64
3.2.1 Les jeux comme réseau de transition récursif . . . 65
3.2.2 Les jeux comme unités de type initiative-réponse cohérentes . . . 66
3.2.3 Les jeux comme structures pour l'engagement dans le dialogue . . . 69
3.3 Discussion . . . 74
3.3.1 Vers un modèle normatif des jeux de dialogue basé sur les engagements . . . 74

3.3.2 Avantages et limites des jeux de dialogue . . . 78
3.3.3 Vers des modèles mixtes fondés sur des jeux de dialogue . . . 81

II Analyse des interactions humaines 83

4 Corpus Cogni-CISMeF 85
4.1 Constitution du corpus . . . 85
4.1.1 Objectifs . . . 86
4.1.2 Recueil du corpus . . . 87
4.1.3 Transcription . . . 87
4.2 Processus de recherche d'information . . . 88
4.2.1 Modèle standard . . . 88
4.2.2 Un processus opportuniste . . . 88
4.2.3 Un processus stratégique . . . 89
4.3 Analyse des entretiens du corpus . . . 90
4.3.1 Situation dialogique . . . 90
4.3.2 Structure globale des entretiens . . . 91
4.3.3 Stratégies et tactiques . . . 94
4.3.4 Rôles des participants . . . 99
4.4 Synthèse . . . 103

5 Annotation du corpus et extraction de motifs 105
5.1 Processus d'annotation . . . 106
5.1.1 Segmentation du dialogue . . . 106
5.1.2 Configuration du processus d'annotation . . . 108
5.1.3 Résultats du processus d'annotation . . . 112
5.1.4 Synthèse . . . 125
5.2 Processus d'extraction de motifs dialogiques . . . 126
5.2.1 Configuration du processus d'extraction . . . 127
5.2.2 Résultats du processus d'extraction . . . 128
5.2.3 Synthèse . . . 141
5.3 Discussion . . . 142

III Modélisation des interactions humaines et mise en œuvre 143

6 Formalisation des motifs d'interaction 145
6.1 Modèle de l'engagement social et tableau de conversation . . . 145
6.1.1 Modèle de l'engagement social . . . 146
6.1.2 Tableau de conversation . . . 149
6.1.3 Interprétation des engagements du tableau de conversation . . . 156
6.2 Jeux de dialogue et jeux de communication . . . 158
6.2.1 Modèle de jeu de dialogue . . . 158
6.2.2 Modèle de jeu de communication . . . 161
6.2.3 Combinaisons et établissement de jeux de dialogue . . . 161
6.2.4 Interprétation des jeux de dialogue dans le tableau de conversation . . . 163

6.3 Spécification empirique de jeux depuis le corpus COGNI-CISMEF . . . 163
6.3.1 Représentation du contenu sémantique . . . 164
6.3.2 Actes de dialogue . . . 168
6.3.3 Jeu de communication et jeux de dialogue . . . 169
6.3.4 Couverture du corpus par les jeux définis . . . 176
6.4 Discussion . . . 179

7 Mise en œuvre du modèle 183
7.1 DOGMA : « DialOgue Game MAnager » . . . 183
7.1.1 Architecture d'un système de dialogue Homme-Machine . . . 184
7.1.2 Exploitation des jeux de dialogue par un gestionnaire de dialogue . . . 184
7.1.3 Présentation de DOGMA . . . 185
7.1.4 Implémentation . . . 188
7.2 Traces de fonctionnement de DOGMA . . . 189
7.2.1 Tableau de conversation . . . 189
7.2.2 Exemple de jeu de requête . . . 189
7.2.3 Exemple de jeu de question à choix multiples . . . 194
7.2.4 Exemple d'emboîtement . . . 201
7.3 Vers une validation de DOGMA . . . 205
7.3.1 Test « à la Turing » . . . 206
7.3.2 Développement de systèmes de dialogue . . . 207
7.4 Discussion : vers un agent dialogique utilisant les jeux . . . 207

Conclusion 211

Annexes 215

A Annexes de l'analyse du corpus Cogni-CISMeF 217
A.1 Le schéma XML d'annotation pour un acte de dialogue dans Gate . . . 217
A.2 Alignement entre la taxonomie d'actes de dialogue de Loisel et DIT++ . . . 221
A.3 Annexes des résultats du processus d'annotation . . . 223
A.3.1 Exemples de dialogues annotés du corpus COGNI-CISMEF . . . 223
A.3.2 Proportion des tours de parole annotés . . . 229
A.3.3 Accords inter-annotateur . . . 229
A.3.4 Analyse au niveau des segments fonctionnels . . . 233
A.3.5 Analyse au niveau des dimensions . . . 234
A.3.6 Co-occurrence des dimensions . . . 236
A.3.7 Analyse au niveau des fonctions communicatives . . . 237
A.4 Annexes du processus d'extraction . . . 247
A.4.1 Répartition entre le corpus d'extraction et le corpus de référence . . . 247
A.4.2 Cas de la fonction $ExecNegativeAutoFB$ . . . 248
A.4.3 Présence des motifs dans les entretiens du corpus d'extraction . . . 248
A.4.4 Exemples supplémentaires d'instance de motif . . . 250

B Annexes du modèle **261**
B.1 Événéments internes issus des jeux de dialogue 261
B.2 Jeu de communication de contextualisation 261
B.3 Jeu de dialogue de question à choix multiples 261

C Annexes de la mise en œuvre **265**
C.1 Exemples de code 265
C.2 Traces de fonctionnement du système 268
C.2.1 Refus d'entrée dans le jeu 268
C.2.2 Échec d'un jeu de question/réponse 269
C.2.3 Jeu d'évaluation : exemple de correction 273

Liste des dialogues **275**

Liste des codes sources **278**

Liste des algorithmes **280**

Bibliographie **283**

Liste des tableaux

1 Organisation préférentielle des paires adjacentes 5

1.1 Niveaux de la gestion de l'interaction selon Clark 15
1.2 Actes conversationnels multi-niveaux 22

2.1 Exemple de représentation d'un acte "Informer" comme une action de plan 33
2.2 Exemple de spécification d'une tâche dialogique de salutations dans DTask . . . 37
2.3 Gestion de feedbacks du système dans GoDiS 52

3.1 Exemple de jeu de dialogue d'aide dans l'approche de Levin et Moore 59
3.2 Exemples de jeux de dialogue dans l'approche de Mann 61
3.3 Exemple de macrojeu de présentation d'information dans l'approche de Mann . . 62
3.4 Exemple du jeu de dialogue d'explication dans l'approche de Maudet 72
3.5 Jeu de communication d'évaluation dans l'approche de Maudet 73

4.1 Conventions de transcription du corpus Cogni-CISMeF 88
4.2 Occurrences des différents types de verbalisation dans le corpus 93
4.3 Répartition des tactiques de réparation dans le corpus 100

5.1 Segmentation et multifonctionnalité 107
5.2 Répartition des dialogues par annotateur dans le processus d'annotation 110
5.3 Nombre de fonctions communicatives annotées par corpus et par annotateur . . . 112
5.4 Proportion des tours de parole annotés dans le corpus 113
5.5 Exemples de fonctions communicatives par dimension associées à un exemple typique 114
5.6 Accord inter-annotateur pour la tâche de segmentation 115
5.7 Accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension 115
5.8 Accord inter-annotateur pour les tâches de segmentation et d'étiquetage pour les fonctions générales 117
5.9 Nombre de fonctions par segment fonctionnel par annotateur, par corpus et au total 117
5.10 Répartition des segments fonctionnels en fonction du rôle 118
5.11 Nombre d'actes de dialogue par tour en fonction du rôle 118
5.12 Proportions de fonctions annotées par dimension sur l'ensemble du corpus Cogni-CISMeF 119
5.13 Proportion de fonctions annotées par catégories 121
5.14 Répartition des fonctions dans la dimension *Task* par catégorie pour l'expert et le demandeur. 121

5.15 Répartition des fonctions dans la dimension *Task* pour la catégorie des fonctions fournissant de l'information . . . 122
5.16 Répartition des fonctions dans la dimension *Task* pour la catégorie des fonctions de discussion d'actions . . . 123
5.17 Répartition des fonctions dans la dimension *Task* pour la catégorie des fonctions de recherche d'information . . . 124
5.18 Synthèse des fonctions annotées dans la dimension *Auto-feedback* . . . 125
5.19 Extrait des motifs dialogiques de vérification (catégorie de recherche d'information) 131
5.20 Motifs dialogiques de question ouverte et de question à choix multiples . . . 134
5.21 Motifs dialogiques de discussion d'action . . . 135
5.22 Motif dialogique d'accord (catégorie de transfert d'information) . . . 138

6.1 Événements et combinaisons d'événements pour les engagements dialogiques en action . . . 148
6.2 Fonctionnalités générales du tableau de conversation . . . 150
6.3 Opérations générales sur les engagements en action et propositionnel (création, suppression et annulation) . . . 151
6.4 Opérations sur les engagements en action (contextualisés ou non) . . . 151
6.5 Prédicats concernant les descriptions d'événement . . . 153
6.6 Relations entre un engagement en action dialogique et l'occurrence d'un événement 154
6.7 Jeu de dialogue de requête . . . 159
6.8 Opérations générales sur les engagements conjoints sur un jeu de dialogue . . . 160
6.9 Fonctions des actes de dialogue de contextualisation . . . 162
6.10 Exemples de propositions dans notre représentation sémantique . . . 166
6.11 Exemples de question dans notre représentation sémantique . . . 167
6.12 Relations de résolution et de pertinence . . . 167
6.13 Fonctions des actes de dialogue considérés dans notre formalisation . . . 168
6.14 Type de contenu sémantique des actes de dialogue . . . 169
6.15 Jeu de communication d'évaluation : effets directs . . . 170
6.16 Jeu de communication d'évaluation . . . 171
6.17 Jeu de communication d'évaluation : cas de la correction . . . 171
6.18 Conditions d'entrée, de succès et d'échec des jeux de vérification, de vérification négative et de question oui/non . . . 171
6.19 Jeu de dialogue de vérification . . . 172
6.20 Jeu de dialogue de vérification négative . . . 172
6.21 Jeu de dialogue d'interrogation polaire . . . 173
6.22 Jeu de dialogue d'interrogation ouverte . . . 174
6.23 Jeu de dialogue de suggestion . . . 175
6.24 Jeu de dialogue d'offre . . . 175
6.25 Jeux utilisables par un interlocuteur afin de modifier les états des engagements extra-dialogiques . . . 176
6.26 Résultats de l'étude de la couverture des jeux . . . 178

Table des figures

1 Vision schématique des usages d'un corpus de traces enrichies pour la conception d'un agent interactif. . . . 3
2 Étapes de notre méthodologie basée sur un corpus 4

1.1 Fonctions générales de DIT++ 25
1.2 Extraits des fonctions spécifiques à certaines dimensions de DIT++ 26

2.1 Automate de l'initiateur pour la demande d'information dans l'approche de Pauchet 45

3.1 Jeu de dialogue de demande d'information dans l'approche de Lewin 65
3.2 « Initiative-response units » 67
3.3 Structure générale d'un jeu de dialogue dans l'approche d'Hulstijn 67
3.4 Échange de recherche d'information dans l'approche d'Hulstijn 68
3.5 Séquencement de jeux dans l'approche d'Hulstijn 68
3.6 Exemple de séquencement de jeux de recherche d'information dans l'approche d'Hulstijn 68

4.1 Récolte et transcription de corpus dans notre méthodologie 85
4.2 Situation dialogique entre l'expert, l'utilisateur et CISMeF dans l'expérimentation 91
4.3 Structure des dialogues de recherche d'information collaborative observée 91
4.4 Enchaînement idéalisé des tactiques de réparation en fonction des résultats de la requête 99

5.1 Annotation du corpus Cogni-CISMeF et extraction de régularités 105
5.2 Répartition observée des instances de motifs dialogiques par catégorie dans le corpus d'extraction 129
5.3 Répartition observée des instances de motifs dialogiques de recherche d'information dans le corpus Cogni-CISMeF 131
5.4 Répartition observée des instances de motifs dialogiques de discussion d'action dans le corpus Cogni-CISMeF 135
5.5 Répartition observée des motifs dialogiques de transfert d'information dans le corpus Cogni-CISMeF 138
5.6 Motif dialogique de correction 139

6.1 États d'un engagement propositionnel 146
6.2 États d'un engagement en action. 147
6.3 États d'un engagement conjoint sur un jeu de dialogue 148
6.4 Principe de l'évolution du tableau de conversation en fonction de l'occurrence d'événements 149
6.5 Principe du jeu de communication de contextualisation pour un jeu de dialogue . 163

6.6 Structure sous-jacente commune aux jeux de dialogue d'interrogation ouverte et d'interrogation à choix multiples 173

7.1 Architecture de DOGMA 187

Introduction

Des agents en interaction avec les humains, limités par leurs capacités dialogiques

Les interactions entre des humains et des agents logiciels sont de plus en plus répandues, bénéficiant du rapprochement entre l'Interaction Homme-Machine (IHM) et l'Intelligence Artificielle (IA) [Horvitz 1999a, Horvitz 2007, Lieberman 2009]. L'éventail des agents visant à offrir des capacités d'interaction proche des capacités humaines est important. Nous pouvons citer les agents interface [Lieberman 1997], les agents virtuels intelligents [Swartout 2006], les agents conversationnels animés [Cassell 2000] et les agents assistants d'initiative mixte [Tecuci 2007] [1]. Dans ce document, nous parlons plus généralement d'*agent interactif* pour désigner un agent destiné à interagir avec un humain.

La conception d'agent interactif implique l'intégration de nombreux domaines de l'IA [Swartout 2006, Tecuci 2007] incluant la représentation des connaissances, la résolution de problème et la planification, l'acquisition de connaissances et l'apprentissage, les systèmes multiagent, la théorie du discours et l'interaction Homme-Machine. Sans surprise, la conception d'agents en interaction directe avec les humains est une tâche réputée difficile (voir par exemple les problématiques soulevées par l'interaction d'initiative mixte [Tecuci 2007] ou encore les environnements mixtes [Swartout 2006]). Les humains utilisent des processus de communication et de raisonnement complexes auxquels les agents doivent s'adapter. Un défi dans la conception des interactions Homme-Machine se situe en particulier au niveau de la *problématique de la communication*. La gestion du dialogue est vue comme un point clé car c'est un moyen de communication efficace pour les humains qui requiert peu ou pas d'entraînement pour être utilisé [Tecuci 2007]. De plus, c'est le moyen le plus à même de permettre la réalisation d'un véritable système collaboratif d'initiative mixte [Ferguson 2007].

La gestion du dialogue reste cependant un point délicat pour les Agents Conversationnels Animés (ACA) [Swartout 2006, Ales 2012]. Ainsi, la plupart des systèmes existants n'intègrent que des modules de gestion de dialogue basiques (e.g., voir le projet SEMAINE [Schroder 2010]). Une première approche consiste à rechercher des motifs textuels dans les énoncés de l'utilisateur vus comme une séquence de mots (e.g., le célèbre ELIZA [Weizenbaum 1966] ou l'« Artificial Intelligence Markup Language » (AIML) [Wallace 2003]). Des *règles* spécifient le motif textuel recherché et la réponse à produire par le système. Elles permettent au système de produire sa réponse en confrontant les règles à l'énoncé de l'utilisateur. Ce type d'approche ne donne que l'illusion d'un dialogue cohérent. La deuxième approche est particulièrement répandue (voir, e.g., le cas des systèmes de dialogue dans le domaine médical [Bickmore 2006]). Elle consiste à représenter la structure du dialogue via un automate fini (éventuellement hiérarchique) où chaque état représente l'état du dialogue et chaque transition, étiquetée par un énoncé, conduit à un nouvel

1. Respectivement, traductions de : « Autonomous Interface Agent », « Intelligent Virtual Agents » (IVAs), « Embodied Conversational Agents » (ECAs) et « Mixed-Initiative Assisting Agent » (MIAA)

état (voir, e.g., [McTear 2004]). Cette approche décrit la structure de l'ensemble du dialogue. En pratique, elle s'avère rigide et limitée à des dialogues menés par le système. Enfin, l'approche « par cadre » [2] voit le dialogue comme un processus de renseignement d'un formulaire contenant un ensemble d'emplacements (cf., e.g., [Aust 1995] et le standard VoiceXML [Oshry 2007]). Les emplacements correspondent généralement à des informations nécessaires au système. La rigidité du modèle est réduite par l'*algorithme de contrôle* qui est chargé de déterminer les interventions du système sur la base du cadre. Par exemple, l'utilisateur peut remplir plusieurs emplacements avec un seul énoncé (ce qui était impossible avec l'approche précédente). Néanmoins, toutes ces approches se limitent aux dialogues menés par le système et dont les contributions sont fixées par avance. Elles ne laissent que peu de place à l'initiative de l'utilisateur.

Une méthodologie pour concevoir le comportement des agents interactifs

Notre point de vue sur la modélisation des interactions Homme-Machine se fonde sur le postulat que la conception d'agents en interaction avec les humains peut être améliorée par l'étude et l'analyse des interactions Homme-Homme et Homme-Machine. Dans ce but, nous adoptons une démarche interdisciplinaire mêlant psychologie cognitive et intelligence artificielle [Chaignaud 1996, Pauchet 2006].

La modélisation de la cognition humaine passe selon nous par l'étude de corpus de traces Homme-Homme [Chaignaud 1996, Pauchet 2006]. Dans cette perspective, nous laissons de côté les approches qui visent à travailler sur des exemples artificiellement créés (comme les « donkey sentences » [Geach 1962] ou bien des dialogues artificiels). Cette décision est motivée, d'une part, par le fait que l'accord sur l'acceptabilité des exemples créés est difficile à obtenir. D'autre part, ce genre d'approches conduit à la modélisation de phénomènes sans se préoccuper de savoir s'ils apparaissent fréquemment dans l'usage. Au contraire, une approche par corpus permet de se concentrer sur des phénomènes *observables*, *attestés* et *quantifiables*. De plus, l'accord est plus facile à obtenir que sur des exemples artificiels. Néanmoins, il est nécessaire d'être conscient des limites des approches par corpus. Premièrement, l'étude d'un corpus résulte d'une construction élaborée dans un cadre précis. En tant que telle, la généricité des résultats obtenus est à démontrer. Ensuite, un corpus ne fait pas apparaître de manière exhaustive l'ensemble des phénomènes que l'on cherche à étudier. L'étude de corpus est une *approche inductive* qui va de pair avec l'intervention de connaissances extérieures. Enfin, il faut bien garder à l'esprit qu'un corpus est une source de connaissances à partir de laquelle il est possible d'élaborer des modèles. Cependant, la quantité de phénomènes qui y occurrent est souvent loin de la quantité des phénomènes *in fine* modélisés. En ce sens, et en particulier dans le cas du dialogue, il est pour le moment illusoire de penser pouvoir modéliser l'ensemble des phénomènes observés car ils impliquent les champs de la *pragmatique* et de la *sémantique* dont les frontières restent mal connues [Maudet 2003b, Prévot 2004, Ginzburg 2010, Ginzburg 2012].

Nous prescrivons néanmoins l'usage de corpus de véritables dialogues Homme-Homme pour la modélisation des interactions Homme-Machine. La principale motivation est que les interactions Homme-Homme sont riches et naturelles. En conséquence, leur étude permet d'améliorer les systèmes d'interaction Homme-Machine existants afin de tendre vers une interaction plus fluide et plus naturelle. De plus, un certain nombre d'études met en évidence l'utilité de corpus Homme-

2. Traduction de « Frame-based approach »

Homme pour la modélisation des interactions Homme-Machine [Bilange 1991a, Pauchet 2006, Loisel 2008, Orkin 2013].

Notre méthodologie s'appuie sur un corpus de *traces Homme-Homme*. Ces traces sont récoltées suite à une expérimentation conçue spécifiquement et mettant en interaction des *participants*. Elles incluent généralement l'enregistrement [Chaignaud 2000, Pauchet 2006, Orkin 2013] (i) des *actions physiques* liées à la tâche (e.g., ajout d'un mot clé à une requête d'un moteur de recherche, envoi de la requête, etc.), (ii) des *interactions dialogiques* entre les participants. Schématiquement (cf. figure 1), il est possible d'analyser sur deux niveaux ce corpus pour la

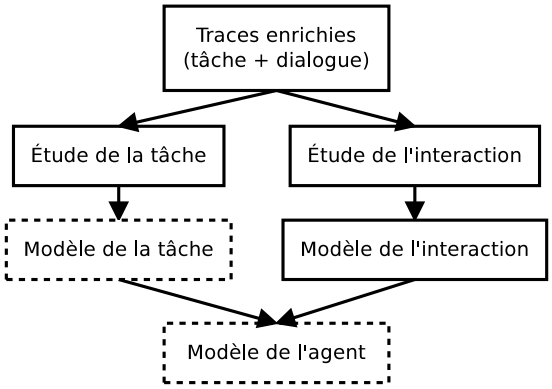

FIGURE 1: Vision schématique des usages d'un corpus de traces enrichies pour la conception d'un agent interactif. Les cadres en trait plein représentent les points abordés dans cette thèse. Les cadres en pointillé représentent les suites potentielles de cette thèse.

production du modèle d'un agent prenant le rôle d'un des participants [Chaignaud 2000, Pauchet 2006, Orkin 2013] : au niveau de la tâche (e.g., en terme de planification) et au niveau des interactions entre les participants. Ces analyses se focalisent sur la recherche de *régularités* dans les traces sous forme de *motifs récurrents* au niveau de la tâche et au niveau de l'interaction dialogique [Chaignaud 2000, Pauchet 2006, Orkin 2013].

L'approche « Collective Artificial Intelligence » [Orkin 2009] est proche de notre démarche aussi bien au niveau méthodologique (récolte de traces Homme-Homme et étude des *motifs récurrents*) que par l'objectif d'amélioration du comportement des agents interagissant avec des humains. Elle se concentre sur les agents interagissant avec des humains dans des environnements virtuels (EV) comme un jeu vidéo. Leur approche permet d'apprendre automatiquement des scripts [Schank 1977] (connaissances socialement établies) sans intervention humaine depuis un grand nombre de logs d'interaction Homme-Homme incluant des actions physiques dans un EV et des énoncés saisis via un chat. Cette approche n'implique pas de structures de haut niveau (e.g., des buts et sous-buts). Les *plans d'actions* automatiquement appris sont définis en terme de séquences d'actions issues des logs. De même pour les *instances de dialogue* qui sont définies en terme de séquence de motifs textuels de surface. Plans d'action et instances de dialogue sont stockés dans une *mémoire collective*, qui est exploitée par une architecture d'agent appelée *proposition-critique* [Orkin 2009]. Dans cette architecture, l'agent réévalue les plans disponibles dans la mémoire collective dès qu'il détecte une anomalie dans son plan courant (partie proposition) pour sélectionner le plan qui dévie le moins de la norme (partie critique). La conclusion de cette approche après expérimentation est que des agents basés sur ces structures de bas niveau permettent d'assurer une *faible cohérence locale* au niveau de l'interaction mais pas de *cohérence globale* [Orkin 2009]. Cette étude montre qu'il est nécessaire de considérer des structures de haut

niveau pour la tâche (e.g., hiérarchie de buts) et le dialogue (e.g., formaliser les énoncés en terme de types de phrase et de contenu sémantique) pour atteindre une cohérence globale. *Il est ainsi nécessaire d'exploiter un corpus via des structures de haut niveau intégrables dans le processus délibératif d'un agent interactif.* Contrairement à leur objectif initial, elle montre la nécessité d'intégrer l'humain dans le processus global afin d'enrichir les annotations avec de nouvelles connaissances au niveau de la tâche et du dialogue. L'intervention humaine peut être *directe* de façon *manuelle* ou *semi-automatique* (e.g., [Orkin 2010, Ales 2012]). Elle peut également être *indirecte* via l'usage d'outils d'annotation automatique entraînés sur des annotations effectuées par des humains (e.g., [Stolcke 2000] en annotation automatique d'actes de dialogue).

À la lumière de ces résultats, notre méthodologie vise à représenter un corpus de traces sous *forme matricielle* via une étape d'enrichissement du corpus réalisée par intervention humaine directe ou indirecte. Cette forme matricielle permet l'extraction de *motifs récurrents* définis comme une séquence d'annotations dont l'arrangement se produit dans plusieurs traces [Ales 2012]. Ces régularités doivent ensuite être exploitées pour spécifier des structures de plus haut niveau.

Ce document se focalise sur les interactions dialogiques entre les participants et vise la production d'un modèle d'interaction pour un agent interactif capable de s'intégrer dans un processus délibératif plus général intégrant le niveau de la tâche (représenté par les traits pleins sur la figure 1). L'application de la méthodologie dans cette thèse se décompose en étapes présentées en figure 2 [Ales 2012]. Elle est fondée sur la *collecte* d'un corpus de dialogues à partir

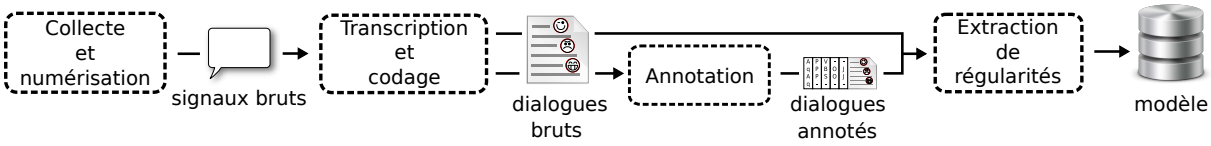

FIGURE 2: Étapes de notre méthodologie basée sur un corpus de dialogues [Ales 2012]

d'une expérimentation impliquant des utilisateurs visés par le futur système. La collecte peut être effectuée en format audio, vidéo ou directement textuel (e.g., via un chat). Puisque notre méthodologie est *ascendante*, les expérimentations doivent dépendre de la tâche visée par le futur agent. Nous encourageons l'usage de corpus de *dialogues finalisés* (i.e., orientés vers la réalisation d'une tâche précise). L'étape de *transcription* permet de transcoder les données brutes dans un format exploitable pour l'analyse. Elle permet de conserver un certain niveau de détails (énoncé, prosodie, pauses, etc.). Les caractéristiques à conserver dépendent des phénomènes que l'agent doit exhiber. L'étape d'*annotation* est l'étape d'enrichissement du corpus réalisée de manière directe ou indirecte par intervention humaine. Elle permet d'ajouter des connaissances en fonction de critères définis par un schéma d'annotation prédéfini (e.g., actes de dialogue, émotion, etc.). Cette étape inclut généralement une phase de *segmentation* visant à décomposer le corpus en unités définies par le schéma d'annotation choisi. À l'issue de cette étape, une représentation multidimensionnelle de chaque unité d'interaction est obtenue, conduisant à l'obtention d'une *représentation matricielle* du corpus. Les *motifs d'interaction* sont extraits par la suite, et constituent la base du *modèle d'interaction* de l'agent. Compte tenu de notre représentation matricielle, un *motif dialogique* est défini comme une séquence d'annotations dont l'arrangement se produit dans plusieurs dialogues.

Motifs d'interaction et structure du dialogue

Notre étude se concentre sur l'extraction et la modélisation de *motifs récurrents* dans les traces d'interaction dialogique appelés *motif d'interaction* ou *motif dialogique*. Allwood constate trois types de régularités dans le dialogue [Allwood 1994] : (i) celles qui dépendent des relations au sein d'un énoncé entre ses différentes parties, (ii) celles qui dépendent des relations entre énoncés, (iii) et celles qui dépendent des relations entre des facteurs globaux et les énoncés. Ici, énoncé est à prendre au sens d'*unité de conversation*. Les régularités qui nous intéressent sont celles qui dépendent des relations entre les énoncés. Autrement dit, un *motif dialogique* est une séquence ordonnée d'énoncés qui réapparaît fréquemment dans plusieurs dialogues (par exemple, une paire question/réponse) [Hulstijn 2000a].

L'étude des corpus de dialogues Homme-Homme fait apparaître un certain nombre de motifs dialogiques identifiables par des caractéristiques de surface des énoncés (voir, e.g., [Levin 1977, Mann 1988, Hulstijn 2000b, Pauchet 2006]). Parmi ces études, les plus influentes proviennent du *champ de l'analyse de la conversation* [Schegloff 1973, Sacks 1974]. Ce dernier se concentre sur l'étude de corpus de conversations retranscrites, sans a priori théorique tel que des schémas d'annotation. Le but est de produire une analyse objective issue de l'*observation* de régularités dans les corpus. L'analyse conversationnelle a notamment mis en avant la notion de *tour de parole* et a décrit précisément le processus de *gestion de tour de parole* entre les interlocuteurs. Ces travaux sont généralement utilisés comme point de départ dans les systèmes de dialogue Homme-Machine (voir, e.g., [Kronlid 2008]). Ce champ a également montré que les énoncés ont tendance à intervenir sous forme de *paires* dans une conversation : les *paires adjacentes* [Schegloff 1973]. L'exemple classique est la paire question/réponse. Selon [Schegloff 1973], les paires adjacentes ont les propriétés suivantes : (i) Elles consistent en deux énoncés ordonnés : la *première partie* et la *seconde partie*. (ii) Les deux parties sont produites par des locuteurs différents. (iii) Chaque partie est typée de manière à spécifier quelle partie vient en premier et laquelle en second. (iv) La forme et le contenu de la seconde partie dépendent du type de la première partie. (v) Étant donnée la première partie, la seconde partie de la paire est *conditionnellement pertinente* – i.e., pertinente et prévisible – en tant qu'énoncé suivant. Ces observations ont conduit certains chercheurs à proposer l'idée plus large d'*organisation préférentielle* [Levinson 1983]. L'idée est que les secondes parties des paires adjacentes peuvent être classées selon un ordre préférentiel. Cet ordre est lié au caractère habituel de l'énoncé. Publiquement, les initiateurs d'une paire préfèrent son accomplissement à sa modification, sa modification à son refus et son refus à sa non-considération. Le tableau 1 présente quelques exemples d'organisation préférentielle de paires adjacentes.

Première partie	Question	Requête	Offre
Seconde partie			
. . . préférée	Réponse attendue	Acceptation	Acceptation
. . . non-préférée	Réponse inattendue	Refus	Refus

Tableau 1: Organisation préférentielle des paires adjacentes

Le champ de l'analyse de la conversation est fécond. Il a produit de nombreux résultats dont la mise en avant de la notion de *paire adjacente*, le motif d'interaction dialogique minimal. Il a également montré la structuration de la conversation sous forme de combinaisons de paires adjacentes (pré-séquence et emboîtement). Cette approche est descriptive et l'utilisation directe

de ces résultats est difficile. Elle montre néanmoins que le dialogue est un *objet structuré* sans imposer pour autant une structure rigide. L'idée pour un modèle de dialogue est de produire des structures qui vont *expliquer* l'émergence de structures telles que celles mises en avant par l'analyse de la conversation. *Localement*, le dialogue est structuré au niveau des paires adjacentes. Au *niveau intermédiaire*, le dialogue est structuré par les combinaisons de séquences. *Globalement*, le dialogue est structuré par l'agencement des séquences (e.g., salutations/réalisation de la tâche/clôture).

Notre problématique

Comme nous l'avons déjà dit, cette thèse vise à observer et modéliser les *motifs d'interaction* dans l'objectif de leur intégration dans une architecture d'agent délibératif interagissant avec un humain. En outre, les questions centrales de cette thèse sont : sur quelle base théorique peut-on interpréter ces motifs d'interaction ? Ces motifs d'interaction ont-ils un intérêt pour la modélisation du dialogue pour un agent interactif ? Si oui, quelles sont les structures adaptées à la modélisation des motifs d'interaction pour produire un modèle de l'interaction pour un agent en interaction avec un humain ?

Le point défendu est que les *motifs récurrents* sont les manifestations de *conventions d'interactions dialogiques* observées dans et parfois *au-delà du corpus*. Les conventions d'interaction dialogiques sont alors l'objet d'étude privilégié. Nous défendons l'intérêt de la modélisation de ces motifs récurrents dans le cadre plus général de la modélisation des conventions dialogiques. La gestion de l'interaction avec un utilisateur humain pour la réalisation d'une tâche est alors conçue comme mêlant deux niveaux : un haut niveau constitué de raisonnements délibératifs, et un bas niveau constitué de comportements réactifs guidés par les conventions dialogiques. Dans cette optique, il est possible de concevoir un modèle conventionnel des interactions enrichi par les motifs récurrents observés dans plusieurs corpus. Ce modèle peut être exploité par un agent en interaction avec un humain pour gérer la part conventionnelle de l'interaction.

Plan de la thèse

Ce document est constitué de trois parties. La *première partie* présente notre positionnement par rapport à certains travaux antérieurs. Nous y parcourons les différentes approches de modélisation du dialogue pour parvenir à la conclusion de la nécessité d'aller vers des approches mixtes. Nous nous intéressons en particulier à la métaphore des jeux de dialogue qui semble prometteuse pour ce genre d'approche. La *deuxième partie* décrit les étapes de recueil, d'annotation et d'extraction de motifs d'interaction de notre méthodologie appliquées à un corpus de dialogues entre un expert et un utilisateur pour une tâche de recherche d'information dans le domaine médical. La *troisième partie* présente le formalisme adopté pour modéliser les conventions d'interaction dialogique basé sur les jeux de dialogue. Nous définissons les jeux de dialogue correspondant aux motifs que nous avons extraits. Puis nous présentons la mise en œuvre de DOGMA [3] : un module normatif de l'interaction conventionnelle basé sur les jeux de dialogue et utilisable par un agent interactif pour la gestion de la part conventionnelle du dialogue.

Ces trois parties se décomposent en les 7 chapitres suivants :

3. DOGMA pour « DialOgue Game MAnager »

— Le chapitre 1 introduit des notions indispensables à la modélisation du dialogue pour l'interaction Homme-Machine. Il détaille les caractéristiques du dialogue Homme-Homme qui entrent en jeu dans nos travaux. Nous nous intéressons à la théorie des actes de langage, fondement des modèles de l'énoncé en intelligence artificielle. Nous soulignons l'intérêt de passer à une approche contextuelle des actes de dialogue.

— Le chapitre 2 présente deux grandes catégories d'approche de modélisation du dialogue qui diffèrent par leur interprétation des motifs d'interaction. Les approches intentionnelles interprètent ces régularités comme la preuve d'un plan de la part des interlocuteurs tandis que les approches conventionnelles les voient comme des conventions établies. Après avoir exposé des exemples représentatifs de chacune de ces approches, nous soulignons les arguments en faveur de leur complémentarité.

— Le chapitre 3 présente les approches mixtes qui cherchent à combiner les avantages des approches intentionnelles et conventionnelles. Nous nous concentrons sur la métaphore des jeux de dialogue qui vise à définir une structure permettant de formaliser les motifs d'interaction observés dans des conversations humaines en vue de leur intégration dans un processus délibératif. Nous défendons l'intérêt d'aller vers une architecture d'agent mixte réactive/délibérative basée sur les jeux de dialogue vus comme structure d'engagements.

— Le chapitre 4 illustre les phases de récolte et de transcription de notre méthodologie sur le corpus Cogni-CISMeF. Les entretiens du corpus sont décrits par l'exposition d'une structure haut niveau commune, relative à la tâche de recherche d'information collaborative.

— Le chapitre 5 présente la réalisation des processus d'annotation du corpus Cogni-CISMeF et d'extraction de motifs dialogiques sur le corpus annoté. Les résultats de chacun de ces processus sont analysés et récapitulés. La réalisation de ces étapes conduit à la constitution d'une bibliothèque de motifs d'interaction récurrents prenant majoritairement la forme de paires adjacentes.

— Le chapitre 6 expose un cadre pour modéliser les motifs dialogiques observés dans le corpus Cogni-CISMeF, basé sur l'approche des jeux de dialogue comme structure d'engagements. La spécification des jeux de dialogue depuis les motifs dialogiques que nous avons extraits est illustrée. Ces jeux forment la base du modèle d'interaction d'un agent interactif. La couverture de ces jeux est validée face au corpus de référence conservé à cet effet lors du processus d'extraction.

— Le chapitre 7 montre l'intérêt des jeux de dialogue pour la gestion du comportement communicatif conventionnel d'un agent interactif. Il présente la mise en œuvre du modèle dans Dogma, un module basé sur les jeux de dialogue exploitable par un agent interactif afin de réguler son comportement communicatif. Des exemples illustrant le fonctionnement du module sont exposés. Nous décrivons plusieurs scénarios d'évaluation pour Dogma à moyen et long terme. Puis, nous donnons quelques pistes pour l'intégration de ce système dans le processus délibératif d'un agent interactif.

Première partie

Positionnement

Chapitre 1

Fondements pour la modélisation du dialogue

Sommaire

1.1 Généralités sur le dialogue 11
1.1.1 Le dialogue est une activité conjointe et opportuniste 11
1.1.2 Gestion de la tâche sous-jacente et gestion de la communication 14
1.2 Actes de langage et actes de dialogue 16
1.2.1 Actes de langage 17
1.2.2 Des actes de langage aux actes de dialogue 19
1.3 Synthèse 27

Ce chapitre introduit les notions indispensables à la modélisation du dialogue pour l'interaction Homme-Machine. La section 1.1 se concentre sur la présentation des caractéristiques importantes du dialogue dans cette optique. Puis, nous abordons en section 1.2 la théorie des actes de langage qui est à la base du modèle de l'énoncé dominant en intelligence artificielle. Enfin, nous dressons une synthèse en section 1.3 qui apporte un regard critique sur la théorie des actes de langage à la lumière des caractéristiques du dialogue que nous avons présentées.

1.1 Généralités sur le dialogue

Dans cette section, nous présentons les caractéristiques importantes du dialogue Homme-Homme pour la modélisation de l'interaction Homme-Machine *à propos d'une tâche*. Nous limitons nos considérations au dialogue à deux participants. Le but de cette section est de présenter un ensemble de notions utilisées dans la suite du manuscrit. Dans un premier temps, nous voyons que le dialogue est une *activité conjointe* où les interlocuteurs cherchent à se coordonner (cf. section 1.1.1). Nous soulignons la *vision opportuniste* du dialogue qui insiste sur le fait que le dialogue est co-construit et co-contrôlé par les interlocuteurs. Puis, nous traitons de la *multi-dimensionnalité* du dialogue (cf. section 1.1.2). Le dialogue sert à gérer la tâche sous-jacente mais également l'interaction en elle-même. Conséquence de la multidimensionnalité du dialogue, nous constatons que les énoncés des interlocuteurs sont souvent multifonctionnels et abordent plusieurs dimensions de la gestion du dialogue simultanément.

1.1.1 Le dialogue est une activité conjointe et opportuniste

Les *activités conjointes* sont un type particulier d'*activités collectives* (une activité dans laquelle plusieurs individus prennent part). Pour Clark [Clark 1996] (p. 37-38), une activité

conjointe est une activité : (i) qui est effectuée par deux participants ou plus ; (ii) dans laquelle les participants jouent un rôle public ; (iii) dans laquelle les participants essayent d'établir et d'atteindre des buts publics ; (iv) dans laquelle les participants peuvent poursuivre des buts privés ; (v) qui est composée d'actions ou d'activités conjointes ; (vi) dans laquelle les participants sont susceptibles d'utiliser des procédures spécifiques pour atteindre leurs buts ; (vii) dans laquelle les participants s'accordent sur un début et une fin ; (viii) qui peut être simultanée ou séquencée. L'exemple classique d'une activité conjointe est celle d'un duo en musique.

Les activités conjointes se décomposent en *actions conjointes*. Ces dernières sont elles-même constituées d'*actions individuelles*. Celles-ci peuvent se décomposer en deux sous-types qui se différencient par la notion clé de *coordination* : les *actions autonomes* et les *actions participatives*. L'individu qui réalise une action autonome ne cherche pas à se coordonner avec un autre participant. Au contraire, une action conjointe requiert la *coordination* des actions individuelles de deux personnes ou plus, i.e. répondre à la question « qui fait quoi et quand ? ». Une action conjointe est donc plus que la somme des actions individuelles. Au-delà même de la coordination, un individu qui agit conjointement doit prendre en compte les actions des autres participants et les considérer comme faisant partie de l'action conjointe. Ces actions individuelles d'une action conjointe sont les *actions participatives*.

[Clark 1996] suggère qu'une action conjointe peut être coordonnée parce qu'elle peut se voir attribuer : (i) une *entrée* dans l'action conjointe, (ii) un *corps*, i.e. l'action conjointe en elle-même, et (iii) une *sortie* de l'action conjointe. Autrement dit, les participants doivent aller d'un état où ils ne sont pas engagés dans l'action conjointe, à un état où ils y sont engagés puis de nouveau dans un état hors de l'action conjointe. Dans une telle action, les participants doivent se coordonner sur ces trois caractéristiques. Pour assurer la *synchronisation* de leurs actions, les participants doivent se coordonner sur l'entrée et la sortie de chaque action conjointe. Selon [Clark 1996], les participants y arrivent en suivant des stratégies de coordination (cadence, enchaînement, ...) qui sont caractérisées par le *principe de synchronisation* : dans les activités conjointes, les participants synchronisent leurs processus principalement en coordonnant les temps d'entrée et les actions participatives pour chaque action conjointe. Autrement dit, les actions conjointes sont principalement organisées autour de leurs entrées et des actions participatives attendues.

Clark avance l'idée de *projet conjoint*. Un projet conjoint est une action conjointe proposée par un des participants qui peut être acceptée et réalisée par tous (ou bien refusée). Considérons les deux énoncés du dialogue 1.1 qui forment une paire adjacente. Celle-ci peut être vue comme la réalisation de l'activité conjointe suivante :

1. S et H participe à un échange d'information.
2. S participe à 1 en posant une question.
3. H participe à 1 en répondant à la question.

S1 : ca date de quand ça ?
H2 : oh il est de 82

Dialogue 1.1 – Un exemple de projet conjoint d'échange d'information (entretien VD06)

Selon Clark, la première partie de l'échange est une question qui forme une proposition d'entrée dans un *projet conjoint*. Elle établit les participants (le locuteur et son interlocuteur),

le temps d'entrée (marqué par la réalisation de la question) et enfin le contenu de l'action conjointe. En outre, la question produite est également l'action participative de l'interlocuteur S attendue dans l'action conjointe. La réponse à la question réalise l'acceptation du projet conjoint en plus de la réalisation de l'action participative attendue. Les projets conjoints mettent en avant le fait que le dialogue est *contrôlé conjointement* par les interlocuteurs : l'entrée dans un projet commun requiert l'engagement des deux participants. L'auteur va même plus loin en posant que les paires adjacentes sont les *projets communs minimaux*. L'idée est qu'une paire adjacente établit avec un *effort conjoint minimum* les participants, le temps d'entrée et le contenu de l'action conjointe. Les *participants* sont établis par l'initiateur de la première partie de la paire et son destinataire. Le *temps d'entrée* est marqué par le déclenchement de la première partie de la paire. Le *contenu* est déterminé par la première partie de la paire et la seconde partie attendue.

Enfin, Clark défend une *vision opportuniste* du dialogue : les participants s'engagent dans une activité conjointe complexe sans pouvoir connaître à l'avance ce qu'ils vont y faire. Le dialogue est vu comme une activité dans laquelle les actions des participants sont *locales* et *opportunistes*. Pour Clark, le dialogue est contrôlé conjointement par les interlocuteurs via des projets conjoints. Il émerge alors sous la forme de *projets communs étendus*. Ces derniers sont obtenus par *combinaison* de *projets communs minimaux*, i.e. les paires adjacentes. Le champ de l'analyse de la conversation fait apparaître que la stricte adjacence n'est pas toujours respectée. Les paires adjacentes sont en fait utilisées pour créer des séquences plus longues via des *combinaisons* de séquences : la pré-séquence et l'emboîtement. Les *pré-séquences* sont des séquences qui *préparent* une autre séquence en vérifiant que des conditions préparatoires sont vérifiées (e.g., pré-question, pré-invitation, pré-requête). Le dialogue 1.2 présente un exemple de pré-séquence de clôture.

S1 : Est-ce que vous voudriez préciser quelque chose ?
H2 : Non c'est bon
S3 : donc on va s'arrêter là
H4 : Ok.

Dialogue 1.2 – Exemple de pré-séquence de clôture (entretien AL06). La séquence d'énoncés [1;2] prépare la séquence d'énoncés [3;4].

Les *séquences emboîtés*[1] sont des exemples typiques qui ne respectent pas l'adjacence stricte. Classiquement, elles prennent la forme d'une ou plusieurs séquences s'emboîtant entre deux parties d'une paire adjacente (e.g., pour introduire une clarification, une explication, une reformulation). Le dialogue 1.3 présente un exemple d'emboîtement de paires adjacentes.

S1 : Tu viens au cinéma avec nous ce soir ?
H2 : Vous allez voir quel film ?
S3 : On va voir « Holy Motors ».
H4 : OK, je viens.

Dialogue 1.3 – Exemple de séquences emboîtées. Entre la première partie de paire **S**1 et la seconde partie **H**4 est emboîtée la séquence [2;3].

Enfin, Clark propose la notion d'*enchaînement* [Clark 1996]. L'idée est que la seconde partie

1. Traduction de « Side sequence ».

d'une paire forme généralement la première partie d'une autre. Le dialogue 1.4 présente un exemple de séquence en trois temps de type question/réponse/évaluation.

S1 : donc c'est des douleurs articulaires à quel niveau ?
H2 : au niveau des genoux et autres
S3 : d'accord

Dialogue 1.4 – Exemple de séquences enchaînées (entretien AL06) de type question/réponse/évaluation.

Ainsi, les projets communs minimaux jouent le rôle de briques minimales dans la construction d'une interaction étendue.

En résumé, le dialogue peut être vu comme une *activité conjointe opportuniste* [Clark 1996]. Plus précisément, le dialogue implique la combinaison d'actions conjointes à différents *niveaux*.

1.1.2 Gestion de la tâche sous-jacente et gestion de la communication

Axe régissant et axe incident

La complexité de l'étude et de la modélisation du dialogue est due à l'une de ses grandes forces : le dialogue sert à faire progresser la tâche qui le motive, mais surtout il est utilisé pour la gestion des processus de communication. En d'autres termes, le dialogue sert à gérer la tâche sous-jacente, et le dialogue en lui-même !

À ce sujet, Luzzati distingue deux *axes* de déroulement du dialogue [Luzzati 1989]. L'axe *régissant* est dédié à la réalisation de la tâche tandis que l'axe *incident* est dédié à la gestion de l'interaction (e.g., les sous-dialogues de clarification). En cas de problème de communication, le dialogue change d'axe pour passer sur l'axe incident. On parle alors de *dialogue incident* (e.g., les deux derniers tours de parole de : *« Je m'appelle Guillaume. – Pardon ? – Je suis Guillaume. »*).

Clark propose la notion de *track* qui affine la vision proposée par Luzzati [Clark 1996]. Soulignons la présence de deux *tracks* : la première pour gérer la tâche sous-jacente (ce que Clark nomme l'« official business »), et la deuxième pour gérer la communication qui a lieu sur la track 1. Il précise que ces tracks sont récursives mais dépassent rarement les profondeurs 3 ou 4. Enfin, il note que la présence de *meta-communication* n'a pas pour unique but de corriger des problèmes. Au contraire, les interventions meta-communicatives sont *systématiques* et *essentielles* au bon déroulement de la communication. Il note la présence de signaux spécialement dédiés à la meta-communication comme les « ok », « mmh mmh », les hochements de tête, les sourires, etc. Ainsi, il ne faut pas réduire la meta-communication à une vision simpliste où le dialogue se déroulerait en segment strictement sur un axe ou l'autre. Les signaux produits recouvrent généralement les deux axes.

La dissociation entre les deux axes de déroulement du dialogue est une première étape vers la considération de sa complexité. Mais la gestion de la communication intègre différents processus qui peuvent intervenir sur plusieurs niveaux.

Processus de gestion de l'interaction et niveaux

La gestion de l'interaction fait apparaître différents *processus*. Allwood propose le concept d'« Interactive Communication Management (ICM) » pour désigner la communication en rapport avec la gestion de l'interaction dialogique [Allwood 1995]. Il distingue la *gestion des tours*

de parole, la *structuration du dialogue* (comme l'introduction de sujet de discussion, e.g., « Abordons la question de l'interopérabilité des systèmes ») et les *feedbacks* (signaux de retour). En plus de l'ICM, il distingue une dimension de *gestion de la communication personnelle*[2] dédiée aux processus que le locuteur met en jeu lors de la production de son énoncé (e.g., des auto-corrections). Bunt divise la gestion de l'interaction en plusieurs *dimensions* [Bunt 2009]. Il considère les dimensions de feedbacks, de structuration du dialogue, de gestion de tour de parole et de gestion de la communication personnelle envisagées par Allwood. Il ajoute la gestion du temps (e.g., les pauses), la gestion du contact, la gestion de la communication du partenaire (e.g., la complétion d'une réponse).

Plusieurs auteurs ont mis en évidence le fait que ces processus interviennent sur plusieurs *niveaux*. Allwood liste quatre niveaux sur lesquels peut intervenir la gestion de la communication : le contact (gestion de l'attention), la perception, la compréhension et le niveau de réaction [Allwood 1995]. De la même manière, Clark indique que l'usage du langage implique *au moins* quatre niveaux différents d'*actions conjointes* [Clark 1996] présentés dans le tableau 1.1.

Niveau	Action du locuteur (A)	Action de l'allocutaire (B)
4	A propose un projet conjoint w à B	B considère la proposition w de A
3	A signifie p à B	B comprend p
2	A présente un signal s à B	B identifie s
1	A exécute un comportement c pour B	B est attentif à c

Tableau 1.1: Niveaux de la gestion de l'interaction selon Clark [Clark 1996]

Le niveau 1 est le niveau de gestion de l'attention. Les interlocuteurs s'assurent qu'ils sont en mesure de percevoir les messages des participants du dialogue. Le niveau 2 peut être vu comme le niveau de perception, tandis que le niveau 3 est celui de la compréhension. Enfin le dernier niveau consiste en la proposition d'une action conjointe considérée par le partenaire. Selon Clark, ces niveaux forment une *échelle d'actions* avec les propriétés de *causalité ascendante*, de *réalisation ascendante* et d'*évidence descendante*. La causalité ascendante spécifie que les actions d'un niveau inférieur sont exécutées *afin de* réaliser les actions du niveau supérieur. Ainsi, B est attentif au comportement de A afin de pouvoir identifier le signal présenté par A, et ainsi de suite jusqu'au niveau 4. Cette propriété conduit à celle de « réalisation ascendante » qui stipule que les actions ne peuvent être réalisées que du niveau le plus bas vers un des niveaux supérieurs dans l'échelle d'actions. Pour finir, cette dernière propriété implique l'*évidence descendante* : dans une échelle d'actions, l'évidence de réussite d'un niveau est également la preuve de réussite des niveaux qui lui sont inférieurs. Ainsi, comprendre une question (niveau 3) implique que la question a été perçue (niveau 2) et que le contact a été établi (niveau 1). Cependant, la question peut ne pas être acceptée (niveau 4).

Bunt affine ces *niveaux* et considère : l'attention, la perception, l'*interprétation*, l'*évaluation* et l'*exécution* [Bunt 2009]. Selon nous, les niveaux d'interprétation et d'évaluation sont à rapprocher du niveau de compréhension de Clark (niveau 3). Il s'agit de comprendre le signal d'un point de vue sémantique (établir la fonction communicative et le contenu sémantique) et pragmatique (e.g., la question a-t-elle déjà été répondue ?). Ces niveaux sont préparatoires au niveau d'exécution qui correspond au niveau de considération chez Clark, et de réaction chez Allwood.

Le dialogue est multidimensionnel : il est utilisé pour faire progresser sa tâche sous-jacente et

2. Traduction de « Own Communication Management » (OCM)

différents processus de gestion de l'interaction. La prochaine section traite de l'impact de cette multidimensionnalité sur les signaux produits par les interlocuteurs.

Multifonctionnalité des énoncés

Les énoncés de dialogue sont souvent *multifonctionnels* [Allwood 1992, Traum 1992, Bunt 2011b] : un énoncé aborde plusieurs *dimensions* du dialogue simultanément. Cela est principalement dû à la multidimensionnalité du dialogue qui est lui-même utilisé afin de gérer la tâche sous-jacente (obtenir une information, négocier un accord, etc.) mais aussi l'interaction (gestion des tours de parole, du contact, de la compréhension, etc.). Une étude empirique sur des corpus de dialogues Homme-Homme a montré dans le plus contraignant des cas qu'en moyenne un énoncé sur trois aborde deux dimensions du dialogue [Bunt 2011b]. « Le plus contraignant des cas » est à comprendre au sens où un énoncé est la plus précise des unités segmentables dans le dialogue. Le dialogue 1.5 présente un exemple d'énoncé multifonctionnel.

S1 : [...] on peut enlever analyse
H2 : alors enlevons analyse
S3 : et diagnostic
H4 : oui

Dialogue 1.5 – Exemple de dialogue entremêlant avancement de la tâche et gestion de l'interaction (entretien VD06)

Il est tiré de notre corpus de réalisation d'une tâche de recherche d'information entre un utilisateur et un expert du moteur de recherche CISMeF. L'énoncé **S**1 est une suggestion d'ajouter le mot-clé « analyse » à la requête. Elle est acceptée par l'énoncé **H**2. Remarquons que cette acceptation est réalisée en *répétant une partie de l'énoncé de suggestion*. Stricto sensu, cette répétition n'est pas indispensable comme le démontre l'énoncé **H**4 qui est une acceptation par un simple « oui » à une suggestion de suppression du mot-clé « diagnostic » (énoncée en **S**3). L'énoncé **H**2 est en fait sur les deux axes de communication : sur l'axe régissant, c'est une acceptation d'une suggestion et sur l'axe incident, c'est un feedback positif précisant que la suggestion a été perçue et comprise. C'est un énoncé *multifonctionnel*. Nous approfondissons la question de la multifonctionnalité des énoncés lors du processus d'annotation de notre corpus (cf. section 5.1.1).

La modélisation du dialogue nécessite la considération de ses constituants : les énoncés échangés par les interlocuteurs. Intéressons-nous maintenant aux modèles existants permettant leur modélisation.

1.2 Actes de langage et actes de dialogue

Dans cette section, nous présentons le modèle de l'énoncé dominant en intelligence artificielle : l'*acte de langage* (cf. section 1.2.1). Nous soulignons l'inadéquation du modèle de l'acte de langage pour la modélisation du dialogue Homme-Homme au regard des propriétés présentées dans la section 1.1 (cf. section 1.2.2). Nous présentons une alternative qui consiste à considérer plus généralement des *actes de dialogue* (cf. section 1.2.2).

1.2.1 Actes de langage

La théorie des *actes de langage* [Austin 1962, Searle 1969] issue de la philosophie du langage est à la base de nombreux travaux en intelligence artificielle sur la modélisation du dialogue et des systèmes de dialogue. L'idée fondatrice de cette théorie peut se résumer à la célèbre expression « Dire c'est faire » [Austin 1962]. L'intérêt de cette théorie pour l'intelligence artificielle est que l'usage de la langue peut être vu comme la réalisation d'une action « classique ».

Notion d'acte de langage

Austin remarque que les énoncés performatifs contiennent l'action qu'ils dénotent. Par exemple, dire « Je baptise ce bateau le Queen Elizabeth » c'est faire l'action du baptême à condition qu'on ait la légitimité de le faire. Pour Austin, tout énoncé est un acte appelé *acte de langage* contenant trois composantes [Austin 1962] :

— l'*acte locutoire* : acte physique d'énonciation (le dire).

— l'*acte illocutoire* : acte qui traduit les intentions du locuteur envers son ou ses interlocuteurs (le faire). La reconnaissance de l'acte illocutoire est appelée *effet illocutoire* [Searle 1969]. Les question, affirmation, suggestion sont des actes illocutoires et leurs reconnaissances des effets illocutoires.

— l'*acte perlocutoire* : acte par lequel le locuteur cherche à produire des *effets perlocutoires* (ou *perlocutions*) sur son interlocuteur. Les effets perlocutoires peuvent être voulus (e.g., la réalisation de l'ordre « Ferme la porte ! ») ou non (e.g., éclatement de rire du destinataire suite à une menace).

Ainsi, ordonner « Haut les mains ! » à un interlocuteur, c'est réaliser :

1. l'acte locutoire par l'énonciation de l'énoncé ;
2. l'acte illocutoire d'ordre dont l'effet illocutoire est la reconnaissance de cet ordre par l'interlocuteur ;
3. l'acte perlocutoire qui est l'acte par lequel le locuteur cherche à obtenir l'effet perlocutoire (ici le fait que l'interlocuteur lève les mains).

Notons que la réalisation de l'effet perlocutoire est postérieure à la reconnaissance de l'acte illocutoire (effet illocutoire). Ainsi, le partenaire peut très bien avoir reconnu l'ordre et refuser de s'y soumettre. D'autre part, signalons que l'expression « acte de langage » est souvent utilisée abusivement pour désigner l'acte illocutoire.

Apport de la logique illocutoire

À la suite d'Austin, Searle replace la théorie des actes de langage dans une théorie des intentions [Searle 1969]. Énoncer une proposition résulte de l'intention de la produire tandis que la comprendre c'est interpréter l'intention sous-jacente, dans le contexte où elle a été produite. Il pose la notation fonctionnelle $F(p)$ où p représente le *contenu propositionnel* et F la *force illocutoire* appliquée à p. Ces travaux ont été poursuivis dans la formalisation d'une logique illocutoire du discours [Searle 1985, Vanderveken 1990].

Il propose une taxonomie de forces illocutoires des actes de langage basée sur la notion de *but illocutoire*. Le but illocutoire d'un acte illocutoire peut être vu comme l'effet perlocutoire public voulu (e.g., faire faire quelque chose, engager le locuteur sur la réalisation d'une action, etc.).

Notons qu'un locuteur accomplissant un acte illocutoire peut avoir d'autres buts perlocutoires en plus du but illocutoire. Par exemple, un locuteur peut vouloir embarrasser une personne tout en poursuivant le but illocutoire de lui faire faire quelque chose. De plus, chaque but illocutoire peut être associé à une *direction d'ajustement*. La taxonomie est la suivante :

— Les *assertifs* qui servent à décrire des faits (e.g., « Il pleut. »). Le but est de rendre le contenu propositionnel conforme au monde (direction « mots vers monde »).

— Les *directifs* qui visent à faire faire quelque chose à l'interlocuteur (e.g., « Sers moi une tasse de café ! »). Le but est de rendre le monde conforme au contenu propositionnel (direction « monde vers mots »).

— Les *promissifs* qui obligent le locuteur à réaliser une action (e.g., « Je te promets que je viendrai »). Le but est de rendre le monde conforme au contenu propositionnel (direction « monde vers mots »).

— Les *déclaratifs* qui rendent vrais des faits par leur simple énonciation (e.g., « Je vous déclare unis par les liens sacrés du mariage »). La direction d'ajustement est double : le monde s'ajuste au contenu propositionnel et réciproquement.

— Les *expressifs* qui expriment l'état d'esprit du locuteur (e.g., « Je suis désolé »). La direction d'ajustement est vide.

Deux notions principales sont associées à un acte de langage : son *succès* et sa *satisfaction*. Les *conditions de succès* d'un acte sont déterminées par sa force illocutoire et son contenu propositionnel. Elles précisent les circonstances dans lesquelles le locuteur a réalisé l'acte avec succès. Elles incluent des *conditions préparatoires* et des *conditions de sincérité*. Par exemple, un certain nombre de conditions doivent être remplies avant de réaliser la requête « Ferme la porte ! ». Le locuteur doit être sincère : il veut que la porte soit fermée. En outre, d'autres conditions doivent être vérifiées comme le fait qu'il doit y avoir une porte et un interlocuteur.

Les *conditions de satisfaction* déterminent les circonstances dans lesquelles l'acte de langage est satisfait. Ces circonstances sont décrites dans le contenu propositionnel de l'acte. Ainsi, un acte est satisfait si son contenu propositionnel est rendu vrai selon la direction d'ajustement au monde déterminée par son but illocutoire. Par exemple, une requête est satisfaite quand son destinataire produit la situation décrite dans le contenu propositionnel de l'acte (dans notre cas, s'il ferme la porte).

Actes de langage indirects

Les actes de langage indirects remettent en cause l'hypothèse selon laquelle il est possible de déterminer le type d'un acte en se basant uniquement sur la forme de surface de l'énoncé (i.e., la forme linguistique). Un acte indirect est un acte non littéral accompli au moyen d'un autre acte, appelé *acte littéral*. Cela inclut des exemples comme :

— « Il fait froid ! » dont l'acte littéral est de type assertif et dont l'acte indirect peut être, par exemple, une requête pour fermer la fenêtre.

— « Est-ce que tu peux me passer le sel ? » qui prend la forme d'une question mais est très certainement une requête.

Une observation intéressante est qu'il est possible de répondre aussi bien à l'acte littéral, l'acte indirect ou aux deux en même temps ! [Clark 1996] (pp. 216–218). Ainsi, il paraît tout à fait possible d'acquiescer l'assertion « Il fait froid ! », d'acquiescer l'assertion et de fermer la

fenêtre, ou de simplement fermer la fenêtre. L'objet d'un modèle de dialogue incluant des actes de langage indirect est de préciser sur quelle base ces inférences peuvent être faites.

1.2.2 Des actes de langage aux actes de dialogue

Dans cette section, nous présentons un regard critique sur les actes de langage classiques. Face aux difficultés rencontrées par les actes de langage, nous présentons une vision alternative : l'approche contextuelle. Nous voyons ensuite comment cette approche contextuelle est mise à profit par les *actes de dialogue*.

Critique des actes de langage

La théorie des actes de langage est celle qui s'est imposée afin de représenter les unités conversationnelles (i.e., les énoncés) en intelligence artificielle. Cependant, la prise en compte du cadre dialogique soulève un certain nombre de problèmes. Nous soulignons en particulier les point suivants :

— le *caractère non monologique du dialogue* : les interlocuteurs sont engagés dans une activité conjointe complexe (cf. section 1.1.1). L'allocutaire n'est pas un auditeur passif (voir, e.g., la notion d'*uptake* [Austin 1962, Clark 1996]).

— le *caractère multidimensionnel du dialogue* : le dialogue sert à gérer la tâche sous-jacente mais aussi l'interaction elle-même (cf. section 1.1.2).

— le *caractère multifonctionnel des énoncés* : dès que le caractère multidimensionnel du dialogue est pris en compte, on constate que les énoncés peuvent réaliser plusieurs actes communicatifs contrairement à ce que prétend la théorie des actes de langage (cf. section 1.1.2).

La prise en compte de la multidimensionnalité du dialogue et de la multifonctionnalité pose problème aux approches traditionnelles des actes de langage. Ces approches se sont principalement intéressées au cas idéalisé où les processus de gestion de l'interaction n'interviennent pas. Cela a conduit certains chercheurs à s'orienter vers l'approche contextuelle des actes de langage qui permet de prendre plus largement en compte le contexte du dialogue plutôt que de se limiter aux intentions et croyances des interlocuteurs.

Contexte du dialogue

La notion de *contexte* du dialogue a été étudiée par différents auteurs (e.g., [Clark 1996, Bunt 2000, Ginzburg 2012]). La définition qui nous semble la plus claire et la plus intéressante pour notre travail est celle de Bunt [Bunt 1996, Bunt 2000, Bunt 2011a]. Elle est issue d'un travail liant à la fois l'*analyse de dialogue* et la conception de *système de dialogue*. Bunt définit le contexte du dialogue de la manière suivante :

> « Le contexte du dialogue est la totalité des conditions qui influencent l'interprétation ou la génération d'énoncés dans le dialogue. » [Bunt 2000]

Notons tout d'abord que cette notion de contexte dépasse amplement le cadre du contexte en linguistique qui est généralement réduit au co-texte. Cette définition inclut un grand nombre de facteurs généraux qui peuvent influencer le comportement communicatif d'un interlocuteur comme sa fatigue ou son éventuelle animosité pour l'autre interlocuteur.

Pour rendre cette notion plus tractable, Bunt différencie le *contexte global* (qui ne peut pas être changé par le dialogue) du *contexte local* (qui peut être changé par le dialogue). Plus précisément, le contexte local est défini de la manière suivante :

> « Le contexte local est la totalité des conditions qui peuvent être changées par l'interprétation d'énoncés du dialogue. » [Bunt 2011a]

Le temps qu'il fait (soleil, pluie, ...) fait partie du contexte global tout comme le fait que le dialogue prenne place entre un médecin et un patient, que le dialogue soit un débat philosophique ou un dialogue de réservation de billet de train, etc. En contrepartie, l'historique du dialogue courant fait partie du contexte local puisqu'il est changé par le cours du dialogue.

Bunt propose de diviser le contexte en cinq catégories [Bunt 1996] : cognitif, physique et perceptuel, social, sémantique et linguistique. Pour chaque catégorie, il faut distinguer l'aspect global « figé » de l'aspect local « dynamique ».

D'un point de vue global, le contexte cognitif inclut le but qui a motivé l'entrée dans le dialogue appelée la *tâche sous-jacente*. Il contient également un *modèle des autres participants* au dialogue tels que leurs identités, leurs capacités communicatives (machine, enfant, adulte, ...), leurs niveaux d'expertise vis-à-vis de la tâche sous-jacente (e.g., qui peuvent dépendre de la relation patient/docteur, élève/professeur, client/vendeur, etc.). À un grain plus fin, ce contexte inclut la représentation des croyances attribuées aux interlocuteurs relativement à la tâche et aux processus de communication. Par exemple, il peut représenter les problèmes qui peuvent intervenir à certains niveaux dans les processus de reconnaissance et de génération des interlocuteurs du participant au dialogue. C'est ce contexte local qui indique la présence d'un problème de perception du dernier énoncé.

Le contexte physique et perceptuel représente la situation de dialogue. Globalement, il s'agit d'un ensemble de paramètres qui ont une influence directe sur la disponibilité des canaux communicatifs et perceptuels. Certains paramètres comme la co-présence, la visibilité et l'audibilité des partenaires, ont une influence sur les modalités de communication comme les gestes, les regards, la parole. Ainsi, un dialogue face à face (en co-présence) permet aux interlocuteurs de s'exprimer via des gestes, des regards, et bien sûr la parole. Tandis qu'un dialogue téléphonique (médiatisé) réduit les moyens d'expression à la parole. Localement, ce contexte inclut des informations comme la présence, l'attention et la disponibilité des partenaires de l'activité dialogique.

Le contexte social représente les droits, obligations et contraintes relatifs aux aspects sociaux du dialogue. Cela inclut des conventions dialogiques (e.g., « de rien » après un remerciement) et aussi la relation sociale des participants (élève/professeur, client/vendeur, etc.).

Les contextes sémantique et linguistique regroupent des conditions principalement dynamiques. Le contexte sémantique contient les informations relatives à l'état de la tâche sous-jacente au dialogue. Au niveau local, le contexte d'un individu représente sa perception de la tâche en cours, ainsi que ses objectifs courants en rapport avec cette tâche. Enfin, le contexte linguistique représente l'état du dialogue. C'est-à-dire qu'il inclut les événements communicatifs passés (historique du dialogue) et les événements communicatifs en cours de réalisation (e.g., la production d'un énoncé d'un participant du dialogue).

Approche contextuelle des actes de langage

L'approche contextuelle des actes de langage emprunte la vision de la théorie des actes de langage selon laquelle l'usage de la langue peut être vu comme la réalisation d'actions commu-

nicatives. Cela a conduit aux approches *contextuelles* [Hamblin 1970, Bunt 1996, Bunt 2000] ou « information-state approach » [Larsson 2000b, Traum 2003]. Ces approches envisagent les effets des actes en terme de changement sur les *états d'information* ou *contexte* des participants. Autrement dit, les actions communicatives sont des fonctions de contexte vers contexte [Bunt 1996].

L'idée est de se concentrer sur le contexte dans lequel l'acte est survenu en plus de sa forme de surface. L'acte est ensuite utilisé pour mettre à jour le contexte. Ainsi, un énoncé comme « Sais-tu quelle heure il est ? » peut en fonction du contexte être interprété comme une véritable question ou une menace (le destinataire est en retard à son rendez-vous). S'il est interprété comme une question, le destinataire met alors à jour son état d'information en incluant le fait qu'une question demandant s'il connaît l'heure a été posée par son partenaire. Néanmoins, si le locuteur a utilisé cet énoncé pour reprocher le retard de son interlocuteur, l'état d'information sera mis à jour sur la base du reproche. Dans ce cas, il est possible que l'état d'information ne contienne pas le fait qu'une question sur l'heure a été explicitement posée.

Selon Bunt [Bunt 1996, Bunt 2009], l'acte de langage est une *fonction communicative* (e.g., une question, un reproche, une requête) appliquée à un *contenu sémantique*. La fonction communicative exprime ce que le locuteur tente de réaliser, et le contenu sémantique décrit l'information qui est traitée. De manière plus formelle, un acte de dialogue est un opérateur de mise à jour de l'état d'information produit par l'application d'une fonction communicative à un contenu sémantique.

Il est intéressant de noter que ce qui différencie l'approche « classique » des actes de langage et l'approche contextuelle tient au contenu de l'*état d'information*. Si celui-ci se résume aux croyances et intentions de l'ensemble des interlocuteurs, l'approche contextuelle se transforme en une reformulation de la théorie classique. L'approche contextuelle devient particulièrement intéressante lorsque le contexte est plus large et englobe plusieurs aspects de la multidimensionnalité du dialogue (cf. section 1.1.2). En fonction de ce que représente le « contexte », on parle d'acte de langage ou bien d'*acte de dialogue*.

Actes de dialogue

Les actes de dialogue tentent de dépasser les limites des actes de langage en incluant la multidimensionnalité du dialogue. Le contexte considéré va alors prendre en compte les états mentaux des interlocuteurs et plus largement l'état du dialogue (e.g., l'historique du dialogue). De nouveaux actes vont faire leur apparition pour gérer les processus d'interaction comme la gestion des tours de parole, de la structuration du dialogue, etc. (cf. section 1.1.2). C'est le cas des actes de *feedbacks* qui permettent de gérer l'établissement d'un énoncé à différents niveaux de communication (contact, perception, compréhension, réaction). Par exemple, Larsson [Larsson 2003] présente une mise en œuvre du concept d'« Interactive Communication Management » [Allwood 1995].

Nous distinguons deux principaux angles d'attaque pour la définition des actes de dialogue : l'approche multi-niveaux qui se concentre sur les niveaux d'interaction et l'approche multidimensionnelle qui se concentre sur les dimensions de la gestion de l'interaction.

Taxonomie multi-niveaux d'actes de dialogue Une première tentative de modélisation d'actes de dialogue a été réalisée par les actes de conversation multi-niveaux [Traum 1992, Poesio 1997]. Ils entendent généraliser les actes de langage en éliminant certaines hypothèses non réalistes pour le dialogue comme le fait qu'un acte de langage est produit de manière isolée

par un locuteur pour un allocutaire passif.

Ils se concentrent sur la notion de *niveaux* de l'interaction (cf. section 1.1.2). Ils distinguent quatre niveaux reproduits dans le tableau 1.2 : sous-énonciation (Sub UU), énonciation (UU), unité de discours (DU) et discours (Multiple DUs). À chaque niveau correspond un ensemble d'actes : les actes de tour de parole au niveau sous-énonciation, les actes d'établissement [3] au niveau énonciation, les actes noyaux au niveau unité de discours qui correspondent aux actes illocutoires « classiques », et enfin les actes d'argumentation au niveau le plus élevé. Il faut bien noter que les trois premiers *niveaux* représentent des processus différents et ne sont pas réductibles les uns aux autres [Traum 1992] (p. 4). Le « niveau argumentation » ne semble pas être un *niveau* au sens où nous les avons présentés. Ce niveau est en fait généré par des *séquences d'actes noyaux*, contrairement aux autres niveaux qui sont eux générés par les actes locutoires [Poesio 1997] (p. 318). Ce niveau inclut d'autres structures telles que les *jeux de dialogue* [Poesio 1998b] (p. 2).

Niveau	Type d'acte	Exemples
Sub UU	Tour de parole	`take-turn`, `keep-turn`, `release-turn`, `assign-turn`
UU	Établissement	`initiate`, `continue`, `ack`, `repair`, `reqRepair`, `reqAck`, `cancel`
DU	Actes « noyaux »	`inform`, `WHQ`, `YNQ`, `accept`, `request`, `reject`, `suggest`, `offer`, `promise`
Multiple DUs	Argumentation	`Elaborate`, `Summarize`, `Clarify`, `Q&A`, `Convince`, `Find-Plan`

TABLEAU 1.2: Actes conversationnels multi-niveaux [Traum 1992, Poesio 1997]

Dans la suite de leurs travaux [Poesio 1997], le sous-niveau d'acte locutoire est ajouté. Il consiste en un acte `utter` permettant la production d'énoncés. Précisons qu'un acte locutoire peut générer *plusieurs actes* [Poesio 1997, Poesio 1998b]. Ainsi, un simple « ok » peut générer les actes `ack` au niveau établissement et `accept` au niveau DU. Les énoncés deviennent alors *multifonctionnels* (cf. section 1.1.2).

Pour comprendre les interactions entre ces niveaux, intéressons-nous au processus d'établissement proposé. Celui-ci a été modélisée par Traum [Traum 1994a] en s'appuyant sur le modèle de Clark et Schaefer [Clark 1989]. Il est vu comme la réalisation d'actes d'établissement. Un DU comprend une phase initiale de *présentation* par le locuteur et une phase d'*acceptation* par l'allocutaire (qui peut être implicite). Il correspond à une *contribution* au sens de [Clark 1989]. La phase de présentation peut s'étendre sur plusieurs énonciations (UU), par exemple via l'acte `continue`. Le changement radical par rapport à la théorie classique des actes de langage est qu'un acte de niveau noyau ne prend effet qu'une fois qu'il a été établi.

Le dialogue 1.6 est un exemple tiré de [Traum 1992] et présente l'établissement d'une suggestion. Le tour **M**1 commence par un acte d'établissement `initiate` qui ouvre un nouveau DU. Cet acte est poursuivi par une succession d'actes `continue` qui ajoutent des éléments au DU en cours. La production d'un acte `ack` au tour **S**2 permet au contenu du DU d'intégrer le terrain commun. À la suite de ce tour de parole, l'acte `suggest` a été établi. Néanmoins, les auteurs

3. Traduction du terme « grounding » [Clark 1989]

précisent que, dans ce cas particulier, le tour **S**2 n'est pas un `accept` au niveau DU. En d'autres termes, la suggestion n'a pas été acceptée au niveau de la tâche sous-jacente au dialogue.

	UU	DU
M1 : um and hook up the boxcar to the engine,	`initiate`	
move it from Dansville to Corning,	`continue`	
load up some oranges into the boxcar,	`continue`	
and then move it on to Bath.	`continue`	`suggest`
S2 : okay	`ack`	

Dialogue 1.6 – Exemple de dialogue annoté avec des actes de niveaux UU et DU (tiré de [Traum 1992]).

DIT++ Le cadre DIT++ [Bunt 2009] inclut une taxonomie *multidimensionnelle* de fonctions communicatives qui étend la taxonomie de la théorie « Dynamic Interpretation Theory » (DIT) [Bunt 1996]. Cette théorie propose une taxonomie de fonctions indépendantes de l'application pour l'analyse du dialogue Homme-Homme et pour la conception de système de dialogue. Elle adopte une approche multidimensionnelle de l'analyse du dialogue. La participation au dialogue est vue comme la réalisation de plusieurs activités en parallèle de la part des interlocuteurs (cf. section 1.1.2).

Acte de dialogue, fonction communicative et contenu sémantique Les interlocuteurs prennent part à ces activités en réalisant des *actes de dialogue*. Dans la continuité des approches contextuelles, un *acte de dialogue* est vu comme une opération de mise à jour de l'*état d'information* des interlocuteurs. Un acte de dialogue est constitué d'une *fonction communicative* et d'un *contenu sémantique*.

La *fonction communicative* exprime ce que le locuteur essaye de faire tandis que le *contenu sémantique* décrit l'information traitée. En d'autres termes, la fonction communicative spécifie la façon dont va être utilisé le contenu sémantique pour mettre à jour l'état d'information. Plus précisément, l'opération de mise à jour de l'état d'information dépend du locuteur, de l'interlocuteur et de la dimension [Bunt 2011c].

Notion de dimension Chacune des activités se déroule dans une *dimension*. Une dimension est définie comme étant un aspect de la participation au dialogue [Bunt 2011b] :

1. dans lequel les interlocuteurs peuvent intervenir au moyen d'actes de dialogue ;
2. qui peut être traité indépendamment des autres aspects considérés.

La première condition spécifie que les aspects de la communication considérés doivent avoir une existence empirique. La seconde condition précise que les dimensions doivent être *orthogonales*. L'orthogonalité est entendue dans le sens de [Bunt 2011b] où les fonctions communicatives attribuables à une dimension ne sont pas pleinement déterminées par les fonctions dans les autres dimensions.

Chaque activité est concernée par un type d'information différent (e.g., la gestion des tours de parole est concernée par l'attribution du tour de parole, le contenu des actes de gestion de la tâche est spécifique à la tâche sous-jacente, etc.). Ainsi, le type de contenu sémantique d'un acte de dialogue est déterminé par la dimension dans laquelle il intervient.

Enfin, l'ensemble des fonctions communicatives pouvant être utilisées dans une dimension respecte la contrainte suivante : pour chaque paire de fonctions communicatives qui peuvent être utilisées dans une dimension, soit les deux fonctions sont mutuellement exclusives, soit l'une est une *spécialisation* de l'autre (e.g., *Confirm* de *Answer*).

DIT++ propose l'utilisation de dix dimensions issues de l'étude d'un ensemble de corpus de dialogue et respectant la définition précédente [Bunt 2009] :

Task/Activity Actes de dialogue qui contribuent à l'avancement de l'activité sous-jacente au dialogue.

Auto-feedback Actes de dialogue par lesquels le locuteur fournit des informations sur son traitement de l'énoncé précédent.

Allo-feedback Actes de dialogue par lesquels le locuteur exprime son avis concernant le traitement de son précédent énoncé par l'autre interlocuteur, ou qu'il sollicite de l'information à propos de ce traitement.

Contact Management Actes de dialogue afin d'établir ou de maintenir le contact.

Turn Management Actes de dialogue concernés par l'attribution du rôle de locuteur.

Time Management Actes de dialogue signalant que le locuteur a besoin d'un peu de temps pour formuler sa contribution.

Discourse Structuring Actes de dialogue permettant de structurer explicitement la conversation (e.g., en annonçant le prochain acte de dialogue ou en proposant un changement de sujet).

Own Communication Management Actes de dialogue qui indiquent que le locuteur est en train de modifier sa contribution courante.

Partner Communication Management Actes de dialogue produits par le participant n'étant pas le locuteur afin de corriger ou d'assister le locuteur dans la formulation de sa contribution au dialogue.

Social Obligations Management Actes de dialogue qui s'occupent des conventions sociales telles que les salutations.

Les fonctions communicatives DIT++ propose une taxonomie de fonctions communicatives influencée par des schémas d'annotation précédents (e.g., DAMSL [Allen 1997] et TRAINS [Allen 1995]). Cette taxonomie inclut 88 *fonctions communicatives* et est constituée de deux parties : les fonctions *générales* et les fonctions *spécifiques à certaines dimensions* (e.g., *Apology*, *Turn Grab*). Les fonctions générales sont représentées sur la figure 1.1. Un extrait des fonctions spécifiques est représenté sur la figure 1.2.

Les fonctions générales sont divisées en quatre hiérarchies. Les deux premières sont les fonctions de recherche d'information (qui incluent différents types de question) et les fonctions fournissant de l'information (e.g., *Inform*, *Agreement*). Les fonctions de discussion d'action sont classiquement décomposées entre les fonctions promissives (e.g., *Promise*, *Offer*) et les fonctions directives (e.g., *Request*, *Suggestion*). Notons que chacune de ces catégories forme une *hiérarchie* où les fonctions sont soit mutuellement exclusives (e.g., *Confirm*/*Disconfirm*, *AcceptOffer*/*DeclineOffer*), soit des *spécialisations* (e.g., *PosiCheck* spécialise le type de question oui/non représenté par l'acte *PropositionalQuestion*).

Au contraire d'une fonction générale, une fonction spécifique ne peut intervenir que dans une dimension donnée. La figure 1.2 présente des exemples de fonctions spécifiques pour neuf

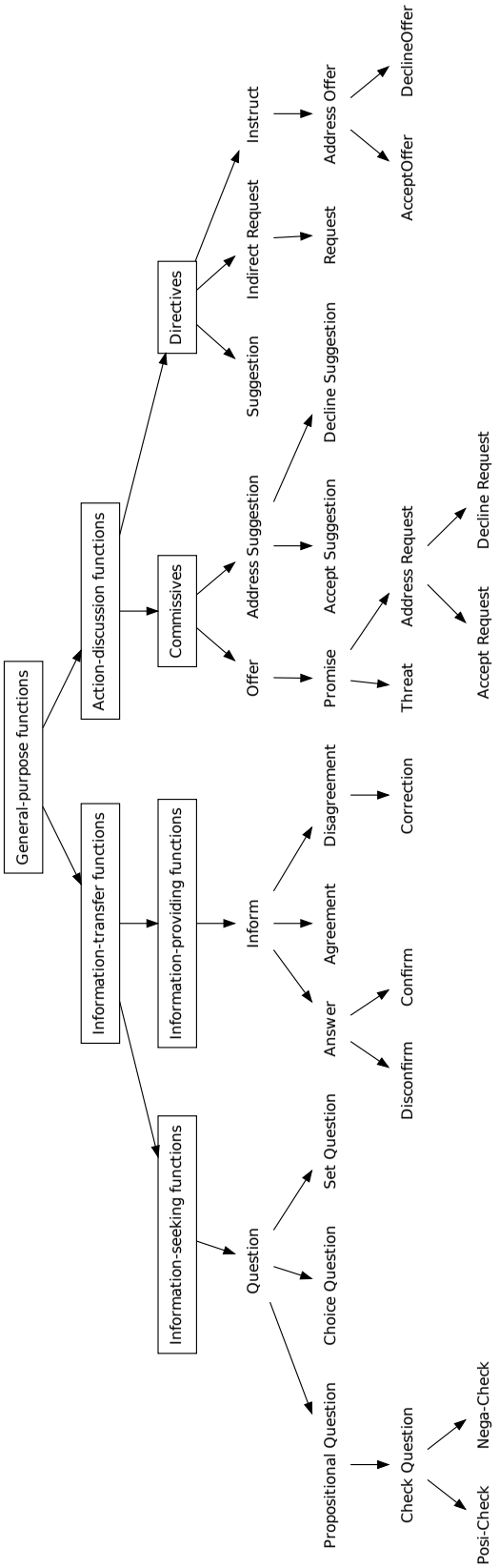

FIGURE 1.1: Fonctions générales de DIT++ [Bunt 2009]

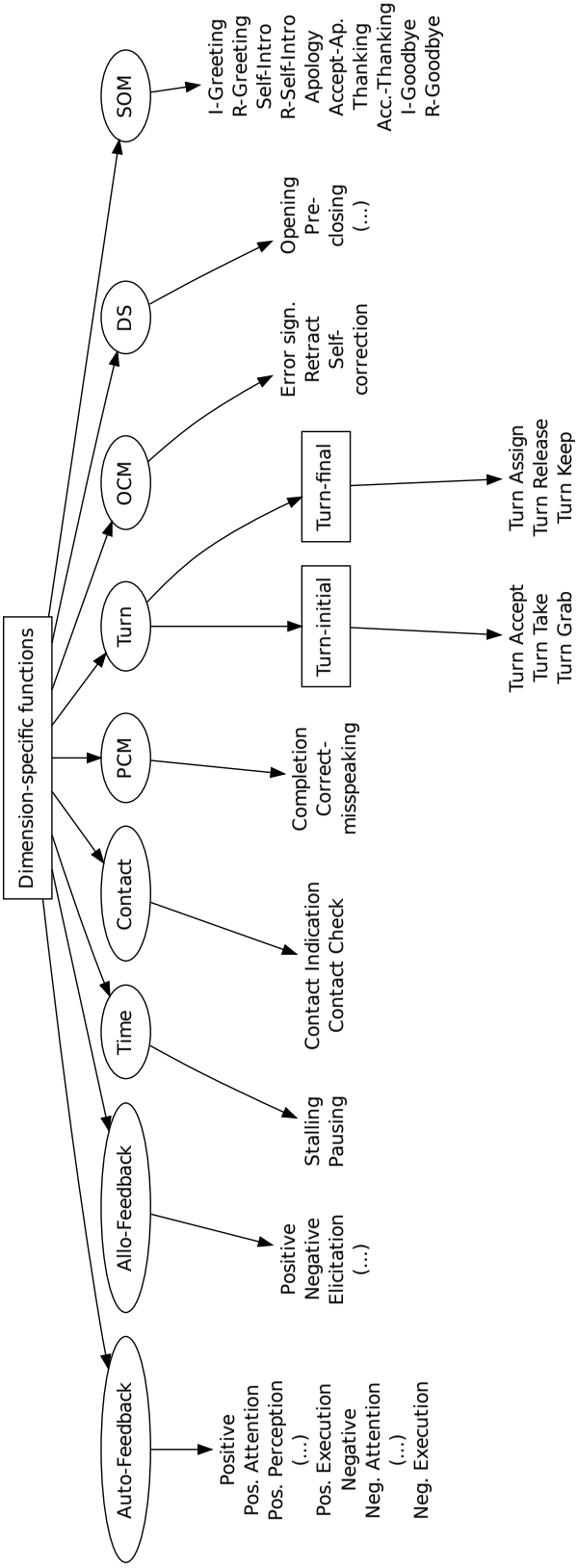

FIGURE 1.2: Extraits des fonctions spécifiques à certaines dimensions de DIT++ [Bunt 2009]

dimensions (*Task* exclue) comme la fonction *Turn Accept* dans la dimension *Turn Management* et la fonction *Thanking* dans la dimension *Social Obligation Management*.

Segment fonctionnel Enfin, Bunt propose de segmenter le dialogue en *segments fonctionnels*, un segment fonctionnel étant défini comme « ... une portion minimale de comportement communicatif qui a une fonction communicative. De telles portions n'ont pas besoin d'être grammaticalement correctes ou contiguës et peuvent avoir plus d'une fonction communicative. » [Bunt 2011b]. Notons qu'un segment possède au plus une fonction par dimension. En conséquence, les « énoncés multifonctionnels » dans DIT++ sont des segments fonctionnels qui peuvent être vus comme des combinaisons d'opérations de mise à jour de l'état d'information correspondant à chaque acte de dialogue [Bunt 2009, Bunt 2011b].

Structuration de l'état d'information Une question qui se pose est le lien entre l'état d'information et les dimensions présentées. En particulier, quel est le contenu de cet état d'information et comment est-il structuré ? Comme nous l'avons déjà dit (cf. section 1.2.2), l'approche contextuelle devient particulièrement intéressante dès le moment où le contexte intègre des notions qui vont au-delà des états mentaux des interlocuteurs comme des informations sur le cadre dialogique (e.g., historique du dialogue). Une première approche pourrait être de considérer un état d'information contenant un composant par dimension afin de représenter les informations indispensables à chaque aspect de l'activité dialogique. À ce propos, [Bunt 2011a] propose de regrouper certaines dimensions dans une version de l'état d'information en cinq composants : linguistique (historique du dialogue, ...), sémantique (état de la tâche sous-jacente, ...), cognitif (état des processus d'interprétation et de génération des énoncés, ...), physique et perceptuel (état de la situation interactive, ...) et social (gestion des aspects sociaux comme les salutations).

1.3 Synthèse

Dans un premier temps, ce chapitre a présenté le dialogue comme une *activité conjointe opportuniste* où les interlocuteurs cherchent à coordonner leurs contributions de manière à co-construire et à co-contrôler le dialogue. Nous avons abordé la *multidimensionnalité* du dialogue qui sert à gérer la tâche sous-jacente mais également l'interaction en elle-même. En conséquence, les énoncés des interlocuteurs sont souvent *multifonctionnels* et abordent plusieurs dimensions de la gestion du dialogue simultanément.

Puis, ce chapitre a été l'occasion de présenter le modèle de l'énoncé qui est à la base de nombreux travaux en intelligence artificielle : l'acte de langage. Historiquement, l'étude des énoncés par la théorie des actes de langage a précédé son intégration dans le cadre plus large de la modélisation du dialogue. Seulement cette intégration a soulevé de nombreux problèmes parmi lesquels nous avons souligné les aspects non-monologiques du dialogue, sa multidimensionnalité et la multifonctionnalité des énoncés. Face à ces problèmes, l'alternative semble d'étendre la portée des actes de langage à la prise en compte plus large du *contexte du dialogue*. Il s'agit alors de considérer une approche contextuelle des *actes de dialogue*. Parmi les taxonomies existantes, nous avons présenté les actes multiniveaux et DIT++ qui permettent de prendre en compte la multidimensionnalité du dialogue et la multifonctionnalité des énoncés.

Notons que les actes de langage et de dialogue sont des modèles de l'énoncé et non du

dialogue. Il est nécessaire d'expliquer comment les motifs dialogiques et des structures comme celles mises en avant par l'analyse conversationnelle émergent.

Chapitre 2

Modélisation du dialogue : intention versus convention

Sommaire

2.1 Approches intentionnelles 29
2.1.1 Fondements des approches intentionnelles 30
2.1.2 Approches par planification 32
2.1.3 Principales mises en œuvre 36
2.1.4 Avantages et limites des approches intentionnelles 38
2.2 Approches conventionnelles 39
2.2.1 Fondements des approches conventionnelles 39
2.2.2 Modèles fondés sur la structuration 42
2.2.3 Modèles fondés sur les questions en discussion 46
2.2.4 Avantages et limites des approches conventionnelles 52
2.3 Discussion 54

Dans ce chapitre, nous présentons deux grandes catégories d'approches de modélisation du dialogue généralement perçues comme opposées. Nous débutons par un aperçu des approches intentionnelles qui se focalisent sur les intentions des interlocuteurs pour expliquer la structure du dialogue (cf. section 2.1). Nous poursuivons ensuite par la présentation des approches conventionnelles qui délaissent les intentions pour se concentrer sur l'aspect conventionnel de l'interaction dialogique (cf. section 2.2). Enfin, nous discutons la prétendue opposition de ces approches (cf. section 2.3).

2.1 Approches intentionnelles

Dans cette section, nous effectuons un bref survol du vaste champ des approches intentionnelles de la modélisation du dialogue. Dans ce type d'approche, les motifs d'interaction sont interprétés comme la preuve d'un plan de la part des interlocuteurs [Hulstijn 2000a]. Les régularités sont vues comme un épiphénomène des intentions sous-jacentes et de la rationalité des interlocuteurs. Nous commençons cette section par une présentation des fondements théoriques de ce type d'approche (cf. section 2.1.1). Nous présentons ensuite trois approches représentatives (cf. section 2.1.2). Puis, nous abordons les principales mises en œuvre inspirées des modèles intentionnels (cf. section 2.1.3). Enfin, nous discutons les avantages et limites de ce type de modèle (cf. section 2.1.4).

2.1.1 Fondements des approches intentionnelles

Intention individuelle

Les travaux sur l'intentionnalité comme ceux de Bratman [Bratman 1987] ont donné naissance au modèle BDI [Rao 1995], influant dans le champ de l'intelligence artificielle. Ce modèle tire son nom des trois attitudes mentales qu'il considère pour fonder le comportement rationnel d'un agent : les *croyances* (B pour « Belief »), les *désirs* (D pour « Desire ») et les *intentions* (I).

Ces notions définissent une sémantique aux algorithmes de planification. Les *croyances* représentent la vision que l'agent a du monde (environnement et autres agents). Ces croyances peuvent être incorrectes, incomplètes ou incertaines. Elles évoluent au fur et à mesure des interactions entre l'agent et le monde (notamment via sa perception du monde qui peut amener à changer ses croyances). Les *désirs* représentent les états du monde souhaités par l'agent. Ils peuvent être contradictoires ou irréalisables. Le processus de *délibération* est le processus par lequel l'agent choisit les désirs qu'il va poursuivre (ses buts). Les intentions d'un agent sont donc les désirs qu'un agent souhaite accomplir. En d'autres termes, une intention est un but que l'agent s'engage à tenter de réaliser personnellement. Une architecture BDI permet d'assurer une certaine stabilité à l'agent. Les intentions permettent à l'agent de ne pas reconsidérer ses actions en permanence. En effet, une fois une intention adoptée, l'agent tente de la réaliser jusqu'à son accomplissement ou jusqu'à ce qu'elle devienne irréalisable. D'autres critères d'abandon peuvent bien entendu entrer en jeu. Notons que cette notion d'intention individuelle semble insuffisante pour caractériser les intentions « collectives » qui surgissent lors d'activités conjointes. Cela conduit à la considération de dérivé de l'intention comme l'« *intention que* » [Grosz 1996] (cf. section 2.1.2).

Il est important de distinguer les *plans* des *recettes* [Pollack 1990]. Une recette est une suite d'actions permettant de décomposer une action complexe en actions plus simples. Elle peut se présenter sous une forme hiérarchique constituée de buts et de sous-buts : c'est un arbre appelé graphe de recette. Elle correspond au *savoir-faire* de l'agent. Par exemple, je peux avoir une recette pour aller de mon appartement à la gare de ma ville qui consiste en une suite d'actions : aller à pied jusqu'à la station Joffre-Mutualité, prendre le métro direction Boulingrin, descendre à la station Gare Rue Verte, etc. Néanmoins, ce n'est pas parce que je connais cette recette que j'ai l'intention de la faire. La nuance est qu'un plan, version attitude mentale, réfère à l'adoption d'une intention qui guide les actions futures de l'agent (une sorte d'« engagement » à la réalisation d'une action future). Si je planifie d'aller à la gare de ma ville, j'ai alors cette intention. J'ai donc le plan de le faire (en plus de la recette précédemment décrite). Précisons que les premières approches par planification que nous présentons ne font pas la distinction (cf. sections 2.1.2 et 2.1.2) et considèrent les plans sous l'angle de la recette.

Dans le dialogue, la notion d'intention intervient au moins au niveau de l'acte illocutoire d'un acte de langage (cf. section 1.2.1). La notion d'*intention communicative* de Grice [Grice 1957] peut également entrer en jeu. Néanmoins, la distinction entre la *force illocutoire* et l'*intention communicative* ne fait pas consensus dans les approches intentionnelles [Maudet 2002a]. Dans tous les cas, une fois l'intention sous-jacente à l'acte du locuteur reconnue, il est nécessaire d'expliquer ce qui pousse l'interlocuteur à répondre.

Hypothèses de coopération

La notion de coopération dans le dialogue a été définie par Grice [Grice 1975]. Selon cet auteur, les interlocuteurs doivent se conformer au *principe de coopération* pour permettre l'échange d'information :

> « Que votre contribution conversationnelle corresponde à ce qui est exigé de vous, au stade atteint par celle-ci, par le but ou la direction acceptée de l'échange dans lequel vous êtes engagés. » [Grice 1975]

Les célèbres « maximes de Grice » sont issues de ce principe de coopération : les maximes de quantité (faites que votre contribution soit aussi informative que requise par le but de l'échange mais pas plus), de qualité (ne pas dire ce que vous pensez être faux, ou ce pour quoi vous manquez de preuve), de pertinence (soyez pertinent), de manière (évitez de vous exprimer de manière obscure, évitez l'ambiguïté, soyez brefs et ordonnés).

Ces maximes sont des règles « normatives » qui sont souvent transgressées. Leur transgression amorce un processus d'inférence chez l'interlocuteur appelé *implicature* qui permet de dépasser le sens littéral des énoncés. Prenons par exemple, un échange comme :

S1 : Tu viens à la soirée ce soir ?
H2 : J'ai du boulot.

Le deuxième énoncé viole la maxime de quantité. En reconnaissant cette transgression, S comprend que H ne viendra pas en supposant qu'il est coopératif.

Sperber et Wilson réduisent les maximes de Grice au principe de présomption optimale de pertinence [Sperber 1989]. La pertinence d'un énoncé est évaluée en fonction du rapport entre les effets qu'il produit et les efforts que la production de ces effets demande. Les éléments permettant d'évaluer cette pertinence sont néanmoins difficile à déterminer.

Structure intentionnelle

L'idée sous-jacente de toutes les approches intentionnelles est que la structure du dialogue résulte des intentions et de la coopérativité des interlocuteurs [Maudet 2002a,Pasquier 2005]. La *structure intentionnelle* a été exhibée et mise en relation avec deux autres structures par Grosz et Sidner [Grosz 1986]. Pour ces auteurs, la structure d'un dialogue peut se décomposer de la façon suivante :

— la *structure linguistique* (les énoncés agrégés en *segments* (DS pour *Dialogue Segment*)) ;
— la *structure intentionnelle* (ensemble des buts des segments du dialogue qui constituent des sous-buts contribuant au but global du dialogue. Le but d'un DS est un DSP pour *Dialogue Segment Purpose*) ;
— l'*état attentionnel* (focus dynamique du dialogue, i.e. les objets, propriétés et relations qui sont saillants à un moment donné du dialogue, en connexion avec les deux autres structures).

Les DSP sont vus comme des intentions au niveau du discours qui ont vocation à être reconnues à l'instar des intentions au niveau des énoncés (cf. section 2.1.1). Des exemples sont « avoir l'intention qu'un agent réalise une action », « avoir l'intention qu'un agent croie un fait », « avoir l'intention qu'un agent connaisse les propriétés d'un objet », etc.

Grosz et Sidner présentent deux relations possibles entre les DSP :

domination : DSP_1 *domine* DSP_2 si et seulement si la réalisation de DSP_2 contribue à la réalisation de DSP_1.

satisfaction-précédence : DSP_1 *pré-satisfait* DSP_2 si et seulement si la satisfaction de DSP_1 est une condition nécessaire à la réalisation de DSP_2.

Le dialogue entre deux participants (ou plus) est motivé par une tâche sous-jacente. La variété des tâches sous-jacentes au dialogue est importante. Elle peut être tout simplement « passer le temps », négocier, réaliser une activité collaborative comme « monter un meuble ensemble », etc. Le dialogue est alors *utilisé* pour l'activité [Clark 1996].

Nous pouvons nous interroger sur les liens entre la structure linguistique et la structure intentionnelle ainsi qu'entre la structure intentionnelle et la structure de la tâche sous-jacente. Pour Grosz et Sidner, la structure linguistique traduit la structure intentionnelle. Cette dernière se reflète dans la structure linguistique tout particulièrement dans les imbrications entre DS qui correspondent à des imbrications au niveau de la structure intentionelle. Quid des liens entre le structure intentionnelle et la structure de l'activité sous-jacente ? Grosz et Sidner sont très claires sur ce point : la structure intentionnelle n'est *ni identique*, *ni isomorphe* à la structure de l'activité sous-jacente. La raison en est que la structure intentionnelle contient des DSP qui sont *particuliers* à l'exécution de la tâche et du dialogue. Autrement dit, certains événements vont survenir pendant le dialogue, qui trouveront leur écho dans la structure intentionnelle mais pas dans la structure de la tâche (qui n'est pas capable de prévoir tous les cas). Un des exemples proposés par Grosz et Sidner est un dialogue orienté tâche où un expert tente d'apprendre à un apprenti à démonter un volant de voiture. Le plan de la tâche inclut des sous-plans pour trouver les vis de fixation, les dévisser, trouver un tournevis, etc. Néanmoins, le plan ne peut pas inclure des plans pour des évènements imprévus (que faire quand on ne trouve pas les vis ou que le seul tournevis disponible n'est pas adéquat ?).

2.1.2 Approches par planification

La théorie des actes de langage (cf. section 1.2.1) a naturellement conduit à envisager les actes de langage comme des actions dans un système de planification « classique ». En effet, elle permet de faire le lien entre d'une part *buts* et *intentions*, et *actions* et *dialogue* d'autre part.

Nous avons choisi de nous restreindre à la présentation de trois approches intentionnelles qui nous semblent représentatives : l'approche par plan « fondatrice » (cf. section 2.1.2), l'approche par plan face à la gestion du dialogue (cf. section 2.1.2), et enfin, l'approche par plan partagé (cf. section 2.1.2). En outre, toutes ces approches ont contribué à des systèmes interactifs avec un utilisateur humain. Compte tenu de la masse importante de références bibliographiques sur les approches intentionnelles, nous invitons le lecteur à se référer aux lectures suivantes s'il souhaite de plus amples détails [Maudet 2002a, Pasquier 2005]. En particulier, nous n'abordons pas les approches de la *théorie de l'interaction rationnelle* [Cohen 1990a, Sadek 1991]. Cette décision est motivée par le fait que ces approches sont trop complexes pour être mises en œuvre telles quelles dans des systèmes réels [Pasquier 2005]. En outre, les simplifications opérées pour leur implémentation font tendre ces approches vers celles de la planification « classique » [Pasquier 2005, Ferguson 2007].

Approche par planification classique

Les approches par planification s'inspirent de la formalisation de STRIPS [Fikes 1972]. Il s'agit de trouver le meilleur chemin dans un *espace d'états* pour passer d'un *état initial* à un *état final* en appliquant des *opérateurs* (i.e. des actions) permettant de passer d'un état à un

autre. Les états sont représentés par un ensemble de *faits* dans une représentation logique. Les actions sont définies par des *préconditions* (qui doivent être vérifiées afin de pouvoir appliquer l'action) et des *effets* (qui déterminent quels sont les faits qui sont ajoutés ou supprimés). À cela s'ajoutent un entête (un nom et un ensemble de paramètres) et un corps qui permet de décomposer une action en sous-action. Un *plan* est défini comme étant une séquence bien formée d'actions de telle sorte que les préconditions des actions sont atteintes par les effets des actions précédentes.

Le modèle pionnier de l'utilisation de la planification pour la modélisation du dialogue est celui de Allen, Cohen et Perrault [Cohen 1979, Allen 1980]. Il se base sur la capacité du locuteur à reconnaître le plan de son interlocuteur afin de produire la réponse la plus pertinente. Dans cette approche, les interlocuteurs sont dotés d'états mentaux (buts, et différents types de croyances). Les actes de langage sont formalisés comme n'importe quelle autre action. Le tableau 2.1 présente un exemple de l'acte « Informer » présenté avec le formalisme d'une action de plan (L est le locuteur, I l'interlocuteur et P la proposition).

INFORMER(L, I, P)	
Préconditions :	L VEUT INFORMER(L, I, P) L SAIT P
Effets :	I SAIT P
Corps :	L DIRE P à I

TABLEAU 2.1: Exemple de représentation d'un acte "Informer" comme une action de plan. L est le locuteur, I l'interlocuteur et P la proposition.

Dans cette approche, la reconnaissance du plan par l'interlocuteur peut être considérée comme un processus de planification inversée. L'interlocuteur reconnaît le plan à partir de la connaissance des opérateurs et des règles de construction de plan :

— L VEUT *Action* ⇒ L VEUT *Effet*

— L VEUT *Préconditions* ⇒ L VEUT *Action*

— L VEUT *Corps* ⇒ L VEUT *Action*

La reconnaissance du plan du locuteur inclut la *reconnaissance à l'insu*. Celle-ci implique l'application de règles d'inférence basées sur ces règles de construction. Par exemple (où *Pred* est une précondition de l'action *Act*) :

$$(\text{I CROIT (L VEUT } \textit{Pred})) \Rightarrow (\text{I CROIT (L VEUT } \textit{Act}))$$

Cette règle d'inférence signifie que si L veut obtenir les préconditions de l'action *Act*, c'est qu'il souhaite sans doute réaliser *Act*.

Une fois que le but et le plan du locuteur ont été inférés, l'interlocuteur peut construire un plan permettant d'aider à la réalisation de ce but. La coopérativité de l'interlocuteur est ainsi assimilée à l'*adoption de but*.

La reconnaissance de plan en cours de conversation permet de déterminer des réponses coopératives. Par exemple, la question « À quelle heure est la prochaine séance pour le film "Superman" ? » laisse supposer que le locuteur possède le but d'aller voir le film en question. Pour cela, il possède un plan lui permettant de déterminer l'horaire de la prochaine séance mais aussi le numéro de la salle. La reconnaissance de ce plan de la part de l'interlocuteur lui permet

de générer une réponse coopérative du type : « La prochaine séance est à 17h50. Le film sera diffusé en salle 12. ».

Cette approche admet deux limites. La première est que la reconnaissance se limite à l'analyse d'un seul énoncé. Or la reconnaissance de plan repose plus généralement sur plusieurs énoncés [Carberry 1990]. Ensuite, la structure intentionnelle est identique à la structure de la tâche sous-jacente. Cependant, nous avons vu que ces deux structures n'étaient pas isomorphes (cf. section 2.1.1). Cette deuxième critique nous conduit à considérer l'approche de Litman et Allen qui tente de différencier les apports du domaine et les apports spécifiques à la gestion du dialogue.

Approche par plans du domaine et du discours

Afin de prendre en compte la non-isomorphie entre la structure de la tâche et la structure intentionnelle (e.g., par l'intervention de dialogue de clarification), Litman et Allen proposent de différencier deux types de plans [Litman 1990]. D'un côté, il s'agit de considérer les *plans du domaine* qui modélisent la tâche sous-jacente. De l'autre côté, il s'agit de considérer les *plans du discours* qui sont des meta-plans permettant de manipuler la structure des autres plans.

Les auteurs listent trois types de relation que peuvent entretenir les plans : la relation de continuation (qui permet de commencer l'exécution du plan du domaine ou de poursuivre son déroulement), la relation de clarification (qui permet d'identifier un paramètre du plan ou de proposer une correction au plan courant), et la relation de changement de sujet (qui permet d'introduire un nouveau plan). Ils proposent alors cinq meta-plans de dialogue issus de ces trois relations : suivre-plan, identifier-paramètre, corriger-plan, introduire-plan et modifier-plan.

La structure du dialogue est modélisée par une pile de plans du domaine et de meta-plans. La pile globale est composée de sous-piles pour chaque étape du plan du domaine. Chaque nouveau plan introduit est soit un plan correspondant à un nouveau pas de plan du domaine, soit un meta-plan relié au dernier plan introduit. De cette manière, il est possible de gérer les demandes de clarification emboîtées.

Ce modèle permet de gérer des dialogues qui ne sont pas directement dans le plan du domaine. Par exemple, via la relation de clarification, il est possible de gérer des dialogues comme :

S1 : Où peut-on assister à la projection de « Superman » ?
H2 : Dans la salle au fond à droite.
S3 : Quelle salle ?
H4 : La salle 12 située au fond à droite.

Les tours 3 et 4 constituent un sous-dialogue de clarification. Ce sous-dialogue ne fait pas partie d'un plan du domaine. Il s'agit d'identifier un paramètre du plan en cours (ici la salle où le film est projeté). Il s'agit donc d'un empilement d'un meta-plan identifier-parametre relié par une relation de clarification au plan du domaine.

Il est cependant possible de trouver des interventions dialogiques de telle manière qu'elles ne soient ni une étape de plan de domaine, ni reliée les unes aux autres par les relations proposées. Par exemple :

S1 : Je désire voir un film. Quand commence la séance pour « Superman » ?
H2 : À 19h30.
S3 : Et pour « Batman » ?
H4 : À 19h50.

Ici l'interlocuteur souhaite évaluer plusieurs alternatives, i.e. plusieurs plans à sa disposition. Des

extensions considérant un troisième niveau de *résolution de problème* ont été proposées [Lambert 1991].

Approches par planification collaborative

Grosz et Kraus ont présenté dans leur article « Collaborative plans for complex group action » un formalisme de plans pour l'action d'un groupe d'agents collaboratifs [Grosz 1996]. En partant des travaux de Pollack [Pollack 1990], ce formalisme spécifie deux types de plans définis en terme d'attitudes mentales (croyances et désirs) : les *plans partagés* et les *plans partiels*. Elles introduisent la notion d'*intention que* qui permet de représenter l'engagement de chaque participant envers le but commun.

Comme le nom le laisse à penser, les *plans partagés* sont des plans qui incorporent les actions de plusieurs participants. Le processus de planification est rendu d'autant plus compliqué que le nombre d'agents pris en compte dans le plan est important. Pour simplifier ce problème, les agents ont généralement une base de données de plans pré-calculés, appelés *recettes*, dans laquelle ils peuvent sélectionner des plans sans avoir besoin de tout recalculer. Certains aspects ont besoin d'être établis avant la réalisation de l'activité conjointe par un plan partagé. Les participants doivent :

1. se mettre d'accord sur le but conjoint à atteindre ;
2. se mettre d'accord sur l'ensemble des recettes possibles (i.e., le *par quel moyen* réaliser l'action) ;
3. se mettre d'accord sur la répartition des tâches et l'ordonnancement de la réalisation (i.e., le qui va faire quoi quand) ;
4. et enfin, s'engager sur la réussite des actions des autres participants selon les paramètres précédemment sélectionnés.

Cette approche permet de justifier les comportements coopératifs dans l'activité sans référer à une notion d'intention collective [Searle 1990]. L'aspect collectif de l'activité est capturé par le plan partagé et la notion d'*intention que*. Néanmoins, ces plans partagés doivent être adoptés par les participants. Une hypothèse de coopération doit être faite : il est nécessaire de supposer une entente pour élaborer un plan partagé.

Il est d'autant plus difficile de calculer à long terme des plans partagés que les actions dépendent d'autres participants. Si on ajoute à cela un environnement dynamique, comme par exemple le dialogue, il devient clair que les agents ne peuvent pas planifier « du premier coup » à long terme. Cela a conduit à l'introduction de la notion de *plans partiels* qui vont permettre de représenter partiellement les actions à entreprendre et qui ont vocation à être complétés au fur et à mesure. De la même manière, les agents peuvent avoir des *recettes partielles* qui sont des spécifications partielles pré-calculées de combinaisons d'actions pour atteindre un certain but.

Résumons la situation. L'approche par planification collaborative permet de distinguer les habituels plans « individuels » des plans partagés. Ce sont ces plans partagés qui permettent de capturer la dimension collective de l'activité. En d'autres termes, les participants collaborent sur la mise en place d'un plan partagé qui forme alors l'activité conjointe. Néanmoins, les participants ne sont pas contraints de tout prévoir à l'avance car ils peuvent manipuler des plans partiels, i.e. des plans qui ne sont pas complètement spécifiés. Et pour le dialogue ? L'idée est que les interlocuteurs s'embarquent dans une interaction avec un plan partiel, qu'il s'agit de préciser au fur et à mesure de l'interaction.

L'application de cette théorie au dialogue a été étudiée par Lochbaum [Lochbaum 1994, Lochbaum 1998]. Son point de vue est que les interlocuteurs dialoguent pour compléter des plans partiels (individuels ou partagés, au niveau du discours ou au niveau de la tâche). Cette approche considère l'intervention de sous-dialogues comme étant soit *contributoire* à un autre plan (e.g., un sous-dialogue pour s'accorder sur une recette), soit la *satisfaction* d'un plan est nécessaire à l'exécution de l'autre (pré-satisfaction). Lochbaum considère notamment que les sous-dialogues de clarification sont des sous-dialogues qui doivent être satisfaits afin de déterminer un paramètre pour le plan suivant. Selon Lochbaum, les plans partagés modélisent exactement les DSP (cf. section 2.1.1) en un seul et même formalisme.

Les modèles intentionnels sont à la source de la mise en œuvre de systèmes d'interaction Homme-Machine sur lesquels nous allons maintenant nous pencher.

2.1.3 Principales mises en œuvre

Une grande force des approches intentionnelles est d'apporter une réponse globale unifiée autour de la notion d'intention à plusieurs problématiques du dialogue. En schématisant, nous pouvons résumer la situation de la manière suivante. Un énoncé est vu comme une action « classique » grâce à la théorie des actes de langage. L'interprétation d'un énoncé est vue comme l'inférence de son intention sous-jacente. Et la réponse est générée par un mécanisme de planification associé à des hypothèses de coopération. Ce cadre a permis la réalisation de nombreux systèmes qui ont une forte influence dans le domaine de l'interaction Homme-Machine comme les systèmes TRAINS [Ferguson 1996, Allen 2000] et Collagen [Rich 2001].

Les systèmes fondateurs

Le système TRAINS [Ferguson 1996, Allen 2000] a été l'un des premiers systèmes à gérer un dialogue avec un humain dans le cadre de la résolution collaborative de problème. Il est issu d'un travail important de récolte et d'analyse de corpus de dialogues oraux orientés tâche. Il a conduit à la conception de plusieurs systèmes dénommés TRAINS servant à interroger en langue naturelle une base de données contenant des horaires de trains. Le système TRIPS [Ferguson 1998], successeur de TRAINS, est conçu pour assister un humain dans une tâche de gestion de situations de crise. Le système et l'humain collaborent à la construction de plans d'évacuation. Ce système ajoute au dialogue naturel l'utilisation de graphiques (cartes, tableaux, etc.) pour améliorer la compréhension mutuelle. En se basant sur ces travaux, Ferguson et Allen ont récemment proposé une approche pour la conception et la mise en œuvre d'agent assistant pour la résolution collaborative de problèmes basée sur une architecture BDI [Ferguson 2007].

Les travaux sur la planification collaborative et son application à la modélisation du dialogue (cf. section 2.1.2) ont conduit à la réalisation de Collagen (*Coll*aborative *Agent*) [Rich 2001]. Ce système inclut un *modèle de la tâche* basé sur la planification collaborative [Grosz 1996] ainsi qu'un *modèle de l'état de l'interaction* référençant les croyances et intentions des interlocuteurs (cf. état attentionnel du dialogue, section 2.1.1). Ces deux modèles sont ensuite exploités par les algorithmes d'interprétation [Lochbaum 1998] et de génération de dialogue. Collagen est utilisé en tant que *middleware* pour la réalisation de systèmes collaboratifs en interaction avec un utilisateur humain. Par exemple, il a été utilisé pour la réalisation de DiamondHelp [Rich 2007], un système collaboratif permettant d'assister un utilisateur sur une tâche. Ce système est illustré sur des tâches d'assistance d'un utilisateur telles que la programmation d'une machine à laver moderne ou la configuration d'un thermostat.

L'émergence de nouveaux systèmes inspirés de l'approche par planification

Plus récemment, de nouveaux systèmes s'inspirant de l'approche par planification ont émergé. Nous citons RavenClaw et DTask.

RavenClaw [Bohus 2009b] est un gestionnaire de dialogue indépendant de la tâche basé sur une approche par planification. La gestion du dialogue est opérée de manière indépendante de la tâche sur la base d'un réseau hiérarchique de tâches [1]. Les aspects spécifiques à la tâche sont spécifiés via ce plan hiérarchique. Ce dernier inclut la planification des actions du système (e.g., informer l'utilisateur, rechercher dans une base de données, etc.) ou des actions bilatérales à l'initiative du système telles qu'une demande d'information. RavenClaw intègre des processus de gestion de l'interaction indépendant de la tâche sous-jacente au dialogue. Ces processus sont principalement liés à la reconnaissance vocale. Par exemple, ils permettent au système de demander des confirmations ou des reformulations, tandis que l'utilisateur peut demander au système de répéter les derniers énoncés. En somme, RavenClaw permet de gérer des dialogues fortement liés à une tâche pouvant être décrite simplement par un plan hiérarchique.

DTask [Bickmore 2009] est un gestionnaire de dialogue conçu pour modéliser et exécuter des dialogues *menés par le système* fournissant des *entrées à choix multiples* pour saisir les contributions de l'utilisateur humain. Dans le même esprit que RavenClaw, la structure du dialogue est liée à la structure de la tâche sous-jacente décrite par un plan décomposant les buts et sous-buts des participants. Le point qui nous intéresse particulièrement dans cette approche est que la plus petite tâche conjointe dialogique est décrite sous la forme d'une *paire adjacente*. Celle-ci consiste en un énoncé du système associé à une liste de réponses possibles de l'utilisateur humain. Le tableau 2.2 présente un exemple de spécification d'une paire adjacente tirée de [Bickmore 2009].

Utilisateur	Énoncé
Système	How are you ?
Utilisateur	1. I am good. How are you ? 2. Good.

Tableau 2.2: Exemple de spécification d'une tâche dialogique de salutations dans DTask [Bickmore 2009]

Cette paire décrit un échange de salutations ritualisé entre les interlocuteurs. Elle consiste en la production de l'énoncé initiatif « How are you ? » par le système. L'utilisateur est alors confronté à un choix : il peut choisir de répondre par une des deux propositions présentées dans le tableau 2.2. Notons que la réponse 1 (« I am good. How are you ? ») clôture la paire par l'énonciation de « I am good. » et initie une nouvelle paire par « How are you ? » (bien que cela ne soit pas explicitement modélisé de cette manière dans cette approche). Cette nouvelle initiation peut être suivie par une réponse du système telle que « Great. Thanks for asking ! » [Bickmore 2009]. L'approche DTask a été récemment étendue dans Disco for Games (D4g) [Rich 2012], présenté comme le successeur de Collagen. Dans cette dernière itération, le système ne se limite pas aux paires adjacentes mais intègre des sous-dialogues complets spécifiques à l'application.

Maintenant que nous avons vu les principaux modèles intentionnels et leurs mises en œuvre plus ou moins directes, intéressons-nous aux avantages et limites d'une telle approche.

1. De l'anglais « Hierarchical Task Network » (HTN).

2.1.4 Avantages et limites des approches intentionnelles

Les approches intentionnelles possèdent de nombreux avantages [Maudet 2002a, Pasquier 2005]. Tout d'abord, les approches intentionnelles ont la capacité de produire des réponses *coopératives* qui devancent les attentes de l'interlocuteur en se basant sur la reconnaissance du plan sous-jacent à l'interaction (e.g., « – Vendez-vous des billets de train pour Lille ? – Tout à fait ! Ils coûtent 20 euros. »). Ensuite, en analysant la partie du plan de l'interlocuteur qui est erronée, le système est capable d'établir une stratégie de reprise pertinente vis-à-vis du but de l'interlocuteur [Pollack 1990] (e.g., « – Je vais voir "Superman". Donnez moi un billet pour la salle 10. – La salle 10 ne diffuse pas "Superman". Vous devriez prendre un billet pour la salle 12. »). Enfin, ces approches ne dépendent pas de règles issues d'observations empiriques sur la structuration des conversations qu'il est souvent facile de mettre en défaut. En ce sens, elle dispose d'une grande flexibilité au prix peut-être d'une dépendance au domaine renforcée.

Un ensemble de critiques a été émis à l'encontre des approches intentionnelles [Traum 1994b, Clark 1996, Maudet 2002a, Pasquier 2005]. Parmi toutes ces critiques, nous retenons celles qui sont liées à un point de vue théorique et celles liées à un point de vue pratique.

D'un point de vue théorique, le rôle central de l'intention dans ces approches est à remettre en cause. En effet, la plupart des dialogues présente des phases d'interaction ritualisée qui ont été mises en avant par l'analyse de la conversation (e.g., remerciements, échange de salutations, etc.). La reconnaissance de l'intention sous-jacente semble exagérée dans ces cas. Ceci soulève la question de la pertinence des attitudes mentales considérées (croyances, désirs et intention) et de leur adéquation pour modéliser complètement le comportement conversationnel d'un système. Ces attitudes ne permettent pas de prendre convenablement en compte le caractère collectif et social du dialogue. Ainsi, elles ne suffisent pas pour expliquer des réponses telles que « Je ne sais pas. » à une question. Deux solutions principales sont considérées. D'une part, une stratégie consiste à augmenter ces attitudes d'autres notions comme les *pressions interactives* [Bunt 1996] ou les *obligations* [Traum 1994b]. D'autre part, il s'agit de considérer le dialogue comme une *activité conjointe* [Lochbaum 1994, Grosz 1996]. Néanmoins, cette approche du problème est sujette à controverse sur la notion d'intention collective et de l'introduction de dérivés de l'intention [Searle 1990, Grosz 1996]. De plus, les formalismes proposés pour formaliser la notion d'action conjointe (où l'action des participants est expliquée par la participation à l'activité jointe) restent limités à la considération d'une activité collaborative. En conséquence, elles ne permettent pas de prendre en compte les situations dialogiques conflictuelles comme le débat où il n'y a pas de but commun établi entre les interlocuteurs [Traum 1994b]. Notons que doter un système de dialogue coopératif de capacités lui permettant de rentrer dans un débat avec un utilisateur peut avoir un intérêt. Par exemple, un agent assistant peut essayer de convaincre son utilisateur que la date qu'il a choisi pour un rendez-vous n'est pas la plus adaptée. En somme, les approches intentionnelles n'indiquent pas comment caractériser le fait que les interlocuteurs prennent part à des activités conjointes (collaboratives ou non) dans lesquelles ils peuvent compter sur certaines actions les uns des autres.

Enfin, la dernière critique tient en la nature opportuniste du dialogue [Clark 1996] (cf. section 1.1.1). La vision opportuniste défend l'idée selon laquelle le dialogue n'est pas une activité planifiée : le dialogue ne semble planifié qu'*a posteriori*. Les participants s'engagent dans le dialogue sans savoir à l'avance ce qu'ils vont y faire. Bien entendu, cela n'empêche pas le fait que les participants entretiennent des objectifs préalables à la conversation. La raison principale de la nature opportuniste du dialogue est qu'il est co-construit et co-géré par ses participants. En

conséquence, un participant ne peut rien faire sans l'autre. Compte tenu du fait qu'un participant ne peut pas savoir à l'avance ce que l'autre va faire, le dialogue ne peut pas être planifié. Le dialogue semble donc être une activité conjointe nécessitant quasiment d'être replanifiée au tour par tour, réduisant ainsi l'intérêt de la planification. Cette observation est corroborée par l'observation que certaines séquences d'actes ne peuvent pas être planifiées [Pulman 1998] (typiquement, les demandes de clarifications). Une réponse partielle à ce problème a été apportée par les plans partagés [Lochbaum 1994, Grosz 1996] qui ont vocation à être complétés par les participants.

D'un point de vue pratique, il est indispensable de préciser que les approches intentionnelles sont difficiles à mettre en œuvre dans des systèmes réels. D'une part, le mécanisme de reconnaissance de plans est très complexe et difficile à mettre en place [Cohen 1997]. Qui plus est, les algorithmes de reconnaissance de plans sont combinatoirement intractables dans le pire des cas et indécidables dans certains cas [Bylander 1991]. Enfin, les approches par plans sont dépendantes du domaine. Il est indispensable de définir les recettes couvrant l'ensemble des cas possibles.

2.2 Approches conventionnelles

Dans cette section, nous abordons l'aspect conventionnel de certaines approches de modélisation du dialogue pour l'interaction Homme-Machine. Là où les approches intentionnelles interprètent les motifs d'interaction comme un épiphénomène des intentions des interlocuteurs, les approches conventionnelles les voient comme des *conventions établies* et ne se focalisent pas sur les intentions sous-jacentes. Dans un premier temps, nous présentons les fondements des approches conventionnelles (cf. section 2.2.1). Ensuite, nous abordons les approches étudiant la structuration des motifs d'interaction et qui visent à produire des règles décrivant des séquences admissibles de types d'énoncés (cf. section 2.2.2). Puis, nous voyons les modèles fondés sur les questions en discussion qui illustrent l'ensemble des concepts des approches conventionnelles (cf. section 2.2.3). Enfin, nous discutons les avantages et limites de ce type de modèle (cf. section 2.2.4).

2.2.1 Fondements des approches conventionnelles

Alors que les approches intentionnelles se focalisent sur la « partie privée » en s'intéressant aux attitudes mentales des interlocuteurs, les approches conventionnelles se concentrent sur ce qui est *partagé* par les participants durant le dialogue. En d'autres termes, les approches conventionnelles s'intéressent à la partie « publique ». C'est généralement ce dont il est question lorsqu'on aborde la notion de *fonds commun*.

Fonds commun et tableau de conversation

La notion de fonds commun L'idée de *fonds commun*[2] remonte à Stalnaker [Stalnaker 1979] et Lewis [Lewis 1979]. Stalnaker utilise le concept de fonds commun qui garde une trace de l'état courant du dialogue. Il est vu comme un ensemble non structuré de propositions enrichi par les assertions des interlocuteurs. Ainsi, un interlocuteur réalisant une assertion modifie le fonds commun (e.g., en ajoutant la proposition). Lewis, quant à lui, dresse un parallèle entre

2. Traduction de l'anglais « common ground », également traduit « terrain commun »

le dialogue et le baseball. Il imagine le fonds commun (nommé « conversational scoreboard » pour l'occasion) comme un tableau permettant de garder une trace de l'interaction dialogique au même titre qu'un panneau d'affichage sur un terrain de sport.

Le fonds commun pour le dialogue est généralement vu comme incluant un *fonds commun général* établi par le contexte global (cf. section 1.2.2) et un *fonds commun conversationnel* (en lien avec le contexte local, cf. section 1.2.2) qui est établi au cours du dialogue par le processus d'établissement (le *grounding*).

Plusieurs formalisations du fonds commun existent dont certaines sont sujettes à controverse sur leur plausibilité cognitive [Clark 1996]. La formalisation communément établie est celle de *base partagée* [Lewis 1969] :

> p est une information commune pour les membres d'une communauté C si et seulement si :
>
> 1. Chaque membre de C possède l'information que la base b est établie.
> 2. b indique à tous les membres de C que tous les membres de C possèdent l'information que b est établie.
> 3. b indique à tous les membres de C que p.

La notion de base partagée doit être vue comme une situation commune à partir de laquelle il est possible de dériver certaines informations, et que ces informations sont également partagées avec les membres de la communauté C. Ainsi, si je me trouve à un concert de rock avec une amie, et qu'un groupe s'installe sur la scène, je peux déduire de la situation que le fait « il y a un groupe sur la scène » fait partie du fonds commun.

Soulignons que les informations contenues dans le fonds commun ne sont pas exactement des *informations partagées* mais plutôt des *informations supposées partagées*. C'est pourquoi les versions du tableau de conversation des interlocuteurs peuvent varier au cours du dialogue.

Tableau de conversation En terme de modélisation du dialogue par les approches conventionnelles, il est plus courant de parler de *tableau de conversation*. C'est la partie du fonds commun qui représente l'état du dialogue entre les participants. Le contenu du tableau de conversation inclut des « éléments classiques » comme l'historique du dialogue (ce qui a été dit). D'autres éléments peuvent y être inclus comme certains éléments du contexte local (cf. section 1.2.2). Le contenu du tableau dépend des éléments pris en compte dans la modélisation du dialogue.

Lewis avance l'idée d'*accommodation* pour décrire le principe selon lequel le tableau de conversation peut évoluer de manière à rendre correct un coup dialogique (i.e., un acte contextuel). Imaginons un tableau de conversation contenant une représentation des questions en cours de discussion entre les interlocuteurs sous forme d'une pile. Il est possible qu'un interlocuteur choisisse de répondre non pas à la dernière question posée (au sommet de la pile) mais à l'avant dernière question. Une accommodation de tableau est alors effectuée afin de faire passer l'avant dernière question en tête, de manière à l'interpréter comme la question répondue (cf. section 2.2.3).

Contexte et tableau de conversation Les liens entre contexte dialogique (cf. section 1.2.2) et tableau de conversation nécessitent d'être précisés. Le tableau de conversation réfère généralement à la *partie publique* (ou partagée) de l'*état d'information* qui modélise le contexte dialogique. À ce sujet, Ginzburg présente cinq types de structuration possibles du *contexte dialogique*

en terme d'informations publiques et privées intervenant dans le comportement conversationnel des interlocuteurs (A et B représentent les interlocuteurs) [Ginzburg 2012] (p. 64)[3] :

1. A : ⟨A.private⟩, B : ⟨B.private⟩
2. A : ⟨public⟩, B : ⟨public⟩
3. A : ⟨A.private, A.B.private⟩, B : ⟨B.private, B.A.private⟩
4. A : ⟨public, A.private⟩, B : ⟨public, B.private⟩
5. A : ⟨A.public, A.private⟩, B : ⟨B.public, B.private⟩

Ces modèles permettent de distinguer les éléments privés (e.g., attitudes privées) et publics (e.g., tableau de conversation) pris en compte dans la modélisation du dialogue. Le modèle 1 renie l'existence d'une partie publique. C'est un extrême où le comportement dialogique des agents n'est expliqué que par leurs propres attitudes privées. En conséquence, le tableau de conversation n'existe pas. Le modèle 2 considère l'autre extrême : le comportement dialogique est uniquement expliqué par des facteurs publics. Le contenu du tableau de conversation est entièrement partagé entre les interlocuteurs. C'est ce dernier qui explique tous les comportements dialogiques. Le modèle 3, comme le premier modèle, renie l'existence d'une partie publique. Les agents ne considèrent que leurs attitudes privées et leur représentation des attitudes privées de leur interlocuteur. Il s'agit de la vision communément admise dans les approches intentionnelles. Le modèle 4 divise les facteurs jouant sur le comportement dialogique entre une partie publique partagée entre les interlocuteurs et une partie privée. La partie publique représente le « conversational scoreboard » de Lewis. Enfin, le dernier modèle relativise la partie publique du point de vue de l'agent : le contenu du tableau de conversation peut admettre des variations. Le comportement dialogique d'un agent est expliqué par *sa* représentation de la partie publique et ses attitudes privées. La différence principale entre les approches 4 et 5 réside dans la posture adoptée face à l'ajout d'une information incertaine dans le tableau conversationnel. Dans la première approche, il s'agit de n'enregistrer que les informations établies. Dans la seconde, il s'agit d'enregistrer toutes les informations en prévoyant des mécanismes de révision. Cela nous amène à nous intéresser au mécanisme d'établissement de l'information (le *grounding*).

Mécanisme d'établissement

Le processus par lequel le fonds commun est établi et par lequel de nouvelles informations intègrent le fonds commun est appelé *grounding* [Clark 1989]. L'objectif de ce processus est d'établir ce qui a été dit comme étant *public*. Cela inclut la forme d'un énoncé, sa fonction, son contenu et ses implications. Ce processus conduit les interlocuteurs à produire des *feedbacks* (cf. section 1.1.2).

Dans ce processus, les interlocuteurs essayent d'atteindre un critère – le critère de *grounding* [Clark 1989] – qui spécifie qu'ils doivent atteindre un niveau où le locuteur et l'allocutaire croient que l'allocutaire a compris l'énoncé de manière suffisante pour le but courant. Pour cela, les interlocuteurs réalisent des *contributions* qui se décomposent en deux phases (A et B sont les interlocuteurs) :

Phase de présentation A produit un énoncé u vers son interlocuteur B. A attend alors un *indice* de la part de B qui peut lui faire croire que B a compris ce qu'il voulait dire par u.

3. A.B.private est la représentation de A des attitudes privées de B (et réciproquement pour B.A.private).

Phase d'acceptation B accepte l'énoncé en montrant qu'il a compris (ou pas) ce que A veut dire par u. B suppose alors qu'une fois que A enregistre l'*indice*, il croira que B a compris (ou pas).

Ce n'est qu'après la phase d'acceptation que l'énoncé est ajouté dans le fonds commun. Précisons que la phase d'acceptation ne suit pas nécessairement de manière immédiate la phase de présentation. Des contributions peuvent venir s'intercaler entre les deux phases. Les indices produits peuvent être *positifs* ou *négatifs*. Concernant les indices positifs, Clark en propose une liste hiérarchisée (du moins fort au plus fort) : (a) B montre qu'il continue d'être attentif. (b) B produit un énoncé pertinent à la suite. (c) B acquiesce. (d) B reformule ou complète l'énoncé de A. (e) B répète l'énoncé de A. De même pour les indices négatifs : (a) B engage un dialogue de clarification. (b) B répète l'énoncé en faisant une erreur. (c) B n'est plus attentif. (d) B produit un énoncé non pertinent à la suite. (e) B déclare ne pas comprendre. Notons que le processus est récursif : une phase d'acceptation est elle-même une phase de présentation pour une nouvelle contribution. Pour éviter une récursion infinie, les auteurs invoquent le principe selon lequel les indices diminuent progressivement de force.

Ce modèle a reçu beaucoup d'attention en intelligence artificielle. Pour plus de détails, voir [Traum 1992, Traum 1994a, Poesio 1998b, Traum 1999] dont nous avons présenté un aperçu en section 1.2.2.

Le fonds commun permet l'établissement de régularités au sein d'une communauté généralement désignées par le terme *convention*.

Conventions et motifs d'interaction

Une *convention* est une règle générale adoptée par une communauté pour la résolution d'un problème de coordination [Lewis 1969]. Par exemple, se serrer la main est une solution conventionnelle dans une communauté au problème récurrent de comment se saluer. En d'autres termes, une convention est une régularité qui existe au sein d'une communauté, sans avoir nécessairement fait l'objet d'un accord explicite. Ces régularités font partie du *fonds commun* des membres de cette communauté [Clark 1996]. Les régularités qui nous intéressent sont les *motifs d'interaction* définis en introduction de ce document.

Au contraire des approches intentionnelles où le lien provient de l'intention et des hypothèses de coopération, on suppose ici l'existence d'un lien conventionnel entre les énoncés. Le problème n'est pas de savoir comment ces régularités sont apparues. On suppose leur *existence* (i.e., leur appartenance au fonds commun) et on suppose que les *interlocuteurs sont prêts à s'y conformer*. Ces régularités présentent un intérêt simplificateur pour la modélisation du dialogue. Dans une régularité de ce type, il n'est par exemple pas nécessaire de réaliser des analyses complexes sur l'intention sous-jacente.

Nous nous intéressons en premier lieu aux approches qui se concentrent sur les propriétés structurelles de ces régularités.

2.2.2 Modèles fondés sur la structuration

Nous débutons cette section par les grammaires de dialogue qui cherche à déterminer la structure du dialogue en terme de constituants. Puis nous traitons des approches par protocole qui sont généralement utilisées afin de modéliser les motifs d'interaction.

Grammaires de dialogue

La modélisation du dialogue fondée sur la structuration cherche à déterminer la structure sous-jacente du dialogue en terme de *constituants*. L'idée est alors de modéliser les liens que ces constituants entretiennent. Cette idée est analogue à la décomposition grammaticale d'une phrase (e.g., de type « Sujet + Verbe + Complément »). Ici, on cherche à faire la même chose mais avec le dialogue. L'idée est donc de trouver des *grammaires de dialogue* [Polanyi 1984, Roulet 1987].

Le modèle genevois est l'exemple par excellence du modèle structurel du dialogue humain [Roulet 1987, Moeschler 1989]. Il définit un cadre strict d'analyse en constituants hiérarchiques et d'analyse fonctionnelle. Les constituants du dialogue considérés sont les suivants (du plus simple au plus complexe) :

— L'*acte* est l'unité d'analyse minimale (cf. section 1.2.1).

— L'*intervention* est constituée d'au moins un acte. Elle est dite *simple* si elle ne contient qu'un acte, sinon elle est dite *composée*.

— L'*échange* est la plus petite unité interactive. Il est constitué au maximum de trois interventions. La première intervention possède la fonction *initiative*, la deuxième *réactive* et l'éventuelle troisième *évaluative*.

— La *séquence* est constituée d'un ou plusieurs échanges reliés par un fort degré de cohérence sémantique et/ou pragmatique.

— L'*interaction* est composée d'une ou plusieurs séquences. Elle est délimitée par la rencontre et la séparation des deux interlocuteurs.

En plus des règles structurelles qui définissent les enchaînements possibles des constituants du dialogue (e.g., sous forme de grammaire hors contexte), le modèle ajoute des *fonctions illocutoires* (initiative, réactive, évaluative, ...).

Le système de dialogue Homme-Machine SUNDIAL [Bilange 1991b] exploite le modèle structurel genevois. L'idée de ces travaux est d'utiliser les structures proposées pour construire un contexte du dialogue. Ce contexte de dialogue est ensuite exploité pour prédire les actes qui peuvent être joués par le système ou l'utilisateur à un moment donné de la conversation. Plus précisément, une initiative ouvre un échange. L'interlocuteur est alors mis en situation de réaction. Cela le pousse à produire une intervention réactive qui peut elle-même conduire l'initiateur de l'échange à produire une intervention évaluative. C'est un modèle en trois temps initiatif/-réactif/évaluatif. La force de ce système est d'isoler dans son architecture un module du dialogue, qui représente la structure courante du dialogue, des autres modules (tâche, linguistique, etc.). En tenant compte de la structure du dialogue, le module du dialogue produit des descriptions des prochains énoncés possibles de la part du système et des prédictions à propos des prochains énoncés possibles de la part de l'utilisateur. Néanmoins, le module de la tâche conserve un rôle important dans ces étapes de prédictions et de générations. En outre, aucune contrainte sur le contenu des actes de dialogue n'est représentée dans la structure du dialogue.

Protocoles de communication

Définition Un protocole de communication est un type de *politique de conversation* définie comme un ensemble de « contraintes générales sur les séquences de messages sémantiquement cohérents menant à un but » [Greaves 2000]. Un protocole spécifie des séquences d'actes communicatifs attendus en fonction de l'état de la conversation pour la réalisation d'une tâche précise.

Le formalisme généralement adopté pour représenter un protocole est l'automate à états finis (e.g., le célèbre « Request for Action » de Winograd et Flores [Winograd 1986]). Dans un tel automate, les états représentent l'état de la conversation et les transitions représentent les actes communicatifs réalisables.

Un protocole de communication peut être vu comme un *projet conjoint conventionnel* entre les interlocuteurs [Pasquier 2005]. Un projet conjoint est une activité conjointe proposée par l'un de ses participants et acceptée par le reste des participants (cf. section 1.1.1). Un tel projet possède une phase d'entrée, un corps et une sortie. Chaque participant doit s'engager dans l'action conjointe. Une fois que cela est fait, les participants exécutent leurs actions participatives. Sous cet angle, un protocole peut être vu comme une spécification d'un enchaînement d'activités participatives (ou d'autres actions conjointes). Notons que les phases d'engagement et de désengagement de l'activité conjointe sont généralement contraintes par la tâche qui implique l'usage du protocole.

Les protocoles de communication sont particulièrement utilisés dans le domaine de la communication multiagent (voir, e.g., KQML [Finin 1994] et FIPA-ACL [FIPA 2000]). Certains protocoles ont été utilisés pour la modélisation des interactions Homme-Homme. Nous présentons l'une de ces approches.

Modéliser les communications humaines par des protocoles Dans le cadre de la modélisation d'interactions humaines pour la planification multiagent, Pauchet a proposé un modèle de l'interaction basé sur l'utilisation d'automate temporisé [Pauchet 2006]. Le modèle d'interaction se base sur l'étude d'un corpus d'interactions Homme-Homme (constitué d'échanges de mails pour une tâche de réservation de voyage). Cette étude a fait apparaître la présence de motifs d'interactions appelées *échanges* qui se répartissent en quatre catégories : demande d'information, proposition d'information, envoi spontané d'information et traitement des erreurs. Ces échanges sont caractérisés par : (i) le rôle des participants : *initiateur* de l'échange ou partenaire, (ii) un *but de l'initiateur* qui motive l'échange, (iii) un état de satisfaction (réalisation ou non du but de l'initiateur), et (iv) un état de succès (établissement de l'échange entre les interlocuteurs).

Pauchet propose une modélisation sous forme d'*automates* temporisés des échanges. Ainsi, chaque échange admet une paire d'automates décrivant les *règles conventionnelles* pour l'initiateur et l'autre participant de l'échange[4]. Les transitions dans les automates temporisés sont associées à 3 éléments : (i) une *garde* portant sur les valeurs des chronomètres (aussi appelés *horloge*), utilisée pour spécifier des contraintes de temps, (ii) une *étiquette* représentant une action, et (iii) des *actions de remise à zéro* de certaines horloges. La notion de temps dans les échanges (et donc dans les automates temporisés) a été introduite pour considérer la clôture des échanges en cas d'inactivité en terme d'occurrence d'actes de langage au bout d'un certain délai (cf. [Pauchet 2006], p. 109). Les actions des transitions sont des *actions dialogiques* d'envoi ou de réception d'actes de langage. Une action d'envoi (*send*) dans l'automate de l'initiateur de l'échange (X_{ini} où X est une lettre identifiant l'échange) admet son action conjuguée (*receive*) dans l'automate de l'interlocuteur de l'échange (X_{int}). La figure 2.1 (p. 45) présente l'automate de l'initiateur pour l'échange de demande d'information. Ce dernier présente un motif d'interaction permettant à l'initiateur de demander des informations à son interlocuteur. Cet

4. Comme le signale Pauchet, « Chaque échange aurait pu être représenté avec un automate. Cependant, utiliser deux automates par échange permet de mettre en avant le point de vue de chaque interlocuteur. » [Pauchet 2006] (pp. 110-111).

échange permet d'utiliser des performatives de demande d'information (*query*), de raffinement de la demande (*refine*) ou d'annulation (*cancel*).

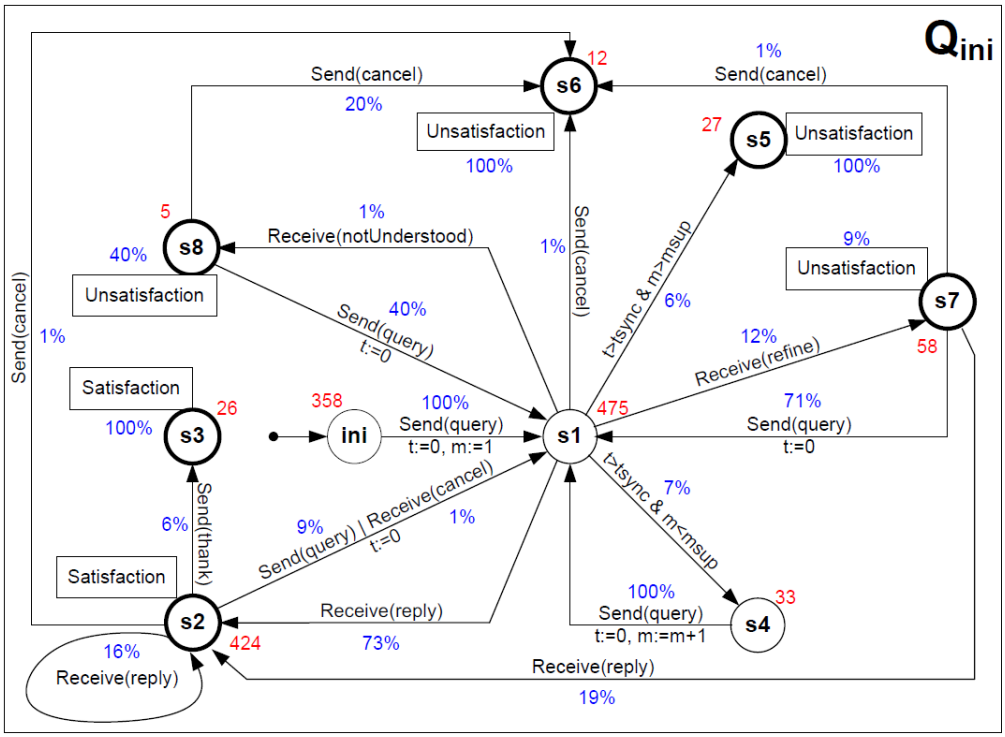

FIGURE 2.1: Automate de l'initiateur pour la demande d'information dans l'approche de Pauchet [Pauchet 2006]

L'approche de Pauchet dépasse le cadre conventionnel strict en proposant un lien avec les attitudes mentales de l'agent. Dans l'esprit des approches intentionnelles, Pauchet présente une sémantique *mentaliste* [Singh 1998] aux actes de langage utilisés. Ils sont vus comme des opérateurs ayant des *préconditions* et des *effets* en terme d'états mentaux des participants. Par exemple, les préconditions de l'acte *inform* spécifient que le locuteur croit la proposition P (hypothèse de sincérité) et a le désir d'informer l'interlocuteur de cette proposition. L'envoi du message conduit le locuteur à ajouter à ses croyances la proposition $pSent(M)$ où M est le message. La réception du message conduit le partenaire à ajouter dans ses croyances la proposition P.

Pauchet note la présence de *dialogues incidents* dans le corpus étudié (cf. section 1.1.2). Ceux-ci correspondent à des « énoncés non-attendus » dans le cours idéal de l'interaction. Le principal type de dialogue incident est dû à une demande d'information sous spécifiée nécessitant une clarification. Pour gérer ce cas, Pauchet a mis en place un acte de dialogue spécial *refine* qui permet de compléter la demande (cf. états $s1$ et $s7$ de l'échange représenté figure 2.1). De manière générale, « Il y a un dialogue incident à chaque émission d'une performative de type *refine* » [Pauchet 2006] (p. 98). La solution proposée pour gérer ces digressions est donc sur l'axe *intra-échange* plutôt que sur l'axe *inter-échange*. Le principal avantage de cette solution est d'être simple et efficace. Néanmoins, on peut se demander si cette solution ne nuit pas à la portée de l'échange qui semble être sur l'*axe de la tâche* sous-jacente du dialogue alors que le *refine* semble agir sur l'*axe interactionnel* du dialogue. Pauchet note par ailleurs que cette

solution ne couvre pas l'ensemble des cas observés dans le corpus, à savoir une imbrication de *refine*.

2.2.3 Modèles fondés sur les questions en discussion

Dans cette section, nous présentons la théorie « Question Under Discussion » (QUD) [5] du linguiste Ginzburg. Puis, nous voyons comment cette théorie a été mise en œuvre dans le système de dialogue implémenté GoDiS.

Questions en discussion

La théorie « Question Under Discussion » (QUD) est une théorie conventionnelle fondée sur une sémantique formelle des questions [Ginzburg 1994, Ginzburg 1996, Ginzburg 2012]. Ce qui fait l'originalité de ce modèle est que Ginzburg propose une version structurée d'un *tableau de conversation* nommé « dialogue gameboard » (DGB) en plus de la sémantique des questions. Ginzburg étudie les effets des couples question-réponse sur le tableau conversationnel. Le tableau est enrichi par des coups dialogiques [6] qui sont des actes de dialogue contextuels (cf. section 1.2.2).

Nous allons maintenant aborder plus en détails cette théorie. Nous nous intéressons tout d'abord à la sémantique des conditions de résolution de ces questions. Puis, nous voyons la structure de la DGB inclue dans le cadre plus large du « State Of Affair » (SOA). Enfin, nous présentons brièvement les effets des questions et assertions sur la DGB.

Relations entre questions et réponses Nous présentons de manière informelle les relations entre les questions et les réponses déterminées par Ginzburg. Une présentation plus complète est disponible dans [Ginzburg 1995a, Ginzburg 1995b, Ginzburg 1996]. Ginzburg propose une vue théorique de la sémantique formelle sur la notion de *question* basée sur la théorie des situations de Barwise et Perry [Barwise 1981]. Dans le modèle de Ginzburg, une question est vue comme une expression en attente d'un ou plusieurs arguments (la ou les réponses à la question). La réponse peut être une proposition ou une simple expression (une réponse courte). Une question est représentée par une lambda expression typée. La réponse à cette question doit être du type de la question. Ginzburg propose un ensemble de relations liant une question et ses réponses et les questions entre elles que nous allons maintenant présenter.

Réponses à propos Les réponses *à propos* [7] sont des réponses qui sont liées à une question indépendamment de leur exactitude ou du niveau de détails. L'avantage de considérer les questions *à propos* est d'éliminer les réponses n'ayant aucun rapport avec la question. Considérons l'exemple du dialogue 2.1.

5. Cette théorie est également appelée KoS [Ginzburg 2012].
6. Traduction de « dialogue move »
7. Traduit de l'anglais *aboutness* et *about*

S$_1$: Est-ce que Paul vient demain ?
H$_2$: oui
H'$_2$: non
H"$_2$: peut-être
H"'$_2$: J'adore le chocolat.

Dialogue 2.1 – Exemple de réponses à propos dans l'approche de Ginzburg (réponses **H'**, **H"** et **H"'**).

Dans le dialogue 2.1, les trois premières réponses de H sont *à propos*. Le dernier énoncé abordant le chocolat n'est pas considéré comme étant une réponse.

Réponses résolvantes La notion de *réponse résolvante* à une question est définie par Ginzburg comme permettant de capturer le point de vue relatif d'un agent qui estime que sa question a été discutée suffisamment pour être considérée comme terminée et passer à un autre sujet. Ce qui signifie que : (i) la réponse résout positivement ou négativement la question sémantiquement, (ii) les inférences liées à cette réponse permettent de remplir les buts de l'agent qui a posé la question. La notion de résolution est nécessairement *relative à l'agent*. En d'autres termes, le fait qu'une réponse soit résolvante dépend des buts de l'agent.

S$_1$: Quelle heure est-il ?
H'$_2$: Pas loin de 19h !
H"$_2$: Il est 10h04min32sec.

Dialogue 2.2 – Exemples de la relativité de la notion de résolution dans l'approche de Ginzburg

Le dialogue 2.2 présente un exemple de deux réponses illustrant la relativité de la notion de résolution. Le première réponse (**H'**$_2$) est résolvante dans le contexte de deux collègues de bureau s'interrogeant sur l'heure pour savoir s'il est temps de rentrer chez eux par exemple. La deuxième réponse (**H"**$_2$) est résolvante dans un contexte de deux chimistes tentant de mesurer avec précision le temps d'exécution d'une réaction chimique. Dans le premier contexte, la deuxième réponse (« Il est 10h04min32sec. ») semblerait trop précise. Alors que dans le deuxième contexte, la première réponse (« Pas loin de 19h ! ») ne serait pas résolvante.

Relation de dépendance entre questions Ginzburg définit également une relation de *dépendance* entre deux questions : q_1 dépend de q_2 si et seulement si q_1 est résolue par un fait τ seulement si q_2 est aussi résolue par τ. L'exemple classique pour illustrer la relation de dépendance entre questions est reporté dans le dialogue 2.3.

S$_1$: Qui a assassiné John ?
H$_2$: Qui était en ville hier ?

Dialogue 2.3 – Exemple de questions dépendantes dans l'approche de Ginzburg

Dans le dialogue 2.3, la question « Qui a assassiné John » dépend de « qui était en ville hier ». De manière symétrique, il est possible de définir la relation d'*influence*. Ainsi, si q_1 dépend de q_2 alors q_2 influence q_1.

Le tableau de conversation Afin de structurer le fonds commun conversationnel de chacun des interlocuteurs au cours du dialogue, Ginzburg introduit la *dialogue gameboard* (DGB) [Ginzburg 1994, Ginzburg 1996, Ginzburg 2012]. Cet objet est dit *quasi-partagé* (cf. section 2.2.1). En effet, chaque interlocuteur possède sa propre version de la DGB. Des différences peuvent apparaître entre les DGB des participants du fait d'une communication imparfaite. Selon Ginzburg, si les deux tableaux sont différents les agents vont chercher à identifier et corriger ces différences. La DGB présente trois champs :

— FACTS : ensemble de faits mutuellement acceptés ;

— QUD (*Question Under Discussion*) : ensemble partiellement ordonné qui spécifie les questions en cours de discussion. Si q est maximale dans QUD, il est possible de fournir une information spécifique à q (notamment en passant par des réponses courtes comme des ellipses). Ainsi, cette structure permet de mettre en avant les questions qui sont saillantes dans la conversation. Larsson remarque que les questions dans QUD possèdent les caractéristiques suivantes [Larsson 2002a] (p. 156) : (a) elles sont ouvertes à la discussion ; (b) elles sont disponibles pour une résolution elliptique ; (c) elles sont explicitement posées ; (d) elles sont non encore résolues.

— LATEST-MOVE : le ou les derniers *coups dialogiques*.

En outre, Ginzburg propose de représenter les buts de l'interlocuteur et ses capacités inférentielles. Contrairement à la DGB qui est quasi-partagée, cette partie est privée. Elle est appelée *unpublicized mental situation (UNPUB-MS(DP))*[8]. Nous avons vu que le fait qu'une réponse soit résolvante était relative à l'interlocuteur. Cela revient à dire que c'est relatif à UNPUB-MS(DP). L'ensemble DGB et UNPUB-MS(DP) est nommé *State of Affair (SOA)*. Il forme une structuration du contexte de type 5 vu en section 2.2.1 avec une partie publique relative au point de vue de l'agent et une partie privée.

À chaque instant, un interlocuteur peut donc choisir entre ajouter quelque chose à la DGB ou aborder une question du champ QUD. Les règles de mise à jour de la DGB s'appuient sur la sémantique précédemment définie (cf. section 2.2.3). A priori, n'importe quelle question peut être posée tant qu'elle n'est pas *résolue* (i.e., si une réponse résolvante a déjà été donnée ou bien que cette réponse se trouve dans FACTS). Dès que le locuteur pose une question q, le locuteur et son interlocuteur l'ajoute au champ QUD. L'allocutaire peut alors accepter la question. Dans ce cas il doit produire une *réponse spécifique* à q. Les énoncés qui sont disponibles comme réponse spécifique à une question sont les suivants : (a) une réponse *résolvante* ; (b) des réponses *à propos* ; (c) une question *influençant* q ; (d) des réponses *à propos* des questions influençantes.

Enfin, la question maximale q dans QUD est dépilée lorsque que : (a) un fait résout q relativement à UNPUB-MS(DP) (où DP est celui qui a produit la question) ; (b) aucune information à propos de q ne peut être fournie.

L'avantage de QUD est de proposer une version structurée d'un tableau de conversation ainsi que des protocoles de mises à jour. Cette théorie est le point de départ choisi par Larsson pour le système GoDiS que nous allons maintenant aborder.

« Issue-based Dialogue Management »

GoDiS [Larsson 2000a] et son successeur IBiS [Larsson 2002a] sont des modèles de dialogue basés sur QUD et implémentés en Prolog avec le framework TRINDIKIT [Larsson 2000b, Lars-

8. DP = « Dialogue Participant »

son 2002b]. Nous référons à ces systèmes en utilisant le nom GoDiS. GoDiS est un exemple complet de gestionnaire de dialogue basé sur une approche contextuelle (cf. section 1.2.2). La génération et l'interprétation des énoncés sont perçus comme la mise à jour et l'exploitation d'un *état d'information* représentant le contexte et l'état du dialogue à un instant donné. En outre, GoDiS intègre des mécanismes d'*accommodation* de l'état d'information (cf. section 2.2.1). Pour finir, GoDiS gère certains processus de gestion de la communication (cf. sections 1.1.2 et 2.2.1). En somme, ce système est la parfaite illustration de l'approche contextuelle associée à la mise en œuvre de mécanismes d'accommodation et de gestion du dialogue.

La présence de mécanismes de planification dans GoDiS peut interroger sur sa classification en tant qu'approche conventionnelle de la modélisation du dialogue. GoDiS utilise des *plans de dialogue* figés qui ont un double rôle : fonder le comportement coopératif du système et représenter les relations de dépendance entre les questions. Ces plans de dialogue sont à différencier des *plans du domaine* utilisés dans les approches intentionnelles. Les plans du domaine sont utilisés comme base de raisonnement pour la production du comportement coopératif (éventuellement dialogique) du système. Ainsi, les approches intentionnelles ne nécessitent pas de *plan de dialogue*. Grossièrement, un plan dans GoDiS représente une décomposition d'une question en un ensemble de *questions influençantes*. Pour répondre à une question sur le prix d'un voyage, le plan représente le soulèvement d'autres questions (villes de départ et d'arrivée, moyen de transport, ...). Ces plans sont *figés* et représentent une connaissance procédurale pour mener le dialogue. GoDiS évite ainsi les processus de reconnaissance d'intention et d'inférence de plan [Larsson 2002a] (pp. 33-34). En outre, les énoncés de l'utilisateur ne sont pas analysés en terme d'intention sous-jacente. Un ou plusieurs coups dialogiques sont attribués à un énoncé sur la base de marqueurs linguistiques de surface. En somme, GoDiS fonde son comportement coopératif dialogique sur des connaissances procédurales conventionnelles. L'absence de plan du domaine et de mécanisme de reconnaissance d'intention nous pousse à ranger GoDiS parmi les approches conventionnelles.

L'état d'information GoDiS s'articule autour de l'*état d'information* (IS). C'est une représentation structurée du contexte et de l'état du dialogue à un instant donné. L'état d'information de GoDiS se structure de la façon suivante :

$$\left[\begin{array}{lll} \text{PRIVATE} & : & \left[\begin{array}{lll} \text{AGENDA} & : & \text{OpenQueue(Action)} \\ \text{PLAN} & : & \text{OpenStack(PlanConstruct)} \\ \text{BEL} & : & \text{Set(Proposition)} \\ \text{TMP} & : & \left[\begin{array}{lll} \text{USR} & : & \textit{Tmp} \\ \text{SYS} & : & \textit{Tmp} \end{array}\right] \\ \text{NIM} & : & \text{OpenQueue(Pair(DP,Move))} \end{array}\right] \\ \text{SHARED} & : & \left[\begin{array}{lll} \text{COM} & : & \text{Set(Proposition)} \\ \text{ISSUES} & : & \text{OpenStack(Question)} \\ \text{QUD} & : & \text{OpenStack(Question)} \\ \text{ACTIONS} & : & \text{OpenStack(Action)} \\ \text{PM} & : & \text{OpenQueue(Move)} \\ \text{LU} & : & \left[\begin{array}{lll} \text{SPEAKER} & : & \text{Participant} \\ \text{MOVES} & : & \text{Set(Move)} \end{array}\right] \end{array}\right] \end{array}\right]$$

L'état d'information suit le principe du *State of Affair* de Ginzburg. Ainsi, il est séparé en

deux parties : une privée et une partagée.

La partie privée (`PRIVATE`) équivaut au UNPUB-MS de Ginzburg. Elle est principalement utilisée pour la gestion de la planification. Elle est constituée d'un champ `PLAN` qui stocke les plans qui sont en train de se dérouler (les questions qui vont être posées, etc.). Ces plans définissent des relations de dépendance entre les questions. Par exemple, une question pour connaître le prix d'un voyage est dépendante de questions concernant la ville de départ, la ville d'arrivée et le moyen de transport. Le champ `PLAN` a une portée sur le « long terme » contrairement au champ `AGENDA` dont la portée est plus courte. Ce dernier va en effet gérer les actions à faire dans un futur très proche (poser une question, répondre à une question, etc.). Le champ `BEL` (*belief*) est un ensemble de propositions représentant les connaissances du système et qui ne sont pas encore partagées (e.g., le prix du billet de train avant d'être communiqué à l'utilisateur). Le champ `NIM` (*non-integrated moves*) est une structure permettant de stocker les coups dialogiques qui n'ont pas encore été intégrés par le système. Il est utilisé lors des mécanismes de *grounding*. Enfin, le champ `TMP` est une structure temporaire permettant de sauvegarder les champs importants de l'IS afin de permettre un retour en arrière. En effet, le système suppose que l'utilisateur va comprendre ce qu'il dit. Cependant, si l'utilisateur marque son incompréhension par un *feedback négatif* (e.g., « Pardon ? »), le champ `TMP/SYS` permet de revenir en arrière. De la même manière, le système va supposer qu'il comprend tout ce que lui dit l'utilisateur. Si ce n'est pas le cas, c'est le champ `TMP/USR` qui va permettre de restaurer les champs et de fournir une réaction appropriée.

La partie partagée (`SHARED`) est l'équivalent de la DGB de Ginzburg. C'est la représentation des informations partagées par les interlocuteurs pendant le dialogue. Elle est constituée du champ `COM` (*commitments*) qui est un ensemble de propositions qui ont été établies. Le champ `ISSUES` représente les questions qui sont en cours, qui ne sont pas résolues et qui ne sont pas QUD-maximales (ce qui revient à dire qu'elle ne sont pas disponibles pour une résolution par une réponse courte). Le champ `QUD`, quant à lui, représente les questions qui ont les mêmes propriétés que les précédentes mais qui sont QUD-maximales. Le champ `ACTIONS` regroupe les actions en cours. Le champ `PM` (*previous moves*) contient l'ensemble des coups dialogiques associés aux précédents énoncés qui ont été établis (i.e., l'historique du dialogue). Enfin, le champ `LU` (*Last Utterance*) stocke l'interlocuteur et l'énoncé précédent qui a été intégré par le système.

Gestion du dialogue Le système est constitué d'un ensemble de *modules* et de *ressources*. Les modules ont accès en lecture et/ou écriture à l'état d'information. Les *ressources* servent à inclure les éléments spécifiques au domaine d'application. La gestion du dialogue dans GoDiS est effectuée en fonction de l'état d'information et en mettant à jour l'état d'information à l'aide de *règles*. Ces règles sont spécifiées dans les modules du « Dialogue Move Engine » (DME). Ces dernières sont constituées d'un nom, d'une liste de préconditions et d'une liste d'effets. Si les préconditions d'une règle sont vérifiées, alors les effets de la règle sont appliqués.

Le DME inclut deux modules principaux. Le premier est le module *update* qui regroupe un ensemble de règles dont le rôle est de gérer le processus d'interprétation des coups dialogiques effectués par les interlocuteurs. Il inclut les règles d'intégration qui permettent d'intégrer les coups dialogiques (*answer*, *ask*, etc.) du système et de l'utilisateur dans l'IS. Le second module *select* contient les règles permettant au système de (i) sélectionner les prochains coups dialogiques qu'il va effectuer, et (ii) trouver une nouvelle action à effectuer (en fonction du plan courant ou de l'état de l'IS). L'application des règles est coordonnée par des *algorithmes de mise à jour* spécifiques à chaque module. Ces algorithmes contraignent l'ordre d'exécution des règles. Dans

le cas des modules *select* et *update*, ces algorithmes vont appeler les règles de chaque module dans un ordre fixé. L'étape de sélection permet au système de générer des actions dialogiques ou de se mettre en attente d'une action de la part de l'utilisateur. Après chaque action dialogique, l'algorithme de mise à jour de l'état d'information est appelé. Le système retourne alors à la phase de sélection.

En sus, Larsson propose un ensemble de règles permettant d'accommoder l'état d'information à un coup dialogique inattendu de l'utilisateur.

Gestion de l'accommodation Larsson propose un ensemble de mécanismes permettant de gérer l'*accommodation* de l'état d'information en fonction des réponses données par l'utilisateur [Larsson 2002a, Cooper 2010]. Ce mécanisme permet de prendre en compte des coups dialogiques inattendus par l'état courant du dialogue.

Un premier type d'accommodation permet de répondre à une question qui n'a pas encore été posée. L'*accommodation de plan* permet à l'utilisateur de répondre à une question qui n'est pas en discussion. Le système infère alors la question et intègre sa réponse. L'*accommodation globale* permet à l'utilisateur de répondre en avance à une question dès lors qu'elle fait partie du plan courant. Cela permet de prendre en compte des « sur-réponses » comme dans l'exemple : « – Où souhaitez-vous aller ? – À Paris, depuis Rouen, en train ». Au niveau de l'IS, cette accommodation se déroule du champ `/private/plan` vers le champ `/shared/issues`.

Un second type d'accommodation permet à l'utilisateur de changer la réponse qu'il a donnée à une question et cela à n'importe quel moment du dialogue. Ce mécanisme, appelé ré-accommodation, permet de *remplacer* la réponse présente dans `/shared/com` par la nouvelle réponse. Le système est capable de re-soulever (*reraising*) un plan. En effet, il est nécessaire de re-considérer les réponses des questions qui sont dépendantes de la question dont la réponse a été modifiée.

Gestion de la communication Larsson propose des mécanismes de gestion des feedbacks en partant des niveaux de communication d'Allwood et de Clark (cf. section 1.1.2) [Larsson 2002a, Larsson 2003]. Les feedbacks sont des coups dialogiques sur différents niveaux. Le système est capable de produire des feedbacks positifs du niveau *perception* jusqu'au niveau *acceptation*. Il est également capable de produire des feedbacks négatifs sur l'ensemble des niveaux. En outre, le système peut produire des demandes explicites de clarification au niveau *compréhension*. Ces feedbacks sont illustrés dans le tableau 2.3.

Les feedbacks utilisables par l'utilisateur et pouvant être traités par le système sont plus réduits. Celui-ci peut produire des feedbacks négatifs au niveau *perception*. De plus, il peut accepter ou rejeter des questions. Certains travaux ont étudié plus en détails les demandes de clarification de l'utilisateur et leur intégration à GoDiS [Purver 2004].

Synthèse sur le système GoDiS

GoDiS est un exemple complet de système de dialogue implémenté basé sur la structuration d'un contexte dialogique. La structuration et la sémantique des questions et des réponses sont issues de la théorie QUD du linguiste Ginzburg. Il a été utilisé avec succès comme base pour construire d'autres systèmes. Par exemple, il a permis de concevoir un système de dialogue pour un lecteur multimédia [Hjelm 2005] ou encore un système d'interaction multimodale embarqué dans une voiture [Larsson 2011]. Il est également à la base de Cogni-CISMeF, un prototype

Niveau		Exemples
Contact	-	« I didn't hear anything from you. »
Perception	-	« Pardon ? », « I didn't hear what you said. »
	+	« I heard "to Paris". »
Compréhension (sémantique)	-	« I don't understand. »
	+	« Paris. » (répétition/reformulation)
Compréhension (pragmatique)	-	« I don't quite understand. »
	n	« To Paris, is that correct ? »
	+	« To Paris. »
Réaction/Acceptation	-	« Sorry, Paris is not a valid destination. »
	+	« Okay. »

TABLEAU 2.3: Gestion de feedbacks du système dans GODIS [Larsson 2003]. « - » = feedback négatif, « + » = feedback positif, « n » = feedback neutre.

d'agent assistant pour la recherche d'information auquel nous avons contribué [Loisel 2011, Loisel 2012]. Nous défendons l'idée que GODIS est un système de dialogue proposant des mécanismes de gestion de phénomènes dialogiques avancés indépendants de la tâche en s'appuyant sur un état d'information. Il reste néanmoins limité à une tâche de recherche d'information simple tout en étant difficile à étendre.

2.2.4 Avantages et limites des approches conventionnelles

Nous avons présenté certains aspects conventionnels représentatifs de modèles du dialogue pour l'interaction Homme-Machine. Au contraire des approches intentionnelles qui se focalisent sur les attitudes privées des agents, les approches conventionnelles se basent sur les notions de fonds commun et de convention mettant en avant ce qui est *partagé* par les interlocuteurs. Les approches conventionnelles supposent l'existence d'un lien conventionnel entre les énoncés qui se concrétise par l'existence de régularités, les *motifs d'interaction*. Le problème n'est pas de savoir comment ces régularités sont apparues. Leur existence est supposée, tout comme le fait que les interlocuteurs sont socialement poussés à s'y conformer. Ce sont ces dernières suppositions qui peuvent laisser penser que les approches conventionnelles ne feraient que décrire des motifs expliqués par les approches intentionnelles. Nous revenons sur ce point en section 2.3. De manière générale dans ces approches, un énoncé est cohérent s'il correspond à ce qui est *conventionnellement attendu* au cours du dialogue.

Les approches conventionnelles *structurelles* étudient les motifs d'interaction sans se focaliser sur les intentions sous-jacentes. Cela provient de l'observation qu'un grand nombre de types d'énoncés ne semble pas être consciemment émis mais plutôt conventionnellement déclenché par le contexte (l'exemple canonique étant les salutations). Ces études conduisent à la production de règles décrivant les séquences admissibles de types d'énoncés (e.g., sous forme de protocoles ou de grammaires de dialogue). La faiblesse la plus criante de ces approches structurelles est très certainement leur manque de flexibilité. Elles sont souvent mises en défaut par un énoncé non-attendu comme illustré par les problèmes de dynamique des échanges de Pauchet (cf. section 2.2.2). Cela nous amène à une critique plus générale des protocoles de communication [Maudet 2002b]. Nous avons pu voir que les protocoles manquaient de *flexibilité*. En particulier, il est indispensable de préférer un ensemble de petits protocoles qu'il est possible de *composer*. En

outre, les protocoles pour l'interaction Homme-Machine mêlent généralement la modélisation de plusieurs *niveaux de l'interaction* (cf. section 1.1.2). Dans l'idéal, les protocoles devraient se contenter de gérer les actions participatives propres au niveau sur lequel ils interviennent. Or, nous constatons l'intégration dans leur modélisation d'autres préoccupations comme la gestion des entrée/sortie de l'activité conjointe et des processus de gestion de l'interaction (voir, par exemple, les actes *cancel*, *refine* et *notUnderstood* de l'approche de Pauchet). En tant qu'activité conjointe, il est nécessaire d'étudier les mécanismes qui permettent aux participants du dialogue d'établir le protocole en usage. Ensuite, le formalisme d'automate semble trop rigide. Il est en particulier difficile de considérer des messages inattendus par le protocole. Enfin, les protocoles représentent des projets conjoints *conventionnels*. À cet égard, l'approche de Pauchet (comme bien d'autres approches) réserve un rôle ambigu aux protocoles. D'un côté, les protocoles sont des dispositifs de coordination *partagés* par les interlocuteurs (i.e., faisant partie du fonds commun). D'un autre côté, la sémantique mentaliste spécifie des conditions d'applicabilité des actes de langage en terme d'attitudes privées. Le télescopage des niveaux public et privé brouille la portée de tels protocoles. En tant que dispositif partagé de coordination, il semble nécessaire de fournir des spécifications à un haut niveau d'abstraction et idéalement indépendantes des spécificités des participants impliqués dans la communication (plus particulièrement, des dispositions privées comme les croyances et les désirs qui correspondent davantage à un niveau lié à l'intentionnalité et à la tâche sous-jacente). Une alternative consiste à spécifier une sémantique publique [Singh 1991, Yolum 2004].

Nous avons également abordé l'approche QUD qui se focalise sur la formalisation du *tableau de conversation* en terme de questions et de leurs réponses. En tant que telle, nous pouvons la considérer comme une approche se focalisant sur l'étude du motif d'interaction question/réponse. Un énoncé est cohérent en fonction du *tableau de conversation* s'il soulève une nouvelle question ou s'il est spécifique à une question posée (réponse résolvante, à propos, question dépendante, etc.). Notons à ce propos qu'elle envisage le cas d'une question adjacente à une autre dépassant le strict motif question/réponse. Cette approche étudie finement le rôle des questions dans le dialogue mais délaisse les autres types d'énoncés (requêtes, suggestions, salutations, assertions, etc.). Enfin, l'approche de GoDiS basée sur QUD nous a permis d'illustrer des concepts comme la structuration du tableau de conversation, une approche contextuelle des actes de dialogue, l'accommodation, et la gestion de la communication. La force de ce système est de gérer de nombreux phénomènes dialogiques de manière indépendante de la tâche. L'accommodation tire partie des relations de dépendance entre les questions et de la partie publique de l'état d'information représentant les questions en discussion. Elle permet de prendre en compte certains coups dialogiques inattendus. La gestion de l'interaction intègre des coups dialogiques sur plusieurs niveaux inspirés d'Allwood et de Clark. Cela permet au système et à l'utilisateur de gérer des cas simples de problèmes de perception, compréhension et acceptation. L'interaction est gérée de manière contextuelle sans avoir recours à des meta-plans. En outre, le modèle du dialogue se concentre sur les questions en leur donnant un statut sémantique privilégié. Les plans de dialogue se résument à des actions permettant de poser ou de répondre à des questions. Ainsi, l'intervention du domaine se cantonne aux ressources qui spécifient des lexiques et des plans de dialogue (i.e., les questions à poser et leurs relations de dépendance). Néanmoins, GoDiS se limite à la modélisation des questions/réponses dans le dialogue en se basant sur QUD. La taxonomie d'actes de dialogue liés à la tâche est fortement réduite. L'ajout de nouveaux actes de dialogue est une tâche loin d'être triviale comme le démontre la tentative de Loisel d'intégration des actes *inform*, *suggest* et *offer* [Loisel 2008]. Le défaut majeur de l'approche adoptée par GoDiS de

mise à jour de l'état d'information sous forme de règles est qu'il est difficile de prédire les effets des règles et de leurs interactions lorsque leur nombre devient trop important. Ainsi, une règle gère souvent plusieurs aspects du dialogue (mise à jour conventionnelle du tableau, gestion de l'interaction et de l'accommodation, gestion des plans de dialogue). En conséquence, la système est difficile à étendre. L'ajout de règles augmente la difficulté de prédire le comportement du système.

Pour conclure, précisons que les approches conventionnelles capturent la *cohérence locale* des énoncés. Il reste néanmoins nécessaire d'expliquer comment émerge une *cohérence globale* sur l'ensemble du dialogue. En d'autres termes, il est nécessaire d'expliquer la structure sous-jacente à l'agencement des motifs d'interaction. Par exemple, Pauchet tente de dépasser le cadre conventionnel strict en définissant un but de l'initiateur motivant l'échange. Ce but est relié à l'intention générée par l'architecture BDI de l'agent utilisant les protocoles. Un germe d'idée apparaît : les approches intentionnelles capturent la cohérence à haut niveau tandis que les protocoles assurent la cohérence locale. Néanmoins, cette approche repose sur de fortes hypothèses de coopération. En ne définissant qu'un but de l'initiateur motivant l'échange, cette approche explique la participation du partenaire en terme d'adoption de but. Or, cette hypothèse n'est pas tenable pour des dialogues où les buts des interlocuteurs divergent (e.g., argumentation). Rappelons qu'il est possible d'entretenir des dialogues conflictuels tout en étant parfaitement coopératif (cf. section 2.1.4). De manière générale, il est nécessaire d'expliquer le lien entre l'aspect conventionnel local du dialogue (i.e., les motifs d'interaction), ce qui motive les interlocuteurs à participer à ces activités locales et l'émergence de structure de plus haut niveau comme la structure intentionnelle du dialogue.

2.3 Discussion

Les approches intentionnelles et conventionnelles de la modélisation du dialogue sont généralement vues comme rivales [Pulman 1998, Hulstijn 2000a, Maudet 2001]. Alors que les approches intentionnelles se concentrent sur l'agent et ses états mentaux, les approches conventionnelles se focalisent sur la structure du contexte dialogique. D'autre part, les unités d'interaction utilisées diffèrent dans leur définition. D'un côté, l'approche intentionnelle relie les actes de langage aux intentions des agents via la force illocutoire. De l'autre côté, on parle plus volontiers de *coups dialogiques* qui sont vus comme des fonctions modifiant le contexte dialogique au sens large [9]. Enfin, les approches conventionnelles telles que les grammaires de dialogue sont généralement perçues comme des approches descriptives à l'opposé des approches intentionnelles qui sont vues comme génératives.

Pourtant, un certain nombre de chercheurs proposent de voir ces approches comme *complémentaires* et d'aller vers des agents dialogiques délibératifs/réactifs [Traum 1997, Pulman 1998, Hulstijn 2000a, Lewin 2000, Maudet 2001]. Cette complémentarité vient du fait que les processus de communication peuvent être considérés comme des actions conjointes entre un locuteur et ses allocutaires [Clark 1996]. La caractéristique clé d'une action conjointe est la *coordination* d'*actions participatives* par au moins deux personnes. En partant du constat que les humains ne peuvent pas délibérer indéfiniment dans une activité opportuniste et dynamique telle

9. Comme nous l'avons vu (cf. section 1.2.2), ces approches diffèrent en fait en fonction de ce que représente le contexte dialogique. Si celui-ci se résume aux états mentaux, alors les coups dialogiques correspondent aux actes de langage classiques.

que le dialogue, on parvient à la conclusion que la coordination doit reposer sur des dispositifs tels que les conventions.

De manière intéressante, les théories de l'action et de la planification conjointe (e.g., [Grosz 1996]) font également apparaître ce besoin d'enchaînement stéréotypique d'actions (appelé *recettes*) et de protocoles conventionnels de régulation de la coopération [Hulstijn 2000b]. Si les processus de communication sont eux-mêmes vus comme des actions conjointes, alors ils doivent également admettre leurs propres recettes. Il est donc nécessaire de déterminer ces recettes et les structures pour les manipuler. Comme le précise [Maudet 2001], ces modèles de l'action conjointe pour la communication doivent capturer l'idée de coopération sur plusieurs niveaux comme au jeu d'échec où il est possible de *coopérer* sur l'activité du jeu tout en poursuivant des *buts opposés* (chaque joueur souhaite gagner la partie). Or, ces modèles sont pour le moment limités aux actions conjointes strictement coopératives.

En outre, il est clair que les approches intentionnelles ne capturent pas tous les phénomènes dialogiques dès lors que le dialogue est vu comme une activité *opportuniste* et *sociale*. Non seulement certains énoncés sont susceptibles d'être déclenchés conventionnellement par la situation mais en plus certaines séquences d'actes ne peuvent pas être planifiées [Pulman 1998] (typiquement les demandes de clarification). Les approches conventionnelles ne se contentent donc pas de décrire les motifs expliqués par les approches intentionnelles. Au contraire, elles permettent de capturer d'autres phénomènes comme par exemple des phénomènes sociaux (e.g., répondre « Je ne sais pas. » à une question).

Enfin, il semble excessif que nous planifions toutes nos interventions comme des salutations, des remerciements (e.g., « de rien ! ») ou même nos réponses à des questions (sauf lorsqu'on souhaite être coopératif). D'un point de vue pratique, la prise en compte d'une partie conventionnelle permet de simplifier la gestion du dialogue.

Chapitre 3

Vers des modèles mixtes à base de jeux de dialogue

Sommaire

3.1 Fondements des jeux de dialogue 57
3.1.1 Les jeux de dialogue comme scripts partagés 58
3.1.2 Les jeux de dialogue comme recettes partagées 60
3.1.3 Les jeux de dialogue comme projet conjoint 62
3.1.4 Synthèse sur la métaphore des jeux de dialogue 64
3.2 Les jeux de dialogue pour l'interaction Homme-Machine 64
3.2.1 Les jeux comme réseau de transition récursif 65
3.2.2 Les jeux comme unités de type initiative-réponse cohérentes 66
3.2.3 Les jeux comme structures pour l'engagement dans le dialogue 69
3.3 Discussion 74
3.3.1 Vers un modèle normatif des jeux de dialogue basé sur les engagements 74
3.3.2 Avantages et limites des jeux de dialogue 78
3.3.3 Vers des modèles mixtes fondés sur des jeux de dialogue 81

Parmi les approches mixtes, nous pouvons distinguer celles qui cherchent à adapter les approches intentionnelles en y ajoutant de nouvelles attitudes comme les *obligations* [Traum 1994b] ou les *pressions interactives* [Bunt 1996] de celles qui s'intéressent à capturer dans des structures spécifiques les *motifs d'interaction* apparaissant au cours du dialogue telles que les *jeux de dialogue*. L'approche par jeux de dialogue a été défendue de manière convaincante comme étant une structure à considérer pour lier approche intentionnelle et approche conventionnelle [Hulstijn 2000a, Maudet 2001]. C'est donc naturellement que nous nous intéressons à cette approche en particulier.

La section 3.1 parcourt les fondements de la notion de jeux de dialogue pour l'interaction Homme-Machine. La section 3.2 présente plusieurs formalisations envisagées afin de capturer cette notion. La section 3.3 souligne les avantages des jeux de dialogue pour la modélisation des interactions Homme-Machine, et montre l'intérêt du formalisme basé sur les engagements sociaux.

3.1 Fondements des jeux de dialogue

L'intuition des jeux de dialogue provient des *jeux de langage* de Wittgenstein qui conçoit la communication dans son activité sociale [Wittgenstein 1958] : les énoncés sont des coups dans une activité qui la font progresser et celle-ci contraint les coups possibles en retour.

Les jeux de dialogue ont reçu beaucoup d'attention dans le domaine de la dialectique formelle [Hamblin 1970, Walton 1995] et dans le domaine des systèmes multiagent [Maudet 2002b, Pasquier 2005]. Pour notre part, nous nous intéressons aux jeux de dialogue utilisés pour expliquer le dialogue humain et pour générer des dialogues artificiels à destination des humains. L'idée de ces jeux est de fournir une structure permettant de formaliser les motifs d'interaction observés dans des conversations humaines.

La section 3.1.1 présente les jeux de dialogue comme scripts partagés. Puis, la section 3.1.2 précise les jeux de dialogue en les envisageant comme recettes partagées. La section 3.1.3 affine cette dernière vision en présentant les jeux de dialogue comme projets conjoints. Enfin, la section 3.1.4 synthétise les caractéristiques essentielles des jeux de dialogue.

3.1.1 Les jeux de dialogue comme scripts partagés

Une des premières études de corpus de dialogues Homme-Homme qui a conduit à la structuration de jeux de dialogue dans un but computationnel a été présentée par Levin et Moore [Levin 1977]. L'analyse d'une soixantaine de transcriptions de communications Homme-Homme via un *chat* a permis de mettre en avant la présence de nombreuses régularités dans les échanges sous la forme de *motifs récurrents* s'étendant sur plusieurs tours de parole. Les efforts se sont concentrés sur les caractéristiques essentielles de ces motifs sous la forme d'une structure appelée *jeu de dialogue*.

La structure d'un jeu de dialogue est composée de 3 champs :

— Paramètres : ils représentent les éléments qui varient dans les *instances* d'un jeu de dialogue. Ici, il s'agit des participants et du sujet de l'interaction.

— Spécifications des paramètres : elles consistent en un ensemble de prédicats sur les paramètres permettant d'expliciter les buts des participants ainsi que leurs connaissances. Ces aspects restent *constants* lors de l'interaction.

— Composants : ils représentent les aspects qui évoluent lors de l'interaction. Il s'agit d'un ensemble de sous-buts des participants, partiellement ordonnés dans le temps. Ils contiennent aussi bien des actions *dialogiques* que des actions *physiques* sur le monde.

Les auteurs illustrent leur propos avec le jeu de dialogue d'aide reproduit dans le tableau 3.1. Ce jeu de dialogue d'aide possède en paramètres les participants (assistant et demandeur) et l'objet du jeu qui est la tâche t. La spécification des paramètres précise le contexte d'occurrence de ce motif d'interaction. Elle indique que le demandeur d souhaite réaliser la tâche t et qu'il en a le droit. Cependant, il n'est pas en mesure de la réaliser seul. L'assistant souhaite aider d à réaliser t et en a la capacité. Les composants présentent une décomposition en sous-buts. Dans ce cas, d doit décrire la situation (quel événement attendu a été observé) et le problème (quel événement attendu n'a pas été observé). a doit ensuite lui proposer la solution sous la forme d'une action qui permet de déclencher l'événement voulu. Le dialogue 3.1 présente une instanciation d'un tel jeu (énoncés [5;6]). Les auteurs ont également défini d'autres jeux de dialogue : demande d'action, demande d'information, investigation, instruction, mécontentement. Ces jeux sont tous *collaboratifs* dans le sens où ils présentent des sous-buts des participants complémentaires les uns aux autres.

En plus de la structuration des jeux de dialogue, Levin et Moore se sont intéressés à l'entrée et à la sortie des jeux de dialogue par les interlocuteurs. L'entrée s'effectue par une phase de proposition du jeu (*bid*). Elle permet d'identifier le jeu, d'indiquer l'intérêt de l'initiateur à

Paramètres	a : Assistant d : Demandeur t : Tâche	
Spécifications des paramètres	d est une personne d veut réaliser la tâche t d est autorisé à réaliser la tâche t d n'est pas en mesure de réaliser t	a est une personne a veut rendre d capable de réaliser t a est capable de permettre à d de réaliser t
Composants	d veut que a sache quel événement attendu a été observé d veut que a sache quel événement attendu e_1 n'est pas survenu a veut que d connaisse l'action act_1 qui va lui permettre de réaliser e_1 a veut que d réalise cette action act_1	

Tableau 3.1: Exemple de jeu de dialogue d'aide dans l'approche de Levin et Moore

poursuivre ce jeu et d'en identifier les paramètres. La caractéristique de cette phase est qu'elle peut être effectuée de plusieurs manières qui sont souvent brèves, voir complètement implicites. Le dialogue 3.1 présente un exemple d'entrée explicite. Une phase de *bid* (**S**1) est suivie d'une phase d'*acceptation* (**H**2) qui permet de reconnaître que le *bid* a été fait, d'identifier le jeu proposé et ses paramètres et d'accepter de poursuivre le jeu ainsi paramétré. Les auteurs soulignent que la fin d'un jeu peut prendre la même forme que son entrée : une phase de *bid* suivie d'une phase d'*acceptation*. Ils notent la présence de trois autres possibilités : l'interruption ou la terminaison spontanée causée soit par la satisfaction des buts du jeu, soit par l'impossibilité de la satisfaction de ces buts.

S1 : Je peux te poser une question ?
H2 : OK
S3 : Tu connais bien les systèmes GNU/Linux ?
H4 : Oui !
S5 : Je me suis connecté en SSH sur mon serveur web. J'ai voulu lancer Firefox mais il ne s'est pas affiché. Ça m'a mis : « Error : no display specified ».
H6 : Il faut que tu actives la redirection du serveur X. Fais un « ssh -X » [...]

Dialogue 3.1 – Jeux de dialogue de demande d'information puis d'aide, *bid* explicite avec les énoncés 1 et 2

Au-delà de la structuration intra-jeu, les auteurs abordent deux agencements possibles entre les jeux (i.e. structuration inter-jeu) : le pré-séquencement et l'imbrication des jeux de dialogue. Deux jeux de dialogue peuvent s'articuler sous forme d'une séquence. Le premier jeu sert à *initier* l'autre jeu. Par exemple, le dialogue 3.1 présente un pré-séquencement de deux jeux (énoncés [1;2] et [3;4]). Le premier permet d'effectuer un *bid* pour entrer dans le second. Le jeu de recherche d'information dans les énoncés [3;4] sert à vérifier certains paramètres du jeu de dialogue d'aide dans les deux derniers énoncés [5;6]. Ce dernier a pour but de résoudre un problème sur un système GNU/Linux avec la commande SSH. Les auteurs précisent que les jeux peuvent s'imbriquer les uns dans les autres. Ils ajoutent que l'imbrication *stricte* des jeux n'est

pas nécessaire, et qu'en fait les participants passent souvent d'un jeu à un autre.

Cette première approche des jeux de dialogue, fondée sur l'étude d'un corpus de dialogue Homme-Homme, permet de dresser les contours de la structure sous-jacente. Levin et Moore ont mis en évidence des éléments de la structure intra-jeu et de la dynamique inter-jeu. Cette approche admet plusieurs limites. Les jeux de dialogue sont structurés en terme de buts et sous-buts sous forme de *scripts partagés* [Schank 1977]. Ils consistent en une liste – *partagée entre les interlocuteurs* – d'actions à réaliser, aussi bien *physiques* que *dialogiques*. C'est ce genre de structure que cherche à apprendre automatiquement l'approche de [Orkin 2013]. Ici, les jeux de dialogue ne capturent pas uniquement les conventions des interactions dialogiques. Ils sont une anticipation des modèles de l'action et de la planification conjointe [Grosz 1996]. Enfin, ils sont tous *collaboratifs* puisque les buts décrits sont complémentaires. Ils ne permettent pas en l'état de prendre en compte des situations dialogiques conflictuelles comme une dispute ou un débat.

3.1.2 Les jeux de dialogue comme recettes partagées

Mann (collègue de Levin et Moore) a ensuite formalisé plus clairement la notion de jeu de dialogue [Mann 1988] [1]. Sa formalisation présente quelques variations par rapport à la version de Levin et Moore que nous allons souligner.

Les participants ont des *rôles* à l'intérieur d'un jeu de dialogue : I pour *Initiateur*, et R pour *Responder* que nous traduisons par *Partenaire*. Un jeu de dialogue est vu comme un triplet (IP, GR, CC) :

— IP (*Illocutionary Point*) : le but de l'initiateur du jeu

— GR (*Goal Responder*) : ensemble non vide de buts du partenaire

— CC (*Conventional Conditions*) : ensemble de descriptions du monde du point de vue de I appelé conditions conventionnelles du jeu.

Un jeu de dialogue entre deux participants est vu comme un moment particulier de l'interaction où : (i) les rôles des participants sont fixés (initiateur ou partenaire), (ii) chacun des participants poursuit ses propres buts à travers la réalisation de ce jeu (pas nécessairement compatibles), et (iii) un ensemble de conditions conventionnelles sont adoptées pendant la durée du jeu. Mann propose des exemples de jeu de dialogue (non exhaustifs) [Mann 1988] (p. 515), reportés dans le tableau 3.2. En plus des conditions conventionnelles spécifiques à chaque jeu de dialogue, Mann a extrait un ensemble de conditions conventionnelles communes à tous les jeux :

1. I poursuit l'IP comme but.
2. I croit que réaliser l'IP est possible.
3. I croit que réaliser les GR est possible pour R.
4. I a le droit de réaliser l'IP.
5. I a le droit d'utiliser le jeu de dialogue.
6. L'IP n'a pas déjà été réalisé.
7. R a le désir de réaliser les GR.

De la même manière que Levin et Moore, Mann décompose l'entrée dans un jeu de dialogue en deux phases : une proposition d'entrée (un *bid*) et son *acceptation* (souvent implicites). Proposer

1. Le papier a été publié en 1978 dans un livre qui n'est jamais paru, il a été re-publié en 1988 après avoir circulé librement dans les laboratoires.

Jeu	IP	GR	CC
Investigation	I sait si R sait Q	R informe I de sa connaissance de Q	I sait Q
Aide	I est capable d'effectuer A	I est capable d'effectuer A	R est capable d'aider I à effectuer A I a le droit d'effectuer A
Demande d'information	I sait Q	I sait Q	R sait Q
Dispute	R croit P	R justifie que I ne devrait pas croire P	I croit P R ne croit pas P
Demande de permission	I sait que R approuve que I réalise A.	R choisit si I a le droit d'effectuer A I sait ce choix	I n'a pas le droit de réaliser A sans la permission de R I veut réaliser A
Demande d'action	R effectue A	R effectue A	R n'effectuerait pas A sinon.
Présentation d'information	R sait P	R sait P	Les connaissances de R et P peuvent être réconciliées. I sait P

Tableau 3.2: Exemples de jeux de dialogue selon Mann. I est l'initiateur du jeu, R le partenaire. Q représente une information. A représente une action. P représente une proposition. IP = but de l'initiateur, GR = buts du partenaire, CC = conditions conventionnelles du jeu.

l'entrée d'un jeu consiste notamment à : (i) consentir à poursuivre l'IP du jeu, (ii) demander à ce que R poursuive les GR, et (iii) proposer d'adopter les CC du jeu pendant toute sa durée (ou jusqu'à ce que d'autres conditions les remplacent). Mann signale que plusieurs jeux peuvent être ouverts en même temps. En conséquence, une proposition de clôture d'un jeu peut concerner un jeu (ouvert) différent du dernier jeu ouvert. De plus, il note que la fermeture d'un jeu peut entraîner des fermetures en cascade des autres jeux ouverts. C'est le cas des jeux imbriqués où les participants peuvent se rendre compte dans un jeu fils que l'IP du jeu parent n'est pas atteignable, ce qui peut provoquer la clôture de jeux fils en chaîne.

Mann s'intéresse principalement à l'imbrication des jeux de dialogue en définissant la notion de *compatibilité* d'une proposition d'entrée dans un jeu avec son *contexte*. En somme, une proposition d'emboîtement de jeu est valide si l'IP de ce nouveau jeu *contribue* à un jeu déjà ouvert.

Les jeux de dialogue formalisés par Mann présentent de nouvelles caractéristiques intéressantes par rapport à la version de Levin et Moore. La première différence de taille est que les structures définies ne représentent que des actions *dialogiques* des interlocuteurs. Les jeux de dialogue voient ainsi leur cadre recentré : ils passent de scripts partagés (mêlant actions non-dialogiques et dialogiques) à des « recettes » qui cristallisent les caractéristiques essentielles des échanges dialogiques récurrents. Mann a exhibé une partie générique des jeux sous la forme de conditions conventionnelles standards qui peuvent être complétées par des conditions conventionnelles spécifiques à chaque jeu. Enfin, en dissociant IP et GR, Mann permet d'appréhender

des jeux qui *ne sont pas collaboratifs*. C'est-à-dire des jeux dont les buts des participants ne sont pas compatibles (voir compétitifs) comme le jeu de dispute ou de demande de permission.

3.1.3 Les jeux de dialogue comme projet conjoint

Mann a proposé plus récemment une généralisation de la théorie des jeux de dialogue sous le nom de « Dialogue Macrogame Theory » [Mann 2002] pour modéliser des dialogues naturels Homme-Homme. Cette généralisation raffine la vision du jeu de dialogue précédemment présenté : il est vu comme une activité conjointe impliquant la poursuite d'un but conjoint (au sens de [Clark 1996]). En conséquence, cela amène Mann à revoir sous cette perspective les notions d'*établissement* du jeu et les actions qu'il qualifie d'« unilatérales ».

Un macrojeu de dialogue est vu comme une activité conjointe, groupant les intentions des participants du dialogue pour la coordination de leur activité. Plus précisément, un macrojeu est une *convention* définie comme un ensemble de trois buts :

1. le but de l'initiateur,
2. le but du partenaire,
3. et un but conjoint.

Mann précise que ces trois buts n'ont pas de rapport hiérarchique. Néanmoins, il est attendu des joueurs qu'ils maintiennent une structure hiérarchique de leurs propres buts (en particulier dans les emboîtements de jeu). 19 jeux ont ainsi été définis parmi lesquels le jeu de présentation d'information (cf. tableau 3.3) tiré de [Mann 2002]. Les buts des jeux ne sont pas figés et incluent des *arguments non spécifiés* tels que l'« information particulière » du macrojeu de présentation d'information. De plus, Mann précise que les macrojeux sont voués à décrire des dialogues généraux et pas uniquement des dialogues orientés tâche. Ainsi, les buts d'une instance d'un macrojeu peuvent être des buts d'une tâche particulière, mais pas nécessairement.

But de l'initiateur	fournir une *information particulière* au destinataire
But du partenaire	identifier et recevoir l'*information particulière* fournie
But conjoint	le partenaire vient à posséder l'*information particulière*

Tableau 3.3: Exemple de macrojeu de présentation d'information dans l'approche de Mann [Mann 2002]

Lorsque le jeu est joué, les participants vont maintenir dans leur mémoire deux buts : leur but propre et le but conjoint. De manière générale, le cours du dialogue est sous le contrôle des deux participants. L'établissement, l'abandon et la clôture des jeux sont réalisés *conjointement* et *en coordination* à l'aide d'*actions particulières*. Mann propose de voir ces mécanismes sous la forme de négociation : un jeu est proposé par l'initiateur, et le partenaire peut refuser ou accepter cette proposition. L'acceptation conduit à l'ouverture du jeu. De la même manière, la fermeture du jeu est proposée par un des deux participants. Cette fermeture peut être acceptée ou refusée. Mann souligne que la phase d'entrée dans le jeu est la plupart du temps *implicite*. Néanmoins, certaines situations d'incompréhension peuvent les rendre explicites. Dans le cas où la proposition d'entrée dans le jeu est rejetée, le but conjoint n'est pas établi. Cela peut intervenir dans la situation où une personne fournit une information qui n'est pas nécessaire à son destinataire. De la même manière, Mann précise que les jeux se terminent le plus souvent par l'accomplissement apparent du but conjoint du jeu. Ainsi, répondre correctement à une question sera généralement perçu comme le moment où poursuivre la recherche d'une réponse

à la question prend fin. Le dialogue 3.2 présente un exemple de phases d'entrée et de sortie implicites.

S1 : As-tu faim ?
H2 : Oui !

Dialogue 3.2 – Jeu de dialogue de demande d'information, phase d'entrée et de sortie implicite

Afin de synthétiser ces éléments, nous proposons d'analyser le dialogue 3.2 selon [Mann 2002]. L'énonce **S**1 réalise les deux actions suivantes :

1. la proposition d'entrée dans le jeu, et
2. la poursuite du but conjoint *proposé*.

Tandis que l'énoncé **H**2 réalise trois actions :

1. l'acceptation d'entrée dans le jeu,
2. la poursuite du but conjoint *établi*, et
3. une proposition de sortie du jeu par satisfaction.

Les jeux de dialogue ainsi définis sont des *projets conjoints* au sens de Clark (cf. section 1.1.1).

Mann précise que les jeux peuvent s'imbriquer les uns dans les autres. Néanmoins, cette imbrication n'est qu'une affaire de *portée* du macrojeu : un jeu est imbriqué dans un autre si son ouverture et sa fermeture sont situées entre l'ouverture et la fermeture du jeu parent. Contrairement à sa précédente théorie [Mann 1988], cette définition n'intègre pas la notion de buts. La définition est alors généralisée par rapport à la théorie des jeux de dialogue. En contrepartie, les macrojeux ne représentent qu'une vue partielle de la structure du dialogue.

Mann note la présence dans les dialogues d'une classe d'actions qui n'impliquent pas la poursuite de buts conjoints et qui tiennent généralement en un tour de parole. Il les nomme *actions unilatérales* et les regroupe en 3 catégories : « Media Management », « Politeness », « Acknowledgement ». Il s'agit d'actions telles que des actions de gestion de l'interaction comme l'établissement du contact illustré dans l'exemple suivant : « – Apollo 13, Houston. – Go ahead – Roger. » (tiré de [Mann 2002], p. 9). Dans ce dernier exemple, chaque énoncé réalise une action de la catégorie « Media Management ».

Mann propose la théorie des macrojeux de dialogue destinée à la représentation des dialogues naturels Homme-Homme. Il indique que la théorie des macrojeux présentée se limite à un dialogue bipartite (i.e. à deux participants). En tant que telle, elle est insuffisante à la représentation d'un dialogue multipartite et nécessite un travail d'adaptation. Les macrojeux de dialogue proposés par Mann apportent des éléments intéressants supplémentaires aux précédentes théories. Tout d'abord, notons que ces nouveaux jeux englobent la définition qu'il avait précédemment présenté. Mann adopte une vision des jeux de dialogue comme étant un projet conjoint, le conduisant ainsi à définir un jeu en associant un but conjoint aux participants à un ensemble de buts spécifiques à chaque participant. Cette vision a deux grandes conséquences. La première est que le contrôle du dialogue est coordonné dans le cas général par les deux participants. L'établissement du jeu est vu comme étant une négociation : les buts constituants les jeux sont acceptés et rejetés de manière conjointe plutôt que de manière individuelle. Ensuite, Mann dégage lors de son analyse l'existence d'*actions unilatérales* qui, selon lui, ne participent pas à la poursuite d'un but conjoint. Enfin, cette nouvelle vision des jeux de dialogue couvre des dialogues qui ne sont pas nécessairement orientés tâche. Néanmoins, les buts décrits dans la

définition du jeu peuvent recouper des buts liés à la tâche. En conséquence, les buts décrits par la structure intentionnelle sont inclus dans les buts des macrojeux. En outre, ces jeux permettent de représenter la cohérence du dialogue *d'une manière indépendante de la tâche*. Cependant, Mann précise que ces jeux n'ont pas vocation à expliquer la cohérence de tout dialogue naturel et qu'un ensemble de théories sera nécessaire afin d'y parvenir.

3.1.4 Synthèse sur la métaphore des jeux de dialogue

Dans cette section, nous avons étudié les travaux fondateurs des jeux de dialogue pour l'analyse et la modélisation du dialogue humain. Ces travaux se basent sur l'étude de corpus de dialogues Homme-Homme, et plus particulièrement sur l'étude des conventions émergeant sous forme de motifs d'interaction identifiables au niveau de surface des énoncés. À la lumière de ces travaux, nous dressons le bilan des caractéristiques essentielles des jeux de dialogue.

La métaphore du jeu de dialogue est essentiellement *normative* (à l'opposé d'une recette). C'est une structure permettant de décrire les droits et obligations ainsi que les buts et les actions dialogiques possibles des participants en fonction de l'avancement d'une activité d'un certain type. Cette métaphore permet de voir l'aspect opportuniste du dialogue comme la combinaison d'activités vues par la métaphore des jeux.

Par raffinement successif, les propriétés suivantes ont émergées. Un jeu de dialogue est une *activité conjointe bornée* avec un début, un corps et une fin [Clark 1996]. Deux participants prennent part à cette activité en endossant un *rôle* (*initiateur* du jeu ou *partenaire*). L'initiateur projette l'exécution du jeu qui doit être acceptée par le partenaire. Autrement dit, le jeu est établi conjointement (c'est un *projet commun*). Dans le cas général, le contrôle du dialogue est coordonné par les deux participants. Les *règles du jeu* spécifient les coups dialogiques autorisés (les actions participatives) pour chaque participant. Les règles ne se limitent pas à la spécification de séquences de coups et peuvent également spécifier le *contenu* des coups dialogiques [Hulstijn 2000a] ou des *interdictions* [Maudet 2001]. Les participants sont appelés à jouer leurs rôles en réalisant les coups attendus par l'*état courant du jeu*. L'occurrence d'un coup détermine en retour un nouvel état du jeu. Les participants poursuivent des *buts non nécessairement compatibles*. Certains jeux non-collaboratifs peuvent faire intervenir un aspect gagnant/perdant. En outre, les jeux sont des structures *combinables* (e.g., pré-séquence, emboîtement [2], ...). Dans leur dernière évolution, les jeux de dialogue permettent de représenter des conventions d'une manière indépendante de la tâche. Enfin, les jeux de dialogue nécessitent d'être associés à d'autres théories, notamment pour assurer la cohérence de la structure intentionnelle du dialogue.

3.2 Les jeux de dialogue pour l'interaction Homme-Machine

Dans cette partie, nous voyons comment la métaphore des jeux de dialogue a été utilisée pour produire des modèles de dialogue pour l'interaction Homme-Machine. La section 3.2.1 présente la formalisation des jeux de dialogue comme réseau de transition récursif. La section 3.2.2 s'intéresse à la formalisation des jeux de dialogue par des unités de type initiative-réponse cohérentes. Enfin, la section 3.2.3 décrit une formalisation des jeux en tant que structures capturant les engagements des interlocuteurs.

2. Nous utilisons de manière interchangeable « emboîtement » et « imbrication »

3.2.1 Les jeux comme réseau de transition récursif

Nous présentons dans cette partie l'approche formalisée et *implémentée* des jeux de dialogue de Lewin [Lewin 2000] dans le cadre du projet TRINDI [Larsson 2000b].

Dans l'approche de Lewin [Lewin 2000], le contexte dialogique inclut un *tableau d'engagements* (qui se résume à un ensemble de propositions) et un ensemble de « Proposition Under Discussion » (PUD).

Les actes de dialogue sont des actes contextuels constitués d'une *fonction* et d'un *contenu sémantique*. La fonction détermine *comment* mettre à jour les PUD et le contenu sémantique définit *avec quoi* mettre à jour cet ensemble.

Les *jeux de dialogue* ont un *type* (e.g., information, salutations, interrogation, ...) et un *contenu* sous forme de proposition. Les *règles du jeu* sont formalisées sous la forme d'un réseau de transition récursif. Les états du réseau de transition représentent l'état du jeu. Les arcs sont étiquetés par un coup dialogique ou par un autre jeu de dialogue qui sera alors emboîté. L'emboîtement de jeux n'est pas limité. La figure 3.1 présente un exemple de jeu de demande d'information. Ce jeu peut être ouvert par deux coups dialogiques – *qw* (pour *query*) ou *qw-r* (pour *restricted-query*) – suivis d'une réponse (*rw*). Le demandeur peut alors terminer le jeu par un coup de reconnaissance (*ack*) ou demander une confirmation (*cnf*). Les réponses possibles à une demande de confirmation sont des confirmations positives (*ryes*) ou négatives (*rno*), ou une correction de réponses (*rmod*). Les couples confirmation/correction de réponse envisagés sont du type : « Tu rentres à Reims ce week-end ? – Non, à Rouen ! ».

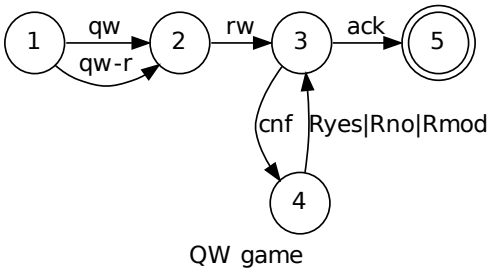

FIGURE 3.1: Jeu de dialogue de demande d'information dans l'approche de Lewin [Lewin 2000]

Les jeux de dialogue mettent à jour les engagements lorsqu'ils se terminent. Ils sont sémantiquement vus comme des structures permettant de prendre des engagements dans le dialogue. Enfin, une séquence d'énoncés réalise une instance d'un jeu si la séquence des types de coup générée est acceptée par le réseau de transition. Notons que le premier coup dialogique d'un jeu établit le jeu et son contenu.

Cette approche se concentre sur la formalisation des règles du jeu de dialogue sous la forme d'un réseau de transition récursif qui définit les enchaînements d'actes. La dynamique des jeux de dialogue considérée est celle de l'emboîtement de jeux. Elle permet de distinguer deux niveaux de gestion du dialogue pour l'agent : le niveau des actes de dialogue et le niveau des jeux de dialogue. Enfin, bien qu'il considère des jeux avec un nombre d'états assez important, Lewin conclut que, de son point de vue, les jeux sont essentiellement des structures planifiées de la forme initiative-réponse ou initiative-réponse-feedback.

Ce point de vue est illustré dans les travaux de [Morge 2013]. Ils envisagent la régulation de la communication entre un agent virtuel vendeur et un humain par des *échanges*, formalisés par des automates à états finis. Ils proposent un échange de recherche d'information (une question

pouvant être suivie d'une réponse ou d'une admission d'ignorance) et un échange de délibération (une proposition qui peut être acceptée ou refusée). Ces échanges sont combinés en *phase* qui les enchaîne séquentiellement et dynamiquement. Une phase de recherche d'information est par exemple constituée de plusieurs échanges de recherche d'information. Le comportement de l'agent consiste à détecter les besoins de l'utilisateur via une phase de recherche d'information, puis à proposer des produits via une phase de délibération.

3.2.2 Les jeux comme unités de type initiative-réponse cohérentes

Hulstijn a proposé de modéliser les motifs récurrents d'actes de dialogue par des jeux de dialogue (suivant la définition de Mann) [Hulstijn 2000a]. Ses travaux sont mis en application par [Dastani 2001] qui propose une méthodologie pour construire des protocoles de négociation flexibles basés sur les jeux de dialogue. Le framework proposé est prétendu assez générique pour être applicable à d'autres types de dialogue (cf. [Hulstijn 2000a, Hulstijn 2000b]).

Les agents considérés peuvent avoir des intérêts concurrents, mais ils partagent au moins un but commun qui est de coordonner leurs actions. Pour représenter les actions coordonnées, les auteurs utilisent des représentations partielles sous forme de *recettes* [Grosz 1996]. Un type particulier de ces recettes sont les *jeux de dialogue*.

Un dialogue est vu comme une suite *cohérente* d'énoncés. Les énoncés peuvent être décomposés en actes de dialogue comprenant une *fonction communicative* appliquée à un *contenu sémantique*. La fonction communicative de l'acte de dialogue peut avoir un effet sur la *tâche* (activité sous-jacente qui motive le dialogue) et/ou sur les *processus d'interaction* (comme les *feedbacks*). L'échange d'énoncés entre les participants enrichit un *tableau conversationnel* qui garde les traces des déclarations faites et des engagements pris. L'objectif d'un modèle du dialogue est alors de combiner les actes de dialogue en des échanges cohérents et utiles. Les auteurs définissent la notion clé de *cohérence*. Un acte de dialogue dans un dialogue de négociation est *cohérent* avec le *contexte du dialogue* si [Dastani 2001] :

1. la fonction relative à la tâche de l'acte correspond à un plan de la tâche qui peut permettre à l'agent d'atteindre ses buts apparents, et
2. la fonction relative à l'interaction de l'acte se conforme aux règles de l'interaction courante.

En fonction de l'attitude de l'agent et de qui a l'initiative, 1 ou 2 prime.

La cohérence est donc décomposée en deux niveaux, celui de la tâche et celui de l'interaction. La cohérence au niveau de la tâche peut impliquer des inférences complexes relatives à la planification tenant compte du contexte du dialogue. Mais comme le fait remarquer Hulstijn [Hulstijn 2000b], le niveau de l'acte de dialogue n'est pas celui qui est intéressant au niveau de la planification. Par exemple, l'effet escompté d'une question est d'obtenir la réponse. Dans la théorie des actes de langage, cela signifie que la condition de succès de l'acte de question implique sa réponse (un autre acte de langage). Autrement dit, une question n'a d'intérêt que dans un *échange* de type question-réponse. Hulstijn en conclut que les actes de dialogue n'ont de sens pour la planification que comme partie d'un échange et plus particulièrement d'une unité initiative-réponse. Il ajoute que ces unités sont les plus petites actions qui peuvent être encore appelées conjointes. Ce qui conduit à la conclusion suivante : la cohérence au niveau de l'interaction peut être cernée par des recettes pré-compilées modélisées par les jeux de dialogue.

La fonction communicative liée à l'interaction est un coup dans un jeu de dialogue. Ce coup peut être *initiatif* ou *réactif*. Les auteurs notent que les *initiatives* et les *réponses* sont souvent reliées par une structure en triangle (figure 3.2) composée d'une *initiative* suivie d'une réponse

qui peut être *positive* ou *négative*, ou sinon d'un nouvel essai. Par exemple, une proposition est une initiative, une acceptation de cette proposition correspond à une réponse positive, un refus à une réponse négative et une contre-proposition peut être vue comme un nouvel essai. L'exécution par les interlocuteurs d'une *IR-unit* conduit à un ajout d'information dans leur *fonds commun* (modélisé par le tableau conversationnel). Ainsi, une réponse positive a un effet de *grounding* [Clark 1989]. Par exemple, bien répondre à une question est un *feedback* fort et implicite indiquant la bonne attention, perception, interprétation et compréhension de cette dernière.

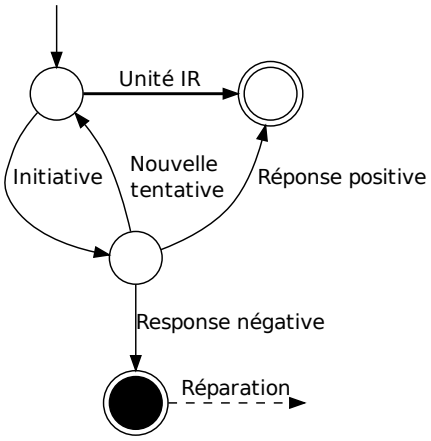

FIGURE 3.2: « Initiative-response units » tiré de [Hulstijn 2000a]

Les jeux de dialogue pour la négociation

La structure générale d'un jeu de dialogue suit cette structure en triangle. Elle est présentée en figure 3.3.

$$\begin{aligned} exchange(a,b,\zeta) = initiative(a,b,\eta);(&pos_response(b,a,\zeta) \\ &\mid neg_response(b,a,\zeta) \\ &\mid retry(a,b,\xi)), \\ &\text{where } M_{a,b} \vDash coherent(\eta,\zeta) \end{aligned}$$

FIGURE 3.3: Structure générale d'un jeu de dialogue dans l'approche d'Hulstijn tirée de [Dastani 2001]. Le « ; » symbolise la séquentialité.

Un échange entre deux interlocuteurs a et b, initié par a, est suivi par une réponse de b soit positive, soit négative ou d'un nouvel essai de a. η, ζ et ξ représentent les contenus sémantiques. $M_{a,b}$ représente le contexte (partagé) du dialogue. En plus des enchaînements d'actes, les jeux de dialogue spécifient des contraintes sémantiques sur les contenus. Ainsi, un échange est *permis* si la contrainte de cohérence entre les contenus sémantiques des initiatives et des réponses est remplie (ce qui est représenté par $M_{a,b} \vDash coherent(\eta,\zeta)$ dans la figure 3.3).

La figure 3.4 présente un exemple d'échange basé sur la structure décrite en figure 3.3. Cet exemple décrit un échange de demande d'information. La relation de cohérence porte dans ce

$$
\begin{aligned}
information_seeking(a,b,\psi) = & question(a,b,?\phi);(answer(b,a,\psi) \\
& \mid clarification_question(b,a,?\chi)), \\
& \text{where } M_{a,b} \not\models \neg\psi \text{ (consistent)} \\
& \text{where } M_{a,b} \not\models \psi \text{ (informative)} \\
& \text{where } M_{a,b} \models relevant(?\phi,\psi) \\
& \text{where } M_{a,b} \models licensed(?\phi,\psi)
\end{aligned}
$$

FIGURE 3.4: Échange de recherche d'information dans l'approche d'Hulstijn tiré de [Dastani 2001]

cas sur le contenu sémantique de la réponse. Celui-ci doit être consistant, informatif, pertinent et suffisant par rapport à la question.

Les jeux de dialogue peuvent être combinés dynamiquement et statiquement

La structure en triangle des *IR-units* peut laisser à penser qu'une réponse doit intervenir immédiatement après une initiative. Les auteurs précisent que ce n'est pas le cas : plusieurs jeux peuvent être ouverts en même temps. Ainsi, une structure présentant des *emboîtements* de type $(I(IR)(IR)R)$ est possible. Aucune contrainte de cohérence n'est clairement explicitée pour ce type de combinaison. Notons que ce type de combinaison est dynamique.

Les jeux peuvent également être statiquement *séquencés* ou *chaînés*. De la même manière que pour les actes de dialogue, des relations de cohérence vont permettre d'expliciter la façon dont les jeux peuvent être combinés.

$$
\begin{aligned}
game(a,b,(\eta.\zeta)) = & exchange(a,b,\eta);game(a,b,\zeta); \\
& \text{where } M_{a,b} \models coherent(\eta,\zeta)
\end{aligned}
$$

FIGURE 3.5: Séquencement de jeux dans l'approche d'Hulstijn tiré de [Dastani 2001]. Le résultat d'un tel échange est la combinaison des résultats exprimée par le « . ».

La figure 3.5 présente la définition du séquencement dans cette approche. Les jeux peuvent ainsi être séquencés autant de fois que nécessaire pourvu que la relation de cohérence soit respectée.

$$
\begin{aligned}
& information_exchange(a,b,\phi \wedge \psi) = \\
& information_seeking(a,b,\phi);information_exchange(a,b,\psi) \\
& \text{where } M_{a,b} \models ?\chi \text{ such that} \\
& M_{a,b} \models relevant(?\chi,\phi) \text{ and } M_{a,b} \models relevant(?\chi,\psi)
\end{aligned}
$$

FIGURE 3.6: Exemple de séquencement de jeux de recherche d'information dans l'approche d'Hulstijn tiré de [Dastani 2001]

S1 : Est-ce que j'ai besoin d'un visa pour voyager ?
H2 : Tu pars d'où ?
S3 : De France.
H4 : Tu vas où ?
S5 : En Chine.
H6 : Alors oui !

Dialogue 3.3 – Exemple de dialogue de séquencement de jeux de recherche d'information aux énoncés [2;3] et [4;5] par rapport à la question de plus haut niveau posée au tour 1

La figure 3.6 présente un exemple qui spécifie que des échanges de demande d'information peuvent être séquencés quand ils continuent d'être pertinents (via la relation « relevant ») par rapport à une question de plus haut niveau (des questions dépendantes dans la terminologie de Ginzburg, cf. section 2.2.3). Le dialogue 3.3 illustre informellement le type de séquencement capturé par cette définition. Une question sur la nécessité d'un visa est soulevée au tour de parole **S**1. Pour y répondre, deux jeux de recherche d'information sont séquencés (énoncés [2;3] et [4;5]) dont les réponses sont *pertinentes* par rapport à la question initialement posée.

Les échanges peuvent également être chaînés. Deux jeux J_1 et J_2 sont chaînés si et seulement si l'acte de dialogue réponse du jeu J_1 est l'acte initiatif du jeu J_2. L'exemple le plus démonstratif de cette combinaison est l'enchaînement proposition/contre-proposition/... où la contre-proposition joue le rôle à la fois de réponse négative dans l'échange n, et le rôle initiatif (en tant que proposition) dans l'échange $n + 1$.

3.2.3 Les jeux comme structures pour l'engagement dans le dialogue

Sur le modèle des systèmes dialectiques [Hamblin 1970, Walton 1995], Maudet envisage les jeux de dialogue comme des structures capturant les engagements que les interlocuteurs contractent pendant le dialogue [Maudet 2001].

Notion d'engagement social

Les *engagements sociaux* sont à distinguer des *états privés* des agents tels que croyance et désir. Les engagements sociaux sont des engagements qui lient un agent à une communauté [Singh 1991, Castelfranchi 1995]. Les caractéristiques des engagements sociaux sont les suivantes (reprises et adaptées de [Walton 1995, Singh 1999, Maudet 2001, Pasquier 2005]).

Les engagements sont sociaux. Ils sont contractés envers d'autres membres d'une communauté (à distinguer des engagements d'un agent envers lui-même [Cohen 1990b]. Dans le cas du dialogue, les engagements sont pris envers les autres interlocuteurs. On appelle *débiteur* la personne qui s'engage et *créditeur* la personne envers qui l'engagement est pris.

Les engagements sont publics et *accessibles* aux membres de la communauté. Ils sont généralement stockés dans un tableau appelé *commitment store* qui a pour vocation de faire partie de la zone publique de l'état d'information d'un système de dialogue.

Les engagements sont propositionnels ou en action. Usuellement, les engagements sont distingués entre ceux dirigés vers le *présent* (les engagements propositionnels) et ceux dirigés vers le *futur* (les engagements en action). Les engagements propositionnels portent sur l'état du

monde (e.g., *« Paris est la capitale de France »*) tandis que les engagements en action portent sur des actions à réaliser dans le futur (*« Demain, j'irai nager. »*). Cette vision est toutefois simplificatrice puisqu'il est tout à fait possible de contracter des engagements propositionnels dirigés vers le futur, i.e., décrivant un état du monde futur (*« La vaisselle sera faite avant midi. »*). Il existe une tendance à aller vers une uniformisation du traitement de ces deux types d'engagements [Singh 2008].

Les engagements peuvent être conditionnels. À l'instar d'une règle (*« si tu me payes, je te livre la commande »*), un engagement peut être conditionnel. Un engagement conditionnel tient si sa condition est remplie. La considération de ces types d'engagements peut être effectuée via des engagements d'ordre supérieur (e.g., des engagements permettant de créer de nouveaux engagements) [Singh 1999, Maudet 2001, Pasquier 2005], ou en envisageant les engagements conditionnels comme fondamentaux [Singh 2008].

Les engagements sont dialogiques ou extra-dialogiques. Dans le dialogue, on distingue les engagements *dialogiques* qui sont pris dans un contexte local particulier (e.g., un jeu de dialogue), des engagements *extra-dialogiques* qui sont pris dans le contexte de la tâche sous-jacente du dialogue.

Les engagements sont explicites ou implicites. Les engagements *explicites* sont créés par l'occurrence d'actes communicatifs. Les engagements *implicites* sont associés à une connaissance partagée de la communauté qui n'a pas été explicitée (e.g., les règles d'un cadre dialogique comme un protocole).

Les engagements sont datés, i.e., pris à un moment donné.

Enfin, *les engagements peuvent être ordonnés.* Certains engagements en action sont prioritaires sur d'autres car ils doivent être satisfaits en priorité. Par exemple, on répond généralement à la dernière question qui a été posée, remplissant l'engagement de réponse créé par la question, avant de considérer la suivante.

Le tableau de conversation et les engagements sociaux

Dans l'approche de Maudet, l'état du dialogue est représenté par un tableau de conversation qui enregistre les *engagements sociaux* pris par les interlocuteurs à un instant du dialogue.

Maudet distingue les engagements sociaux pour deux interlocuteurs entre ceux qui sont *propositionnels* et ceux qui sont *en action.* Les engagements en proposition $C(x, p)$ expriment qu'un agent x s'engage dans le présent sur une proposition p envers y (non précisé pour alléger l'écriture). Par exemple, si x dit à y : *« Paris est la capitale de la France »*, il contracte l'engagement $C(x, capital(france, paris))$ signifiant que x est engagé envers y sur la proposition $capital(france, paris)$. Les engagements en action sont contextualisés par le jeu j dans lequel ils sont pris. Ils peuvent prendre les formes suivantes :

— $C_j(x, \alpha)$: exprime que x s'engage à ce que α survienne : *« Je vais en Roumanie l'été prochain. »*

— $C_j(x, \alpha_1|\alpha_2)$: exprime que x s'engage à ce que l'événement α_1 ou α_2 survienne.

— $C_j(x, \neg\alpha)$: exprime que x s'engage à ce que l'événement α ne survienne pas.

La formalisation considère les *engagements conditionnels* utilisés pour modéliser la conséquence (contextualisés par le jeu j dans lequel ils sont pris) :

— $C_j(x, \alpha \Rightarrow \beta)$: x s'engage à ce que, si l'événement α se produit, l'opération β est réalisée sur le tableau.

— $C_j(x, \alpha \overset{*}{\Rightarrow} \beta)$: x s'engage à ce que, *à chaque fois que* l'événement α se produit, l'opération β est réalisée sur le tableau.

Enfin, les engagements conjoints sont utilisés pour engager les interlocuteurs sur un jeu j : $C(\{x, y\}, j)$.

Le tableau d'engagements évolue grâce à un ensemble d'opérations (création, satisfaction, annulation, prioritarisation) qui permettent de créer, supprimer, annuler et ordonner les engagements.

Les actes de dialogues

Les actes de dialogues considérés par Maudet sont des actes contextuels *multiniveaux* constitués d'une fonction et d'un contenu sémantique. Les niveaux considérés sont le niveau *informationnel* et le niveau de *contextualisation*. Le premier niveau permet aux interlocuteurs d'échanger de l'information et leur contenu est propositionnel. Par exemple, $affirme(x, beau)$ est l'acte par lequel x affirme la proposition « Il fait beau. ». Le deuxième niveau permet aux interlocuteurs de modifier le jeu de dialogue courant (i.e. le contexte local de l'interaction). Par exemple, $prop.entrée(y, interrogation(p))$ est l'acte par lequel y propose d'entrer dans un jeu d'interrogation.

Les jeux de dialogue sont des structures qui capturent les engagements créés durant le dialogue

Les jeux représentent des *conventions* entre des interlocuteurs et sont des structures *partagées* et *bilatérales*. Maudet propose la distinction de deux types de jeux : les *jeux de dialogue* et les *jeux de communication*. Un jeu de dialogue est un cas particulier d'*activité conjointe* activée *temporairement* pendant le dialogue dans un *but spécifique* (e.g., pour l'obtention d'une information, pour la demande de réalisation d'une action, etc.). Certains représentent un motif d'interaction local dans le dialogue. Ils doivent être établis par les participants du dialogue (les jeux de dialogue peuvent être *ouverts*, *fermés* ou *proposés*). Les jeux de communication sont dédiés à des processus de gestion de l'interaction plus généraux (comme la contextualisation, la compréhension mutuelle, la gestion des tours de parole, etc.) et sont *toujours* activés.

Les jeux de dialogue sont définis par des *conditions objectives* qui portent sur des engagements sociaux, et non sur des dispositions subjectives des interlocuteurs comme des états mentaux. Celles-ci sont de trois types :

entrée définit l'état du tableau de conversation afin de permettre l'entrée dans le jeu.

succès définit l'état du tableau qui correspond au succès du jeu.

échec définit l'état du tableau qui correspond à l'échec d'un jeu. Maudet adopte une hypothèse simplificatrice concernant les conditions d'échec : elle consiste en l'énonciation d'un coup dialogique vide [Maudet 2001] (p. 125).

Les jeux sont également définis par des *règles* qui précisent les engagements conversationnels que les interlocuteurs contractent lorsqu'ils entrent dans le jeu. Par exemple, ces engagements précisent les actes attendus ou interdits dans le cadre du jeu. Ils vont guider le comportement dialogique des interlocuteurs.

Cette définition permet de distinguer deux propriétés des jeux. Un jeu est *collaboratif* si et seulement si les conditions de succès pour l'initiateur et le partenaire sont identiques. Il est *symétrique* si et seulement les règles qui s'appliquent aux deux joueurs sont identiques.

Le tableau 3.4 présente l'exemple du jeu d'explication. Ce jeu est collaboratif et asymétrique. Il permet à l'initiateur de demander une explication au partenaire sur une proposition p. Les conditions d'entrée se résument à ce que le partenaire soit engagé sur la proposition p. Celle-ci traduit le fait qu'on ne peut pas demander une explication sur une proposition à quelqu'un s'il n'est pas engagé sur celle-ci. Les règles peuvent se lire comme suit. L'initiateur s'engage à jouer un *défi* sur la proposition. Le partenaire s'engage à jouer une explication (sous la forme d'un *affirme*) ou à retirer son engagement si le *défi* est joué. Ce jeu incorpore une règle différente d'une règle de production d'actes de dialogue. Ici, il s'agit d'une interdiction. Le partenaire ne peut pas justifier la proposition sur laquelle il est engagé en affirmant cette même proposition. Le jeu est un succès lorsque le partenaire est engagé sur un ensemble de propositions qui permet de supporter la proposition p.

	initiateur (x)	partenaire (y)
entrée		$C(y, p)$
succès	$C(y, s)$ tq. $s \vDash p$	$C(y, s)$ tq. $s \vDash p$
règles	défi(x, p)	défi$(x, p) \Rightarrow$ créer$(y, C_j(y,$ affirme(y, s)|retrait(y, p)|vide$(y)))$
		$\neg$affirmer(y, p)

TABLEAU 3.4: Exemple du jeu de dialogue d'explication dans l'approche de Maudet [Maudet 2001]. $j =$ EXPLICATION(p).

Les jeux de dialogue constituent des contextes dialogiques au sein desquels les interlocuteurs contractent des engagements conversationnels les uns envers les autres. Ces engagements vont *guider leurs comportements dialogiques*. Notons que la différenciation des conditions de succès entre l'initiateur et le partenaire du jeu permet de considérer des jeux *collaboratif* ou *non*. Cette formalisation admet l'avantage de pouvoir spécifier les jeux à *plusieurs niveaux de granularité*. Les engagements peuvent être utilisés pour spécifier précisément des enchaînements d'actes attendus comme dans les approches de Lewin (cf. section 3.2.1) ou d'Hulstijn (cf. section 3.2.2). Mais ils peuvent également spécifier des prérogatives générales qui devront être suivies (e.g., ne pas se contredire dans un jeu d'argumentation).

Les jeux de communication

Les jeux de communication sont spécifiés sous la forme d'engagements conditionnels persistants ($C_j(-, \alpha \overset{*}{\Rightarrow} \beta)$). Notons que les jeux de communication n'ont ni condition d'entrée ni condition de sortie puisqu'ils sont activés en permanence. Maudet propose deux jeux de communication, celui de contextualisation et celui d'évaluation, tout en précisant que d'autres phénomènes comme l'établissement (cf. section 2.2.1) ont vocation à être spécifiés sous cette forme.

Le *jeu d'évaluation* permet de capturer l'idée que les interlocuteurs tentent de s'accorder sur l'état du monde durant le dialogue. Les interlocuteurs ont donc tendance à *évaluer* ce qui est dit en concédant ou contredisant. Le tableau 3.5 présente la spécification de ce jeu de communication. À chaque affirmation (par l'acte *affirme*), l'allocutaire contracte l'engagement d'évaluer positivement (via l'acte *concède*) ou négativement (via l'acte *contredit*) la proposition. Un interlocuteur contredit contracte l'engagement lui permettant de retirer son engagement sur la proposition affirmée (via l'acte *retrait*).

Événements (α)	Opérations (β)
affirme(x,p)	créer$(x, C_{ev}(y, \text{concède}(y,p)\|\text{contredit}(y,p)))$ créer$(x, C(x,p))$
concède(x,p)	créer$(x, C(x,p))$
retrait(x,p)	annuler$(x, C(x,p))$
contredit(x,p)	créer$(x, C_{ev}(y, \text{retrait}(y,p)))$ créer$(x, C(x,\neg p))$

Tableau 3.5: Jeu de communication d'évaluation dans l'approche de Maudet [Maudet 2001]

L'approche de Maudet est la seule à considérer explicitement l'*établissement* du jeu de dialogue via le jeu de *contextualisation* [Maudet 2003a]. Le jeu correspond au mécanisme de négociation proposé par Mann (cf. section 3.1.3). Il traduit le fait que les interlocuteurs négocient l'entrée et la sortie du jeu sous la forme d'une proposition suivie d'une acceptation ou d'un refus. Pour ce faire, les interlocuteurs peuvent produire des actes de contextualisation pour ouvrir un jeu (*prop.entrée*, *acc.entrée*, *ref.entrée*) ou pour fermer un jeu ouvert (*prop.sortie*, *acc.sortie*, *ref.sortie*).

La dynamique des jeux de dialogue

Maudet envisage deux possibilités de combinaison de jeux de dialogue : le pré-séquencement et l'emboîtement.

La notion de pré-séquencement traduit le fait que le premier jeu *prépare* la réalisation du second. La difficulté de ce type de combinaison est que la pré-séquence est souvent observée a posteriori, une fois que le deuxième jeu a été joué. Maudet s'intéresse à un type particulier (et plus simple) de pré-séquence qu'il nomme *pré-séquence informée*. Il s'agit du cas où le jeu préparé est connu à l'avance, i.e., j_1 pré-séquence j_2 si j_1 est *ouvert* alors que j_2 est *proposé*. Une telle combinaison est envisageable pour s'assurer que les conditions d'entrée du jeu sont vérifiées. C'est la cas où une proposition d'entrée dans un jeu a été jouée de manière sous-spécifiée (les conditions d'entrées ne sont pas toutes vérifiées) et où un second jeu a été ouvert pour établir les conditions d'entrée manquantes.

Nous avons rencontré les emboîtements de jeux dans l'ensemble des approches que nous avons présentées. L'idée peut se résumer ainsi : un jeu j_2 est emboîté dans j_1 si le jeu j_2 est ouvert alors que le jeu j_1 était déjà ouvert. Quelles sont les conséquences d'un emboîtement de jeux ? Maudet en liste trois. Premièrement, les engagements contractés dans le jeu emboîté sont prioritaires sur les engagements du jeu parent. Cela traduit l'idée que le contexte dialogique le plus spécifique est aussi le plus saillant. Généralement, on répond à la dernière question qui a été posée avant de répondre aux autres (cet exemple devient contre-exemple quand on parle de *questions coordonnées*). Ensuite, le jeu parent domine ses jeux emboîtés. La conséquence de cette observation est que l'annulation du jeu parent conduit à l'annulation de ses jeux emboîtés. Enfin, les engagements du jeu emboîté peuvent être contradictoires avec les engagements du jeu parent. Dans ce cas, la règle de priorité s'applique. Un acte de dialogue peut violer un engagement d'un jeu j uniquement dans le cas où cela lui permet de satisfaire un engagement d'un jeu emboîté dans j.

3.3 Discussion

Dans cette discussion, nous défendons l'intérêt d'aller vers une architecture d'agent mixte réactive/délibérative basée sur les jeux de dialogue vus comme des structures d'engagements. Nous montrons dans la section 3.3.1 l'intérêt du formalisme basé sur les engagements. Puis nous soulignons les avantages qui font des jeux de dialogue une structure à considérer pour des agents interactifs (cf. section 3.3.2). Enfin, nous synthétisons notre point de vue sur les modèles mixtes à base de jeux de dialogue (cf. section 3.3.3).

3.3.1 Vers un modèle normatif des jeux de dialogue basé sur les engagements

Dans cette section, nous explorons les liens entre les formalisations de jeux de dialogue présentées et la structure intentionnelle. Deux points de vue sur le lien entre structure du dialogue et jeux de dialogue se dégagent. Ensuite, nous comparons les approches présentées sur certaines caractéristiques générales des jeux de dialogue (vues en section 3.1) : les phases de vie d'un jeu, la structuration intra et inter-jeu et la prise en compte de l'axe de l'interaction. Enfin, nous soulignons l'intérêt de concevoir les jeux de dialogue comme structures pour l'engagement dans le dialogue.

Jeux de dialogue et structure intentionnelle du dialogue

Lewin, Hulstijn et Maudet abordent explicitement le problème des jeux de dialogue et des liens avec les processus de délibération des agents. Deux points de vue semblent se distinguer. Le premier est partagé par Lewin et Hulstijn : les jeux de dialogue sont essentiellement des *unités de type initiative-réponse* planifiées par les agents. Le second point de vue est celui de Maudet : certains jeux sont planifiés et trouvent pleinement leur ancrage dans la structure intentionnelle du dialogue, et certains jeux et *certaines combinaisons de jeux* s'ancrent dans une structure intermédiaire *motivée dialogiquement* (plutôt qu'intentionnellement).

Nous allons présenter plus précisément ces deux visions.

« Dialogue games are recipes for joint action » La vision des jeux de dialogue de Hulstijn [Hulstijn 2000b, Hulstijn 2000a] provient de la réponse à la question suivante : si l'on considère les processus de communication eux-mêmes comme des actions conjointes représentées dans un formalisme de l'action et de la planification conjointe, alors quelles sont les *recettes partielles* ? Pour Hulstijn la structuration du dialogue est fondamentalement sur deux niveaux :

Structure locale : Ensemble de règles utilisées localement de manière réactive par les agents (par exemple, les règles à l'intérieur des jeux).

Structure globale : Donnée par la structure inter-jeu issue de la structure intentionnelle.

Pour résumer, le dialogue est vu comme une activité conjointe entre les interlocuteurs. Le dialogue est une activité extrêmement dynamique, il paraît donc peu plausible que les interlocuteurs puissent délibérer à l'infini. La coordination de cette activité doit reposer sur un précédent établi ou sur certaines conventions. Comme dans toute activité répétitive, certaines situations sont récurrentes et deviennent par conséquent familières. La coordination des actes de dialogue peut se baser sur des motifs d'interaction conventionnels formalisés par les jeux de dialogue. Pour Hulstijn, les jeux de dialogue (tels que définis section 3.2.2) sont les *recettes pour l'action conjointe.*

L'idée est que les *agents planifient au niveau des jeux*. Notons que ce point de vue est également partagé par Lewin (cf. section 3.2.1). Le modèle DTask (cf. section 2.1.3) peut être vu comme une mise en œuvre simplifiée de cette approche au niveau de surface des énoncés.

Néanmoins, la vision des jeux de dialogue comme recettes pour l'action jointe est réductrice [Maudet 2001]. Le premier point concerne le *statut des jeux*. Si ces derniers sont vus comme des prêt-à-planifier, ils ne permettent pas de prendre en compte une variété de sous-dialogues qui ne peuvent pas être planifiés comme les dialogues de clarification (e.g., dialogue 3.5, p. 76). Et de manière plus générale, les jeux de la version d'Hulstijn sont des structures qui ne fournissent finalement que la part conventionnelle des plans des modèles de l'action conjointe collaborative. De cette dernière remarque naît la seconde critique. Certains types de dialogue ne peuvent pas trouver leur place dans un modèle de l'action conjointe *collaborative*. Il suffit de prendre le cas du débat où il semble difficile de soutenir qu'il y a effectivement *collaboration* entre les interlocuteurs. Ceux-ci coopèrent effectivement sur une *activité commune* qui va les contraindre, sans pourtant que cette activité ne soit un plan partagé.

De prêt-à-planifier à guide de comportement Le mécanisme sous-jacent des jeux de dialogue doit être plus flexible tout en contraignant les actions dialogiques. C'est tout l'objet de la formalisation des jeux de dialogue comme des structures pour l'engagement de Maudet (cf. section 3.2.3). Les jeux de dialogue ne sont plus vus comme des recettes à inclure dans les plans partagés. Ils sont vus comme des *guides* pour le comportement communicatif que les agents peuvent décider de suivre ou pas. Le comportement des agents est guidé par les engagements qu'ils contractent au fur et à mesure de la conversation. Cette vision permet de gagner en *flexibilité* puisque des jeux opportunistes (clarification, évaluation, etc.) peuvent être ouverts à tout moment.

Que devient alors la relation entre les jeux de dialogue et la structure intentionnelle ?

La réponse la plus aboutie à notre connaissance est apportée par Maudet [Maudet 2001]. Il souligne que les jeux peuvent être combinés pour former une structure plus complexe *sans besoin de se référer à la structure intentionnelle*, par exemple par des *emboîtements* ou des *préséquences* (éventuellement informées). La rupture avec la vision précédente tient donc en cette idée qu'au cours d'un dialogue certains jeux sont planifiés et d'autres sont ouverts de manière plus réactive.

Le dialogue 3.4 présente une structure d'emboîtement dominé par les buts. L'enchaînement des jeux est alors expliqué par la structure intentionnelle. La motivation à jouer au jeu j_2 est dominée au niveau de la structure intentionnelle par la motivation à jouer au jeu j_1.

J1 : Est-ce que Diana vient avec nous au cinéma ?
G2 : Est-ce que Zach vient ?
J3 : Non !
G4 : Diana ne viendra pas alors.

Dialogue 3.4 – Emboîtement intentionnellement fonctionnel : j_2 ([2;3]) est dominé par j_1 ([1;4])

Les jeux ouverts de manière réactive incluent les *dialogues incidents*. Il s'agit par exemple de dialogues de clarification, correction, d'évaluation, etc. Le dialogue 3.5 présente un tel exemple. Ce type d'emboîtement consiste en un changement d'axe dans le dialogue pour passer dans la gestion de l'interaction. Il s'agit d'un emboîtement réactif d'un jeu de communication.

Pour synthétiser, Maudet distingue trois niveaux de structuration [Maudet 2001] :

S1 : Comment t'appelles-tu ?
G2 : GDD
S3 : Pardon ?
G4 : Je m'appelle Guillaume !

Dialogue 3.5 – Emboîtement interactionnellement fonctionnel : le jeu de communication ([2;4]) est emboîté dans le jeu de dialogue ([1;4]).

Structure locale : Elle provient des règles contractées lors des jeux de dialogue.

Structure intermédiaire : Elle est issue de règles et contraintes qui n'ont pas besoin de se référer au niveau intentionnel. Elle traduit l'occurrence de *dialogues incidents*.

Structure globale : Elle provient de la *structure intentionnelle*, il s'agit des jeux planifiés par des structures de délibération discursive.

Comparaison des approches présentées

Nous analysons désormais les points communs et les divergences des formalismes de jeu de dialogue au regard de la métaphore décrite en section 3.1.4. Nous comparons les approches sur la prise en compte des « phases de vie » d'un jeu identifiées par Levin, Moore et Mann : entrée/pratique/sortie (cf. section 3.1). Puis nous étudions le pouvoir expressif des formalismes concernant les *règles* de jeu (i.e., la structure *intra-jeu*). Nous constatons ensuite les différentes combinaisons de jeux proposées par les formalismes. Pour finir, nous examinons la prise en compte des axes du dialogue (tâche et gestion de l'interaction, cf. section 1.1.2) dans les jeux de dialogue.

Les phases de vie d'un jeu de dialogue Des trois approches que nous avons considérées, la seule à modéliser les phases d'ouverture, pratique et fermeture d'un jeu est l'approche de Maudet via le jeu de communication de contextualisation (cf. section 3.2.3).

La structuration intra-jeu (règles) Les trois versions apportent des contraintes *structurelles* plus ou moins forte. La *version automate* permet de spécifier facilement des enchaînements d'actes attendus. En contrepartie, il est nécessaire de spécifier les jeux de la manière la plus complète possible, le risque étant de perdre en flexibilité et réutilisabilité. La *version unité IR* part d'un a priori fort, à savoir la structure triangulaire générique. La partie spécifique des jeux s'appuie sur les actes de dialogue initiatifs et réactifs ainsi que sur les *contraintes sémantiques*. C'est la seule approche présentée à considérer ce genre de contraintes. Enfin, la *version engagement* semble être celle qui offre la plus grande souplesse structurelle car elle permet de spécifier très localement et partiellement les séquences attendues tout en permettant une plus grande variété de règles (e.g., les interdictions de coup dialogique). De plus, c'est le tableau de conversation qui est plus largement pris en compte dans la *version engagement* pour déterminer les prochains coups possibles. Cela inclut le dernier coup dialogique joué mais également d'autres informations comme les engagements contractés au cours du dialogue. Cette richesse peut être un avantage par rapport à la version automate au contexte beaucoup plus réduit.

La structuration inter-jeu Trois types de structuration inter-jeu sont évoqués : (i) emboîtement (ou imbrication), (ii) pré-séquence, et (iii) enchaînement. Le dernier n'est abordé que par la formalisation d'Hulstijn.

Comme dans la structuration intra-jeu, Lewin et Maudet ne s'intéressent qu'aux *contraintes structurelles* qui pèsent sur les structures inter-jeu. La formalisation de Lewin permet d'emboîter des jeux de manière infinie grâce aux réseaux de transition récursifs. Selon ce dernier, les contraintes sur la profondeur de l'emboîtement proviennent de la structure intentionnelle qui permet de combiner les jeux.

La formalisation de Maudet précise les contraintes structurelles dans le jeu de contextualisation. Celles-ci permettent de gérer les intercalements de jeu (c'est un type de négociation entre jeu), les pré-séquences et les emboîtements. Il s'agit de contracter des engagements adéquats qui vont permettre d'accepter l'entrée des jeux dès l'atteinte des conditions de succès du jeu intercalé ou des conditions d'entrée du jeu pré-séquencé dans le cas du pré-séquencement. Ce dernier précise qu'il est nécessaire de définir des *contraintes de cohérence* au niveau des *types* de jeux et des *sujets* des jeux. Ces aspects peuvent nécessiter de se référer aux structures intentionnelles des interlocuteurs.

Enfin, la formalisation d'Hulstijn permet aux jeux de s'emboîter dynamiquement (par exemple $(I(IR)(IR)R)$). De la même manière que Lewin, cette structuration dynamique provient de la structure intentionnelle des interlocuteurs (i.e., comment les jeux sont planifiés). Similairement à la structuration intra-jeu, Hulstijn définit récursivement le séquencement de jeu de manière générique : il consiste en une séquence d'un échange (IR) suivi d'un jeu (qui peut lui même être une séquence). Le séquencement doit respecter une *contrainte sémantique* sur les contenus sémantiques des actes. La structuration d'enchaînement est un cas limite des séquences : c'est le cas où l'acte réponse d'un échange est également initiatif de l'échange qui suit. Typiquement, les structures enchaînées sont de type question/réponse/évaluation ou de type proposition/contre-proposition/etc. Ces deux dernières structurations sont *statiques* (les structurations doivent être prédéfinies avant l'utilisation).

Axes de dialogue et jeux de dialogue Le dialogue est une activité complexe, notamment parce qu'elle évolue sur deux *axes*, celui de la tâche et celui de l'interaction, et que le dernier axe peut lui-même être décomposé en *dimensions* et *niveaux* (cf. section 1.1.2). Quel est l'impact de ces observations sur les différentes approches de formalisation des jeux de dialogue ?

Seule l'approche de Maudet fait apparaître explicitement le niveau de la gestion de l'interaction via les *jeux de communication*. Ces derniers modélisent les conventions implicites présentes dans tout type de conversation comme la fonction évaluative, le *grounding*, la gestion des tours de parole, etc. Maudet ne présente pas explicitement des jeux de communication pour le *grounding* ou la gestion des tours de parole, mais signale que l'alliance entre les engagements et les actes multi-niveaux paraît suffisante pour modéliser ces processus.

Les autres approches mêlent ces deux axes. C'est le cas par exemple des automates présentés par Lewin, et plus particulièrement de celui de demande d'information (figure 3.1, p. 65). En effet, les transitions entre les états 3 et 4 sont en fait un mécanisme minimal de *feedback* permettant de gérer des couples confirmation/correction via les actes *cnf*, *Ryes*, *Rno*, *Rmod* (cf. section 3.2.1).

En conclusion, sur les 4 approches présentées, seule l'approche de Maudet distingue jeux de communication et jeux de dialogue dans le but de modéliser les processus de gestion de l'interaction. Cette distinction est motivée par la généricité des jeux de communication qui sont présents dans toute conversation.

Vers un modèle normatif des jeux de dialogue basé sur les engagements

De l'ensemble des formalisations présentées, celle de Maudet semble sortir du lot. Cette approche est plus complète dans la formalisation de la métaphore des jeux de dialogue par sa considération explicite de l'établissement de jeux, sa considération explicite de l'axe de l'interaction du dialogue et son pouvoir expressif plus grand dans les règles (e.g., via les interdictions). En outre, cette approche affine les liens entre la structure du dialogue, la structure intentionnelle et les jeux de dialogue en faisant apparaître un niveau intermédiaire (cf. section 3.3.1). Au-delà de ces caractéristiques, c'est une approche qui propose une structure tractable pour des agents dialogiques via la notion d'*engagement* traduisant l'aspect normatif de la métaphore des jeux de dialogue. Nous rejoignons le point de vue de Maudet et proposons d'aller vers des jeux de dialogue vus comme des structures capturant les engagements que les interlocuteurs contractent pendant le dialogue.

3.3.2 Avantages et limites des jeux de dialogue

Dans cette partie, nous présentons les limites générales des jeux de dialogue et leurs avantages pour la modélisation du dialogue pour un agent interagissant avec un humain. Puis nous nous intéressons aux avantages et limites spécifiques de la vision des jeux de dialogue comme structures pour l'engagement.

Limites des jeux de dialogue

Avant d'aborder les avantages, notons tout d'abord quelques limites de la structure de jeu de dialogue. La métaphore des jeux de dialogue est essentiellement *normative* et conduit à des modèles comme ceux de Maudet (cf. section 3.2.3) *motivant* les agents à se conformer à un comportement plutôt que de les *contraindre* (comme les protocoles). La problématique de l'intégration de notions sociales dans le processus délibératif d'un agent interactif est une question ouverte. Le comportement des agents doit être *normatif* pour prendre en compte les jeux de dialogue en cours et agir en fonction des engagements qu'ils ont contractés, et *délibératif* pour prendre en compte leurs désirs et intentions. Cependant, il est possible de spécifier de manière procédurale la balance entre le comportement réactif et le comportement délibératif dans le cas particulier de certains systèmes [Traum 1997, Larsson 2002a, Orkin 2013].

En outre, les agents interactifs sont de plus en plus amenés à interagir dans des *environnements ouverts* [Bohus 2009a] et *multipartites* [Traum 2004, Kronlid 2008]. À cet égard, la métaphore des jeux de dialogue nécessite un travail d'adaptation non trivial [Mann 2002].

Avantages des jeux de dialogue

Dans cette partie, nous recensons les avantages des jeux de dialogue en complétant des observations antérieures [Hulstijn 2000b, Maudet 2001, Pasquier 2005]. De solides arguments permettent de supporter le fait que les jeux de dialogue sont les structures à considérer pour modéliser le dialogue à partir de motifs d'interaction, et en particulier le dialogue Homme-Machine.

Les jeux ont une réalité empirique Les jeux peuvent être analysés à partir de caractéristiques de surface des énoncés [Kowtko 1991, Carletta 1997]. Des expériences d'annotations sur

le corpus de dialogues d'orientation MAPTASK ont permis de montrer qu'il y avait peu d'erreurs entre des annotateurs novices et des annotateurs experts quand il s'agissait de classifier un énoncé en terme de coups dialogiques et de jeux [Carletta 1997]. Les jeux sont donc perçus de manière intuitive par les interlocuteurs. Néanmoins, cette étude a mis en avant le fait que la notion d'emboîtement était difficile à cerner pour les annotateurs novices [Lewin 2000].

Les jeux peuvent être spécifiés à partir des motifs dialogiques L'idée des jeux de dialogue est de proposer une structure afin de répertorier les conventions d'interactions dialogiques. L'étude des corpus de dialogues Homme-Homme fait apparaître un certain nombre de motifs dialogiques identifiables par des caractéristiques de surface des énoncés (voir, e.g., [Levin 1977, Mann 1988, Hulstijn 2000b, Pauchet 2006]). Ces motifs prennent notamment la forme de *paires adjacentes* [Sacks 1974, Hulstijn 2000a] (e.g., question/réponse, suggestion/acceptation) voire de motifs en trois temps initiatif/réactif/évaluatif [Bilange 1991b, Clark 1996] (e.g., question/réponse/évaluation).

Une métaphore utile pour modéliser ces motifs d'interaction est celle des jeux de dialogue telle que définie en section 3.1 [Hulstijn 2000b, Maudet 2001]. Celle-ci permet de lier ingénieusement les buts des interlocuteurs (non-nécessairement compatibles) aux attentes que les interlocuteurs peuvent avoir les uns envers les autres en terme de participation à l'activité (i.e., les actions participatives). En somme, les jeux de dialogue permettent de capturer des activités dialogiques similaires à une partie d'échec et qui font défaut aux modèles de l'action et de la planification conjointe actuels limités aux activités strictement collaboratives.

L'idée est donc de formaliser des jeux de dialogue à partir de ces motifs d'interaction [Hulstijn 2000b, Hulstijn 2000a]. Ces derniers peuvent être vus comme une manifestation des *règles* du jeu de dialogue. En outre, il semble possible d'extraire des jeux indépendants du domaine [Hulstijn 2000b, Mann 2002]. Il devient alors envisageable de constituer une bibliothèque de jeux pour *modéliser* et *construire* des systèmes de dialogue, ou comme le présente Hulstijn :

> « If we are looking for a library with tools to model and build dialogue systems for different applications, the level of these games are the right level. » [3] [Hulstijn 2000b] (p. 198)

Les jeux sont utiles pour la gestion du dialogue Les jeux sont des structures prédictives qui spécifient entre autres les actions participatives autorisées des interlocuteurs en fonction de l'état du jeu. Une étude sur le corpus MAPTASK démontre à ce propos que prendre en compte la structure de jeu permet d'améliorer significativement les prédictions sur les actes de dialogue [Poesio 1998a].

En outre, les structures de jeux de dialogue peuvent être utilisées par un gestionnaire de dialogue sur deux niveaux [Maudet 2002a] : l'*interprétation* d'actes de dialogue et leur *génération*. Au niveau interprétatif, les jeux rendent possible la production d'un ensemble d'actes *conventionnellement attendus*, et permettent éventuellement de simplifier le processus de reconnaissance d'intentions. Au niveau génératif, les jeux donnent une motivation conventionnelle à la production de certains actes.

Les jeux permettent de décomposer plus finement la structure du dialogue Enfin, comme nous l'avons vu en section 3.3.1, les jeux de dialogue permettent de décomposer plus fine-

3. L'expression « these games » réfère aux jeux de dialogue tels que définis par Mann [Mann 1988]

ment la structure du dialogue. En particulier, ils permettent d'expliquer certains enchaînements de jeux sans avoir besoin de référer à la structure intentionnelle. En outre, ils permettent de raffiner la coopération dialogique avec le mécanisme d'établissement du jeu de dialogue. Dans le cas général, le contrôle du dialogue est coordonné par les deux participants : l'entrée et la sortie des jeux de dialogue sont gérées par un mécanisme de négociation (bien que souvent implicite). Cette caractéristique prend tout son intérêt dans des dialogues d'*initiative mixte* [Horvitz 1999a, Allen 1999] où les interlocuteurs se partagent l'initiative en fonction de leur expertise (voir par exemple les agents assistants d'initiative mixte [Tecuci 2007]).

Avantages et limites de jeux de dialogue vus comme structure d'engagements

En sus des avantages généraux des jeux de dialogue, l'approche par engagements admet certains avantages spécifiques [Maudet 2002b, Pasquier 2005].

L'adoption d'une *sémantique publique* et d'un *tableau de conversation* permet de ne pas contraindre de manière directe les attitudes mentales utilisées par les agents dialogiques. Théoriquement, les jeux ainsi définis peuvent être utilisés dans une grande variété d'architectures d'agents. Par exemple, les jeux peuvent être utilisés de manière procédurale « à la GoDiS » [Larsson 2002a] ou dans une architecture BDI adaptée aux notions sociales. En outre, le tableau de conversation conserve une trace plus riche du dialogue sur laquelle le système peut fonder sa réponse. La contrepartie est que ce niveau public ne dispense pas de la nécessité d'un niveau privé [Pasquier 2005]. Les jeux doivent être utilisés par un agent capable de raisonner sur cette partie publique afin de l'intégrer à leur processus délibératif [Pasquier 2003].

Les approches par engagements sont plus *flexibles*. Les engagements *motivent* les agents dialogiques à se conformer aux conventions sans les contraindre. Elles permettent de gérer plus facilement des messages inattendus qui peuvent s'intégrer dans un autre jeu. À ce sujet, les jeux de dialogue sont des structures *combinables* favorisant la gestion du dialogue par un ensemble de jeux plutôt que par une seule et même structure rigide comme un protocole. Néanmoins, il est nécessaire de spécifier d'une façon ou d'une autre les combinaisons possibles entre les jeux. Cette tâche est d'autant plus ardue que les combinaisons peuvent provenir de sources différentes (*structure intentionnelle* ou raisons conventionnelles). Des éclaircissements sont nécessaires sur ces notions.

Enfin, la notion d'engagement permet de spécifier des jeux à plusieurs niveaux de granularité (e.g., enchaînements précis d'actes ou prérogatives générales). Elle permet de s'adapter à une grande variété de motifs. En outre, l'adoption d'une sémantique publique permet de spécifier les jeux de dialogue de manière déclarative accroissant leur clarté.

L'approche des jeux de dialogue par les engagements sociaux de Maudet admet des limites. Premièrement, la notion d'*engagement social* nécessite d'être approfondie [Pasquier 2005]. En particulier, il est nécessaire de spécifier clairement le *cycle de vie* des engagements (création, satisfaction, violation, etc.) [Chaib-Draa 2006]. Deuxièmement, la structure de jeux proposée par Maudet est limitée. Les conditions d'échec ne sont pas pleinement exploitées (et se résument à la production d'un coup dialogique *vide*). Ensuite, Hulstijn précise que la cohérence au niveau interactionnel est aussi bien une question de *fonctions* des actes de dialogue que de *contraintes sémantiques* sur le contenu des actes (cf. section 3.2.2). Dans la version engagement, les règles des jeux permettent de spécifier des coups attendus (via un engagement en action) ou des enchaînements de coups (via un engagement conditionnel). En l'état, elles ne permettent pas de prendre en compte des *contraintes sémantiques*. Troisièmement, la contextualisation des jeux

est limitée et ne permet pas de proposer une entrée dans un jeu sans en spécifier son but (e.g., *« Je peux te poser une question ? »*) [Maudet 2001]. Enfin, cette formalisation perçoit le tableau de conversation comme étant strictement partagé entre les interlocuteurs (cf. modèle 4, section 2.2.1). Or, la gestion de mécanismes de communication comme le *grounding* peut nécessiter la relativisation de cette partie publique du point de vue de l'agent [Larsson 2003].

3.3.3 Vers des modèles mixtes fondés sur des jeux de dialogue

Nous avons parcouru les approches des jeux de dialogue pour la modélisation du dialogue Homme-Homme et Homme-Machine qui nous paraissent les plus pertinentes.

Les motifs d'interaction observables lors de l'analyse des corpus semblent clairement être des manifestations de conventions, nécessaires à la coordination de l'activité dialogique vue comme une activité conjointe. Nous avons vu que les motifs d'interaction étaient les parties émergées de l'iceberg qui ne rendent pas clairement compte de la contextualisation des jeux, de l'agencement inter-jeu et de la séparation entre les deux axes du dialogue (tâche et interaction). En conséquence, l'analyse des motifs d'interaction peut conduire à plusieurs formalisations de granularités différentes.

Cette étude nous a permis de nous rendre compte que le véritable intérêt des jeux était de cristalliser les conventions dialogiques dans une structure pouvant être intégrée à un processus délibératif. Nous avons vu que les jeux étaient plus que des recettes pour l'action conjointe directement intégrable dans des plans partagés. Une vision plus juste semble de les voir comme des guides pour l'activité dialogique qu'il est possible de suivre, ou de transgresser. En d'autres termes, les jeux de dialogue sont essentiellement *normatifs*.

Pour finir, la formalisation de Maudet semble la plus complète et offre le plus de souplesse. Elle nécessite encore du travail aussi bien sur les relations de cohérence sémantique, le jeu de contextualisation et l'intégration des jeux de dialogue dans un agent délibératif implémenté. Comme [Hulstijn 2000b, Maudet 2001], nous proposons d'aller vers une architecture hybride réactive/délibérative où une théorie de l'action conjointe servira de sémantique aux motifs d'interaction décrits comme des jeux de dialogue. Nous rejoignons le point de vue de Maudet et considérons les jeux de dialogue comme des structures d'engagements [Maudet 2001].

Deuxième partie

Analyse des interactions humaines

Chapitre 4

Corpus Cogni-CISMeF

Sommaire

4.1 Constitution du corpus 85
4.1.1 Objectifs 86
4.1.2 Recueil du corpus 87
4.1.3 Transcription 87
4.2 Processus de recherche d'information 88
4.2.1 Modèle standard 88
4.2.2 Un processus opportuniste 88
4.2.3 Un processus stratégique 89
4.3 Analyse des entretiens du corpus 90
4.3.1 Situation dialogique 90
4.3.2 Structure globale des entretiens 91
4.3.3 Stratégies et tactiques 94
4.3.4 Rôles des participants 99
4.4 Synthèse 103

Dans ce chapitre, nous illustrons les phases de *récolte de corpus* et de *transcription* de notre méthodologie (cf. figure 4.1). Nous présentons le corpus que nous utilisons pour illustrer notre démarche méthodologique. Nous abordons sa constitution (cf. section 4.1). Puis nous nous intéressons à la tâche de recherche d'information (RI) (cf. section 4.2). Enfin, nous présentons les entretiens de notre corpus en étudiant la tâche de RI collaborative sous-jacente dans la perspective à long terme de réalisation d'un agent assistant d'initiative mixte (cf. section 4.3).

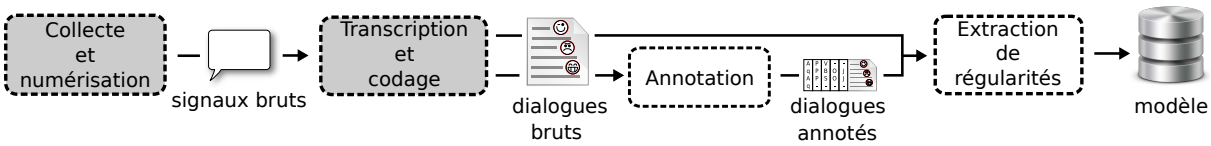

Figure 4.1: Récolte et transcription de corpus dans notre méthodologie. Le fond coloré indique les étapes traitées dans ce chapitre.

4.1 Constitution du corpus

Notre méthodologie encourage l'usage de corpus de *dialogues finalisés* (i.e., orientés vers la réalisation d'une tâche précise). Il est indispensable de préciser que ces dialogues possèdent des

caractéristiques simplifiant leur étude au regard de « conversations libres » (même si certains chercheurs considèrent que tout dialogue possède une tâche sous-jacente, e.g., [Clark 1996]). Lorsque la tâche sous-jacente est bien définie (e.g., plus précise qu'une tâche comme « passer le temps »), les dialogues sont moins complexes qu'une communication humaine en général. Ils fournissent un cadre contraint plus aisément formalisable. Certains travaux ont ainsi montré empiriquement que le vocabulaire employé par les interlocuteurs dans un dialogue est très dépendant de sa tâche sous-jacente et qu'il y a peu de recoupement entre les différentes tâches [Allwood 2001]. En outre, la tâche influence la structure du dialogue [Grosz 1986]. Nous avons vu que c'était cette caractéristique qui était exploitée par les approches intentionnelles (cf. section 2.1). Le lien entre le dialogue et la tâche permet de caractériser plus facilement les rôles des participants et leurs buts que lors d'une conversation libre. Enfin, la variété des interactions est également moindre dans des dialogues finalisés.

Cette section détaille les objectifs à long terme qui ont motivé la constitution du corpus Cogni-CISMeF de dialogues finalisés (section 4.1.1). Elle décrit le recueil du corpus (section 4.1.2) et sa transcription (section 4.1.3).

4.1.1 Objectifs

Notre objectif à long terme est la construction d'un *agent assistant d'initiative mixte* [Tecuci 2007] pour la recherche d'information (RI) avec le système CISMeF [Darmoni 2000].

CISMeF [1] est un moteur de recherche spécialisé dans l'indexation de ressources médicales en langue française. Il propose une interface graphique et un langage de requête permettant de décomposer une requête en éléments issus d'un lexique contrôlé (le MeSH, « Medical Subject Headings »). La terminologie CISMeF comporte des *mots-clés*, des *qualificatifs* (symptômes, traitements, etc.), des *méta-termes* (spécialités médicales) et des *types de ressources* (base de données, périodique, images, ...). Le système s'est étoffé en offrant la possibilité d'effectuer des requêtes étendues. Or, l'utilisateur est souvent peu enclin à l'utilisation de requêtes complexes pourtant plus efficaces.

CISMeF présente des difficultés d'utilisation qui justifient la conception d'un agent assistant [Loisel 2008]. En synthèse, CISMeF propose des fonctionnalités avancées qui ne sont pas à la portée de l'utilisateur lambda sans un apprentissage important. Ce système propose un langage de requête complet (connecteurs booléens, recherche en mots-clés majeurs, recherche limitée, etc.) mais difficile d'utilisation. En outre, l'utilisateur doit connaître la terminologie CISMeF, i.e. le vocabulaire contrôlé utilisé pour indexer les ressources dans le moteur de recherche. En d'autres mots, l'utilisateur doit connaître les rouages du système pour pouvoir l'utiliser à son potentiel maximum. Par exemple, une demande comme « médicaments pour le paludisme » peut être traduite dans une requête constituée du *mot-clé* « paludisme », du *qualificatif* « thérapeutique » et du *type de ressources* « patient ». Un ensemble de fonctionnalités avancées est accessible à l'utilisateur pour mener à bien sa recherche, mais est rarement utilisé en pratique. CISMeF est victime de la *crise de l'utilisabilité* [Rich 2007] : au-delà des fonctionnalités basiques, les utilisateurs sont perdus. Cette observation est appuyée par les expérimentations de [Loisel 2008] qui montrent que même des étudiants de médecine en 2$^{\text{ème}}$ année formés à CISMeF ne sont pas capables d'utiliser les fonctionnalités avancées du moteur de recherche.

Le prototype Cogni-CISMeF basé sur GoDiS [Larsson 2002a] a constitué une première tentative de développer un agent assistant pour une tâche de RI sur CISMeF [Loisel 2008].

1. « Catalogue et Index des Sites Médicaux de langue Française », disponible à l'URL www.cismef.org

Cogni-CISMeF est sujet aux limites de GoDiS, qui est difficile à étendre pour une tâche complexe et pour l'ajout de nouveaux coups dialogiques. Cogni-CISMeF a été victime du passage à l'échelle et n'a pas pu dépasser le stade de prototype [Loisel 2012]. Cette expérience a en partie motivé la mise en place de la méthodologie sur laquelle se base cette thèse.

4.1.2 Recueil du corpus

Nous utilisons dans cette thèse le corpus constitué lors du projet Cogni-CISMeF [2] [Loisel 2008]. Il est constitué de dialogues d'assistance sur une tâche de RI médicale entre un expert CISMeF et un utilisateur. Le recueil de corpus a été effectué par raffinage successif sur plusieurs expérimentations [Loisel 2008]. Lors de ces expérimentations, un *expert* du système CISMeF se retrouve en tête à tête avec un *demandeur* qui apporte une formulation de son besoin d'information d'ordre médical. Celui qui occupe la fonction d'expert dispose d'un accès à CISMeF et est chargé de mener la recherche en coopération avec le demandeur. L'expert doit dans le même temps verbaliser tout ce qu'il est en train de faire. Les dialogues sont enregistrés afin d'être retranscrits et les logs des requêtes sont récupérés par le système. L'entretien se clôt lorsque la réponse satisfait le demandeur, ou qu'il semble qu'aucune réponse ne puisse être trouvée. Deux membres du projet Cogni-CISMeF ont joué les rôles d'expert. Pour éviter toute connivence entre personnes, les membres du projet les moins familiers avec le laboratoire ont réalisé cette expérimentation en jouant le rôle d'expert. Les demandeurs sont des membres du laboratoire LITIS (secrétaires, administrateur réseau, doctorants, enseignants-chercheurs). Les utilisateurs sont représentatifs de l'audience visée par le futur système puisqu'ils ne sont pas spécialistes du domaine médical et qu'ils souhaitent obtenir des réponses à des interrogations d'ordre médical. L'intervention des deux experts a permis de contraster les démarches.

Le corpus est constitué des retranscriptions des 21 dialogues enregistrés entre les 2 experts et les 21 volontaires. Il contient environ 37 000 mots. Il est constitué de deux *sous-corpus* caractérisés par la personne qui a joué le rôle d'expert : le corpus AL (12 entretiens) et le corpus VD (9 entretiens). Nous référons aux dialogues du corpus en précisant l'identifiant du corpus suivi du numéro de l'entretien (par exemple, VD05 réfère à l'entretien n° 5 du corpus VD).

4.1.3 Transcription

Les entretiens ont été manuellement retranscrits par Irina Lenka, étudiante en sociolinguistique chevronnée sur les tâches de retranscription. Des conventions de transcription ont été définies (cf. tableau 4.1). Ces dernières permettent de marquer certaines intonations (affirmative, interrogative, exclamative) et des phénomènes courants dans le dialogue oral (faux départ, pause, insistance). La prosodie détaillée (intonation, rythme, accentuations, etc.) n'a pas été annotée car elle nécessite un effort long et important ainsi que des compétences spécifiques. Une limite notable de ce travail réside dans le fait qu'il se limite à la transcription de la partie verbale d'une interaction en co-présence. Il est important de préciser que l'interaction via d'autres modalités n'a pas été annotée (mouvements de la tête, expressions faciales, ...). Nous revenons sur ce point dans les limites de notre processus d'annotation.

2. Projet financé par le Programme Interdisciplinaire TCAN du CNRS 2004-2007

Symbole	Description	Exemple
?	Intonation interrogative	« alors ensuite c'est des crises / crises d'angoisse / dans le sens banal ? »
!	Intonation exclamative	« Ah ! »
-	Faux départ	« donc là on est un syno- / l'épilepsie ça doit être représenté par synonyme »
/	Pause	« Les symptômes / qu'est-ce que c'est exactement le symptômes »
:	Insistance	« Euh : : »
xxx	Mot ou passage non reconnu	« efficacité de la xxx xxx et des interventions concernant le style de vie »
(...)	Phénomènes non verbaux	(rires)
<...>	Transcription incertaine	« ah on a trois choses / enregistrement <pligraphique> du sommeil »
"..."	Lecture à l'écran	« alors le cinq on a "le diagnostic est pris en charge du trouble anxieux généralisé" »

Tableau 4.1: Conventions de transcription du corpus Cogni-CISMeF

4.2 Processus de recherche d'information

Notre corpus de dialogues est centré sur la réalisation d'une tâche de RI réalisée collaborativement entre un expert du moteur de recherche et un demandeur. Dans cette section, nous présentons quelques modèles du processus de RI réalisé par un individu isolé. Nous insistons sur les caractéristiques essentielles de ce processus. Le lecteur intéressé par le domaine de la RI peut trouver de plus amples informations dans [Hearst 2009].

4.2.1 Modèle standard

Le processus de RI est généralement perçu comme une instance particulière d'un processus de résolution de problème [Marchionini 1989]. Ce processus implique un *chercheur* ayant un *besoin d'information identifié*. Le problème est alors de combler ce manque d'information. Une fois le besoin d'information spécifié, le chercheur établit un *plan de recherche* qu'il exécute au travers de la recherche en elle-même. Il évalue les résultats trouvés pour éventuellement réitérer le processus entier. La RI est vue comme un *processus itératif* décomposable en une succession d'étapes [Sutcliffe 1998,Broder 2002,Marchionini 2008] : (i) identification d'un besoin d'information, (ii) spécification de la requête (formulation du besoin d'information, expression de ce besoin dans le moteur de recherche, etc.), (iii) exécution de la requête, (iv) évaluation des ressources obtenues, (v) reformulation de la requête si nécessaire et répétition du cycle jusqu'à obtention de résultats satisfaisants ou abandon de la recherche.

4.2.2 Un processus opportuniste

Le modèle standard est limité sur deux aspects. D'un côté, le processus est vu comme impliquant un *besoin d'information statique*. De l'autre, l'individu est vu comme raffinant successivement sa requête jusqu'à tomber sur un ensemble de documents répondant à son besoin d'information initial. Certaines études ont démontré justement que le besoin d'information dans

ce processus n'est pas statique et que l'objectif n'est pas de déterminer une unique requête retournant un ensemble de documents répondant au besoin d'information [Bates 1989, O'Day 1993]. [Bates 1989] propose le modèle de « berry-picking » [3] qui insiste sur deux points. Le premier est que le besoin d'information du chercheur évolue en fonction des ressources trouvées lors de la recherche. Une information rencontrée à un moment peut conduire la recherche dans une direction nouvelle et *imprévue*. Le second est que le besoin d'information n'est pas satisfait par un *unique* ensemble de documents obtenu à la fin de la recherche, mais par une sélection de ressources récoltées *tout au long* du processus. En synthèse, le processus de RI est un processus fondamentalement *opportuniste* dont le *cheminement* est prégnant face au résultat final.

4.2.3 Un processus stratégique

Coup, tactique, stratagème et stratégie

Le comportement d'un individu dans un processus de recherche peut être analysé comme un *processus stratégique* [Bates 1979, Bates 1990]. Ce point de vue a pour but de caractériser aussi bien le comportement d'expert de la RI mettant en œuvre des stratégies planifiées et délibérées, que le comportement plus réactif d'un chercheur lambda. Il s'agit de caractériser la base sur laquelle sont choisies les actions pour passer à l'étape suivante de la recherche. [Bates 1990] décompose l'activité de recherche en *niveaux* en s'appuyant sur son précédent travail sur les *tactiques* de recherche [Bates 1979]. Le premier niveau est celui des *coups*. Un coup est l'unité minimale d'analyse. Il s'agit d'une action qui prend part à l'activité de recherche (e.g., « saisir la formulation de la recherche », « saisir un opérateur booléen ET », ...). Le deuxième niveau est celui des *tactiques*. Il s'agit d'un ou plusieurs coups ayant pour but de poursuivre la recherche en prenant en compte l'état courant (e.g., changer un terme en un plus général comme « cholestérol HDL » par « cholestérol », produire une *première requête* incluant la plupart des termes de la formulation du besoin d'information, ou au contraire produire une première requête en minimisant sa taille). Le troisième niveau est celui des *stratagèmes*. Il s'agit d'un ensemble de coups et/ou tactiques conçu pour exploiter un *domaine particulier* de la RI par une *procédure systématique*. Par exemple, le domaine peut être un index de citation et le stratagème peut consister à partir d'une citation pour déterminer l'ensemble des travaux l'ayant citée. Un autre stratagème peut consister à chercher tous les travaux d'un auteur traitant d'un sujet particulier. Enfin, une *stratégie* est un *plan global* pour la recherche entière pouvant contenir coups, tactiques et/ou stratagèmes. Un plan global est difficile à spécifier de par la nature opportuniste du processus de recherche [Bates 1990]. Il faut noter que ces activités représentent des *niveaux différents* non réductibles les uns aux autres (une stratégie n'est pas une simple succession de stratagèmes, eux-mêmes succession de tactiques). Chacun de ces niveaux sont conceptuellement différents. En outre, [Bates 1990] précise que les *tactiques* et les *stratagèmes* ne sont pas les uniques niveaux envisageables au dessus des coups.

Stratégies incrémentales

Deux principaux types de *stratégies incrémentales* ont été observés lors du processus de RI [Hearst 2009]. Ils peuvent être envisagés comme des *stratégies partielles* complétées de manière opportuniste pendant le processus. La première stratégie est nommée « orientee-

3. « Berry-picking » peut être traduit par « cueillette de baies »

ring »[4] [O'Day 1993]. À l'image d'une course d'orientation, cette stratégie vise à partir d'une requête générale pour la raffiner progressivement suite aux indices accumulés. Elle est constituée de trois grandes étapes : (i) spécifier une courte requête générale, (ii) évaluer et inspecter les ressources obtenues, et (iii) modifier la requête en conséquence pour itérer le processus. Cette stratégie est généralement opposée à la stratégie de *téléportation* [Teevan 2004]. Contrairement à l'« orienteering », cette dernière vise à spécifier une requête le plus précisément possible afin de se « téléporter » directement aux ressources répondant au besoin d'information.

4.3 Analyse des entretiens du corpus

Dans cette section, nous analysons la tâche de RI qui motive le dialogue. Comme précédemment évoqué, la tâche sous-jacente contribue à restreindre la complexité des conversations. En outre, elle influence la structure globale du dialogue [Grosz 1986] (voir, e.g., la structure globale d'un dialogue de réservation de billet d'avion [Bilange 1991a] ou de transaction [Hulstijn 2000b]). Nous étudions l'influence de la tâche de RI sur les entretiens du corpus et en particulier sur la structure globale du dialogue.

À notre connaissance, peu de travaux existent sur la tâche de RI en situation de co-présence entre un utilisateur et un expert du moteur de recherche en dehors des travaux de Loisel [Loisel 2008]. Les travaux les plus proches se situent au niveau de la définition de la RI collaborative qui s'inspire du web social [Golovchinsky 2008, Shah 2008]. Dans la suite de cette section, nous présentons une étude originale du processus de RI en situation de co-présence.

Elle se base sur un rapport technique qui décortique les dialogues des entretiens en détaillant les différentes phases et l'évolution de la requête [Dubuisson Duplessis 2013]. Dans la section 4.3.1, nous précisons la situation dialogique entre l'expert et le demandeur. Nous présentons ensuite la structure globale du dialogue que nous avons observée (cf. section 4.3.2). Puis, nous synthétisons les stratégies et les tactiques employées par les interlocuteurs (cf. section 4.3.3). Enfin, nous abordons les rôles des participants dans cette interaction (cf. section 4.3.4).

4.3.1 Situation dialogique

La situation de collaboration entre l'*expert* et le *demandeur* est représentée en figure 4.2. Les deux partenaires font face au mode avancé du moteur de recherche CISMeF. Ils dialoguent pour faire avancer la tâche de RI. L'expert manipule l'ordinateur et l'enquêté l'observe.

[Golovchinsky 2008] propose de caractériser la situation de collaboration entre des partenaires sur une tâche de RI selon trois dimensions pertinentes : intention, simultanéité et situation. L'*intention* caractérise la façon dont le besoin d'information est établi entre les partenaires. Elle peut être *explicite* (le besoin d'information est explicitement établi par les partenaires) ou *implicite* (le besoin d'information est inféré par un des collaborateurs, e.g., via les actions de l'utilisateur). La collaboration peut être *synchrone* (les collaborateurs s'influencent en temps réel) ou *asynchrone*. Enfin, les collaborateurs peuvent être en *situation* de co-présence ou non. Dans notre cas, la situation de collaboration sur la tâche de RI est *explicite*, *synchrone* et en *situation de co-présence*.

4. « Orienteering » peut être traduit par « course d'orientation »

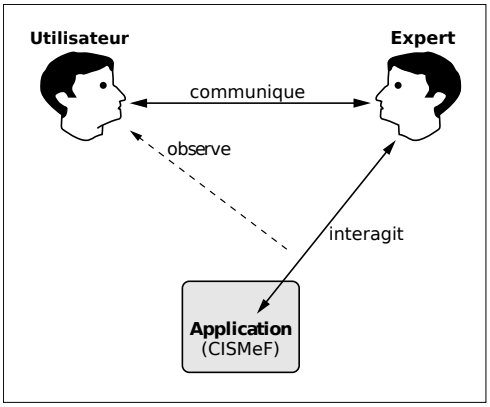

FIGURE 4.2: Situation dialogique entre l'expert, le demandeur et CISMEF dans l'expérimentation (adaptée de [Rich 2001])

4.3.2 Structure globale des entretiens

Les dialogues de notre corpus montrent une même structure globale. Le dialogue se découpe en phases qui jouent chacune un rôle dans l'avancement de la tâche. Nous présentons cette décomposition d'une manière générale puis nous décrivons plus précisément chacune des étapes. L'enchaînement des phases est représenté sur la figure 4.3. Les dialogues commencent généralement par une phase d'ouverture qui consiste en des salutations.

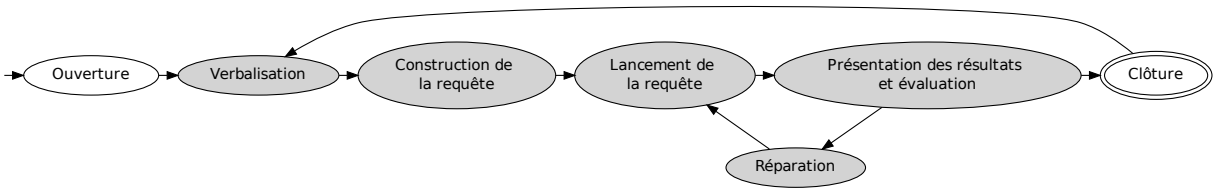

FIGURE 4.3: Structure des dialogues de recherche d'information collaborative observée. Les phases colorées représentent une *séquence* de recherche d'information.

Le dialogue de recherche commence réellement à la phase de verbalisation du besoin d'information qui est toujours présente. Cette dernière rend explicite le besoin d'information du demandeur entre les interlocuteurs. Ce besoin n'est pas figé et peut être raffiné par la suite (e.g., dans les phases d'évaluation et de réparation). La verbalisation est suivie d'une phase de construction de la première requête qui va dépendre du besoin explicité. Une fois cette première requête construite, elle est lancée. Il s'ensuit une phase de présentation des ressources trouvées et d'évaluation des résultats. Suite à cette phase d'évaluation, deux possibilités se présentent : (i) les résultats sont complètement satisfaisants, ou bien (ii) les résultats sont partiellement satisfaisants ou non satisfaisants. Dans le premier cas, le dialogue de recherche se termine par une phase de clôture. Cette phase de clôture peut déboucher sur l'arrêt de la RI ou sur l'exécution d'une recherche sur un nouveau sujet. Par exemple, l'entretien AL09 fait apparaître deux *séquences* de recherche : une sur des problèmes d'articulation, l'autre sur des problèmes d'eczéma athopique. Dans le deuxième cas, l'expert et le demandeur entrent dans une phase de réparation de la requête qui va dépendre des résultats obtenus. La réparation conduit au lancement d'une nouvelle requête et à une nouvelle évaluation des résultats. Il est important de noter la présence du cycle de modification de la requête constitué des phases de lancement de la requête,

de présentation et d'évaluation des résultats, et de réparation de la requête. Ceci est le cœur du dialogue de RI collaborative et met en avant la nature *itérative* du processus. Tous les dialogues du corpus présentent au moins une fois la réalisation de ce cycle de réparation. La phase de clôture est atteinte après la phase de présentation et d'évaluation des résultats soit parce que les résultats obtenus sont satisfaisants, soit parce que la recherche est abandonnée car les interlocuteurs jugent qu'aucun résultat ne pourra être obtenu. Cependant, nous ne pouvons pas exclure le fait que le dialogue puisse se terminer à n'importe quel moment (même si cela n'a pas été observé).

Nous définissons une *séquence de recherche* comme la succession des étapes de verbalisation, de construction de la requête et du cycle de modification de la requête. Les dialogues de RI peuvent inclure plusieurs *séquences* situées entre les phases d'ouverture et de clôture.

Phases d'ouverture et de clôture

La phase d'ouverture consiste en de simples salutations. Cette phase est facultative. Précisons que les entretiens du corpus Cogni-CISMeF commence par une phase de demande d'informations sur le demandeur spécifique à l'expérimentation (nom, prénom, âge, etc.).

La phase de clôture fait apparaître des propositions de lancement d'une nouvelle séquence (e.g., *« est-ce que vous avez une question sur un autre domaine ? »* (AL09)) ou d'arrêts (e.g., *« donc je crois qu'on va en rester là non ? – d'accord ça marche »* (AL04)).

Phase de verbalisation

La phase de verbalisation permet d'établir le *sujet de la recherche* entre les interlocuteurs (i.e., les thématiques de la recherche). La suite de la tâche de construction/réparation de la requête se base sur ce besoin d'information identifié et établi. Nous avons relevé trois principales formes de verbalisation à la suite de la sollicitation de l'expert. La première forme est constituée d'une étape de formulation du demandeur suivie d'un éventuel feedback de l'expert. Par exemple, la phase de verbalisation dans l'entretien AL01 est la suivante :

— Formulation (demandeur) : *« donc j'ai une amie qui est épileptique et je voudrais savoir quel est le risque pour la grossesse »*

— Évaluation (expert) : *« ok, donc la question est suffisamment précise »*

La deuxième forme consiste en une formulation du demandeur reprise et reformulée par l'expert. Cette reformulation peut être suivie de précisions spontanées du demandeur. Par exemple, la phase de verbalisation dans l'entretien AL03 est la suivante :

— Formulation (demandeur) : *« donc il a toujours la périarthrite »*.

— Reformulation (expert) : *« donc vous voulez des informations sur la périarthrite »*.

— Précisions spontanées (demandeur) : *« les soins et puis pour savoir l'évolution »*.

Enfin, la troisième forme consiste en une formulation du demandeur suivie de demandes de précisions de l'expert. Celle-ci est illustrée dans l'entretien VD02 :

— Formulation (demandeur) : *« enfin je voudrais parler de leucémie en fait »*

— Demande de précisions (expert) : *« alors des choses un petit peu plus précises sur la leucémie »*

— Précisions (demandeur) : *« les symptômes / qu'est-ce que c'est exactement ? les symptômes »*

Le tableau 4.2 présente les proportions observées de ces formes de verbalisation dans le corpus. On observe une majorité de formulation « simple » (57%) suivie par la formulation conduisant à des demandes de précisions (24%). Enfin, le dernier type (formulation/reformulation) est observé dans 19% des cas.

Forme	VD	AL	Total
Formulation[1]	4	8	12 (57%)
Formulation/Reformulation[1]	1	3	4 (19%)
Formulation/Demande de précisions/Précisions	4	1	5 (24%)

Tableau 4.2: Occurrences des différents types de verbalisation dans le corpus

Phase de construction de la requête

De manière générale, la problématique de cette phase est de trouver les termes de la terminologie CISMeF adaptés à la verbalisation de l'enquêté. Dans le meilleur des cas, il existe des termes dans la terminologie identiques aux termes de la verbalisation de l'enquêté (e.g., « leucémie », VD02). Dans le pire des cas, le terme est absent de la terminologie. L'expert et l'enquêté doivent donc collaborer pour trouver des termes de la terminologie approchant la verbalisation de l'utilisateur. Nous revenons sur cette problématique dans la section 4.3.3.

Phase de lancement de la requête

La phase de lancement de la requête consiste à exécuter la requête élaborée par les partenaires. Cette phase est souvent implicite. Cependant, il arrive qu'elle soit explicite dans certaines situations, par exemple : *« ok / alors on lance la recherche avec uniquement néphrite [...] »* (VD08).

Phase d'évaluation

La phase de présentation et d'évaluation des résultats est l'une des plus importantes du dialogue de recherche. C'est dans cette phase que l'enquêté et l'expert jugent la qualité des ressources obtenues. De manière générale, l'expert présente les résultats au demandeur. Il le sollicite pour juger l'adéquation des ressources obtenues par rapport à ses attentes (e.g., *« vous allez me dire si ça vous convient ou pas et pourquoi »* (VD02), *« Est-ce que le document n° x vous intéresse ? »* (VD03)). Les résultats obtenus peuvent être : (i) satisfaisant, (ii) partiellement satisfaisant, i.e. pas adapté au profil de l'utilisateur (ressources pour médecin) ou n'abordant pas l'ensemble des thèmes de la requête, ou (iii) pas satisfaisant, i.e. document hors sujet ou déjà obtenu par une précédente requête. L'objet de cette phase est également d'envisager la poursuite de la recherche ou son arrêt.

1. Éventuellement suivie d'une phase de précisions spontanées du demandeur

Phase de réparation

Enfin, les phases de réparation de la requête montrent que l'expert et le demandeur mettent en place un ensemble de tactiques permettant de modifier la requête pour récolter des ressources répondant au besoin d'information. Nous les étudions plus en détails dans la section 4.3.3. Dans cette phase, les partenaires proposent des coups ou des tactiques afin d'établir un plan de modification de la requête. Les dialogues 4.1 et 4.2 sont des exemples de réparation de requête.

Dans le dialogue 4.1, l'expert propose une *tactique* de reformulation de la requête (énoncé **A**1). Il suggère au demandeur d'ajouter « néphrite » à la nouvelle requête. Ce dernier l'accepte (énoncé **B**2). Ensuite, la proposition de l'expert d'enlever le terme « membres inférieurs » (énoncé **A**3) est acceptée par le demandeur (énoncé **B**4). Finalement, l'expert suggère de ne pas enlever « membres inférieurs » mais de le conserver en mot-clé (énoncé **A**5). Cette suggestion n'est pas considérée par l'enquêté qui demande l'exécution de la requête avec « néphrite », i.e. l'exécution du plan de modification précédemment établi (énoncé **B**6). L'expert exécute alors ce plan de modification (énoncé **A**7). En synthèse, les interlocuteurs s'engagent dans une *tactique de reformulation*. Ils co-construisent un plan constitué de *coups* modifiant la requête. Le plan élaboré est finalement exécuté.

A1 : les pistes qu'on a prises ne fonctionnent pas donc on va essayer de partir sur autre chose / on parle d'une néphrite tout à l'heure / vous voulez qu'on essaie de ce côté là
B2 : on peut essayer
A3 : donc j'enlève membres inférieurs
B4 : ouais
A5 : ou je le rajoute en mot clé
B6 : bon allons y sur néphrite voir un peu ce qu'il y a
A7 : en mot clé cismef je tape néphrite après avoir enlevé membres inférieures

Dialogue 4.1 – Exemple de dialogue de réparation de requête (VD08). A est l'expert, B est le demandeur.

Dans le dialogue 4.2, l'expert propose une *tactique* d'élargissement de la recherche (énoncé **A**1). Comme dans le précédent dialogue, les partenaires élaborent conjointement un plan de modification de la requête. Ici, c'est principalement le demandeur qui propose les coups à effectuer (énoncés **B**4 et **B**6) et qui sont validés par l'expert (énoncés **A**5 et **A**7). Dans la suite de l'élaboration du plan, l'expert propose de rajouter le qualificatif « diagnostic » (énoncé **A**8) bien que sa suppression ait été précédemment validée (énoncé **A**7). Le demandeur accepte et propose l'exécution du plan (énoncé **B**9). L'exécution du plan est acceptée et effectuée par l'expert (énoncé **A**10).

4.3.3 Stratégies et tactiques

Nous présentons maintenant la stratégie menée par l'expert observée dans le corpus. Nous appelons *tactique initiale* une tactique qui apparaît dans la phase de construction de la requête et *tactique de réparation* une tactique qui apparaît dans la phase de réparation. De manière générale, nous avons observé que l'expert met en place un processus de recherche incrémental.

A1 : [...] bah peut-être qu'on peut essayer d'élargir la recherche dans ce cas là si on regarde un petit peu les mots qu'on a mis /
B2 : on a quand même pas mis grand chose
A3 : bah non alors
B4 : pourquoi enlever / on peut enlever analyse
A5 : alors enlevons analyse
B6 : et diagnostic
A7 : oui

[...]

A8 : [...] j'aurais presque envie de mettre diagnostic quand même parce que / parce que on va voir ce que ça donne
B9 : oui normalement c'est un diagnostic / ok / essayons comme ça
A10 : on va essayer comme ça sinon on enlèvera encore des choses pour arriver à avoir des / donc je relance la recherche avec l'accès thématique cancéro le mot clé cismef colon et puis le qualificatif diagnostic sans précision du type de ressource qu'on recherche

Dialogue 4.2 – Exemple de dialogue de réparation de requête (VD06). A est l'expert, B est le demandeur.

Stratégie globale

Une *stratégie globale* débute par une *tactique initiale* suivie d'une série de *tactiques de réparation* de la requête permettant de collecter des documents. Les tactiques sont élaborées en collaboration avec le demandeur. Nous avons observé deux types de stratégie globale correspondant aux deux experts. La première consiste à partir d'une requête précise pour ensuite la généraliser au fur et à mesure (corpus VD). C'est une stratégie de *téléportation*. La seconde consiste à partir d'une requête vague pour la préciser successivement ensuite (corpus AL). C'est une stratégie d'*orienteering*. La stratégie globale influence les tactiques de réparation utilisées comme nous le détaillons en fin de section.

Tactiques initiales

Tactique de téléportation L'expert du corpus VD mène une tactique initiale principalement directive et stéréotypée. Elle consiste à remplir un maximum de champs du formulaire de recherche avancée en collaboration avec l'enquêté. Elle se décompose de la manière suivante :

1. Préciser la requête en utilisant l'*accès thématique* de CISMeF i.e. préciser un thème général (e.g. « Anatomie », « Biologie », « cardiologie », etc.).
2. Préciser le ou les *mots-clés* (généralement extrait de la phase de verbalisation).
3. Préciser des *qualificatifs* associés aux mots-clés (e.g. « épidémiologie » associé au mot-clé « paludisme »).
4. Préciser le *type de ressources* (e.g. « article de périodique », « bande dessinée »).

Cette tactique est une tentative de téléportation aux ressources répondant aux besoins d'information. Le dialogue 4.3 présente un exemple complet d'élaboration d'une première requête entre l'expert et le demandeur. Cette phase est dirigée par l'expert. Les partenaires déterminent

l'accès thématique (énoncés $\mathbf{A}_1$ à $\mathbf{A}_5$). Puis, ils fixent le mot-clé (énoncés $\mathbf{A}_5$ à $\mathbf{A}_7$). Enfin, ils précisent des qualificatifs (énoncés $\mathbf{A}_7$ à $\mathbf{A}_{17}$) et un type de ressources (énoncés $\mathbf{A}_{17}$ à $\mathbf{A}_{19}$).

Tactique d'*orienteering* Contrairement au corpus VD, la tactique initiale de l'expert du corpus AL ne semble pas clairement figée. On observe néanmoins un certain nombre de caractéristiques. L'expert ne cherche pas à créer une requête précise mais au contraire, une plutôt générale. Dans ce but, il semble avoir une préférence pour préciser dans un premier temps uniquement une spécialité médicale dans laquelle s'insère la recherche. Cette tactique est une mise en œuvre de l'orienteering. Le dialogue 4.4 présente un exemple de spécification d'une requête générale entre l'expert et le demandeur.

Tactiques de réparation

Une fois la première requête lancée, les interlocuteurs vont évaluer les résultats. Nous avons pu observer dans le corpus les tactiques mises en place par les interlocuteurs pour trouver des ressources adéquates. L'élément central est la phase d'évaluation qui va conduire à choisir entre trois types de tactiques : la *précision*, la *généralisation* et la *reformulation* que nous mettons en relation avec les tactiques initiales de téléportation et d'« orienteering » en fin de section.

Tactiques de précision L'idée des tactiques de précision est d'augmenter l'adéquation entre les ressources obtenues et les attentes de l'enquêté. Généralement, la précision d'une requête conduit à obtenir moins de documents. Ces tactiques sont utilisées quand (i) des résultats sont obtenus (quelque soit le nombre), et (ii) les ressources observées sont partiellement satisfaisantes [5]. Ce type de tactiques dépend du critère qui rend les ressources partiellement satisfaisantes. Nous avons observé deux critères qui motivent les réparations de la requête. Le premier critère dépend de la *spécificité* des résultats par rapport à ce qu'attend le demandeur. Ce point est clairement connecté à son profil. Les résultats peuvent être trop spécialisés (i.e., les ressources abordent des notions trop spécifiques). Au contraire, les résultats peuvent être insuffisamment précis (i.e., les ressources abordent des notions trop générales). Le second critère dépend des thèmes de la recherche. Si la requête est *incomplète* (i.e. elle ne recouvre pas tous les thèmes de la recherche) alors les interlocuteurs vont essayer de la préciser. Les tactiques de précisions visent généralement à ajouter un élément à la requête (ajout d'un mot-clé, d'un qualificatif, d'un type de ressources, d'une spécialité médicale, etc.). C'est par exemple le cas de l'entretien AL02 où l'expert propose de préciser avec le type de ressources « patient » : *« on va voir on lance la recherche/ ce qu'on obtient/ voilà ah / on a beaucoup plus de documents / cinquante-et-une ressources / donc on a peut-être même trop / [...] on peut lancer sur les documents qui sont spécifiques pour les patients »*. Néanmoins, d'autres précisions peuvent exister comme substituer un terme de la requête par un plus précis (e.g., les partenaires substituent « cholestérol » par « cholestérol HDL » lors de l'entretien AL12).

Tactiques de généralisation L'idée des tactiques de généralisation est de tenter d'obtenir une requête qui retourne des résultats pour ensuite utiliser des tactiques de précision/reformulation. Ces tactiques sont utilisées quand (i) aucun résultat n'est obtenu, et (ii) la requête est composée (i.e., il est possible de la simplifier en lui soustrayant des éléments). Dans le cas où il

5. Intuitivement, la précision d'une requête si peu de ressources sont obtenues semble inutile. Cependant quelques cas ont été recensés dans notre corpus.

A1 : donc on va y aller / alors accès thématique on va dire que c'est grosso modo l'ensemble des *spécialités médicales* [...] est-ce que vous voyez des choses dans lesquelles vous auriez envie de classer votre demande sur la leucémie / je fais défiler donc
B2 : le menu déroulant
A3 : le menu déroulant [...] si vous ne trouvez rien c'est pas grave on mettra rien / peut-être que
B4 : jusque là j'ai pas vu
A5 : D'accord / ok je continue à faire dérouler / y'a rien / donc nous ne mettons rien en accès thématique / en *mot clé* bah écoutez je vous propose de mettre leucémie
B6 : leucémie
A7 : leucémie / D'ailleurs il me le propose / on va le mettre directement
et le qualificatif permet de préciser un tout petit peu ce mot clé / donc par rapport à votre question / de la même façon je vais faire dérouler le menu et vous allez me dire si vous voyez des choses
B8 : chimiothérapie
A9 : alors chimiothérapie ok / alors vous souhaitez mettre d'autres qualificatifs
B10 : oui
A11 : D'autres précisions / alors on va aller chercher [...]
B12 : diagnostic
A13 : diagnostic ok [...] ok je continue à faire dérouler le menu de qualificatifs
B14 : effets indésirables
A15 : effets indésirables / ok / alors effets indésirables / donc je continue à faire dérouler [...] le menu déroulant
B16 : sang
A17 : allons-y / sang et de toute façon là je pense que ça va être à peu près tout [...]
alors souhaitez-vous qu'on aille regarder dans un *type de ressource* particulier relatif aux associations / base de données / bibliothèque médicale / etc / ou quelque chose qui soit spécifique pour les patients
B18 : ouais
A19 : oui / alors je mets type de ressource : patient / bon on va le lancer comme ça [...]

Dialogue 4.3 – Exemple de dialogue de construction de requête initiale avec la tactique de téléportation (VD02). A est l'expert, B est le demandeur.

A1 : [...] alors donc on a plusieurs domaines / alors déjà *on va commencer par l'accès thématique* / ça nous permet d'avoir accès à différentes branches médicales // donc pour le dos euh : // alors je regarde s'il y a quelque chose qui va convenir sinon on va faire autrement [...]
B2 : c'est peut-être un truc osthéo
A3 : oui c'est ça / alors donc ostéopathie merci /euh : // non ils ont pas ça dans l'annuaire /
B4 : rhumatologue / rhumatologie dans la première /
A5 : d'accord / euh : donc on va essayer ça / avec rhumatologie / donc *je lance la recherche comme ça*

Dialogue 4.4 – Exemple de dialogue de construction de requête initiale avec la tactique d'orienteering (AL05). A est l'expert, B est le demandeur.

n'est pas possible de généraliser la requête, les interlocuteurs vont soit reformuler un des termes (ou plusieurs), soit abandonner la requête. Les tactiques de généralisation conduisent généralement à supprimer un élément de la requête (mot-clé, qualificatif, etc.) ou à substituer un terme par un plus général. Le dialogue 4.2 (p. 95) présente un exemple de tactique de généralisation.

Tactiques de reformulation Les tactiques de reformulation interviennent lorsque les ressources obtenues sont directement rejetées par les interlocuteurs. Il s'agit du cas où les ressources sont clairement non satisfaisantes. Le constat d'inadéquation entre la requête et ce qui est attendu par le demandeur est généralement explicité (e.g., *« les pistes qu'on a prises ne fonctionnent pas donc on va essayer de partir sur autre chose »* (VD08), cf. dialogue 4.1). En outre, les tactiques de reformulation sont choisies dès le moment où les partenaires n'ont pas la possibilité d'appliquer les tactiques de généralisation et de précision. La reformulation de la requête peut reposer sur des termes *précédemment énoncés* durant la discussion et qui n'ont pas encore servi à construire une requête. Les tactiques de reformulation conduisent généralement à altérer ou substituer un élément de la requête (e.g., par l'utilisation de synonyme) ou à utiliser une nouvelle combinaison de termes précédemment évoqués. Le dialogue 4.1 (p. 94) présente un exemple de tactique de reformulation.

Synthèse de l'usage des tactiques La figure 4.4 représente l'enchaînement idéalisé des tactiques en fonction des résultats de la requête (nombre de résultats et satisfaction). Les phases du dialogue sont représentées sur la figure par des rectangles, les décisions par des losanges et les tactiques par des ellipses. Le premier critère sur lequel se base le choix de la tactique est le nombre de résultats obtenus après le lancement de la requête. Si le nombre de résultat est nul, la tactique à choisir dépend de la requête. Si elle est *généralisable*, la tactique choisie est la *généralisation* sinon la *reformulation*. Si le nombre de résultats est strictement positif, le choix de tactique dépend de l'adéquation des résultats avec les attentes de l'enquêté. Si les résultats sont satisfaisants, le dialogue de recherche passe à la phase de clôture. Cela signifie que la recherche est un succès et que les interlocuteurs soit interrompent leur collaboration, soit passent à une nouvelle séquence. Si les résultats récoltés jusqu'alors sont partiellement satisfaisants, la tactique choisie sera la *précision*. Sinon, si les résultats ne sont pas satisfaisants (i.e. hors sujet), la tactique choisie est la *reformulation*.

Ce schéma présente une tendance dans le choix des tactiques qui n'est pas toujours respectée. Marginalement, nous avons observé des modifications de requête issues de l'inspiration du moment d'un des collaborateurs. Par exemple, deux entretiens de notre corpus (AL01 et VD06) font apparaître des modifications de requête combinant une tactique de généralisation et l'ajout spontané d'un terme (motivé par *« on va essayer contre-indications [...] à tout hasard »* dans AL01, et la volonté du demandeur dans VD06).

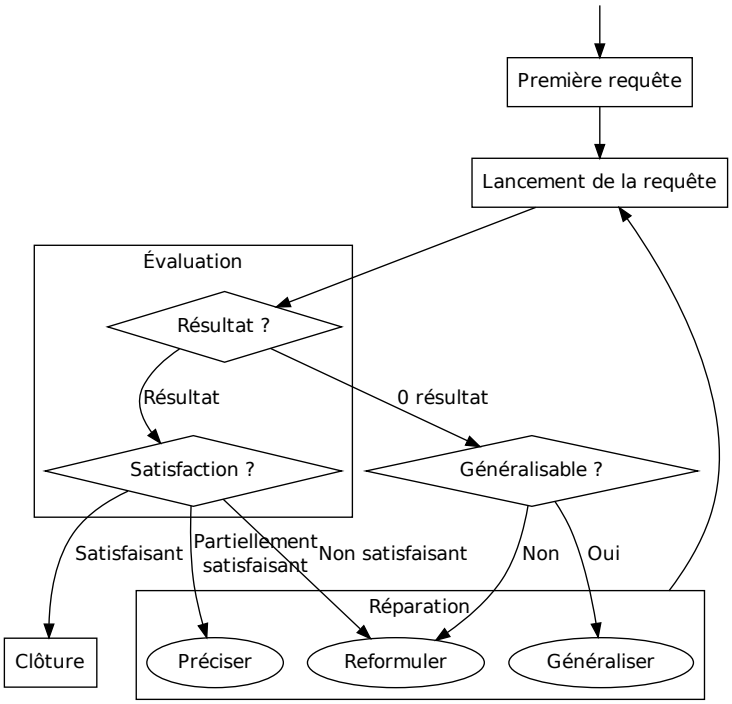

FIGURE 4.4: Enchaînement idéalisé des tactiques de réparation en fonction des résultats de la requête. Les phases du dialogue sont représentées par des rectangles, les décisions par des losanges et les tactiques par des ellipses.

Le tableau 4.3 présente la répartition des tactiques dans le corpus. Globalement, nous constatons que les trois types sont répartis uniformément (approx. 30 instances de tactiques recensées pour chaque type). Dans le détail, nous observons qu'une grande majorité des tactiques de réparation utilisées dans la stratégie globale de *téléportation* sont des généralisations (53% du total, expert VD). Ceci s'explique par le fait que les requêtes trop précises retournent rarement des résultats dans CISMEF. Au contraire, la stratégie globale d'*orienteering* ne conduit qu'à très peu de tactiques de généralisation (8% du total, expert AL). Elle implique principalement des tactiques de précisions (48%) et de reformulation (44%). Ceci s'explique par le fait que la stratégie d'orienteering consiste à adapter la requête en fonction des résultats obtenus. En synthèse, la stratégie globale influence les tactiques de réparation utilisées dans le processus de recherche. La *téléportation* conduit à l'utilisation de tactiques de généralisation tandis que l'*orienteering* conduit à préciser ou reformuler la requête.

4.3.4 Rôles des participants

Dans cette section, nous nous intéressons aux rôles des participants dans la situation de RI collaborative. Dans un premier temps, nous présentons brièvement l'initiative mixte dans un

Tactiques	VD	AL	Total
Précisions	11 (23,4%)	23 (47,9%)	34 (35,8%)
Généralisation	25 (53,2%)	4 (8,3%)	29 (30,5%)
Reformulation	11 (23,4%)	21 (43,8%)	32 (33,7%)
	47	48	95

TABLEAU 4.3: Répartition des tactiques de réparation dans le corpus

contexte de résolution collaborative de problème. Puis nous abordons le type d'interaction majoritairement observé dans le corpus. Enfin, nous présentons quelques éléments clés des processus mis en jeu par l'expert dans ce corpus.

Initiative mixte dans un contexte de résolution collaborative de problème

En interaction Homme-Machine, on différencie généralement trois grands types d'interaction du point de vue du système [Allen 1999]. Le système peut être *réactif*. Il s'agit de systèmes qui se contentent de réagir aux ordres de l'utilisateur. L'*initiative* est conservée par l'humain. Typiquement, l'application se présente sous la forme d'une interface graphique qui va être manipulée par un utilisateur (e.g., un éditeur de texte, navigateur web, etc.). À l'autre extrême, on observe les systèmes *directifs*. Il s'agit de systèmes qui contrôlent rigidement le flot de l'interaction. C'est alors l'utilisateur qui réagit aux sollicitations du système. Les meilleurs exemples sont les centres d'appel où l'utilisateur doit appuyer sur des touches pour naviguer dans le menu. Entre les deux se situe l'*initiative mixte* dans un contexte de résolution de problème entre un humain et des agents qui a émergé de différents travaux [Horvitz 1999b, Allen 1999, Horvitz 2007, Tecuci 2007]. L'initiative mixte suppose un entrelacement naturel et efficient des contributions des utilisateurs et des agents déterminé par leurs connaissances relatives, leurs compétences, et par le contexte de résolution du problème plutôt que par des rôles fixés. Ceci permet à chaque participant de contribuer à ce qu'il fait le mieux au moment approprié. L'idée de l'interaction d'initiative mixte est résumée dans [Allen 1999] (p. 14) : « L'interaction d'initiative mixte laisse les agents travailler de manière plus efficiente en équipe – c'est la clé. Le secret est de laisser les agents qui connaissent sur le moment la meilleure façon de procéder coordonner les autres agents. ». L'initiative mixte se manifeste généralement dans les systèmes de dialogue capables d'avoir à la fois l'initiative (« Quel type de document souhaitez-vous ? ») et de permettre à l'utilisateur de prendre l'initiative (« En fait, je voudrais modifier le mot-clé dans la requête. »). Néanmoins, l'interaction d'initiative mixte va au-delà en incluant plus largement la résolution de problème collaborative. L'initiative mixte dans ce contexte est illustrée par un entrelacement de contributions de ses différents participants.

Rôles de l'expert et du demandeur

Le rôle des interlocuteurs dans la situation de collaboration de notre expérimentation peut se caractériser par les degrés de familiarité avec l'*outil* CISMeF et le *domaine* de la médecine [Golovchinsky 2008]. Dans notre expérimentation, l'expert possède l'*expertise* de l'outil et est *novice* du domaine tandis que l'utilisateur est *novice* de l'outil et *novice* du domaine.

Nous avons constaté que ces rôles d'expert et de demandeur donnent des responsabilités différentes. Le demandeur est à l'origine du besoin d'information d'ordre médical qui motive la RI.

En outre, il est capable de juger de l'intérêt des ressources retournées par le moteur de recherche. L'expert possède les connaissances et les compétences lui permettant d'utiliser efficacement le moteur de recherche CISMeF et de répondre à des questions à propos de CISMeF (e.g., terminologie, fonctionnement, types de ressources indexées, etc.). Il possède un ensemble de stratégies et de tactiques afin de mener la recherche et de récolter des ressources satisfaisantes pour le demandeur. En outre, il a la responsabilité de mener la RI en collaboration avec le demandeur. Il l'aide à construire sa première requête. Il lui présente un ensemble de ressources résultant de l'exécution de la requête. Il évalue, en collaboration avec le demandeur, l'adéquation entre les ressources et le besoin d'information. Enfin, les partenaires modifient conjointement la requête afin de récolter des ressources satisfaisant le besoin d'information du demandeur.

La situation place l'expert en *initiateur* puisqu'il est chargé de mener la recherche en guidant le demandeur. En outre, c'est lui qui manipule le moteur de recherche. Néanmoins, les différentes phases de la tâche donnent l'*initiative* à l'un ou l'autre des participants en fonction de leurs capacités. Les phases de verbalisation et de construction de la première requête sont principalement menées par l'expert (cf. dialogue 4.3, p. 97). La phase d'évaluation donne l'initiative au demandeur qui sait le type d'information qu'il souhaite obtenir. La phase de réparation de la requête est clairement d'initiative mixte (cf. dialogue 4.2, p. 95). En outre, certains dialogues présentent des prises d'initiative spontanées du demandeur dans les phases normalement dirigées par l'expert (cf. dialogue 4.4, p. 98). Par exemple, bien que présentant une procédure systématique pour la conception d'une première requête, l'expert VD accepte les prises d'initiative des demandeurs qui souhaitent lancer la requête avec un unique mot-clé. En somme, l'interaction observée est radicalement d'*initiative mixte*.

Capacités de l'expert mises en jeu

En sus des capacités dialogiques et interactives dont l'expert doit faire preuve, nous notons la présence de capacités spécifiques à la tâche de RI. Dans cette partie, nous synthétisons certaines capacités de l'expert pour proposer des termes complétant la requête. Nous nous intéressons aux critères qui entrent en jeu dans l'évaluation des ressources. Enfin, nous soulignons son expertise vis-à-vis du moteur CISMeF ainsi que sa coopérativité envers les buts proposés par le demandeur.

Proposition de termes Nous avons noté un certain nombre de sources utilisées par l'expert pour proposer des termes au demandeur. La première source est les termes extraits de la verbalisation du besoin d'information. L'expert est capable de proposer des termes en utilisant la terminologie CISMeF. Par exemple, il est capable de proposer « donneur de tissus » à partir du terme « don » dans l'entretien VD01.

L'expert extrait des termes des notices des ressources retournées par une requête. Ces termes sont ceux qui indexent la ressource. L'extraction est souvent explicite, comme par exemple : *« on parle apparemment des 'troubles anxieux' »* (AL02). Elle intervient dans les phases d'évaluation et de réparation de la requête.

L'expert est capable de proposer l'ajout de termes à la requête en se basant sur le profil de l'enquêté. Dans notre corpus, il s'agit principalement pour l'expert de proposer l'ajout du type de ressources « patient ».

L'expert re-propose des termes évoqués pendant l'interaction comme des termes de la verbalisation ou des expressions soulevées lors de l'analyse des ressources dans la phase d'évaluation.

Enfin, l'expert utilise ses connaissances sur le domaine pour proposer spontanément des termes au demandeur. C'est notamment le cas dans l'entretien VD09 où l'enquêté recherche des documents sur la prévention du glaucome. L'expert propose alors spontanément d'ajouter la spécialité médicale « ophtalmologie » compte-tenu du fait qu'il sait que le glaucome est une maladie dégénérative du nerf optique.

Évaluation des ressources L'expert est capable d'évaluer les ressources dans certains cas. Les critères « simples » se basent sur (i) le nombre de ressources retournées par la requête (*trop* ou *pas assez* de documents), et (ii) sur le fait que les résultats obtenus ont déjà été présentés au demandeur. D'autres critères impliquant un raisonnement plus avancé de l'expert peuvent intervenir. Il peut ainsi déterminer que *les résultats sont manifestement hors sujet*. Dans l'entretien VD04, l'expert est capable de juger les résultats retournés comme inadéquats car ils abordent des maladies rares et non un problème de calvitie. Ensuite, l'expert se base sur le fait que *les ressources ne recouvrent qu'un sous-ensemble des thèmes de la recherche* (e.g., les ressources parlent de « sport » mais pas des « inconvénients du sport »). L'expert peut également évaluer que *les résultats sont en inadéquation avec le profil de l'enquêté* (e.g., les documents sont pour des spécialistes). Les autres critères impliquent des raisonnements plus poussés sur les attentes de l'enquêté. Il est cependant intéressant de remarquer des cas particuliers spécifiques au moteur de recherche CISMeF. Par exemple, l'expert est capable de juger un document comme étant trop général dans l'entretien AL05 car il est indexé par un grand nombre de mots-clés de domaines différents.

Expertise vis-à-vis de CISMeF L'expertise de l'expert se manifeste proactivement et réactivement. L'expert propose proactivement des explications sur les éléments de la terminologie à chaque fois qu'il en introduit un. Il définit par exemple ce qu'est un *accès thématique* (e.g., *« alors accès thématique on va dire que c'est grosso modo l'ensemble des spécialités médicales »* (VD02)), un qualificatif (e.g., *« le qualificatif permet de préciser un tout petit peu ce mot-clé »* (VD02)) ou le fonctionnement de CISMeF (e.g., *« bon on va lancer comme ça [...] il va aller chercher donc sur internet l'ensemble des sites que cismef a répertorié »* (VD02)).

En outre, l'expert apporte son expertise sur la requête et la terminologie qui font la spécificité de CISMeF. Ses compétences et connaissances spécifiques à CISMeF lui permettent de mettre à portée du demandeur des fonctionnalités qu'il ne connaît pas (e.g., *« on va essayer au lieu de mettre en* mot-clé*, mettre en* plein texte *genou / pour voir les articles de la médecine du sport où intervient le mot genou »* (AL06)).

Enfin, l'expert est disponible pour répondre aux demandes opportunistes du demandeur sur CISMeF. C'est le cas dans l'entretien AL10 où le demandeur interroge l'expert après la première requête : *« donc document c'est / c'est des documents qui / c'est quel type de document parce que je ne sais pas où se fait la recherche en fait / »*. Il s'ensuit un sous-dialogue de 9 tours de parole entre les interlocuteurs pour répondre aux questions sur CISMeF.

Adoption de but Enfin, l'expert n'hésite pas à adopter les buts du demandeur en laissant de côté le but conjoint courant. Par exemple, le demandeur de l'entretien VD03 indique à l'expert qu'il cherche un terme dont une définition est « l'idée qu'un médicament perde son efficacité quand on l'utilise ». L'expert adopte alors le but du demandeur et lui propose de regarder la liste des mots-clés CISMeF pour essayer de trouver un terme correspondant à la définition.

4.4 Synthèse

Dans ce chapitre, nous avons présenté les étapes de collecte et de transcription du corpus Cogni-CISMeF. Ce dernier est au cœur de l'extraction de motifs d'interaction dialogique que nous souhaitons modéliser dans cette thèse. Nous avons donné une description des dialogues du corpus. Nous avons caractérisé leur tâche sous-jacente qui consiste en la réalisation collaborative d'une recherche d'information sur CISMeF entre un expert et un utilisateur. Cette caractérisation s'est appuyée sur l'étude du processus de RI dans le cas individuel. Nous avons vu que la tâche de RI collaborative met en jeu un processus *itératif*, *opportuniste*, *stratégique* et *interactif* entre les participants. La structure globale des dialogues dans une situation de collaboration entre un expert CISMeF et un demandeur suit des étapes similaires à celles d'un processus de RI individuel : identification du besoin d'information, spécification de la requête, évaluation des résultats puis réitération du processus entier. Chaque phase implique une interaction entre les participants. La phase de verbalisation permet de rendre la collaboration sur la RI *explicite*. La RI est menée *stratégiquement* par les partenaires. La stratégie globale est caractérisée par la tactique initiale (orienteering ou téléportation). La phase d'évaluation conduit à la reconsidération du besoin d'information sur la base des résultats observés. La requête est modifiée par l'élaboration conjointe d'une tactique de réparation. En outre, les participants possèdent un *rôle* définissant leurs compétences complémentaires vis-à-vis de la collaboration. Ces résultats complètent le travail de [Loisel 2008] qui s'était focalisé sur l'étude des questions/réponses apparaissant dans le corpus.

Nous avons souligné un certain nombre d'indices tout au long de notre étude de la tâche qui pourront être utilisés dans l'objectif plus global de conception d'un agent assistant d'initiative mixte pour la RI dans CISMeF. Nous laissons en suspens la question de l'élaboration d'un tel agent. Le reste de notre travail de thèse se concentre plus généralement sur les capacités dialogiques des agents interactifs. Nous soulignons néanmoins que la conception d'un agent assistant pour la RI s'inscrit dans la problématique plus large de la collaboration sur une tâche de RI. Un des objectifs qui en découle est la réalisation de systèmes qui vont collaborer *explicitement* avec l'utilisateur. À l'opposé des systèmes en boîte noire comme le célèbre Google, l'idée est de permettre à l'utilisateur de *diriger* sa recherche [Bates 1990]. Il est alors nécessaire de casser la supposition selon laquelle le système et ses concepteurs *savent mieux* que l'utilisateur. Il s'agit de rendre le contrôle à l'utilisateur en lui cachant le moins de détails (e.g., rendre explicite la façon dont l'information est recherchée par le système, quels documents ont été rejetés et pourquoi, etc.) tout en lui fournissant une assistance automatique améliorant la réalisation de sa tâche.

Chapitre 5

Annotation du corpus et extraction de motifs

Sommaire

5.1 Processus d'annotation 106
5.1.1 Segmentation du dialogue 106
5.1.2 Configuration du processus d'annotation 108
5.1.3 Résultats du processus d'annotation 112
5.1.4 Synthèse 125
5.2 Processus d'extraction de motifs dialogiques 126
5.2.1 Configuration du processus d'extraction 127
5.2.2 Résultats du processus d'extraction 128
5.2.3 Synthèse 141
5.3 Discussion 142

Dans ce chapitre, nous illustrons les phases d'*annotation du corpus* et d'*extraction de régularités* de notre méthodologie en les appliquant au corpus Cogni-CISMeF (cf. figure 5.1). La section 5.1 présente l'étape d'annotation du corpus. Cette étape réalise l'enrichissement du

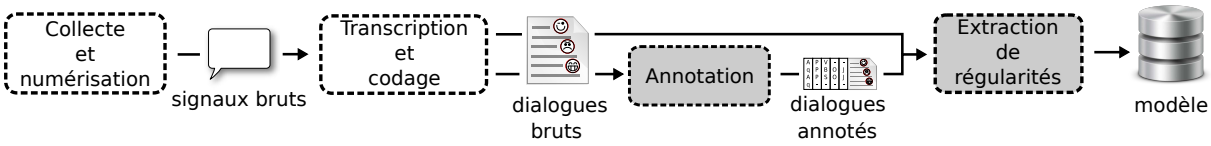

Figure 5.1: Annotation du corpus Cogni-CISMeF et extraction de régularités. Le fond coloré indique les étapes traitées dans ce chapitre.

corpus en s'appuyant sur le schéma d'annotation DIT++. Elle consiste en la segmentation du corpus en unités et à leur annotation avec des fonctions communicatives. À l'issue de ce processus, une représentation multidimensionnelle de chaque unité d'interaction est obtenue. Cette représentation est exploitée afin d'extraire des motifs dialogiques, c'est-à-dire des séquences d'unité d'interaction récurrentes. Le processus d'extraction de motifs dialogiques est développé en section 5.2. Enfin, nous concluons ce chapitre en commentant les évolutions envisageables de notre méthodologie (cf. section 5.3).

5.1 Processus d'annotation

Cette section présente la tâche d'annotation du corpus Cogni-CISMeF. Nous nous intéressons dans un premier temps à la définition de l'unité de sens dans le dialogue humain désignée par le terme « énoncé » (section 5.1.1). Ensuite, nous abordons la mise en place du processus d'annotation (section 5.1.2). Après avoir abordé le schéma d'annotation et les moyens utilisés pour réaliser cette tâche, nous présentons une analyse des résultats obtenus (section 5.1.3). Enfin, nous synthétisons l'essentiel des conclusions de ce processus d'annotation (section 5.1.4).

5.1.1 Segmentation du dialogue

L'étude du dialogue sous l'angle de la réalisation d'actions communicatives présuppose un moyen de déterminer les portions de comportement communicatif réalisant ces actions. Intuitivement, l'énoncé est vu comme l'*unité* réalisant au moins une action communicative (voire plus pour un énoncé *multifonctionnel*, cf. section 1.1.2). L'identification de ces unités est appelée la *segmentation du dialogue*. Cette opération est loin d'être triviale dès lors que l'on considère la multidimensionnalité et la multifonctionnalité du dialogue [Larsson 1998, Bunt 2011b]. Dans cette section, nous nous intéressons à la définition de cette unité de sens dans le dialogue.

Bunt apporte un éclairage enrichissant sur cette question [Bunt 2011b] en s'appuyant sur les apports d'Allwood [Allwood 1992]. Il distingue plusieurs *formes* de multifonctionnalité : *séquentielle*, *discontinue*, *chevauchante*, *entrelacée* et *simultanée*. Les quatre premiers types sont illustrés dans le tableau 5.1 et concernent des cas de multifonctionnalité liés à la segmentation du dialogue. La multifonctionnalité séquentielle consiste en un tour de parole composé d'une séquence de parties distinctes ayant chacune une fonction communicative différente. L'exemple du tableau 5.1 peut être analysé comme la succession suivante : feedback (*« Donc ce que c'est exactement les symptômes »*), feedback (*« OK »*), suggestion (*« Donc on y va aller »*), affirmation (*« Alors accès thématique on va dire que c'est grosso modo l'ensemble des spécialités médicales »*) et question (*« Est-ce que vous voyez des choses »*). Il s'agit du cas le plus simple où le tour peut être décomposé en une succession d'unités réalisant chacune un acte différent. La multifonctionnalité discontinue consiste en un énoncé interrompu par l'occurrence d'un autre. L'énoncé *« on va voir [...] ce qu'on obtient »* est un énoncé discontinu par l'intervention de l'unité *« on lance la recherche »*. Le type « chevauchant » consiste en une *partie* d'un énoncé qui a une fonction communicative propre différente de l'énoncé complet. C'est le cas du feedback positif *« on peut mettre de côté un des documents »* dans la réponse complète *« mais bien sûr on peut mettre de côté un des documents »*. Enfin, il est possible que deux énoncés soient entrelacés. La segmentation des unités de sens dans le dialogue a une influence sur l'attribution de leurs fonctions communicatives. Ces unités doivent être définies de manière à prendre en compte les formes de multifonctionnalité liées à la segmentation du dialogue.

Contrairement aux formes que nous venons d'aborder, la multifonctionnalité *simultanée* n'est pas connectée à la segmentation. Elle réfère au fait qu'un énoncé réalise *simultanément* plusieurs fonctions. Dans l'exemple *« – Quelle est la capitale de France ? – Paris »*, la réponse « Paris » est simultanément une réponse et un feedback fort de compréhension. Cette simultanéité peut s'expliquer de deux façons [Bunt 2011b] : (i) soit parce que l'énoncé contient des indicateurs de surface démontrant la réalisation de fonctions dans des dimensions différentes (appelée multifonctionnalité indépendante), (ii) soit parce que l'énoncé réalise une fonction en impliquant d'autres (comme dans l'exemple précédent).

Type	Exemple
Séquentielle	*« Donc ce que c'est exactement les symptômes / OK / Donc on y va aller / Alors accès thématique on va dire que c'est grosso modo l'ensemble des spécialités médicales / Est-ce que vous voyez des choses »* (VD02)
Discontinue	*« on va voir on lance la recherche/ ce qu'on obtient »* (AL02)
Chevauchante	**S**1 : on peut mettre de côté un des documents là ? **H**2 : mais bien sûr on peut mettre de côté un des documents (VD06)
Entrelacée	*« Trente euros pour une clé USB . . . est-ce que Paul n'en aurait pas une à te prêter. . . c'est un peu cher ! . . . ou peut-être Coralie ? »*

Tableau 5.1: Segmentation et multifonctionnalité

Nous pouvons nous interroger sur le sens à accorder au terme « énoncé ». En effet, la manière de segmenter a un impact direct sur la multifonctionnalité des unités choisies. Intuitivement, il semble nécessaire d'adopter une définition permettant d'éliminer les formes de multifonctionnalité séquentielle, discontinue, chevauchante et entrelacée. Pourtant, l'état de l'art foisonne de définitions très différentes. L'unité parfois employée en tant qu'énoncé est le *tour de parole*. Un tour de parole est « une unité dans le dialogue oral qui correspond à une portion de l'élocution d'un locuteur, délimitée par une absence d'activité ou par l'activité d'un autre interlocuteur » [Allwood 1992]. Cette définition d'énoncé offre une unité d'étude grossière et ignore de fait les formes de multifonctionnalité précédemment présentée. Il est courant de segmenter un tour de parole en unité plus fine étant soit une forme nominale, soit une unité grammaticale bien formée pouvant être vue comme la réalisation linguistique d'un acte de langage (voir, e.g., la notion de « c-unit » [Weisser 2003]). Ces unités sont alors prises pour énoncé. Cette dernière définition permet d'envisager la multifonctionnalité séquentielle mais échoue à considérer les trois autres formes. Alternativement, Bunt propose de voir l'énoncé sous l'angle du *segment fonctionnel* défini comme étant « . . . une portion minimale de comportement communicatif qui a une fonction communicative. De telles portions n'ont pas besoin d'être grammaticalement correctes ou contiguës et peuvent avoir plus d'une fonction communicative. » [Bunt 2011b] (cf. section 1.2.2). Le dialogue 5.1 présente un exemple de segment fonctionnel sur plusieurs tours (tours 1, 3, 5) clôturé par une complétion du partenaire (tour 6), et coupé par des interventions du partenaire (tours 2 et 4). Chacun des autres tours forme également son propre segment fonctionnel (tours 2, 4, 6, 7). Ce dialogue est présenté annoté avec les fonctions communicatives de DIT++ en annexe A.3.1.

Le segment fonctionnel est une unité d'analyse intéressante pour le dialogue. Il permet d'éliminer de par sa définition les formes de multifonctionnalité séquentielle, discontinue, chevauchante et entrelacée. Seule persiste la multifonctionnalité simultanée. Cette dernière est en partie gérée par le fait qu'un segment fonctionnel dans DIT++ peut être annoté par une fonction communicative par dimension (cf. section 1.2.2). Notons qu'un segment fonctionnel peut être très *court* (*« euh : »*) ou très *long*, et s'étaler sur plusieurs tours de parole. C'est l'unité d'analyse que nous adoptons pour notre tâche d'annotation.

Expert$_1$: [...]donc on a dit qu'on mettait
Enquêté$_2$: attention
Expert$_3$: néphrite
Enquêté$_4$: ouais
Expert$_5$: et qu'on gardait
Enquêté$_6$: membres inférieurs
Expert$_7$: D'accord [...]

Dialogue 5.1 – Exemple de segment sur plusieurs tours de parole (entretien VD08).

5.1.2 Configuration du processus d'annotation

Dans cette section, nous présentons les paramètres d'annotation du corpus COGNI-CISMEF. Dans un premier temps, nous décrivons le schéma d'annotation choisi et nous fournissons les détails sur les moyens (humains et autres) mis en œuvre pour l'annotation. Enfin, nous décrivons la stratégie d'annotation choisie. Pour finir, nous soulignons les limites du processus d'annotation.

Schéma d'annotation

Nous avons choisi le cadre DIT++ qui fournit un schéma d'annotation multidimensionnel associé à une taxonomie de fonctions communicatives (cf. section 1.2.2). Le framework DIT++ est attractif pour la conception du comportement communicatif d'un agent interactif pour plusieurs raisons [Bunt 2009].

Il propose une taxonomie de fonctions *indépendante de l'application* pour l'*analyse* du dialogue Homme-Homme et pour la *conception de système de dialogue* (et plus particulièrement du gestionnaire de dialogue). Tout d'abord, DIT++ propose une *taxonomie riche* de fonctions communicatives. Sa force est de clarifier la portée des fonctions communicatives en les regroupant dans les aspects du dialogue auxquels elles prennent part (i.e., les dimensions). L'existence et la pertinence des dix dimensions proposées ont été validées empiriquement [Petukhova 2009a, Petukhova 2009b]. Ensuite, une approche multidimensionnelle de l'annotation du dialogue permet une analyse plus précise de la communication Homme-Homme. Malgré sa richesse, une telle approche facilite l'annotation [Petukhova 2007, Bunt 2009]. Ceci est permis grâce à la distinction de dimensions chacune relative à un aspect de la communication bien défini. Le travail de l'annotateur peut alors se décomposer en deux étapes. Premièrement, il s'agit de considérer les aspects de la communication abordés par l'unité de comportement communicatif. Cela consiste à choisir la ou les dimensions traitées par le segment. Ensuite, l'annotateur détermine à un grain plus fin la ou les fonctions communicatives réalisées. En outre, le choix de la fonction est simplifiée par la structure taxonomique de DIT++ qui met en avant des fonctions incompatibles ou spécialisées (cf. section 1.2.2). La taxonomie permet d'éviter à l'annotateur la considération de fonctions non pertinentes. De plus, les dimensions telles que définies dans DIT++ permettent au schéma d'être utilisé *partiellement*. L'annotation ou l'analyse d'un corpus de dialogues peut n'utiliser qu'un sous-ensemble des dimensions proposées. Enfin, DIT++ est accompagné d'une documentation précise en ligne [Bunt 2010] [1] comprenant (i) l'ensemble de la taxonomie de fonctions communicatives associées à des descriptions détaillées, (ii) un guide complet de conseils pour les tâches d'annotation, et (iii) des exemples de dialogues Homme-Homme et Homme-Machine

1. Disponible à l'URL http://dit.uvt.nl/

annotés.

Au-delà de l'étude de corpus, DIT++ semble attrayant pour la réalisation effective d'un agent interactif. Tout d'abord, l'utilisation de techniques d'apprentissage pour la reconnaissance automatique des fonctions communicatives de DIT++ semble encourageante [Geertzen 2007, Geertzen 2009a, Bunt 2009]. Tout comme l'annotation humaine, elle peut bénéficier des avantages fournis par la décomposition en dimensions et la structure taxonomique des fonctions communicatives. Ensuite, DIT++ est très prometteur pour les agents interactifs tels que les agents conversationnels animés par son *extensibilité* [Petukhova 2007] et son support de la *multifonctionnalité*. En effet, il est possible d'étendre DIT++ par l'ajout de nouvelles dimensions proprement définies pour inclure d'autres aspects tels que les émotions [Ekman 1999]. DIT++ peut aussi être utile pour l'*interprétation* de comportements communicatifs non verbaux (e.g., mouvements de la tête, expressions faciales) [Bunt 2009]. Une étude réalisée dans le projet AMI [2] a montré que l'ensemble des fonctions communicatives réalisées par un comportement non verbal a pu être adéquatement décrit par les fonctions de DIT++ [Bunt 2009]. Pour finir, l'utilisation du schéma DIT++ dans un système de dialogue permet d'envisager la *génération* d'énoncés multifonctionnels [Keizer 2006] qui peuvent être réalisés de manière multimodale par un ACA (via la combinaison d'expressions faciales et de langage parlé).

Dans son travail sur le corpus Cogni-CISMeF, Loisel avait dressé une taxonomie d'actes de langage lors de son analyse [Loisel 2008] en partant de celle de [Weisser 2003]. Notons que cette taxonomie est incluse dans DIT++. L'annexe A.2 présente l'alignement entre les fonctions de la taxonomie de Loisel et DIT++.

Moyens et corpus

Quatre annotateurs ont réalisé cette tâche d'annotation, notés A, G, J et N. Les annotateurs ont déjà réalisé une telle tâche d'annotation mais ne sont pas experts du schéma DIT++. J'ai donc mis en place une formation à la taxonomie. Cette formation a présenté de manière détaillée les concepts et la taxonomie de DIT++, des exemples de dialogue Homme-Homme annotés et la stratégie d'annotation (présentée dans la suite du document). Les annotateurs étaient ensuite invités à réaliser une série d'exercices d'annotations basés sur les exemples fournis dans la documentation de DIT++ [Bunt 2010]. Ces exercices leur ont permis de confronter leurs annotations à celles de référence. Pour cette tâche, les annotateurs avaient à leur disposition les ressources suivantes : (i) la description générale des concepts de DIT++ [Bunt 2009], (ii) la documentation du schéma incluant la description de l'ensemble des dimensions et des fonctions [Bunt 2010], (iii) les hiérarchies de fonctions (cf. figures 1.1 et 1.2), (iv) des exemples de dialogues annotés [3], et (v) le guide d'annotation [4].

Nous avons retiré de l'analyse deux entretiens : AL01 et VD01. Pour une raison inconnue, l'entretien AL01 ne comporte qu'un tour de parole de la part du demandeur et ne présente donc pas d'intérêt pour l'étude du comportement communicatif. L'entretien VD01 fait apparaître un expert rencontrant des difficultés pour mener de front la recherche avec le demandeur et la manipulation de l'interface. Cet entretien a permis à l'expert de s'approprier l'environnement d'expérimentation. En tout, un total de 18 entretiens a été considéré dans le processus

2. AMI=Augmented Multi-party Interaction (http://www.amiproject.org)
3. Disponibles aux URL : http://dit.uvt.nl/is-examples.html et http://dit.uvt.nl/annotated-examples.html
4. Disponible à l'URL : http://dit.uvt.nl/AnnoGuide.html

d'annotation. Le corpus AL contient 10 entretiens tandis que le corpus VD en possède 8.

Le tableau 5.2 présente la répartition des entretiens entre les annotateurs. Chaque dialogue a été annoté par deux personnes. L'annotateur G (moi-même) a annoté l'ensemble du corpus tandis que les trois autres ont chacun annoté un tiers du corpus.

	AL02	AL03	AL04	AL05	AL06	AL07	AL09	AL10	AL11	AL12
Annot. 01	G	G	G	G	G	G	G	G	G	G
Annot. 02	N	J	N	J	A	N	A	J	A	N

	VD02	VD03	VD04	VD05	VD06	VD07	VD08	VD09
Annot. 01	G	G	G	G	G	G	G	G
Annot. 02	N	N	J	J	A	J	A	A

Tableau 5.2: Répartition des dialogues par annotateur dans le processus d'annotation. A, G, N et J sont les identifiants des annotateurs.

L'annotation a été réalisée au moyen de l'outil d'annotation Gate [Cunningham 2011] (« General Architecture for Text Engineering » [5]) en version 6.1. Nous y avons créé un *schéma d'annotation* qui permet de standardiser les annotations manuelles réalisées dans les dialogues. Ce schéma d'annotation sous forme XML est disponible dans le listing A.1 en annexe A.1. Il définit le *type* d'annotation « acte de dialogue » comme étant constitué de trois *features* : (i) le *locuteur* qui peut prendre la valeur « Expert » ou « Enquêté », (ii) la *dimension* qui peut prendre comme valeur l'une des dimensions de DIT++, et (iii) la *fonction communicative* qui peut prendre comme valeur une des fonctions communicatives de DIT++. L'annotation d'un segment fonctionnel consiste en l'annotation de plusieurs « actes de dialogue » sur la même zone de texte (en respectant la contrainte d'au plus une fonction communicative par dimension).

Stratégie d'annotation

Le processus d'annotation est constitué de deux parties : (i) la *segmentation* des tours de parole en *segments fonctionnels* (cf. section 5.1.1), et (ii) l'*étiquetage* de segments fonctionnels avec zéro ou une fonction communicative par dimension. Comme nous l'avons vu en section 5.1.1, un segment fonctionnel peut avoir simultanément plusieurs fonctions soit parce que l'énoncé contient des indicateurs de surface démontrant la réalisation de fonctions dans des dimensions différentes, soit parce que l'énoncé réalise une fonction en impliquant d'autres. Nous avons choisi la stratégie d'annotation « strictement basée sur des indicateurs » [Bunt 2011b]. Il a été demandé aux annotateurs de marquer les fonctions communicatives directement reconnaissables depuis les caractéristiques linguistiques du segment fonctionnel, tout en tenant compte du passé du dialogue. Les indicateurs explicites fournissent une base claire sur laquelle justifier l'attribution d'une fonction communicative. Il est en revanche plus difficile de spécifier sur quelle base une fonction en implique une autre (voir à ce sujet [Bunt 2011b]).

En outre, il a été conseillé aux annotateurs d'annoter les segments en se mettant à la place de l'allocutaire essayant de comprendre ce que le locuteur veut dire. L'idée est d'inférer le contexte dialogique dans lequel est l'allocutaire afin d'interpréter le comportement communicatif du locuteur. Dans l'esprit des approches contextuelles, il a été demandé aux annotateurs de se méfier de la forme linguistique de l'énoncé. L'annotation ne doit pas porter sur ce que le locuteur *dit* mais

5. Site web de Gate : http://gate.ac.uk/

sur ce qu'il *veut dire*. Ainsi, un énoncé comme *« On y va ? »* est très certainement une suggestion plutôt qu'une question. Enfin, il a été demandé aux annotateurs d'assigner des fonctions communicatives aussi précises que possible pourvu qu'ils aient des preuves (contexte, linguistique, etc.) pour justifier le choix d'une fonction plus spécifique. Par exemple, une *CheckQuestion* est plus spécifique qu'une *PropositionalQuestion* (oui/non) dans le sens où elle attend une réponse positive. Dans le doute, les annotateurs ont été invités à sélectionner la fonction parente de la fonction spécifique.

Enfin, il a été demandé aux annotateurs de n'annoter que les segments pour lesquels ils avaient assez d'indices pour déterminer une fonction communicative, ainsi que les parties du dialogue relatives à la tâche de recherche d'information collaborative sur CISMeF.

Limites du processus d'annotation

Notre annotation admet des limites. Tout d'abord, nous prenons la liberté d'amoindrir la clause de minimalité dans la définition de segment fonctionnel en accord avec le guide d'annotation [Bunt 2010]. Ensuite, nous faisons abstraction de tout comportement non-verbal dans le corpus Cogni-CISMeF. Enfin, ce processus se limite à l'annotation de fonctions communicatives et délaisse les aspects sémantiques.

Segmentation Les segments fonctionnels que nous avons générés depuis les annotations ne respectent pas strictement la contrainte de minimalité afin de simplifier la segmentation du dialogue, en accord avec le guide d'annotation [Bunt 2010]. Nous avons enfreint la règle de minimalité des segments fonctionnels pour les courtes interventions dans les dimensions de *Own Communication Management* (OCM) (e.g., hésitations, auto-corrections, ...) et de *Time Management* (e.g., *« euh : »*). Lorsqu'un segment fonctionnel contenant une fonction communicative de ces dimensions est inclus dans un autre segment fonctionnel, alors les fonctions sont automatiquement attribuées au segment fonctionnel englobant en post-traitement (et le petit segment fonctionnel est supprimé). Ensuite, les segments fonctionnels chevauchants (cf. section 5.1.1) ont été fusionnés pour ne former qu'un seul segment. En outre, un segment fonctionnel peut intervenir sur plusieurs tours de parole (cf. dialogue 5.1, p. 108). Or, les annotations dans Gate sont nécessairement continues. Nous adoptons la convention du guide d'annotation qui indique de décomposer le segment fonctionnel multi-tour en autant de segments que de tours possédant les mêmes actes de dialogue.

Transcription Nous annotons la partie verbale retranscrite d'une interaction multimodale (cf. section 4.1.3). Nous n'avons donc pas en notre possession des indices non-verbaux tels que des mouvements de tête ou des expressions faciales. [Bunt 2009] précise que les comportements non-verbaux interviennent pour quatre buts principaux : (i) insister ou articuler le contenu sémantique des actes de dialogue, (ii) insister ou supporter les fonctions communicatives exprimées verbalement, (iii) réaliser un ou plusieurs actes de dialogue en parallèle de la contribution dialogique courante du partenaire (sans prise de tour de parole), ou (iv) exprimer une autre fonction communicative en parallèle de ce que le locuteur exprime verbalement. [Bunt 2009] précise que le dernier point intervient beaucoup plus rarement que les trois autres. Les points (i) et (ii) nous permettent de prendre conscience que nous perdons des indices permettant de choisir les fonctions communicatives. Tandis que les deux derniers points montrent que nous n'annotons pas

l'ensemble des fonctions communicatives exprimées, en particulier celles concernant la gestion de l'interaction (point (iii)).

Sémantique Le processus d'annotation se limite à l'*attribution de fonctions communicatives* à des segments fonctionnels. De ce fait, nous laissons de côté l'aspect sémantique qui nécessite à lui seul un travail hors de portée de cette thèse (voir les conclusions de [Loisel 2008] et le travail de [Prévot 2004]). Ainsi, le *contenu sémantique* des actes n'est pas annoté. De la même manière, nous n'annotons pas de relations entre les segments fonctionnels. Ce genre de relation a été mis en avant par les grammaires de dialogue [Polanyi 1984] ou encore par les relations rhétoriques de la SDRT [Asher 2003] (e.g., narration, conséquence, explication, justification, élaboration). La raison de cette éviction est que nous ne disposons pas de base tractable sur laquelle fonder l'annotation de relations. Par exemple, le schéma DIT++ ne propose actuellement aucune taxonomie claire des relations entre les segments.

5.1.3 Résultats du processus d'annotation

Cette section s'intéresse aux résultats chiffrés et commentés du processus d'annotation. Elle débute par une présentation de chiffres permettant de quantifier globalement le processus. Puis, elle présente les résultats de la confrontation des annotations réalisées par les deux annotateurs. Le calcul de l'accord inter-annotateur et ses résultats sont discutés. Ensuite, la constitution des tours de parole et des segments fonctionnels en terme de fonction communicative est étudiée. Puis, l'importance relative des dimensions observées dans le corpus est analysée. Enfin, nous entrons dans le détail des segments en étudiant les fonctions les plus courantes dans les principales dimensions.

Généralités

Les résultats ont été produits à l'aide de plugins Gate qui ont été développés pour l'occasion. Ces derniers se basent sur les annotations manuelles. Le tableau 5.3 présente le détail du nombre de fonctions communicatives annotées par corpus et par annotateur. Ce processus a conduit à l'annotation de 6343 fonctions communicatives par les couples d'annotateurs. L'annotateur 1 a annoté légèrement plus de fonction que l'annotateur 2 (51% du total). Le corpus VD regroupe approximativement 59% des annotations. Cela s'explique par le fait que les entretiens de ce corpus sont plus longs en terme de tours de parole.

	Corpus AL (10)	Corpus VD (8)	Total
G	1340 (51%)	1904 (51%)	3244 (51%)
A, J ou N	1285 (49%)	1814 (49%)	3099 (49%)
Total	2625	3718	**6343**

Tableau 5.3: Nombre de fonctions communicatives annotées par corpus et par annotateur

Le tableau 5.4 présente le nombre de tours de parole total par corpus ainsi que le nombre de tours de parole finalement annotés. Suite à la stratégie choisie (cf. section 5.1.2), les annotateurs avaient pour indication de ne pas considérer un tour de parole soit par manque d'indice rendant l'intervention d'un interlocuteur difficilement interprétable en dehors du contexte (9% du total des tours de parole n'ont pu être étiquetés pour cette raison), soit si les tours de parole n'étaient

pas en rapport avec l'utilisation de CISMeF (18% des tours de parole). Cette situation est principalement intervenue à la fin des dialogues du corpus VD. Le détail de ces comptes par entretien est disponible en annexe A.3.2. En somme, un total de 1056 tours de parole a été annoté par chaque annotateur représentant 73% du corpus initial. La majorité des tours non annotés a été volontairement exclue de par leur intervention dans une conversation libre hors du cadre CISMeF.

	Nombre de tours	Tours annotés	Hors CISMeF	Incompréhensible
Corpus AL	271	269 (99%)	0	2 (1%)
Corpus VD	1174	787 (67%)	266 (23%)	121 (10%)
Total	1445	1056 (73%)	266 (18%)	123 (9%)

Tableau 5.4: Proportion des tours de parole annotés dans le corpus. La colonne « Hors CISMeF » dénombre les tours non-annotés car ne faisant pas partie de la tâche de RI sur CISMeF. La colonne « Incompréhensible » décompte les tours de parole dont les indices étaient insuffisants pour déterminer des fonctions communicatives.

Le tableau 5.5 présente quelques exemples de fonctions représentatives du corpus. Il indique la fonction communicative, la dimension dans laquelle elle intervient et un exemple illustratif tiré du corpus. L'annexe A.3.1 présente cinq extraits de dialogue annotés. Ces exemples commentés illustrent les six dimensions suivantes : *Task*, *Auto-feedback*, *Allo-feedback*, *Time Management*, *Turn Management* et *Partner Communication Management*.

Accords inter-annotateur

Calcul de l'accord inter-annotateur Nous avons effectué une analyse de l'accord inter-annotateur (IAA) en terme de précision, rappel et F-mesure (F_1 score) [Rijsbergen 1979, Manning 1999]. Nous délaissons le coefficient Kappa (κ) [Cohen 1960, Carletta 1996] limité à une tâche d'annotation où le corpus a été pré-segmenté. L'annexe A.3.3 présente le détail des calculs de précision, rappel et F-mesure. Les résultats du calcul de l'IAA se divisent en trois catégories : *strict*, *lenient* et *average*. Celles-ci varient par leur façon de considérer les annotations qui se chevauchent sans se confondre : le mode *strict* considère les annotations qui se chevauchent comme incorrectes alors que le mode *lenient* les considère correctes. Le mode *average* est une moyenne des deux modes précédents. Le calcul de l'IAA pour la phase d'étiquetage ne prend pas en compte la taxonomie de DIT++ : une fonction `Check Question` et une fonction `Yes/No Question` sont considérées comme étant aussi différentes qu'un `Thanking` et une `Yes/No Question`. Des métriques taxonomiques ont été proposées pour DIT++ mais prennent uniquement en compte la phase d'étiquetage sans considérer la phase de segmentation [Geertzen 2009b]. En conclusion, les IAAs présentés ici peuvent être considérés comme ceux du pire des cas, sauf indication contraire. Tous les IAA ont été calculés en prenant les annotations de l'annotateur 1 comme l'ensemble de référence et les annotations de l'annotateur 2 comme l'ensemble de comparaison. Intervertir ces deux ensembles conduit à échanger précision et rappel. Ce calcul est réalisé par le plugin « Inter-Annotator Agreement » [6] préexistant dans Gate.

Nous nous intéressons dans un premier temps à l'IAA sur la tâche de segmentation seule. Puis, nous abordons le cas de l'IAA pour la combinaison des tâches de segmentation et d'étiquetage.

6. Documentation du plugin « Inter-Annotator Agreement » de Gate disponible à l'URL suivante : `http://gate.ac.uk/sale/tao/splitch10.html#sec:eval:iaaplugin`.

Dimension	Fonction communicative	Exemples
Task	*Inform*	*« on a 115 articles »* (AL07), *« on cherche mais on ne trouve pas toujours »* (VD05)
	Agreement	*« exactement c'est tout à fait ça »* (VD05)
	Suggestion	*« donc ce que je vous propose c'est qu'on remette effort du coup »* (VD05)
	Offer	*« vous avez une autre question ? »* (AL07) (ici, il s'agit d'une offre pour réaliser une nouvelle recherche)
	Request	*« essayez avec clairance »* (VD07)
	SetQuestion	*« comment on fait pour garder les documents ? »* (VD06)
	PropositionalQuestion	*« il y a pas un synonyme de cette maladie ? »* (AL03)
	CheckQuestion	*« vous voulez savoir les traitements de manière générale finalement ? »* (AL12)
	ChoiceQuestion	*« alors on a créatinine quinaz ou créatinine ? »* (VD07)
Time Management	*Stalling*	*« euh »*, *« bah »*
Auto-feedback	*AutoPositive*	*« OK »*, *« oui »*, *« mmhm mmhm »*
	PerceptNegativeAutoFB	*« pardon ? »* (VD07)
	ExecNegativeAutoFB	*« J'en sais rien »* (VD04)
OCM	*Retraction*	*« là on trouve de nombreux / beaucoup plus de choses »* (AL06)
	Self-correction	*« il me semble que c'était / ah non peut-être pas non »* (AL07)
Turn Management	*Turn Grab*	*« S : je me couche tôt et je suis dans mon lit et je dors pas et A : d'accord »* (AL04)
	Turn Assign	*« donc je vous écoute »* (AL03), *« allez-y »* (VD03)
Allo-feedback	*EvalFBElicitation*	*« . . . non ? »* (VD04), *« . . . c'est ça ? »* (VD05), *« . . . on est bien d'accord ? »* (VD05)
SOM	*Apology*	*« excusez-moi »* (VD05)
	Thanking	*« je vous remercie beaucoup »* (VD02)
	Thanking-downplay	*« nickel c'est moi »* (VD02)
PCM	*Completion*	*« S : si on fait ça je crois que ça va nous donner A : un document spécifique »* (AL10)

Tableau 5.5: Exemples de fonctions communicatives par dimension associées à un exemple typique. OCM = Own Communication Management, SOM = Social Obligation Management, PCM = Partner Communication Management.

Accord inter-annotateur sur la segmentation L'IAA sur la phase de segmentation est présenté dans le tableau 5.6. L'accord obtenu sur la phase de segmentation est *fort*. Nous obtenons en effet des scores supérieurs ou égaux à 0.93 dans chaque mode. Les IAA par corpus sont disponibles en annexe A.3.3. L'IAA sur la segmentation est légèrement meilleur sur le corpus VD (F-mesure à 0.97 en mode *average*) que sur le corpus AL (F-mesure à 0.94 en mode *average*). Cette différence peut s'expliquer par la longueur des tours de parole de l'expert AL, plus difficile à segmenter (cf. section 5.1.3).

Strict			Lenient			Average		
R	P	F	R	P	F	R	P	F
0.93	0.94	0.94	0.97	0.98	0.98	0.95	0.96	0.96

Tableau 5.6: Accord inter-annotateur pour la tâche de segmentation. R = Rappel, P = Précision, F = F-mesure (F_1 score).

Ces résultats nous permettent d'approfondir l'étude de l'accord en ajoutant la phase d'étiquetage.

Accord inter-annotateur sur la segmentation et l'étiquetage par dimension Le tableau 5.7 présente l'IAA par dimension pour l'étiquetage des fonctions communicatives. Les proportions de fonction par dimension sont également fournies. Nous revenons sur les dimensions en section 5.1.3. Notons que les dimensions sont ordonnées dans le tableau de résultats par proportion de fonction décroissante.

	Strict			Lenient			Average			
Dimension	R	P	F	R	P	F	R	P	F	Prop.
Task	0.84	0.85	0.84	0.86	0.87	0.87	0.85	0.86	0.86	68.30%
Time Management	0.75	0.8	0.77	0.81	0.86	0.83	0.78	0.83	0.8	9.93%
Auto-Feedback	0.77	0.8	0.79	0.8	0.83	0.81	0.79	0.81	0.8	9.44%
OCM	0.43	0.49	0.46	0.47	0.54	0.5	0.45	0.52	0.48	5.31%
Turn Management	0.37	0.73	0.49	0.41	0.8	0.54	0.39	0.76	0.51	2.76%
Allo-Feedback	0.63	0.54	0.58	0.66	0.56	0.61	0.64	0.55	0.59	1.19%
SOM	0.43	0.69	0.53	0.48	0.76	0.59	0.46	0.72	0.56	1.18%
PCM	0.86	0.91	0.89	0.86	0.91	0.89	0.86	0.91	0.89	1.09%
Discourse Structuring	0.67	0.57	0.62	0.75	0.64	0.69	0.71	0.61	0.65	0.41%
Contact Management	0.67	0.46	0.55	0.89	0.62	0.73	0.78	0.54	0.64	0.35%
Global	0.77	0.81	0.79	0.81	0.84	0.82	**0.79**	**0.83**	**0.81**	

Tableau 5.7: Accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension. R = Rappel, P = Précision, F = F-mesure (F_1 score). OCM = Own Communication Management, PCM = Partner Communication Management, SOM = Social Obligation Management.

Quatre dimensions se distinguent en terme de proportion de fonctions annotées : la dimension *Task* (68.30%), la dimension *Time Management* (9.93%), la dimension *Auto-Feedback* (9.44%) et celle de *Own Communication Management* (OCM, 5.31%). Nous obtenons un *accord fort* pour la dimension *Task* dont les scores dépassent 0.84 dans tous les modes. La F-mesure pour les autres dimensions (exceptée OCM) dépasse 0.8 (mode *average*) ce que nous qualifions d'*accord*

fiable. En outre, les scores en terme de rappel, précision et F-mesure de ces dimensions sont supérieurs à 0.75 dans tous les modes. Nous revenons sur le cas de la dimension OCM dans la suite de cette section.

Les accords sur les autres dimensions sont plus variés. Les dimensions *Allo-feedback*, *Contact Management* et *Discourse Structuring* entretiennent des F-mesures comprises entre 0.59 et 0.65 (mode *average*). Les accords dans ces dimensions sont *modérés*. Ils sont pénalisés par des scores de précision relativement faibles (compris entre 0.55 et 0.61 en mode *average*). Les dimensions *Social Obligation Management* (SOM) et *Turn Management* possèdent des F-mesures situées entre 0.51 et 0.56 (mode *average*). Les accords dans ces dimensions sont *modérés bas*. Notons que ces dimensions possèdent une forte précision (supérieure à 0.69 dans tous les modes) et sont pénalisées par un faible rappel (0.39 et 0.46 en mode *average*). Enfin, notons l'accord fort obtenu par la dimension *Partner Communication Management* (PCM) dont la principale fonction communicative est la complétion de la locution du partenaire (cf. dialogue 5.1, p. 108). Les scores de cette dimension dépassent 0.86 dans tous les modes.

Il reste la dimension *OCM* dont les scores démontrent un accord *faible* équilibré entre rappel et précision.

Impact de la propriété taxonomique de DIT++ Comme nous l'avons précédemment évoqué, le calcul de l'IAA ne prend pas en compte les propriétés taxonomiques de DIT++. Pour étudier l'impact de ces propriétés, nous avons re-calculé l'IAA en considérant que toutes les fonctions partageant la même racine sont équivalentes (cf. figures 1.1 et 1.2, p. 25). Notons que cette méthode réalise des équivalences abusives. Une fonction `Correction` est ainsi équivalente à un `Answer`. Cependant, les fonctions de la dimension *OCM* forment une branche taxonomique de trois fonctions se spécialisant. Les considérer comme équivalentes revient à ne donner aucune importance au degré de spécificité choisi lors de l'étiquetage de ces fonctions. Les résultats de ce nouveau calcul sont présentés en annexe A.9. Trois dimensions voient leur score augmenter (les autres restant inchangées). La dimension *Task* et *Allo-feedback* augmentent légèrement leur F-mesure de 0.02 point (en mode *average*). La dimension *OCM* voit ses scores considérablement croître. En mode *average*, sa précision passe à 0.62, son rappel à 0.72 et sa F-mesure à 0.67 atteignant ainsi un accord *modéré haut*. L'accord faible initialement obtenu est donc principalement dû au degré de spécificité choisi dans l'étiquetage des fonctions.

Accord inter-annotateur sur les fonctions générales Rappelons que les fonctions générales peuvent intervenir dans toutes les dimensions et en particulier dans la dimension *Task*. Le calcul de cet accord permet de vérifier l'uniformité de l'IAA de la dimension *Task* par rapport aux catégories de fonctions y intervenant. Le tableau 5.8 présente le calcul de l'IAA en ne prenant en compte que les *fonctions générales* de DIT++. Nous avons regroupé ces fonctions dans les catégories initiales de la taxonomie : fonctions de recherche d'information (« information-seeking »), fonctions fournissant de l'information (« information-providing ») et fonctions de discussion d'actions (« action-discussion »). L'accord sur les fonctions fournissant de l'information est *fort*. Les scores sont tous supérieurs à 0.84 et la F-mesure en mode *average* est de 0.87. Les deux autres catégories présentent un accord *fiable* avec des scores globalement supérieurs à 0.75. En outre, la F-mesure est de 0.81 pour les fonctions de discussion d'action et de 0.77 pour les fonctions de recherche d'information.

	Strict			Lenient			Average		
Catégories	R	P	F	R	P	F	R	P	F
Action-discussion	0.8	0.79	0.8	0.83	0.82	0.83	0.82	0.8	0.81
Information-providing	0.84	0.88	0.86	0.87	0.9	0.88	0.85	0.89	0.87
Information-seeking	0.75	0.75	0.75	0.78	0.78	0.78	0.77	0.76	0.77

TABLEAU 5.8: Accord inter-annotateur pour les tâches de segmentation et d'étiquetage pour les fonctions générales. R = Rappel, P = Précision, F = F-mesure (F_1 score).

Accord inter-annotateur sur les corpus AL et VD Nous avons également souhaité savoir si des disparités existaient entre l'annotation du corpus AL et celle du corpus VD. L'annexe A.3.3 présente les tableaux de l'IAA par dimension pour le corpus AL et pour le corpus VD. L'accord global sur le corpus VD est *fiable*. La F-mesure en mode *average* est de 0.82 et les scores sont tous supérieurs à 0.8. De même, l'accord sur le corpus AL est *fiable*. La F-mesure en mode *average* est de 0.77 et les scores sont tous supérieurs à 0.72. Les accords sur les deux corpus sont fiables avec un léger avantage au corpus VD.

Analyse au niveau des tours de parole et des segments fonctionnels

Dans cette section, nous analysons les résultats du processus d'annotation en terme d'unités d'interaction produites, i.e. en terme de segments fonctionnels. Puis nous tentons de faire ressortir les caractéristiques du corpus qui apparaissent au niveau des tours de parole.

Analyse au niveau des segments fonctionnels 5486 segments fonctionnels ont été annotés dans le corpus COGNI-CISMEF dont 58.9% appartiennent au corpus VD (3232), et 41.1% au corpus AL (2254). Nous avons cherché à connaître le nombre moyen de fonctions communicatives par segment fonctionnel pour la stratégie choisie et la définition de segment fonctionnel choisie (cf. section 5.1.2). Le tableau 5.9 présente la synthèse des résultats par annotateur et par corpus ainsi que le résultat global. L'annexe A.3.4 présente le détail des données et des calculs par annotateur, par corpus et par entretiens. Un segment fonctionnel est en moyenne constitué de 1.16 fonctions communicatives sans réelle différence entre le corpus AL et VD. Nous pouvons noter que l'annotateur 1 possède un ratio légèrement supérieur à celui de l'annotateur 2. Cela est cohérent avec la différence observée dans le nombre de fonctions annotées. Ces résultats sont légèrement inférieurs à ceux obtenus par [Bunt 2011b] pour la même stratégie. Le ratio obtenu est de 1.3. La principale explication de la différence est que nous manquons la partie non-verbale, absente dans le cas de [Bunt 2011b] (leurs dialogues ne sont pas en situation de co-présence). Néanmoins, ces résultats restent très proches. En outre, un tour de parole est en moyenne constitué de 2.60 segments fonctionnels dans notre corpus. Ceci nous permet de confirmer l'hypothèse de multifonctionnalité défendue dans [Bunt 2011b].

Annotateur	1	2	Total
Corpus AL	1.19	1.14	1.16
Corpus VD	1.17	1.13	1.15
Total			1.16

TABLEAU 5.9: Nombre de fonctions par segment fonctionnel par annotateur, par corpus et au total

Nous nous sommes ensuite intéressés à la répartition des segments fonctionnels entre l'expert et le demandeur. Le tableau 5.10 présente la répartition des segments fonctionnels entre les interlocuteurs pour chaque corpus et au total. Sur l'ensemble du corpus, l'activité communicative de l'expert est plus intense que l'activité du demandeur (trois quarts des segments fonctionnels). Dans le détail, on note une différence claire entre le corpus AL et le corpus VD. L'expert du corpus AL produit un peu plus de 80% des segments fonctionnels alors que celui du corpus VD n'en produit que les deux tiers. La différence entre l'expert et le demandeur s'explique par le fait que l'expert est chargé de mener la recherche tout en verbalisant ses actions. La différence entre les deux experts s'explique, quant à elle, par le fait que l'expert AL est beaucoup plus directif que l'expert VD.

Rôle	AL	VD	Total
Demandeur	17.10%	31.40%	25.50%
Expert	82.90%	68.60%	74.50%

Tableau 5.10: Répartition des segments fonctionnels en fonction du rôle

Analyse au niveau des tours de parole Comme précédemment évoqué, 1056 tours de parole ont été annotés dans ce processus par chaque annotateur (1 et 2). 269 tours ont été annotés dans le corpus AL contre 787 dans le corpus VD. Les entretiens du corpus AL sont constitués en moyenne de 26.9 tours de parole (écart type de 11.5). Les entretiens du corpus VD sont constitués en moyenne de 98.4 tours de parole (écart type de 45.6). Le détail du nombre de tours de parole par entretien est donné en annexe A.3.2. Les entretiens du corpus AL sont en moyenne plus court que les entretiens du corpus VD.

Afin de caractériser plus précisément l'activité communicative de l'expert et du demandeur, nous avons étudié le nombre d'actes de dialogue en moyenne par tour et par corpus. Le tableau 5.11 présente le nombre d'actes de dialogue par tour en fonction du corpus, de l'annotateur et du rôle. Les tours de parole de l'expert AL contiennent en moyenne beaucoup plus d'actes avec un fort écart type (6.88 actes en moyenne pour un écart type de 7.26). Les tours de parole de l'expert VD contiennent en moyenne moins d'actes que l'expert AL (2.80) avec un écart type plus faible (2.68). Le nombre d'actes par tour pour le demandeur est dans le même ordre de grandeur dans les deux corpus et se situe entre 1 et 1.5 actes en moyenne pour un faible écart type.

(a) Corpus AL

Annotateur	1	2	Total
Expert ($\overline{m}$)	6.88	6.87	6.88
Demandeur ($\overline{m}$)	1.46	1.47	1.47
Expert (Δ)	7.20	7.33	7.26
Demandeur (Δ)	0.87	1.00	0.94

(b) Corpus VD

Annotateur	1	2	Total
Expert ($\overline{m}$)	2.83	2.77	2.80
Demandeur ($\overline{m}$)	1.30	1.28	1.29
Expert (Δ)	2.71	2.65	2.68
Demandeur (Δ)	0.69	0.7	0.69

Tableau 5.11: Nombre d'actes de dialogue par tour en fonction du rôle. $\overline{m}$ = moyenne, Δ = écart type.

Analyse au niveau des dimensions

Le tableau 5.12 présente la proportion de fonctions annotées par dimension sur l'ensemble du corpus Cogni-CISMeF. Quatre dimensions se distinguent en terme de proportion de fonctions annotées : la dimension *Task* (68.30%), la dimension *Time Management* (9.93%), la dimension *Auto-Feedback* (9.44%) et celle de *Own Communication Management* (OCM, 5.31%). Deux tiers des fonctions ont pour but d'avancer la tâche sous-jacente au dialogue tandis qu'un tiers concerne la gestion des processus d'interaction. Sur ces quatre dimensions, deux sont liées à des activités *monologiques* concernant les processus d'élocution du locuteur. Il s'agit des dimensions *OCM* (e.g., auto-corrections, hésitations) et *Time Management* (e.g., pauses). La dimension *Auto-feedback* a pour vocation de gérer les retours sur les processus de compréhension des énoncés à différents niveaux (attention, perception, interprétation, évaluation, exécution).

Dimension	Proportion
Task	68.30%
Time Management	9.93%
Auto-Feedback	9.44%
OCM	5.31%
Turn Management	2.76%
Allo-Feedback	1.19%
SOM	1.18%
PCM	1.09%
Discourse Structuring	0.41%
Contact Management	0.35%

Tableau 5.12: Proportions de fonctions annotées par dimension sur l'ensemble du corpus Cogni-CISMeF. OCM = Own Communication Management, PCM = Partner Communication Management, SOM = Social Obligation Management.

L'annexe A.3.5 contient les données sur la proportion de fonctions annotées par dimension, par corpus et par annotateur, desquelles est issu le tableau 5.12. Les corpus AL et VD font ressortir les quatre mêmes dimensions majoritaires sans variation significative entre les annotateurs. Une différence notable entre les deux corpus concerne la dimension *Time Management*. Cette dernière est nettement plus présente dans le corpus AL (17.29%) que dans le corpus VD (4.74%). Ce phénomène s'explique par les longs tours de parole de l'expert AL qui impose l'usage de segments pour combler les temps d'attente (e.g., *« euh »*, *« alors »*). En outre, le corpus VD possède proportionnellement plus de fonctions dans la dimension *Auto-feedback* (11.83%) que le corpus AL (6.07%). Cela s'explique par les interventions plus fréquentes du demandeur dans ce corpus qui nécessitent l'usage de feedbacks afin d'assurer la compréhension mutuelle. Les deux autres dimensions principales (*Task* et *OCM*) interviennent dans les mêmes proportions.

Nous nous sommes intéressés à la co-occurrence des dimensions dans le corpus Cogni-CISMeF. Le tableau A.16 en annexe A.3.6 présente la matrice de co-occurrence des dimensions. Pour les cinq dimensions principales, nous obtenons les co-occurrences suivantes (ordonnées par proportion d'occurrence décroissante) :

Task : *Own Communication Management*, *Time Management*, *Auto-feedback*, *Turn Management*.

Time Management : *Task*, *Turn Management*, *Own Communication Management*, *Auto-feedback*.

Auto-feedback : *Task*, *Partner Communication Management*, *Turn Management*, *Time Management*.

Own Communication Management : *Task*, *Time Management*, *Turn Management*.

Turn Management : *Task*, *Time Management*, *Own Communication Management*, *Partner Communication Management*.

Une fonction de la dimension *Task* co-occurre principalement avec des fonctions permettant de gérer l'élocution ou le tour de parole. Il s'agit des dimensions *OCM*, *Time Management* et *Turn Management*. Elle co-occurre également avec la production de feedback (dimension *Auto-feedback*). Un élément notable est la co-occurrence des dimensions *Auto-feedback* et *Turn Management* avec la dimension *PCM*. Cette dernière dimension contient des fonctions permettant de récupérer le tour de parole en aidant un partenaire à conclure son élocution. Les interlocuteurs saisissent cette occasion pour produire des feedbacks (i.e., compléter pour montrer la compréhension). De manière intéressante, nous obtenons des co-occurrences de dimensions similaires au corpus AMI [Bunt 2009] (voir annexe A.3.6).

Analyse au niveau des fonctions communicatives

Dans cette section, nous nous intéressons aux fonctions intervenant dans les dimensions principales (*Task*, *Time Management*, *Auto-feedback* et *OCM*). L'annexe A.3.7 présente les données pour les autres dimensions.

Le tableau 5.13 présente les proportions de fonctions annotées par catégorie de la taxonomie DIT++. Sans surprise, nous retrouvons en tête les catégories de fonctions générales fournissant de l'information, de discussion d'action et, en cinquième position, de demande d'information. Ces fonctions interviennent en grande majorité dans la dimension *Task* qui est la dimension la plus représentée. Ensuite, nous retrouvons en bonne place les fonctions spécifiques aux dimensions principales observées en section 5.1.3. Il s'agit des catégories de *Time Management*, d'*Auto-feedback*, d'*OCM* et de *Turn Management*. En queue de tableau, nous observons les fonctions des dimensions minoritaires. Sur les 88 fonctions communicatives disponibles dans DIT++, 64 fonctions ont été utilisées dans notre annotation. Une grande partie des fonctions non-utilisées sont les fonctions dans les dimensions *Auto-* et *Allo-feedback* (que nous traitons dans la suite du document).

Dimension *Task* La dimension *Task* représente approximativement deux tiers du total des fonctions communicatives annotées. En tout, 4333 fonctions ont été annotées dans cette dimension. Sur ce total, 3180 ont été produites par l'expert et 1153 par le demandeur.

Nous analysons les fonctions produites par les interlocuteurs au regard des trois catégories de fonctions générales : les fonctions fournissant de l'information (« information-providing »), les fonctions de recherche d'information (« information-seeking ») et les fonctions de discussion d'action (« action-discussion »). Le tableau 5.14 présente les proportions de fonctions par catégorie pour l'expert et pour le demandeur. Les catégories affichent des proportions semblables pour les deux rôles. La catégorie des fonctions fournissant de l'information arrive en tête avec un peu plus de 60% des fonctions. Elle est suivie par la catégorie de discussion d'action qui

Catégorie	Proportion
Information-providing	44.00%
Action-discussion	20.90%
Time management	9.80%
Auto-feedback	8.80%
Information-seeking	5.50%
OCM	5.30%
Turn management	2.80%
PCM	1.10%
Allo-feedback	0.60%
SOM	0.60%
Contact management	0.30%
Discourse Structure	0.30%

Tableau 5.13: Proportion de fonctions annotées par catégories

(a) Expert

Catégorie	Proportion
Information-providing	61.23%
Action-discussion	31.60%
Information-seeking	7.17%

(b) Demandeur

Catégorie	Proportion
Information-providing	65.74%
Action-discussion	27.06%
Information-seeking	7.20%

Tableau 5.14: Répartition des fonctions dans la dimension *Task* par catégorie pour l'expert et le demandeur.

s'établit alentour des 30%. Enfin, la catégorie de recherche d'information affiche une proportion de fonctions autour des 7%.

Les tableaux 5.15 présentent les fonctions fournissant de l'information apparaissant dans la dimension *Task* pour l'expert et le demandeur. Tout d'abord, les mêmes fonctions interviennent dans un ordre identique entre l'expert et le demandeur. La catégorie est dominée par la fonction *Inform*. S'ensuit la fonction *Agreement* permettant d'affirmer un accord. On trouve par la suite les fonctions de réponses aux questions (*Answer*, *Confirm*, *Disconfirm*). Enfin, la queue de la catégorie est constituée des fonctions de désaccord (*Correction* et *Disagreement*) dont la proportion est faible.

(a) Expert

Fonction	Prop. rel.	Prop. abs.
Inform	91.53%	38.20%
Agreement	3.85%	1.60%
Answer	2.16%	0.90%
Disconfirm	1.03%	0.40%
Confirm	0.87%	0.40%
Correction	0.51%	0.20%
Disagreement	0.05%	0.00%

(b) Demandeur

Fonction	Prop. rel.	Prop. abs.
Inform	65.96%	29.80%
Agreement	11.21%	5.10%
Answer	10.16%	4.60%
Confirm	7.65%	3.50%
Disconfirm	3.30%	1.50%
Disagreement	0.92%	0.40%
Correction	0.79%	0.40%

TABLEAU 5.15: Répartition des fonctions dans la dimension *Task* pour la catégorie des fonctions fournissant de l'information. La proportion relative (« Prop. rel. ») désigne la proportion de la fonction dans la catégorie à laquelle elle appartient. La proportion absolue (« Prop. abs. ») désigne la proportion de la fonction dans l'ensemble des fonctions observées produites en fonction du rôle (expert ou demandeur).

Le nombre élevé d'*Inform* aussi bien pour l'expert que pour le demandeur mérite un éclaircissement. Plusieurs raisons expliquent ce résultat. La première raison provient de la segmentation. Nous avons choisi comme unité le segment fonctionnel dont une des propriétés est la minimalité. Ainsi, une longue assertion est souvent ségmentée en plusieurs unités informatives (e.g., l'assertion *« j'aimerais savoir enfin par rapport au sommeil / donc je suis régulièrement fatigué et je n'arrive pas forcément à dormir par rapport à cette fatigue / je me couche tôt et je suis dans mon lit et je dors pas »* (AL04) conduit à l'obtention de quatre segments avec une fonction *Inform*). Ce point fait débat dans la communauté [Larsson 1998]. Nous avons choisi de respecter la minimalité des segments pour notre tâche. Ensuite, l'expert doit verbaliser le déroulement de l'expérimentation. Cela le conduit à informer de la requête en cours et des opérations qu'il réalise. Enfin, de nombreuses phases de la recherche d'information contiennent des interventions informatives. C'est le cas des phases de verbalisation et d'évaluation des ressources. En outre, l'expert a un rôle informatif envers le demandeur au regard de l'outil CISMeF. Toutes ces raisons contribuent à la proportion élevée d'actes *Inform*.

Les tableaux 5.16 présentent les fonctions de discussion d'action pour l'expert et le demandeur. Ils mettent en évidence la prédominance de la fonction *Suggestion* pour les deux rôles. C'est une fonction *directive* permettant d'engager l'allocutaire ou les deux interlocuteurs sur la réalisation d'une action (e.g., *« On y va ? »*). Cette fonction est utilisée pour proposer des tactiques de modification de la requête (e.g., *« on va rentrer comme mot-clé troubles de l'alimentation »* (AL07)). Elle est particulièrement marquée au niveau de surface des énoncés par l'usage du « on » et du « nous ».

(a) Expert

Fonction	Prop. rel.	Prop. abs.
Suggestion	61.99%	13.40%
Offer	14.03%	3.00%
AcceptSuggestion	7.06%	1.50%
Request	4.88%	1.10%
AcceptRequest	4.28%	0.90%
Instruct	3.58%	0.80%
Promise	3.08%	0.70%
DeclineSuggestion	0.70%	0.20%
AcceptOffer	0.20%	0.00%
AddressRequest	0.10%	0.00%
DeclineRequest	0.10%	0.00%

(b) Demandeur

Fonction	Prop. rel.	Prop. abs.
Suggestion	34.29%	6.40%
AcceptSuggestion	25.00%	4.60%
Request	18.27%	3.40%
AcceptOffer	8.01%	1.50%
DeclineOffer	4.81%	0.90%
AcceptRequest	3.21%	0.60%
DeclineSuggestion	1.92%	0.40%
AddressSuggestion	1.28%	0.20%
AddressOffer	0.96%	0.20%
AddressRequest	0.96%	0.20%
Offer	0.64%	0.10%
DeclineRequest	0.32%	0.10%
Instruct	0.32%	0.10%

TABLEAU 5.16: Répartition des fonctions dans la dimension *Task* pour la catégorie des fonctions de discussion d'actions. La proportion relative (« Prop. rel. ») désigne la proportion de la fonction dans la catégorie à laquelle elle appartient. La proportion absolue (« Prop. abs. ») désigne la proportion de la fonction dans l'ensemble des fonctions observées produites en fonction du rôle (expert ou demandeur).

Les fonctions de discussion d'action utilisées par l'expert sont majoritairement des initiatives (*Suggestion*, *Offer*, *Request*, *Instruct* et *Promise*) aussi bien pour l'expert AL que VD. Elles représentent 87.56% des usages dans cette catégorie. Dans une minorité de cas (12.44%), l'expert emploie des fonctions réactives (*AcceptSuggestion*, *DeclineSuggestion*, *AcceptOffer*, *Accept/Decline/Address Request*). En somme, l'expert est principalement une force de proposition d'action.

On pourrait s'attendre à un demandeur principalement réactif à la vue des résultats de l'expert. Ce n'est pas ce que nous avons observé. Le demandeur équilibre ses contributions entre fonctions initiatives (53.52%) et réactives (47.43%). Ainsi, le demandeur propose approximativement autant qu'il réagit. En outre, son comportement réactif est caractérisé par une large variété de fonctions réactives (*Accept/Decline/Address Suggestion*, *Accept/Decline/Address Request*, *Accept/Decline/Address Offer*). Soulignons néanmoins la très faible présence des fonctions *AddressSuggestion*, *AddressRequest* et *AddressOffer* qui représentent moins de 1% des fonctions du demandeur. Son comportement initiatif est caractérisé par l'usage de suggestions et de requêtes (via les fonctions *Suggestion* et *Request*).

Toutes les suggestions d'action ne sont pas verbalement validées. Ainsi, le corpus fait intervenir plus de fonctions initiatives *Suggestion* que de fonctions réactives (*Accept/Decline/Address Suggestion*). Le tableau A.19 en annexe A.3.7 présente le détail des fonctions considérées par l'expert et le demandeur. Sur les 623 suggestions de l'expert, seules 88 sont verbalement considérées. Ce faible nombre peut s'expliquer par (i) l'absence des indices non verbaux (e.g., la suggestion est acceptée ou refusée par un mouvement de la tête) (ii) la position dominante de l'expert dans l'expérimentation qui suggère une action et la réalise en supposant l'accord du demandeur. Au contraire, il est intéressant de remarquer que les suggestions du demandeur sont verbalement considérées par l'expert dans 73% des cas. Cette proportion monte à 79% des cas

pour les requêtes.

Les tableaux 5.17 présentent les fonctions de recherche d'information pour l'expert et le demandeur. Ils font apparaître dans les deux cas une majorité de questions à *réponse déterminée*. En premier lieu arrivent les questions de vérification (*CheckQuestion*, *PosiCheck* et *NegaCheck*) suivies des questions à choix multiples et des questions oui/non. Les *questions ouvertes* (qui, que, quoi, où, quand, comment, etc.) constituent un peu plus de 25% des questions de l'expert et 31% des questions du demandeur.

(a) Expert

Fonction	Prop. rel.	Prop. abs.
Check Question	48.25%	2.40%
Set Question	25.88%	1.26%
Choice Question	8.77%	0.40%
Nega-check	6.58%	0.30%
Propositional Q.	6.58%	0.30%
Posi-check	3.95%	0.20%

(b) Enquêté

Fonction	Prop. rel.	Prop. abs.
Check Question	57.83%	2.90%
Set Question	31.33%	1.55%
Choice Question	3.61%	0.20%
Propositional Q.	4.82%	0.20%
Posi-check	2.41%	0.10%

TABLEAU 5.17: Répartition des fonctions dans la dimension *Task* pour la catégorie des fonctions de recherche d'information. La proportion relative (« Prop. rel. ») désigne la proportion de la fonction dans la catégorie à laquelle elle appartient. La proportion absolue (« Prop. abs. ») désigne la proportion de la fonction dans l'ensemble des fonctions observées produites en fonction du rôle (expert ou demandeur).

Dimension *Auto-feedback* L'annotation de fonctions dans la dimension *Auto-feedback* a mis en avant la difficulté d'analyse des niveaux des processus impliqués (attention, perception, compréhension, évaluation, exécution) à partir des indices de surface des énoncés. Il est en effet difficile de dire à quel niveau agissent exactement des interventions comme *« ok »*, *« oui »*, ou encore *« mmh »*.

Le tableau 5.18 présente une synthèse pour la dimension *Auto-feedback* dont les détails sont donnés en annexe A.3.7. Il regroupe en quatre catégories les fonctions intervenant dans cette dimension : la catégorie *AutoPositive* (représentant la fonction du même nom), les fonctions générales, les fonctions négatives (fonctions spécifiques de la dimension exprimant un problème de communication) et les fonctions positives (fonctions spécifiques de la dimension témoignant d'un succès de compréhension, *AutoPositive* exclu). Sur l'ensemble du corpus, 90% des fonctions annotées dans cette dimension sont des *AutoPositive*. Cette fonction prévue dans la taxonomie DIT++ est un feedback positif sous-spécifié ne précisant pas le niveau de retour. Ce résultat implique deux conclusions. La première est que l'existence des niveaux de communication ne peut pas être empiriquement démontrée depuis l'annotation de notre corpus. Notons néanmoins la présence de signaux négatifs aisément identifiables comme un *« Je ne sais pas »* après une question (*ExecNegativeAutoFB*) ou un *« Pardon ? »* (*PerceptNegativeAutoFB*). Dans tous les cas, nos résultats attestent de l'existence de signaux de retour. Ensuite, notre corpus fait apparaître une communication de bonne qualité entre nos interlocuteurs comme noté par [Loisel 2008]. Les retours négatifs ne recouvrent en effet que 2.7% des fonctions de la dimension *Auto-feedback* et 0.2% des fonctions totales (contre 8.4% pour les *AutoPositive*). Cette observation est supportée par le faible nombre de fonctions annotées dans la dimension *Allo-feedback* qui va de pair avec la dimension *Auto-feedback* lors de mauvaise compréhension. À ces signaux négatifs peuvent

s'ajouter les fonctions générales qui consistent en des questions de vérification. L'un dans l'autre, les retours négatifs n'atteignent que 0.70% du total de fonctions.

Fonction	Prop. relative	Prop. absolue
AutoPositive	89.86%	8.40%
Fonctions générales	5.74%	0.50%
Fonctions négatives	2.70%	0.20%
Autres fonctions positives	1.69%	0.10%

TABLEAU 5.18: Synthèse des fonctions annotées dans la dimension *Auto-feedback*. La proportion relative désigne la proportion de la fonction dans la catégorie à laquelle elle appartient. La proportion absolue désigne la proportion de la fonction dans l'ensemble des fonctions observées produites.

Autres dimensions principales Le cas des dimensions *Time Management* et *Own Communication Management* sont plus simples car elles n'impliquent que peu de fonctions. L'annexe A.3.7 présente les données en détail pour chacune de ces dimensions.

La dimension *Time Management* fait apparaître un usage massif de la fonction *Stalling* qui permet au locuteur de gagner du temps pour formuler sa contribution.

La dimension *Own Communication Management* met en avant deux fonctions : *Retraction* et sa spécialisation *Self-correction*. La première permet au locuteur de retirer une partie de ce qu'il a dit dans le même tour de parole (e.g., hésitations). La seconde permet au locuteur de corriger une erreur faite dans le même tour (e.g., auto-correction).

5.1.4 Synthèse

Nous avons étudié notre corpus en terme d'unités d'interaction multifonctionnelles : les *segments fonctionnels*. Ces derniers ont émergé d'une tentative de doter l'expression « énoncé » d'une définition rigoureuse [Bunt 2011b]. La production de segments fonctionnels dans notre corpus est réalisée au travers du *processus d'annotation*. Celui-ci est constitué de deux parties : la *segmentation* (i.e., l'identification des unités d'interaction dans les tours de parole) et l'*étiquetage* (i.e., l'attribution d'une ou plusieurs fonctions communicatives aux segments). Nous avons adopté la stratégie d'étiquetage de fonctions communicatives strictement basée sur des indicateurs (marqueurs de surface, historique du dialogue, etc.). L'annotation a été réalisée en utilisant la taxonomie DIT++ au moyen de l'outil d'annotation Gate [Cunningham 2011]. Quatre annotateurs ont travaillé sur cette tâche. Les 18 entretiens de notre corpus ont chacun été annotés par deux annotateurs.

Ce processus a conduit à l'étiquetage de fonctions communicatives par les deux annotateurs sur les segments fonctionnels couvrant 73% du corpus initial. La majorité des tours non annotés a été volontairement exclue de par son intervention dans une conversation libre hors du cadre CISMeF. Compte tenu de la décomposition en segmentation et étiquetage de la tâche d'annotation, nous avons exploré la mesure de l'accord entre les annotateurs (IAA) en terme de précision, rappel et F-mesure (F_1 score). Globalement, l'accord obtenu entre les annotateurs est *fiable*. En particulier, l'accord sur la segmentation est *fort*. L'IAA par dimension sur la tâche d'étiquetage révèle un accord *fiable* pour les dimensions principales[7] qui représentent 93% des fonctions communicatives annotées. La dimension *Task*, qui représente deux tiers des fonctions

7. Il s'agit des dimensions *Task*, *Time Management*, *Auto-feedback*, *Own Communication Management*.

annotées, s'illustre avec un accord *fort*. Ce dernier est uniformément réparti sur l'ensemble des catégories de fonctions communicatives intervenant dans cette dimension.

Globalement, les résultats de notre tâche d'annotation permettent de confirmer la multifonctionnalité des énoncés aussi bien pris au sens de segment fonctionnel qu'au sens de tour de parole. L'analyse en terme de dimension permet d'affirmer que deux tiers des fonctions ont pour but d'avancer la tâche sous-jacente du dialogue tandis qu'un tiers concerne la gestion des processus d'interaction. En outre, quatre dimensions se distinguent en terme de proportion de fonctions annotées : la dimension *Task*, la dimension *Time Management*, la dimension *Auto-Feedback* et celle de *Own Communication Management*. Une étude de la co-occurrence des dimensions a montré qu'une fonction de la dimension *Task* co-occurre principalement avec des fonctions permettant de gérer l'élocution du locuteur, de produire des feedback ou de gérer les tours de parole. Enfin, l'annotation de fonctions dans la dimension *Auto-feedback* a mis en avant la difficulté d'analyse des niveaux des processus impliqués (attention, perception, compréhension, évaluation, exécution) à partir des indices de surface des énoncés. La quasi-totalité des fonctions annotées dans cette dimension sont des feedbacks positifs sous-spécifiés. Notre étude atteste néanmoins l'existence de signaux de retour et souligne la difficulté d'y attribuer un niveau. Notre corpus comporte majoritairement des signaux de retour positif confortant l'observation de [Loisel 2008].

Notre analyse a permis de souligner des différences entre le corpus AL et le corpus VD. La première différence concerne la longueur des entretiens. Ceux du corpus AL sont en moyenne trois à quatre fois plus courts que les entretiens du corpus VD en terme de tours de parole. La seconde différence provient des experts. L'expert AL est communicativement plus actif que l'expert VD en terme de nombre de segments fonctionnels produits par rapport au demandeur ou en terme de nombre d'actes de dialogue par tour de parole.

Au-delà des différences entre les experts AL et VD, nous avons observé que l'expert est communicativement plus actif que le demandeur. Cette observation se base aussi bien sur le nombre de segments fonctionnels produits que sur le nombre d'actes de dialogue en moyenne par tour. Une analyse des fonctions de la dimension *Task* a permis de voir que l'expert et le demandeur utilisent proportionnellement les mêmes catégories de fonctions. La différence principale survient dans la catégorie des fonctions de discussion d'action. L'expert produit majoritairement des fonctions initiatives (suggestion, offre et requête). Le demandeur, quant à lui, est équilibré entre les fonctions initiatives (suggestion et requête) et réactives (acception ou refus).

Outre les limites citées p. 111, la limite inhérente à ce type d'analyse est qu'elle est faite *a posteriori*. Malgré les précautions prises, nous ne pouvons pas exclure un biais dans l'interprétation des énoncés dû à la possession de la totalité du dialogue, et en particulier de la réponse suivant l'énoncé en cours d'interprétation. Il est néanmoins difficile de se défaire de cette limite.

Le processus d'annotation a conduit à la représentation du corpus Cogni-CISMeF sous la forme de séquences d'unité d'interaction multidimensionnelle. C'est cette représentation qui est exploitée pour l'extraction de motifs dialogiques.

5.2 Processus d'extraction de motifs dialogiques

Cette section présente le processus d'extraction de motifs dialogiques réalisé sur une partie du corpus Cogni-CISMeF. Nous abordons dans un premier temps la mise en place du processus d'extraction (cf. section 5.2.1). Ensuite, nous présentons les motifs dialogiques obtenus (cf. section 5.2.2). Enfin, nous synthétisons les conclusions de ce processus d'annotation (cf. section 5.2.3).

5.2.1 Configuration du processus d'extraction

Dans cette section, nous présentons la configuration du processus d'extraction de motifs. Nous commençons par décrire la constitution d'un corpus de référence et d'un corpus d'extraction. Puis nous justifions la restriction de notre champ d'étude à la dimension *Task*. Enfin, nous présentons le processus d'extraction de motifs dialogiques.

Corpus de référence et corpus d'extraction

Avant de débuter le processus d'extraction, nous avons constitué un *corpus de référence* en sélectionnant aléatoirement un tiers du corpus. Ce dernier contient 6 entretiens sur 18 dont trois entretiens du corpus AL (AL02, AL05, AL09) et trois entretiens du corpus VD (VD03, VD08, VD09). Les douze autres entretiens forment le *corpus d'extraction* utilisé pour le processus d'extraction. Le corpus de référence est exploité à des fins de validation, notamment en terme de couverture des jeux de dialogue modélisés depuis les motifs dialogiques extraits lors du processus d'extraction. En conséquence, il a été analysé après avoir proposé le modèle de jeux de dialogue.

Choix de la dimension *Task*

Nous concentrons nos efforts sur l'étude de la seule dimension *Task*. Ceci est rendu possible grâce à la propriété de *partialité* du schéma d'annotation DIT++ (cf. section 5.1.2). Chaque dimension a été conçue de manière à ne représenter qu'une activité bien définie du dialogue, dont l'existence empirique a été démontrée et dont l'orthogonalité a été empiriquement validée. Cette décision est motivée par un constat réaliste : le dialogue est une activité complexe mêlant gestion de la tâche et gestion de la communication sur plusieurs niveaux. Il n'est pour le moment pas envisageable de mener dans une seule thèse la modélisation de l'ensemble de ces activités parallèles intervenant dans le dialogue. Ce constat est appuyé par l'existence de thèses consacrées uniquement à certains processus intervenant dans le dialogue (e.g., le *grounding* [Traum 1994a], la gestion des tours de parole [Kronlid 2008]). Il est donc indispensable de délimiter notre champ d'étude tout en gardant à l'esprit l'existence et la nécessité d'intégrer d'autres activités intervenant dans le dialogue.

Nous avons choisi de nous limiter à la dimension *Task* car elle présente de nombreux avantages. Premièrement, la dimension *Task* prévaut sur les autres dimensions en terme de nombre de fonctions communicatives dans le corpus Cogni-CISMeF. Elle représente en effet deux tiers du corpus. Comme précédemment noté, l'accord inter-annotateur sur cette dimension est *fort* et *uniformément réparti* sur l'ensemble des catégories de fonctions communicatives y intervenant. Ensuite, les dimensions *Time* et *Own Communication Management* sont principalement *monologiques* dans le sens où elles concernent des hésitations et des auto-corrections du seul locuteur. Elles ne sont donc pas de bons candidats à l'extraction de motifs d'interaction entre deux interlocuteurs. Enfin, l'annotation dans la dimension *Auto-feedback* a démontré la difficulté d'identification des niveaux de la communication à partir des formes de surface des énoncés. Cette dimension est dominée par l'usage de la fonction de feedback positif sous-spécifié. Elle n'est donc pas un bon candidat pour la modélisation des processus de feedback essentiellement sur plusieurs niveaux (voir, e.g., [Larsson 2003]). Nous ferons néanmoins une unique exception en intégrant à certains motifs dialogiques la fonction *ExecNegativeAutoFB* qui est un feedback *négatif* au niveau *exécution*. Nous préciserons dans les motifs dialogiques où elle intervient les raisons qui nous ont poussé à l'intégrer.

Le *corpus d'extraction* est constitué de 2825 fonctions communicatives dans la dimension *Task* dont 2072 sont produites par l'expert et 753 par le demandeur. Sous ces trois aspects, le corpus d'extraction représente 65% du total du corpus COGNI-CISMEF.

Processus itératif de recherche de motifs dialogiques

L'extraction de motif dialogique, que j'ai moi-même réalisée, est un processus *itératif* principalement *manuel*.

La constitution des motifs dialogiques se base sur l'extraction de *séquences de segments fonctionnels* récurrentes. En conséquence de la restriction à la dimension *Task*, nous nous intéressons uniquement aux segments fonctionnels qui possèdent une fonction annotée dans cette dimension (auxquels nous ajoutons les segments fonctionnels contenant la fonction *ExecNegativeAutoFB* dans la dimension *Auto-feedback*). Les segments fonctionnels que nous considérons ne possèdent qu'un seul acte de dialogue constitué de la fonction annotée dans la dimension *Task* et du contenu sémantique déterminé par la forme de surface du segment. Pour cette raison, nous parlons également de séquences d'actes voire de séquences de fonctions pour désigner la séquence de segments fonctionnels.

La détection de séquences de segments fonctionnels récurrents se fonde pour chaque segment sur (i) la *fonction annotée* dans la dimension *Task*, et sur (ii) la *forme linguistique* du segment fonctionnel qui fournit le contenu sémantique. La prise en compte de ces deux paramètres permet de différencier les séquences de segments fonctionnellement liés (*« – donc c'est des douleurs articulaires à quel niveau ? – au niveau des genoux et autres »* (AL06)) de ceux qui ne le sont pas comme les énoncés ignorés par un des interlocuteurs (*« – mais peut-être que c'est euh trop spécifique ? – donc l'anorexie et les problèmes du même genre ? – oui »* (AL07)).

En outre, nous ne considérons que les séquences qui incluent au moins un segment produit par chaque interlocuteur. Autrement dit, cette séquence doit évoluer sur au moins deux tours de parole. La sélection d'une séquence de segments fonctionnels dépend de sa *récurrence*, i.e., son apparition dans des entretiens du corpus. Pour être récurrente, elle doit apparaître dans au moins deux entretiens.

Un *motif dialogique* est formé par le *regroupement* de certaines séquences. Les critères de regroupement sont liés dans notre cas à l'acte initiatif des séquences. Par exemple, toutes les séquences débutant par un acte de question à choix multiples sont regroupées dans le motif de question à choix multiples. Une *instance* d'un motif dialogique est la réalisation de l'une de ses séquences. Nous verrons que les motifs dialogiques extraits possèdent une instance dans plus d'un tiers des entretiens (soit 4 entretiens sur 12).

Une fois les motifs dialogiques détectés et annotés manuellement dans les entretiens, nous avons calculé les proportions d'occurrence *inter-motif* dialogique (i.e., la proportion d'apparition d'un type de motif par rapport aux autres) et *intra-motif* (i.e., la proportion d'apparition de la séquence de fonctions formant l'instance du motif dialogique).

5.2.2 Résultats du processus d'extraction

Cette section présente les résultats du processus d'extraction. Elle débute par une présentation globale des catégories de motifs dialogiques observés. Les motifs dialogiques sont ensuite présentés et illustrés par des exemples du corpus d'extraction. Enfin, nous abordons la couverture des motifs dialogiques extraits en terme de fonctions communicatives et de tours de parole.

Répartition des motifs dialogiques

Le processus d'extraction de motifs dialogiques a conduit à l'obtention de 11 motifs dialogiques évoluant principalement dans la dimension *Task*. Nous les avons regroupés dans trois grandes catégories. 6 motifs ont été observés dans la catégorie de *recherche d'information*, 3 dans la catégorie de *discussion d'action* et enfin 2 motifs dans la catégorie de *transfert d'information*. Ces régularités sont constituées de fonctions générales. La fonction *ExecNegativeAutoFB* de la dimension *Auto-feedback* n'intervient que dans la catégorie de *recherche d'information*. 431 instances de motifs dialogiques ont été observées dans l'ensemble du corpus d'extraction. Elles impliquent l'usage d'environ 860 fonctions communicatives provenant en large majorité de la dimension *Task*.

Il est indispensable de préciser que la correspondance entre les catégories de motifs dialogiques et les catégories de fonctions communicatives (abordées dans le processus d'annotation) ne peut pas se faire de façon immédiate. Ainsi, les motifs dialogiques de recherche d'information sont constitués de fonctions communicatives des catégories de recherche d'information et de fonctions fournissant de l'information (*Answer*, *Confirm* et *Disconfirm*). Les motifs dialogiques de transfert d'information sont formés d'un sous-ensemble de fonctions fournissant de l'information (*Inform*, *Agreement*, *Disagreement* et *Correction*). Enfin, les motifs dialogiques de discussion d'action sont constitués des fonctions communicatives de la catégorie du même nom.

La répartition observée des instances de motifs dialogiques par catégorie ne fait pas apparaître une catégorie clairement dominante (cf. figure 5.2). Environ 40% des motifs appartiennent à la catégorie de recherche d'information. Un peu plus d'un tiers des motifs proviennent de la catégorie de discussion d'action. Enfin, les instances de motifs de transfert d'information représentent environ 26% des instances totales.

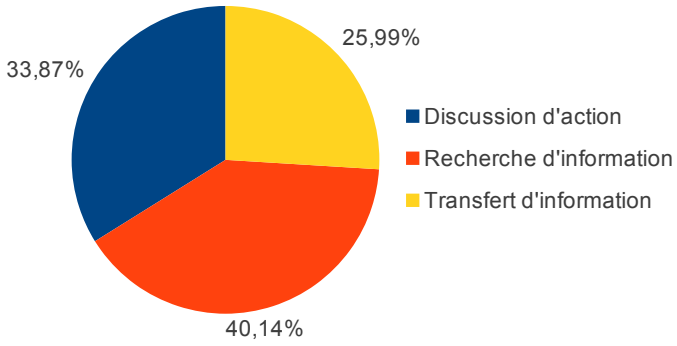

FIGURE 5.2: Répartition observée des instances de motifs dialogiques par catégorie dans le corpus d'extraction

La répartition des catégories de motifs peut sembler surprenante lorsque l'on prend en compte les proportions des fonctions communicatives y intervenant. En effet, les fonctions les plus observées dans le corpus (*Inform* et *Suggestion*) interviennent dans les catégories de motifs dialogiques contenant le moins d'instances. La principale raison tient au fait que l'expert produit des tours de parole contenant beaucoup d'actes de dialogue qui ne sont pas verbalement considérés par le demandeur (cf. section 5.1.3). Or, les motifs dialogiques ne prennent en compte que les séquences d'actes de dialogue au niveau de la dimension *Task* impliquant les deux interlocuteurs. Nous avons précédemment vu que la catégorie de fonctions fournissant de l'information est déséquilibrée par un nombre élevé de fonctions *Inform* (cf. section 5.1.3). Ceci est imputable à la stratégie de segmentation et à la verbalisation de l'expérimentation de la part de l'expert.

En conséquence, un nombre important de fonctions *Inform* n'a pas vocation à être dans un motif dialogique. Nous avions également vu qu'une large majorité des fonctions de *Suggestion* produites par l'expert n'est pas verbalement considérée par le demandeur. Une grande part des fonctions de *Suggestion* produites ne fait partie d'aucun motif dialogique. Ainsi, une proportion importante de fonctions fournissant de l'information et de discussion d'action ne contribuent pas à des motifs dialogiques, ce qui explique la répartition observée des instances de motifs dialogiques par catégorie.

Enfin, tous les motifs dialogiques admettent une de leur instance dans au moins un tiers des dialogues. L'annexe A.4.3 fournit le détail de la présence des motifs dialogiques dans les entretiens du corpus d'extraction.

Motifs dialogiques observés

Forme des motifs Comme envisagé dans [Schegloff 1973, Clark 1996, Lewin 2000, Hulstijn 2000b], nous avons principalement observé des motifs dialogiques de type initiative-réponse. Sur les 11 motifs découverts, 10 sont des motifs en deux temps, et un seul est un motif en trois temps.

Un motif en deux temps est une *paire adjacente* avec des secondes parties de paire préférées ou non. Le motif en trois temps est un *enchaînement* de paires.

Chaque motif en deux temps extrait prend la forme d'un acte de dialogue *initiatif* suivi par un des actes de dialogue *réponse* possibles associé à sa proportion d'occurrence observée dans le corpus d'extraction. L'*initiateur* du motif est l'interlocuteur qui produit la première partie de la paire tandis que le *partenaire* est celui qui réagit dans la seconde partie de la paire.

Nous présentons maintenant pour chaque catégorie, les motifs qui y interviennent. Nous nous efforçons de donner pour chaque motif un ou plusieurs exemples d'instances tirés du corpus. Le lecteur intéressé peut trouver plus d'exemples en annexe A.4.4.

Motifs dialogiques de recherche d'information Les motifs dialogiques de recherche d'information permettent à l'initiateur de s'enquérir d'une information auprès du partenaire. Nous avons observé 6 motifs qui correspondent aux 6 types de questions présents dans la catégorie « information-seeking » de la taxonomie DIT++. Ces motifs sont exclusivement de type initiative-réponse. Chacun de ces motifs est initié par un acte dont la fonction est une spécialisation de la fonction *Question* (i.e., *CheckQuestion*, *PosiCheck*, *NegaCheck*, *SetQuestion*, *ChoiceQuestion* ou *PropositionalQuestion*).

La figure 5.3 présente la répartition des différents motifs dialogiques de cette catégorie. La répartition des motifs dialogiques est cohérente par rapport à la proportion observée des fonctions de la catégorie « information-seeking » dans le corpus (cf. section 5.1.3). Schématiquement, les trois quarts des motifs dialogiques sont initiés par des fonctions représentant des questions à réponse déterminée (*CheckQuestion*, *PosiCheck*, *NegaCheck*, *ChoiceQuestion* et *PropositionalQuestion*). En particulier, nous observons une domination des motifs dialogiques initiés par des questions de vérification (*CheckQuestion*, *PosiCheck* et *NegaCheck*) qui représentent un peu moins de 62% du total. Ces derniers sont suivis du motif de question à choix multiples (8, 67%) et du motif de question oui/non (6, 36%). L'autre quart est représenté par le motif dialogique initié par une fonction représentant une question ouverte (*SetQuestion*).

La particularité des motifs de recherche d'information est la considération de la fonction spécifique *ExecNegativeAutoFB* de la dimension *Auto-feedback*, relative au niveau de l'interaction

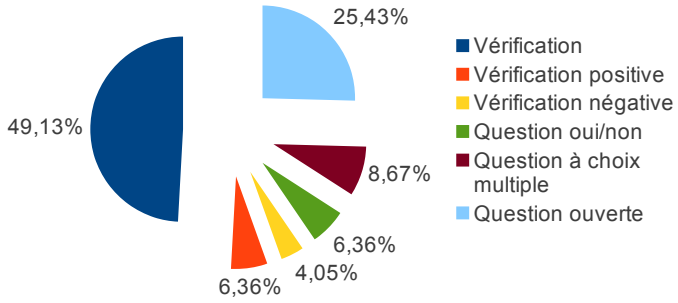

FIGURE 5.3: Répartition observée des instances de motifs dialogiques de recherche d'information dans le corpus COGNI-CISMEF

d'*exécution* (cf. section 1.1.2). Ce niveau est le plus haut niveau dans le traitement d'un énoncé. Il suit les niveaux d'attention, de perception, d'interprétation et d'évaluation. Il s'agit du niveau où la fonction et le contenu sémantique de l'acte ont été établis (via les niveaux d'attention jusqu'à l'interprétation) et l'acte a été vérifié comme étant consistant avec l'état d'information (via le niveau d'évaluation). L'exécution correspond à la *réaction* qui peut rencontrer certains problèmes. Par exemple, « exécuter » une question consiste à déterminer sa réponse. Un échec de cette exécution conduit à un feedback négatif de niveau exécution (*ExecNegativeAutoFB*) traduisant un « je ne sais pas ». L'avantage des feedbacks négatifs est qu'ils sont très souvent explicites et reconnaissables grâce à des marqueurs de surface (« pardon ? », « je ne sais pas ») [Bunt 2010]. Nous avons choisi de considérer l'acte *ExecNegativeAutoFB* parce qu'il est (i) toujours observé après une fonction de type question (l'annexe A.4.2 présente les fonctions qui la précèdent), (ii) aisément reconnaissable par des marqueurs de surface, et (iii) c'est le seul acte de niveau d'exécution observé dans le corpus.

Les motifs dialogiques de vérification Nous avons observé quatre motifs dialogiques de vérification d'une proposition. Ces motifs permettent de vérifier la valeur de vérité d'une proposition. Ils varient par la fonction communicative utilisée dans l'acte initiatif. Un extrait des motifs dialogiques de vérification est présenté dans le tableau 5.19.

	Question oui/non	Vérification d'une information
Première partie	*PropositionalQuestion*	*CheckQuestion*
Seconde partie ... préférée ... non-préférée	 *Answer* (90.91%) *ExecNegativeAutoFB* (9.09%)	 *Confirm* (45.45%) *Disconfirm* (37.88%) *Answer* (15.15%) *ExecNegativeAutoFB* (1.52%)

TABLEAU 5.19: Extrait des motifs dialogiques de vérification (catégorie de recherche d'information)

Le motif dialogique de question oui/non est initié par la fonction *PropositionalQuestion* (cf. tableau 5.19). Cette dernière est la fonction parente des fonctions de vérification (*CheckQuestion*, *PosiCheck* et *NegaCheck*). Elle permet d'obtenir la valeur de vérité de la proposition formant son *contenu sémantique* sans aucun a priori sur celle-ci. Ce motif a été peu observé dans notre corpus (6.36% des instances de motifs de la catégorie recherche d'information). Il possède une instance dans un tiers des entretiens. La seconde partie de paire préférée est un acte *Answer*

donnant la valeur de vérité de la proposition (90.91% des cas). Un exemple d'instance dans ce cas est présenté dans le dialogue 5.2.

	Task
S1 : il y a pas un synonyme de cette maladie ?	*PropositionalQuestion*
H2 : non	*Answer*

Dialogue 5.2 – Exemple d'un motif de type question oui/non avec seconde partie de paire préférée (entretien AL03)

La seconde partie de paire non-préférée intervient dans 9.09% des cas. Il s'agit d'un feedback d'exécution négative illustré par le dialogue 5.3.

	Task	*AutoFB*
S1 : C'est très handicapant vous pensez	*PropositionalQuestion*	*AutoPositive*
H2 : J'en sais rien		*ExecNegativeAutoFB*

Dialogue 5.3 – Exemple d'un motif de type question oui/non avec seconde partie de paire non-préférée (entretien VD04)

Le second motif que nous abordons est celui de *vérification d'une information* initié par la fonction *CheckQuestion*. Cette dernière est une spécialisation de la fonction *PropositionalQuestion* qui se différencie par le fait qu'elle s'attend à ce que la véracité de la proposition formant son contenu sémantique soit confirmée par le partenaire. Ce motif dialogique est celui qui a été le plus observé dans le corpus (49.13% des cas). Il apparaît dans l'ensemble des entretiens du corpus d'extraction. Il est présenté dans le tableau 5.19. L'acte *CheckQuestion* est suivi dans 45.45% des cas du corpus d'extraction par la seconde partie de paire préférée *Confirm*. Celle-ci permet de confirmer que le contenu propositionnel est vrai. Le dialogue 5.4 présente un exemple d'instantiation de ce motif dans ce cas.

	Task
S1 : vous voulez savoir les traitements de manière générale finalement ?	*CheckQuestion*
H2 : voilà / oui c'est ça	*Confirm*

Dialogue 5.4 – Exemple d'un motif de type vérification avec seconde partie de paire préférée (entretien AL12)

Ce motif dialogique admet trois secondes parties de paire non-préférées. La plus courante se manifeste par un acte *Disconfirm* qui déclare la proposition comme étant fausse (37.88% des cas observés). Le dialogue 5.5 propose un exemple d'instantiation du motif dans ce cas.

La seconde partie de paire est dans 15.15% des cas observés une fonction *Answer*. Il s'agit d'un cas d'*altération du projet conjoint* [Clark 1996] : le partenaire altère la question initiale de manière à pouvoir produire une réaction qu'il peut et souhaite produire. Dans le cas d'une question de vérification, le partenaire altère le projet de manière à ne pas confirmer ni infirmer l'objet de la question. Le dialogue 5.6 présente un exemple d'un telle altération. Au tour **S**1, l'interlocuteur demande l'ajout de la spécialité médicale « médecine du sport » à la requête. Son partenaire entame alors une question de vérification afin de s'assurer que la spécialité médicale

	Task
S1 : C'est le même que tout à l'heure ?	*CheckQuestion*
H2 : non c'est pas le même	*Disconfirm*

Dialogue 5.5 – Exemple d'un motif de type vérification avec seconde partie de paire non-préférée (entretien VD05)

est bien en lien avec le besoin d'information (tour **H**2). L'autre interlocuteur réagit en répondant en fait à la question « quelle serait la requête si les interlocuteurs ajoutaient la spécialité médicale », avant d'émettre des réserves (tour **S**3). Notons qu'il n'infirme, ni ne confirme le lien entre la spécialité médicale et le besoin d'information. Son interlocuteur accepte alors la requête (tour **H**4).

	Task
S1 : sinon t'avais des trucs t'avais dans la médecine du sport j'ai vu qu'il y avait médecine de sport	*Request*
H2 : et ça serait lié aussi à la médecine de sport ?	*CheckQuestion*
S3 : caries médecine de sport	*Answer*
enfin [...] je dis ça	*Inform*
H4 : non c'est une très bonne idée	*AcceptRequest*

Dialogue 5.6 – Exemple de motif de type vérification avec altération du projet conjoint (entretien AL11)

Enfin, la dernière seconde paire non-préférée est la production d'un acte *ExecNegativeAutoFB* traduisant l'impossibilité de trouver une réponse (observée dans une minorité de cas).

La fonction *CheckQuestion* admet deux spécialisations : *PosiCheck* et *NegaCheck*. La première spécialisation est une question de vérification sous-entendant de manière plus forte que le partenaire va confirmer la proposition (*« – alors plus précisémment ce serait le genou alors ? – oui »* (AL06)). La seconde spécialisation sous-entend que le partenaire va infirmer la proposition (*« – vous ne voyez pas grand chose non plus ? – [...] rien / je vois rien »* (VD04)). Elle est généralement marquée par l'emploi de la négation. Ces deux spécialisations ont produit deux nouveaux motifs dialogiques. Ces derniers représentent de manière cumulée un peu plus de 10% des motifs de la catégorie recherche d'information. Ils apparaissent tous deux dans un tiers des entretiens du corpus d'extraction. Chacun de ces motifs n'inclut qu'un seul acte initiatif et un seul acte réactif. Nous n'avons observé dans le corpus que des confirmations de la véracité d'une proposition (motif de vérification positive) ou de sa fausseté (motif de vérification négative). Cela peut s'expliquer par la stratégie d'annotation nécessitant des annotateurs des indices clairs pour annoter des fonctions spécialisées.

Le motif dialogique de question ouverte Un peu plus d'un quart des motifs de recherche d'information sont des instances du motif dialogique de question ouverte initié par la fonction *SetQuestion* (cf. tableau 5.20). Ce motif intervient dans 9 entretiens sur 12. La seconde paire préférée est constituée d'une fonction *Answer* et intervient dans 94.87% des cas. Cette fonction permet d'apporter une réponse à la question. Le dialogue 5.7 présente un exemple d'une instance d'un tel motif dialogique. La seconde paire non-préférée a été observée dans 5.13% des

cas. Il s'agit d'un feedback négatif de niveau exécution. Le dialogue 5.8 présente un exemple de dialogue avec une seconde paire non-préférée. De manière intéressante, la réponse à la question se fait en deux temps (tour **H**2). Dans le premier temps, le locuteur fournit un feedback de compréhension positif de la question d'un niveau inférieur au niveau d'exécution (qui a été annoté par la fonction *AutoPositive*). Puis il fournit un feedback négatif d'exécution.

	Question ouverte	Question à choix multiples
Première partie	*SetQuestion*	*ChoiceQuestion*
Seconde partie		
. . . préférée	*Answer* (94.87%)	*Answer* (93.34%)
. . . non-préférée	*ExecNegativeAutoFB* (5.13%)	*ExecNegativeAutoFB* (6.67%)

Tableau 5.20: Motifs dialogiques de question ouverte et de question à choix multiples

	Task
S1 : quel est le spécialiste de la peau c'est qui ?	*SetQuestion*
H2 : dermatologie	*Answer*

Dialogue 5.7 – Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien VD04)

	Task	*AutoFB*
S1 : comment on fait pour garder les documents ?	*SetQuestion*	
H2 : comment on fait pour garder les documents		*AutoPositive*
ça c'est une bonne question		*ExecNegativeAutoFB*

Dialogue 5.8 – Exemple d'un motif de type question ouverte avec seconde partie de paire non-préférée (entretien VD06)

Le motif dialogique de question à choix multiples Le motif de question à choix multiples (cf. tableau 5.20) constitue 8.67% des instances observées de la catégorie. Il apparaît dans 5 entretiens sur 12. Ce motif dialogique est initié par la fonction *ChoiceQuestion*. Le contenu sémantique associé à cette fonction représente une liste de propositions alternatives. La seconde partie de paire préférée consiste en un acte *Answer* dont le contenu sémantique est la proposition choisie comme étant vraie dans les propositions alternatives (93.34% des cas). Le dialogue 5.9 est un exemple d'une telle séquence. De la même manière que pour le motif de question ouverte, la seconde partie de paire non-préférée est l'acte *ExecNegativeAutoFB* informant de l'échec à choisir une proposition parmi les alternatives.

Motifs dialogiques de discussion d'action Les motifs dialogiques de discussion d'action permettent à un interlocuteur de demander ou d'offrir la réalisation d'une action qui doit être validée par le partenaire. Ils représentent 33.87% du total des instances de motifs observées. Nous avons observé trois motifs dialogiques de type initiative-réponse. Chacun de ces motifs est initié par un acte dont la fonction est *Suggestion*, *Request* ou *Offer*. La figure 5.4 présente la répartition observée dans le corpus Cogni-CISMeF des instances de motifs de discussion.

	Task	*AutoFB*
S1 : c'est uniquement sur le genou ? sur les problèmes d'articulation en général ?	*ChoiceQuestion*	
H2 : en général	*Answer*	
S3 : en général		*AutoPositive*

Dialogue 5.9 – Exemple de motif de type question à choix multiples avec seconde partie de paire préférée (entretien AL06)

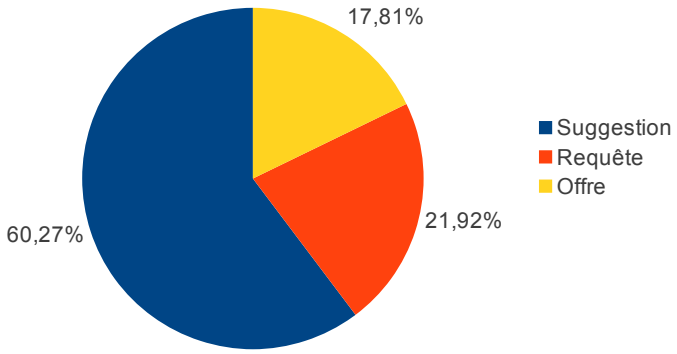

FIGURE 5.4: Répartition observée des instances de motifs dialogiques de discussion d'action dans le corpus COGNI-CISMEF

Cette répartition est cohérente par rapport à la proportion des trois fonctions initiatives observée dans le corpus pour l'expert et le demandeur (cf. section 5.1.3). On observe une majorité d'instances de motifs initiées par une fonction *Suggestion* qui arrive en tête des fonctions de discussion d'action utilisées par l'expert et le demandeur. Les proportions des instances de motifs dialogiques initiées par des fonctions *Request* et *Offer* sont relativement équilibrées et oscillent chacune autour de 20% du total des instances de motifs de discussion d'action. Les motifs dialogiques de discussion d'action sont synthétisés dans le tableau 5.21.

	Suggestion	Requête	Offre
Première partie	*Suggestion*	*Request*	*Offer*
Seconde partie			
... préférée	*AcceptSuggestion* (94.25%)	*AcceptRequest* (96.88%)	*AcceptOffer* (40%)
... non-préférée	*DeclineSuggestion* (5.75%)	*DeclineRequest* (3.12%)	*DeclineOffer* (60%)

TABLEAU 5.21: Motifs dialogiques de discussion d'action

Le motif de suggestion d'action intervient dans 11 entretiens sur 12. Il est initié par la fonction *Suggestion* (cf. tableau 5.21). Celle-ci est associée à un contenu sémantique décrivant une action. À la différence des fonctions *Offer* et *Request*, l'action décrite dans le contenu sémantique peut impliquer le destinataire de l'action *ou bien les deux interlocuteurs*. C'est pourquoi elle est souvent marquée linguistiquement par l'emploi du « nous » ou du « on ». La seconde partie de paire préférée est une acceptation de la suggestion via l'acte *AcceptSuggestion* (observée dans 94.25% des cas). Le dialogue 5.10 présente un exemple de ce cas. L'initiateur suggère à son partenaire de réaliser l'action d'ajout du mot-clé « troubles de l'alimentation » dans le tour **S**1. Cette suggestion est acceptée par le partenaire dans le tour suivant. La seconde partie de paire

non-préférée est un refus de l'action suggérée réalisé via l'acte *DeclineSuggestion*. Cette seconde partie a été très peu observée (5.75% des cas). Le dialogue 5.11 présente un exemple de suggestion déclinée. L'initiateur suggère l'ajout de « colon » à la requête. Le partenaire accepte dans un premier temps avant de s'auto-corriger et de décliner la suggestion (tour **H**2). L'analyse du corpus au niveau des fonctions fait apparaître une seconde paire non-préférée envisageable pour ce motif dialogique : l'acte *AddressSuggestion*. Cette dernière permet d'accepter *conditionnellement* l'action suggérée. Nous ne l'avons pas considérée dans le motif dialogique car elle n'apparaît que très minoritairement dans le corpus (deux apparitions cantonnées au même entretien VD04).

	Task
S1 : on va rentrer comme mot clé troubles de l'alimentation	*Suggestion*
H2 : ça marche	*AcceptSuggestion*

Dialogue 5.10 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien AL07)

	Task
S1 : faudrait colon quand même	*Suggestion*
H2 : moi je le mettrais bien bah quoi qu'on la là	*DeclineSuggestion*

Dialogue 5.11 – Exemple de motif de type suggestion avec seconde partie de paire non-préférée (entretien VD06)

Le motif de demande de réalisation d'action apparaît dans la moitié des entretiens. Il est initié par l'acte *Request*. L'initiateur demande au destinataire de cet acte de réaliser l'action décrite dans son contenu sémantique. La réalisation de l'action est conditionnée par l'acceptation du partenaire ce qui explique les deux secondes parties de paire observées (cf. tableau 5.21). La seconde partie de paire préférée est celle qui intervient dans la grande majorité des cas (96.88%). Il s'agit de l'acceptation de la requête par le partenaire via l'acte *AcceptRequest*. Le dialogue 5.12 illustre un cas de requête acceptée. De la même façon que le motif de suggestion, la seconde partie de paire non-préférée est un refus de réaliser la requête. Ce cas a été très peu observé (3.13%). La fonction *AddressRequest* qui pourrait former un autre cas de seconde paire non-préférée n'a pas été retenue. Elle permet au partenaire d'accepter conditionnellement la réalisation de l'action. Son éviction tient au fait qu'elle n'apparaît que trois fois dans le corpus d'extraction sans réel accord entre les annotateurs.

	Task	*AutoFB*
S1 : essayez avec clairance	*Request*	
H2 : alors je l'essaie en mot clé	*AcceptRequest*	
S3 : ouais c'est clairance	*Inform*	*AutoPositive*
H4 : ok		*AutoPositive*

Dialogue 5.12 – Exemple de motif de type requête avec seconde partie de paire préférée (entretien VD07). L'instance de motif est initiée au tour **S**1 et la requête est acceptée au tour **H**2.

Enfin, le dernier motif dialogique de discussion d'action détecté est celui d'offre de réalisation

d'une action qui intervient dans deux tiers des entretiens (cf. tableau 5.21). Contrairement aux deux précédents motifs qui sont *directifs*, celui-ci est un *promissif*. En produisant l'acte *Offer*, l'initiateur propose à son partenaire de réaliser lui-même l'action décrite dans le contenu sémantique. La réalisation est conditionnée par l'acceptation du partenaire d'où sont issues les deux secondes parties de paires observées. La seconde partie de paire préférée est une acceptation via l'acte *AcceptOffer*. Contrairement aux deux motifs précédents, cette acceptation n'a pas été observée en majorité écrasante puisqu'elle n'apparaît que dans 60% des cas. Le dialogue 5.13 présente un exemple d'une offre acceptée. L'initiateur propose au partenaire de réaliser la recherche automatique du mot-clé « créatinine » dans le texte d'un document. Cette proposition est acceptée. La seconde partie de paire non-préférée est un refus de l'offre via l'acte *DeclineOffer*. Elle suit une offre dans 40% des cas. Le dialogue 5.14 présente un exemple de refus. Il ne faut pas se fier à la forme linguistique. L'offre prend la forme d'une question (tour **S**1) mais est en fait une proposition de lancer une nouvelle *séquence de recherche* collaborative (i.e., l'expert se propose d'assister le demandeur sur un nouveau sujet de recherche). Cette dernière est refusée. Cet exemple illustre le cas le plus fréquent de refus : l'expert offre la possibilité au demandeur de débuter une nouvelle séquence de recherche à la fin de l'entretien. Cette offre est refusée dans la quasi-totalité des entretiens (excepté l'entretien AL09). Nous avions vu la très faible proportion de la fonction *AddressOffer* lors de l'analyse des fonctions du corpus. Cette fonction n'est pas présente dans le corpus d'extraction.

	Task
S1 : alors ce que je vous propose puisqu'il y a pas mal de choses à lire c'est d'aller chercher notre mot clé créatinine dans ce texte	*Offer*
H2 : ouais	*AcceptOffer*

Dialogue 5.13 – Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD07)

	Task
S1 : vous avez une autre question ?	*Offer*
H2 : non non non	*DeclineOffer*

Dialogue 5.14 – Exemple de motif de type offre avec seconde partie de paire non-préférée (entretien AL07)

Motifs dialogiques de transfert d'information Les motifs dialogiques de transfert d'information représentent un échange strictement informationnel. Ils permettent aux interlocuteurs de s'accorder sur les faits qu'ils affirment (i.e. l'état du monde). Nous avons observé deux motifs dialogiques, l'un en deux temps (motif d'accord) et l'autre en trois temps (motif de correction). La figure 5.5 présente la répartition de ces motifs en terme de nombre d'instances observées dans le corpus. Cette catégorie est largement dominée par le motif d'accord (94.64% des instances observées). Nous avons observé marginalement le motif de correction (5.36% des cas).

Le motif d'accord intervient dans 11 entretiens sur 12. Il possède deux actes initiatifs : *Inform* et *Answer* (cf. tableau 5.22). Autrement dit, les actes initiatifs sont des actes *informatifs* engageant leur locuteur sur son contenu propositionnel. Le motif d'accord est ainsi initié par un

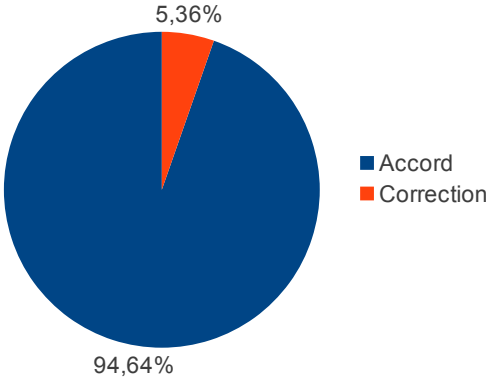

FIGURE 5.5: Répartition observée des motifs dialogiques de transfert d'information dans le corpus COGNICISMEF

acte informatif. Le partenaire peut alors réagir en exprimant son accord ou son désaccord. La seconde partie de paire préférée du motif est un acte *Agreement* (environ 99% des cas). Il permet d'exprimer un accord sur le contenu propositionnel de l'acte initiatif. Le dialogue 5.15 présente un exemple d'une instance du motif dans ce cas. L'initiateur affirme que les résultats retournés par la requête sont insatisfaisants comme les précédents obtenus (tour **S**1). Le partenaire affirme alors son accord (tour **H**2). La seconde partie de paire non-préférée est l'expression d'un désaccord via l'acte *Disagreement*. Elle n'a été observée que marginalement (1.01% des cas).

	Accord
Première partie	Acte informatif (*Inform* et *Answer*)
Seconde partie	
. . . préférée	*Agreement* (98.99%)
. . . non-préférée	*Disagreement* (1.01%)

TABLEAU 5.22: Motif dialogique d'accord (catégorie de transfert d'information)

	Task
S1 : C'est exactement la même chose en fait	*Inform*
H2 : humm, humm tout à fait	*Agreement*

Dialogue 5.15 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien VD02)

Le motif dialogique de correction est un cas particulier de désaccord observé dans un tiers des entretiens (cf. figure 5.6). De la même manière que le motif d'accord, il est initié par un acte informatif (*Inform* ou *Answer*). Cet acte est suivi par un acte de *Correction*. La fonction *Correction* est une spécialisation de la fonction *Disagreement*. En plus d'exprimer un désaccord, cette fonction propose un nouveau contenu sémantique qui a vocation à remplacer le contenu sémantique de l'acte informatif initiatif. Dans deux tiers des cas observés, cette correction est acceptée via l'acte *Agreement*. Dans l'autre tiers, l'instance s'arrête à l'acte de *Correction*. Le dialogue 5.16 présente un exemple de correction en trois temps. Dans le premier tour, le locuteur affirme que l'exécution de la requête retourne les mêmes documents qu'une requête précédente

(tour **S**1). Son partenaire le corrige alors en lui précisant qu'il y a en fait plus de documents (tour **H**2). L'initiateur accepte alors la correction dans le dernier tour.

FIGURE 5.6: Motif dialogique de correction

	Task
S1 : alors nous retombons de nouveau sur la même chose que tout à l'heure	*Inform*
H2 : ah peut-être un peu plus	*Correction*
S3 : oui	*Agreement*

Dialogue 5.16 – Exemple de motif de type correction en trois temps (entretien VD04)

Couverture des motifs dialogiques dans le corpus d'extraction

Jusqu'alors nous avons présenté les motifs dialogiques sans préciser leur taux global d'apparition dans le corpus, i.e., la couverture des motifs.

Une première approche pour quantifier la couverture du corpus d'extraction par les motifs dialogiques extraits est de comparer le nombre de fonctions communicatives intervenant dans une instance de motif au nombre de fonctions communicatives total dans la dimension *Task*. Le ratio obtenu est faible : seules 30.4% des fonctions communicatives de la dimension *Task* interviennent dans un motif. Néanmoins, nos motifs sont principalement des *paires adjacentes* impliquant la production d'une fonction communicative pour chaque interlocuteur. Or, l'expert produit 73.3% des fonctions communicatives dans le corpus d'extraction quand le demandeur en produit 26.7%. Compte tenu du fait que les motifs sont principalement en deux temps, les fonctions communicatives impliquées peuvent difficilement excéder le double des fonctions produites par le demandeur, soit 53.4% du total. Il est donc normal que le taux de couverture soit faible en terme de fonctions communicatives.

Si nous regardons de plus près la constitution d'un tour de parole (cf. section 5.1.3), nous constatons la présence d'un bon nombre de segments fonctionnels isolés, i.e., non verbalement considérés par l'autre interlocuteur. C'est en particulier le cas des tours de parole de l'expert. Le dialogue 5.17 présente un exemple contenant des segments isolés. Le tour **H**2 est une longue prise de parole dans laquelle l'expert accepte la suggestion d'ajouter la spécialité médicale à la requête énoncée par le demandeur dans le tour **S**1. Il modifie également proactivement la requête à la suite des résultats rencontrés. À la fin de l'exemple, les deux interlocuteurs s'accordent sur l'intérêt des ressources. Sur les 13 segments fonctionnels, seuls 4 font partie d'un motif dialogique. Deux forment une instance d'un motif de suggestion et deux autres constituent une instance d'un motif d'accord. Notons que les segments isolés incluent des fonctions *Inform* et *Suggestion* précédemment discutées en section 5.1.3.

	Task
S1 : et des choses comme traumatologie ça serait : :	*Suggestion*
H2 : oui la thraumatologie ça serait bien	*AcceptSuggestion*
ah oui	
effectivement je n'ai pas pensé à mettre ça	*Inform*
on va voir ce qu'on obtient comme ça	*Suggestion*
(l'expert ajoute la spécialité médicale traumatologie à la requête et lance la requête)	
alors trois ressources	*Inform*
apparemment un qui concerne l'épaule / le sport / hmm : : et encore un truc sur xxx	*Inform*
allez on va encore chercher autre chose plutôt que arthralgie	*Suggestion*
on va rester dans thraumatologie et on va mettre genou	*Suggestion*
une nouvelle piste	*Inform*
(l'expert change le mot-clé « arthralgie » par « genou »)	
là on a 33 ressources	*Inform*
alors effectivement / ça s'annonce bien	*Inform*
S3 : ah oui il y a des choses là	*Agreement*

Dialogue 5.17 – Exemple de longue prise de parole de l'expert (entretien AL10)

Afin d'avoir une mesure plus juste de la couverture réelle des motifs dialogiques extraits, nous avons dénombré le nombre de *tours de parole* engagés dans un motif dialogique. Un tour de parole est engagé dans un motif dialogique si et seulement si : (i) il inclut au moins un segment fonctionnel possédant une fonction dans la dimension *Task* (ou une fonction *ExecNegativeAutoFB* dans la dimension *Auto-feedback*), et (ii) il contient au moins un segment fonctionnel faisant partie d'une instance d'un motif dialogique extrait. Les tours de parole ne contenant aucun segment fonctionnel avec une fonction dans la dimension *Task* (ou une fonction *ExecNegativeAutoFB* dans la dimension *Auto-feedback*) sont exclus du compte. Dans le dialogue 5.17, les trois tours de parole sont engagés dans un motif dialogique. Le tour **S**1 fait partie d'un motif de suggestion, le tour **S**3 d'un motif d'accord et le tour **H**2 des deux motifs (suggestion et accord).

Cette mesure met l'accent sur l'enchaînement des tours de parole plutôt que sur une couverture en terme de fonctions communicatives. Elle permet de caractériser le nombre de tours de parole liés par le déroulement de motifs dialogiques (ou une fonction *ExecNegativeAutoFB* dans la dimension *Auto-feedback*).

1129 tours de parole sont éligibles dans le corpus d'extraction, soit 83.4% du total de ce corpus. Les autres tours de parole ne contiennent aucun segment fonctionnel avec une fonction dans la dimension *Task* (ou une fonction *ExecNegativeAutoFB* dans la dimension *Auto-feedback*). Près de deux tiers (64.1%) des tours de parole considérés sont engagés dans au moins un motif dialogique. Les 35.9% non couverts peuvent s'expliquer par plusieurs raisons. La principale est que l'enchaînement de tour de parole est expliqué par une évolution sur l'axe de la gestion de l'interaction (50.2% des cas). Cela inclut des tours de parole suivis par des feedbacks de retour positif (« oui », « ok », « humm humm »), par des complétions de la part du partenaire, par des demandes de clarification ou encore par des prises de parole intempestives (marquées par l'usage de la fonction *Turn Grab* dans la dimension *Turn Management*). La seconde raison

est la présence de tours de parole isolés (39.8% des cas). Il s'agit de tours de parole dont la transcription seule ne permet pas de conclure à un lien avec les tours précédents. Ils ne sont pas verbalement considérés par les interlocuteurs. Les 10% de cas restants sont marginaux. Dans 6.2% des cas, l'enchaînement des tours de parole démontre la présence d'un motif partiel. Les motifs impliqués sont principalement les motifs de discussion d'action. Il s'agit de requête ou de suggestion réalisées sans acceptation verbale (*AcceptSuggestion*/*AcceptRequest*). Dans une minorité de cas (2.6%), l'enchaînement des tours est expliqué par l'usage d'une fonction non couverte par un motif. Il s'agit des fonctions *Instruct*, *AddressSuggestion* et *AddressRequest*. Précisons que ces tours de parole représentent moins de 1% des tours considérés au total. Enfin, nous avons observé quelques phénomènes d'emboîtement de motifs ne respectant pas la stricte adjacence (1.2% des cas).

5.2.3 Synthèse

Nous avons décomposé le corpus COGNI-CISMEF en deux sous-corpus : le *corpus de référence* constitué aléatoirement d'un tiers du corpus et le *corpus d'extraction* formé par les deux tiers restant. Seul le corpus d'extraction a été utilisé lors de l'extraction de motifs dialogiques.

L'annotation multidimensionnelle du corpus avec le schéma DIT++ a permis de prendre en compte la complexité du dialogue qui implique la réalisation de plusieurs activités en parallèle. En outre, la définition des dimensions dans DIT++ rend possible l'exploitation *partielle* du corpus annoté pour l'analyse. Nous avons choisi d'exploiter cette caractéristique en limitant notre travail d'extraction à la principale dimension : la dimension *Task*. Ce choix est motivé par la volonté de ne pas mener de front la modélisation de l'ensemble des processus parallèles intervenant dans le dialogue qui nécessitent chacun une étude rigoureuse. La dimension *Task* semble la plus adaptée : c'est elle qui permet d'avancer la tâche sous-jacente au dialogue et elle prévaut naturellement en terme de fonctions communicatives.

Le processus d'extraction est un processus itératif et manuel consistant en la détection de motifs dialogiques dans les entretiens, puis en l'annotation de ceux-ci. Les motifs dialogiques sont constitués de séquences de segments fonctionnels récurrentes dans les entretiens et qui incluent des interventions de la part des deux interlocuteurs. Ces séquences prennent en compte aussi bien la fonction dans la dimension *Task* que le contenu sémantique au travers de la forme de surface du segment fonctionnel.

11 motifs d'interaction constitués de fonctions générales et de la fonction *ExecNegativeAutoFB* ont été détectés. Nous les avons regroupés en 3 catégories : recherche d'information (6 motifs), transfert d'information (2 motifs) et discussion d'actions (3 motifs). Comme envisagé par de nombreux chercheurs [Schegloff 1973, Clark 1996, Lewin 2000, Hulstijn 2000b], nous avons principalement observé des motifs initiative-réponse constitués d'une paire adjacente avec une seconde partie de paire préférée ou non. Chaque motif extrait prend la forme d'un acte de dialogue initiatif suivi par un des actes de dialogue réponse possibles (associés à sa proportion d'occurrence observée dans le corpus d'extraction).

Enfin, nous avons constaté que deux tiers des tours de parole susceptibles de contribuer à un motif dialogique sont effectivement engagés dans un motif. Le tiers restant est majoritairement impliqué dans des phénomènes interactionnels de non-considération ou d'évolution sur l'axe de l'interaction, volontairement exclus de notre étude.

5.3 Discussion

Ce chapitre a présenté les deux étapes d'exploitation du corpus : son enrichissement (via l'annotation) et l'extraction de motifs dialogiques. Le corpus a été transformé en une séquence d'*unités multidimensionnelles* produites par les interlocuteurs : les segments fonctionnels. La participation de ces unités dans les différentes activités parallèles du dialogue a été vue comme la production de fonctions communicatives dans plusieurs dimensions théoriquement fondées. Le processus d'annotation a ainsi conduit à l'obtention d'une représentation du corpus où chaque unité forme une ligne dont les colonnes sont constituées du locuteur, de la forme de surface du segment ainsi que des fonctions annotées dans les dix dimensions considérées. L'usage du schéma multidimensionnel DIT++ a permis de prendre en compte la richesse qui fait la complexité du dialogue. Le corpus ainsi transformé a rendu possible la recherche et l'extraction de motifs dialogiques récurrents impliquant la participation des deux interlocuteurs. Ces motifs ont été extraits manuellement sur la base de la fonction annotée dans la dimension *Task*, du contenu sémantique tiré de la forme de surface du segment et enfin de l'interlocuteur. En somme, cette étape a permis de déterminer des fragments d'interaction que l'on souhaite voir reproduit par un agent interactif. Nous avons constitué une bibliothèque initiale de 11 motifs d'interaction qui vont nous servir de base pour la création du modèle d'interaction d'un agent interactif.

Le rôle de l'humain dans ces deux étapes de la méthodologie a été prégnant. Les seules interventions automatiques ont eu pour but d'exploiter les annotations réalisées manuellement afin de produire des statistiques. La réalisation manuelle de ces étapes est chronophage et nécessite un effort d'apprentissage non négligeable (notamment du schéma d'interaction pour les annotateurs). Elle a permis la réalisation des étapes cruciales : (i) la segmentation des unités de sens dans le dialogue, (ii) l'étiquetage multidimensionnel de ces unités, et (iii) la découverte et l'extraction de motifs dialogiques récurrents. Selon nous, l'évolution de la méthodologie dépend des possibilités de semi-automatiser les phases d'annotations et d'extraction [Ales 2012]. L'automatisation complète de ces deux étapes en est au stade de travaux de recherche. Les résultats acquis ne permettent pas pour le moment l'utilisation opérationnelle d'outils. Dans cette direction, le corpus Cogni-CISMeF annoté peut servir de base pour l'entraînement d'algorithme d'apprentissage sur la reconnaissance d'actes de dialogue. Nous pensons, comme [Orkin 2013], que la solution la plus réaliste consiste à envisager ces étapes sous la forme d'un processus collaboratif entre l'humain et la machine. Plusieurs équilibres sont alors possibles selon les responsabilités accordées à l'un ou à l'autre. [Orkin 2013] propose de confier à l'humain la responsabilité d'annoter les motifs (sous forme de séquences d'actions représentant une tâche) afin de *spécifier* les relations entretenues entre des fragments de comportements, et de *sélectionner* les fragments que l'agent doit reproduire. À l'exécution, le processus délibératif de l'agent consiste à *rechercher* des fragments de comportement contextuellement pertinent à rejouer à chaque instant, basé sur les motifs de comportements observés.

Troisième partie

Modélisation des interactions humaines et mise en œuvre

Chapitre 6

Formalisation des motifs d'interaction

Sommaire

6.1 Modèle de l'engagement social et tableau de conversation 145
6.1.1 Modèle de l'engagement social 146
6.1.2 Tableau de conversation 149
6.1.3 Interprétation des engagements du tableau de conversation 156
6.2 Jeux de dialogue et jeux de communication 158
6.2.1 Modèle de jeu de dialogue 158
6.2.2 Modèle de jeu de communication 161
6.2.3 Combinaisons et établissement de jeux de dialogue 161
6.2.4 Interprétation des jeux de dialogue dans le tableau de conversation 163
6.3 Spécification empirique de jeux depuis le corpus Cogni-CISMeF 163
6.3.1 Représentation du contenu sémantique 164
6.3.2 Actes de dialogue 168
6.3.3 Jeu de communication et jeux de dialogue 169
6.3.4 Couverture du corpus par les jeux définis 176
6.4 Discussion 179

Dans ce chapitre, nous présentons un cadre pour modéliser les motifs dialogiques observés dans le corpus Cogni-CISMeF. Ce cadre permet de spécifier des enchaînements d'actes attendus. Il est fondé sur l'approche des jeux de dialogue vus comme des structures capturant les engagements que les interlocuteurs contractent pendant le dialogue (cf. section 3.2.3). La section 6.1 décrit notre modèle de l'engagement social pour formaliser les jeux. Ces engagements sont associés à un tableau de conversation qui représente l'état courant du dialogue en stockant les engagements contractés par les interlocuteurs. La section 6.2 présente les structures de jeux de dialogue et de communication basées sur les engagements précédemment définis. Ce formalisme est illustré dans la section 6.3 présentant la spécification des jeux de dialogue et de communication depuis les motifs dialogiques observés dans notre corpus. Enfin, nous concluons ce chapitre en soulignant les avancées et les limites de notre modèle de jeux de dialogue (cf. section 6.4).

6.1 Modèle de l'engagement social et tableau de conversation

Cette section débute par la formalisation de la notion d'engagement social dans le cadre de notre modèle (cf. section 6.1.1). Nous abordons ensuite le tableau de conversation qui contient les engagements sociaux, et son évolution suite à l'occurrence d'événements (cf. section 6.1.2).

Nous terminons cette section en donnant une interprétation des engagements contenus dans le tableau de conversation (cf. section 6.1.3).

6.1.1 Modèle de l'engagement social

Dans cette section, nous présentons notre modèle de l'engagement social inspirés des travaux de référence de [Maudet 2001, Pasquier 2005, Chaib-Draa 2006] (cf. section 3.2.3). Nous envisageons trois types d'engagements : les engagements propositionnels (extra-dialogiques, de type PROPCOMMITMENT), les engagements en action (dialogiques, de type ACTIONCOMMITMENT, et extra-dialogiques, de type GAMEACTIONCOMMITMENT) et les engagements conjoints sur les jeux de dialogue (extra-dialogiques, de type DIALOGUEGAMECOMMITMENT). Dans la suite du document, nous utilisons le symbole « : » afin de représenter la relation « est de type » (expr : Type signifie que *expr* est de type *Type*). Formellement :

c : COMMITMENT ssi c : PROPCOMMITMENT ou c : ACTIONCOMMITMENT ou c : GAMEACTIONCOMMITMENT ou c : DIALOGUEGAMECOMMITMENT

Un engagement *standard* est un engagement propositionnel ou un engagement en action (dialogique ou non). Nous excluons les engagements conjoints sur les jeux de dialogue de ces engagements. Formellement :

c : STANDARDCOMMITMENT ssi c : PROPCOMMITMENT ou c : ACTIONCOMMITMENT ou c : GAMEACTIONCOMMITMENT

Engagement propositionnel

Un engagement propositionnel (PROPCOMMITMENT) capture le fait qu'un interlocuteur s'engage au présent sur une proposition envers un autre interlocuteur. Un tel engagement prend la forme C(x,y,p,t,s) signifiant que l'engagement « x est engagé envers y sur la proposition p » est dans l'état s depuis le temps t. Dans un soucis de simplification, nous ne considérons que les propositions dirigées vers le présent (i.e., décrivant l'état du monde tel qu'il est). Cela nous conduit à n'envisager que deux états pour un engagement propositionnel représenté en figure 6.1. Un engagement propositionnel est initialement *inactif* (**Ina**). À la suite de sa création, l'engagement passe en état créé (**Crt**), dit *actif*. Un engagement créé peut être annulé par son interlocuteur et retourne alors à l'état *inactif*. Dans la perspective d'uniformisation des engagements propositionnels et en action, il semble raisonnable d'étendre ces états vers ceux des engagements en action. Cette perspective nécessite néanmoins des recherches supplémentaires dont les premiers pas peuvent se baser sur [Singh 2008].

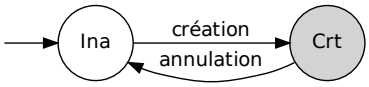

FIGURE 6.1: États d'un engagement propositionnel. L'état grisé est actif.

Engagement en action

Un engagement en action capture le fait qu'un interlocuteur s'engage au présent à ce qu'une action survienne dans le futur.

Un engagement en action *extra-dialogique* prend la forme C(x,y,α,t,s) signifiant que l'engagement « x est engagé envers y sur la réalisation de l'action α » est dans l'état s depuis le temps t. Les états considérés pour les engagements en action sont présentés en figure 6.2. Initialement, un engagement en action est dans l'état inactif (**Ina**). Cet engagement peut alors être créé. La tentative de création peut conduire à l'échec de la tentative de création (**Fal**) ou à la création effective (**Crt**). Un engagement dans l'état **Crt** est alors *actif*. Un engagement actif peut être *violé* le conduisant à l'état **Vio**. Il s'agit de la situation dans laquelle les conditions de satisfaction spécifiées par le contenu ne peuvent plus être remplies. Un engagement actif peut être *satisfait* le conduisant à l'état **Ful**. Un engagement en action est satisfait si son contenu a été effectué.

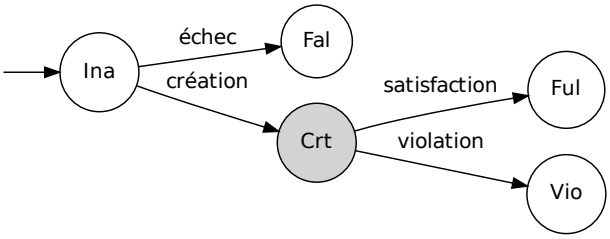

FIGURE 6.2: États d'un engagement en action. L'état grisé est actif.

Les états présentés en figure 6.2 constituent une première approche simplifiée qui a l'avantage d'être *extensible*. En effet, il est possible d'envisager d'autres états comme l'annulation, la mise en attente, l'expiration, etc. (voir, e.g., [Chaib-Draa 2006, Telang 2012]).

Les engagements en action *dialogiques* sont au cœur de la spécification des jeux de dialogue et de communication. Ceux-ci prennent la même forme que les engagements en action *extra-dialogiques* à l'exception du fait qu'ils sont contextualisés dans un jeu j noté en indice : $\mathrm{C_j}(x,y,\alpha,t,s)$. En outre, le contenu de ces engagements dialogiques sont des *descriptions d'événements* (de type DESCRIPTION) devant survenir dans le futur. Ces descriptions peuvent être simples (DESCÉVÉNEMENT) ou composées (cf. tableau 6.1). Par exemple, la description simple de l'événement « inform(x,–) » représente un coup dialogique *Inform* produit par x dont le contenu sémantique est quelconque. Nous supposons l'existence du prédicat :

$$\text{correspond}(e : \text{ÉVÉNEMENT}, \alpha : \text{DESCÉVÉNEMENT})$$

Celui-ci est vrai si l'événement e satisfaisait la description de l'événement α, faux sinon. Par exemple, l'événement dialogique inform(x,p) correspond à la description de l'événement inform(x,–).

Classiquement, ces descriptions d'événements peuvent être combinées de manière à exprimer le choix et la négation (cf. tableau 6.1). En outre, ces engagements peuvent être *conditionnels*. Un tel engagement est exprimé sous la forme d'un engagement d'ordre supérieur [Singh 1999]. $\alpha \Rightarrow \beta$ signifie que si un événement survient et correspond à la description α, alors l'opération β est réalisée sur le tableau de conversation. $\alpha \stackrel{*}{\Rightarrow} \beta$ signifie qu'*à chaque fois* qu'un événement survient et correspond à la description α, alors l'opération β est réalisée sur le tableau de conversation. Les engagements en action dialogiques vont permettre aux interlocuteurs de s'engager sur des *règles de production* conduisant à la modification du tableau de conversation, et donc des engagements des interlocuteurs.

Nom	Type	Notation	Remarques
Description simple	DescÉvénement	α	
Négation	Negation	$\neg\alpha$	α : DescÉvénement
Choix	Choix	$\alpha_1 \vert \alpha_2$	α_1 : DescÉvénement α_2 : DescÉvénement ou Choix
Conditionnelle	Conditionnelle	$\alpha \Rightarrow \beta$	α : DescÉvénement ou Choix β : Operation
Conditionnelle persistante	CondPersistante	$\alpha \overset{*}{\Rightarrow} \beta$	α : DescÉvénement ou Choix β : Operation

Tableau 6.1: Événements et combinaisons d'événements pour les engagements dialogiques en action

Engagement conjoint sur un jeu de dialogue

Enfin, nous considérons les engagements *conjoints* sur les jeux de dialogue. Ces engagements sont *extra-dialogiques*, bien que leur contenu soit un jeu de dialogue. Ils prennent la forme : C($\{x,y\}$,j,t,s) signifiant que l'engagement « x et y sont engagés conjointement sur le jeu de dialogue j » est dans l'état s depuis le temps t. Ces engagements sont *conjointement établis* via le mécanisme de contextualisation que nous présentons en section 6.2.3. Les états de ces engagements correspondent aux différentes étapes d'entrée et de sortie d'un jeu de dialogue (cf. figure 6.3). Initialement, les interlocuteurs ne sont engagés sur aucun jeu de dialogue (état **Ina**). L'entrée dans un jeu peut être suggérée par un des interlocuteurs, faisant passer l'engagement conjoint à l'état **Sugg**. Cette suggestion peut conduire à un refus (retour à l'état **Ina**) ou à la création de l'engagement conjoint sur le jeu de dialogue (état **Open**). Un engagement dans l'état **Open** est *actif* (le jeu de dialogue est établi, et ses *règles* entrent en considération dans le comportement communicatif des interlocuteurs). Un engagement sur un jeu peut alors être déchargé via la fermeture du jeu, conduisant l'engagement à l'état **Closed**.

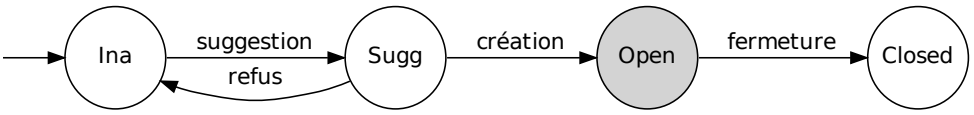

Figure 6.3: États d'un engagement conjoint sur un jeu de dialogue. L'état grisé est actif.

Raccourcis d'écriture

Afin d'alléger l'écriture et la lecture des engagements sociaux, nous réalisons quelques simplifications d'écriture. Puisque nous ne considérons que les engagements pris dans un dialogue avec deux interlocuteurs, nous omettons le *créditeur* de l'engagement et n'indiquons que le *débiteur* de celui-ci. En outre, nous ne précisons pas le paramètre temporel lorsqu'il n'est pas indispensable. Par exemple :

- L'engagement propositionnel extra-dialogique C(x,y,p,t,**Crt**) se réécrit C(x,p,**Crt**)
- L'engagement en action dialogique (contextualisé dans le jeu j) C_j(y,x,$\alpha|\beta$,t,**Ful**) se réécrit C_j(y,$\alpha|\beta$,**Ful**)

6.1.2 Tableau de conversation

Dans cette section, nous introduisons les opérations générales réalisables sur le tableau de conversation en rapport avec les engagements propositionnels et en action. Nous fournissons la spécification de certains algorithmes essentiels permettant de cerner au plus près le fonctionnement du système.

Principe du tableau de conversation

Le tableau de conversation représente l'*état du dialogue* entre les interlocuteurs à un instant donné (cf. section 2.2.1). Puisque nous n'abordons pas les mécanismes de *grounding* et les problèmes de communication [1], le tableau de conversation représente la partie publique du contexte dialogique supposée *strictement partagée* (cf. modèle 4, section 2.2.1).

T_i symbolise le tableau de conversation au temps i (le temps courant). Nous utilisons une simple théorie des instants où « < » est la relation de précédence. Nous distinguons deux grands types d'événements : les *événements externes* (ÉvénementExterne) et les *événements internes* (ÉvénementInterne). Les événements externes peuvent être *dialogiques* (e.g., un événement d'énonciation d'un acte de dialogue) ou *extra-dialogiques* (e.g., un événement comme lumière_allumée dénotant la réalisation de l'action d'activation de la lumière). Les *événements internes* sont des événements déclenchés par l'atteinte d'un état du tableau (présentés dans la suite de cette section).

Le principe de l'évolution du tableau de conversation en fonction de l'occurrence des événements est présenté en figure 6.4. À chaque occurrence d'un événement *externe*, le tableau de conversation évolue et le temps est incrémenté. La modification du tableau de conversation peut provoquer des occurrences d'événements internes. Ces derniers vont contribuer à modifier le tableau sans pour autant faire progresser le temps. Le tableau de conversation et son évolution sont abordés en section 7.2 dans les exemples de fonctionnement du module développé dans cette thèse.

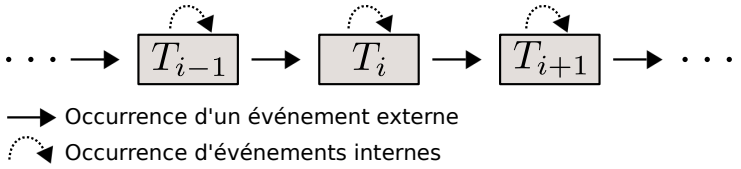

Figure 6.4: Principe de l'évolution du tableau de conversation en fonction de l'occurrence d'événements

Opérations réalisables sur le tableau de conversation

Cette section présente les fonctionnalités générales du tableau de conversation ainsi que les *opérations* (de type Operation) réalisables sur celui-ci pour les engagements propositionnels et les engagements en action.

Fonctionnalités générales du tableau de conversation Le tableau de conversation inclut un *commitment store*, i.e., un ensemble d'engagements *partiellement ordonné*. Ce tableau supporte trois fonctionnalités spécifiées dans le tableau 6.2. Il est possible d'interroger le tableau

1. Nous avons limité notre travail à celui de la dimension *Task* (cf. section 5.2.1).

sur l'appartenance (ou la non-appartenance) de n'importe quel engagement. Notons que le paramètre temps est facultatif lors de ces recherches puisque celui-ci ne se contente que d'établir le temps du dernier changement d'état de l'engagement. En outre, il est possible de déterminer si deux engagements du tableau sont en relation d'ordre ou non, i.e., si un engagement est *prioritaire* vis-à-vis d'un autre. L'*opération* de prioritarisation permet d'ordonner deux engagements du tableau. Après l'application de cette opération, les deux engagements sont ordonnés dans le tableau de conversation. Les jeux de dialogue ont une influence sur la prioritarisation des engagements *dialogiques* que nous précisons en section 6.2.1.

Nom	Fonctionnalité	Préconditions	Résultat
Appartenance [a]	$T_i \vDash c$ c : COMMITMENT		vrai si c$\in T_i$, faux sinon
Non-appartenance	$T_i \nvDash c$ c : COMMITMENT		Équivalent à $\neg(T_i \vDash c)$
Priorité	$T_i \vDash c_1 \prec c_2$ c_1, c_2 : COMMITMENT	$T_i \vDash c_1$, $T_i \vDash c_2$	vrai si c_1 est prioritaire sur c_2, faux sinon
Prioritarisation	prio(c_1,c_2,T_i) c_1, c_2 : COMMITMENT	$T_i \vDash c_1$, $T_i \vDash c_2$ $\neg(T_i \vDash c_2 \prec c_1)$	$T_i \vDash c_1 \prec c_2$ (voir extension en section 6.2.1)

a. Lorsque le temps t n'est pas précisé, l'appartenance signifie qu'il existe un temps t tel que l'engagement appartient au tableau de conversation. Par exemple, $T_i \vDash \mathrm{C}(x,y,p,s) \iff \exists t$ t. q. $T_i \vDash \mathrm{C}(x,y,p,t,s)$.

TABLEAU 6.2: Fonctionnalités générales du tableau de conversation

Opérations sur le tableau de conversation Le tableau 6.3 présente les opérations de création et de suppression applicables sur le tableau de conversation pour des *engagements standards*. L'opération de *création* permet de passer un engagement de l'état inactif à l'état **Crt** dans le tableau courant. Après l'application de cette opération, le tableau de conversation contient l'engagement dont l'état est désormais **Crt** et le paramètre temporel est le temps courant. L'opération de suppression permet de *désactiver* n'importe quel engagement en action ou propositionnel actif (i.e., le faire passer à l'état **Ina**). Il diffère de l'*annulation* pour les engagements propositionnels (qui se contente de désactiver un engagement en état **Crt**). La différence est fine compte tenu des états que nous considérons. Néanmoins, dans la perspective de l'extension de notre modèle en discernant de nouveaux états, les deux opérations ont vocation à être distinguées. L'opération de suppression est principalement utilisée pour désactiver des engagements *dialogiques* dès lors que leur contexte local n'a plus lieu d'exister.

Enfin, le tableau 6.4 présente la spécification des opérations d'échec, de satisfaction et de violation d'engagements en action (dialogique ou extra-dialogique). Ces opérations permettent de changer l'état d'un engagement en action vers les états **Fal**, **Vio** et **Ful** en s'assurant que les conditions sont bien réunies (notamment en terme d'état courant de l'engagement). Par exemple, les préconditions des opérations de satisfaction et de violation spécifient que l'engagement doit être initialement dans l'état **Crt**.

Chaque opération prend le tableau de conversation courant T_i en paramètre. Par application partielle de ses autres arguments, chaque opération peut être réduite en une opération ne prenant que l'argument T_i. Par exemple, une fois l'interlocuteur et l'engagement fixés, l'unique paramètre

Nom	Opération	Préconditions	Effet(s)
Création	créer(x,c,T_i) c : StandardCommitment c=C$_{\{j\}}$(–,–,–,t,s)	$T_i \vDash c$ s=**Ina** $t \leq i$	$T_i \nvDash c$ $T_i \vDash$C$_{\{j\}}$(–,–,–,i,**Crt**) (=c') L'interlocuteur x crée l'engagement c' dans le tableau de conversation T_i.
Suppression	supprimer(x,c,T_i) c : StandardCommitment c=C$_{\{j\}}$(–,–,–,t,s)	$T_i \vDash c$ $s \neq$**Ina** $t \leq i$	$T_i \nvDash c$ $T_i \vDash$C$_{\{j\}}$(–,–,–,i,**Ina**) L'interlocuteur x supprime l'engagement c du tableau de conversation T_i.
Annulation	annuler(x, c, T_i) c : PropCommitment c=C(–,–,–,t,s)	$T_i \vDash c$ s=**Crt** $t \leq i$	$T_i \nvDash c$ $T_i \vDash$C(–,–,–,i,**Ina**) L'interlocuteur x annule l'engagement c du tableau de conversation T_i.

Tableau 6.3: Opérations générales sur les engagements en action et propositionnel (création, suppression et annulation). « – » dénote les champs inchangés dans les opérations. « $\{j\}$ » représente l'éventuel contexte dialogique de l'engagement.

Nom	Opération	Préconditions	Effet(s)
Échec	échec(x,c,T_i) c : ActionCommitment ou c : GameActionCommitment c=C$_{\{j\}}$(–,–,α,t,s)	$T_i \vDash c$ s=**Ina** $t \leq i$	$T_i \nvDash c$ $T_i \vDash$C$_{\{j\}}$(–,–,α,i,**Fal**) La tentative de créer c a échoué.
Satisfaction	satisfaire(c,T_i) c : ActionCommitment ou c : GameActionCommitment c=C$_{\{j\}}$(–,–,α,t,s)	$T_i \vDash c$ s=**Crt** $t \leq i$	$T_i \nvDash c$ $T_i \vDash$C$_{\{j\}}$(–,–,α,i,**Ful**) c est satisfait dans T_i.
Violation	violer(c,T_i) c : ActionCommitment ou c : GameActionCommitment c=C$_{\{j\}}$(–,–,α,t,s)	$T_i \vDash c$ s=**Crt** $t \leq i$	$T_i \nvDash c$ $T_i \vDash$C$_{\{j\}}$(–,–,α,i,**Vio**) c est violé dans T_i.

Tableau 6.4: Opérations sur les engagements en action (contextualisés ou non). « – » dénote les champs inchangés dans les opérations. « $\{j\}$ » représente l'éventuel contexte dialogique de l'engagement.

manquant de l'opération de création est le tableau de conversation. Cette *forme réduite* se révèle très pratique puisqu'elle nous permet de traiter uniformément les opérations réalisables sur le tableau de conversation. Nous exploitons cette caractéristique dans la rédaction des algorithmes de la suite de cette section.

Raccourcis d'écriture Afin d'alléger l'écriture et la lecture des spécifications de jeux de dialogue et de jeux de communication, nous omettons la plupart du temps le paramètre symbolisant le tableau de conversation courant dans les *opérations*. En outre, nous ne précisons pas entièrement les opérations de création, d'échec et d'annulation dans les spécifications pour les combinaisons conditionnelles ($\alpha \Rightarrow \beta$) et conditionnelles persistantes ($\alpha \overset{*}{\Rightarrow} \beta$). Nous nous contentons de préciser l'état de l'engagement tel qu'il va être créé dans le tableau. Ces raccourcis d'écriture sont illustrés dans les trois exemples suivants :

- $\alpha \Rightarrow$ créer(x,C$_{\{j\}}$(x,–,–,**Ina**)) se réécrit $\alpha \Rightarrow$ C$_{\{j\}}$(x,–,–,**Crt**)
- $\alpha \Rightarrow$ échec(x,C(x,α',**Ina**)) se réécrit $\alpha \Rightarrow$ C(x,α',**Fal**)
- $\alpha \Rightarrow$ annuler(x, C(x,p,**Crt**)) se réécrit $\alpha \Rightarrow$ C(x,p,**Ina**)

Évolution du tableau de conversation

Dans cette section, nous décrivons l'évolution des engagements en action dialogique et le mécanisme d'événement interne que nous considérons. Nous donnons deux algorithmes réalisant l'évolution du tableau de conversation suite à l'occurrence d'événements.

Évolution des engagements en action dialogique Nous n'optons pas pour un profil précis des engagements extra-dialogiques. L'établissement de la satisfaction ou de la violation d'un engagement en action *extra-dialogique* est problématique [Pasquier 2005] (p. 134). Il semble en effet difficile de déterminer un processus général de cet établissement. Celui-ci dépend des caractéristiques de l'action et de la situation dans laquelle se trouvent les interlocuteurs. Dans le cas particulier où les interlocuteurs sont en co-présence et que l'action sur laquelle un interlocuteur est engagé produit un résultat immédiat et observable pour les deux interlocuteurs (e.g., « allumer la lumière »), l'établissement peut être effectué sur une base événementielle comme pour les engagements en action dialogique. Néanmoins, ce cas est loin d'être général. La solution pratique communément adoptée consiste à concevoir un jeu de décharge où l'interlocuteur indique s'il a satisfait l'action (e.g., *« J'ai ajouté le mot-clé paludisme avec succès »*) ou non (voir, e.g., [Larsson 2002a] (p. 218)). Nous reportons la tâche de spécification de l'établissement de la satisfaction ou de la violation d'un engagement en action *extra-dialogique* au moment de la spécification de la tâche sur laquelle va intervenir l'agent dialogique.

Une description d'événement peut être *persistante* ou non. Elle peut également être *affectée*, *violée* ou *satisfaite* par l'occurrence d'un événement. Ces différents aspects sont capturés dans notre modèle par des prédicats définis dans le tableau 6.5. Nous discernons les descriptions d'événements entre celles qui sont *persistantes* et celles qui ne le sont pas. Une description persistante n'est jamais satisfaite par l'occurrence d'un événement. Dans notre modèle, seules les combinaisons de type « conditionnelle persistante » sont persistantes. Cette propriété a un impact sur l'évolution des engagements suite à l'occurrence d'un événement. En effet, un engagement dont le contenu est persistant n'est jamais satisfait (i.e., il ne passe jamais à l'état **Ful**). La relation d'*affection* permet de déterminer si une description est affectée par l'occurrence d'un événement.

Cette relation permet de distinguer les descriptions concernées par l'occurrence d'un événement de celles qui ne le sont pas. Une description simple est affectée si l'événement *correspond* à la description. Une négation est affectée si la description simple dont elle est composée est affectée. Un choix est affecté si au moins un de ses éléments l'est. Pour finir, une description conditionnelle est affectée si sa prémisse l'est. Une description est *violée* si c'est une négation qui est affectée par l'occurrence d'un événement. Ces trois relations nous permettent de définir la *satisfaction* d'une description d'événement. Une description d'événement *non persistante* est satisfaite par l'occurrence d'un événement si elle est affectée par cet événement sans être violée.

De la même manière, nous définissons deux prédicats permettant de récupérer une *opération* d'une description d'événement en fonction de l'occurrence d'un événement. Une description d'événement possède une opération si elle est *conditionnelle* et qu'elle est affectée par l'occurrence de l'événement. Dans ces conditions, il est alors possible d'obtenir l'opération détenue par la description d'événement.

Prédicat	Description (α)	Résultat
estPersistant(α)	α' $\stackrel{*}{\Rightarrow}$ β	vrai
	α_1 : DescÉvénement $\neg\alpha$', α_1'$\|\alpha_2$', α' $\Rightarrow$ β	faux
estAffectéPar(α, e)	α_1 : DescÉvénement	correspond(e, α_1)
	$\neg\alpha$'	estAffectéPar(α', e)
	α_1'$\|\alpha_2$'	estAffectéPar(α_1', e) *ou* estAffectéPar(α_2', e)
	α' $\Rightarrow$ β, α' $\stackrel{*}{\Rightarrow}$ β	estAffectéPar(α', e)
estViolépar(α, e)	$\neg\alpha$'	estAffectéPar(α', e)
	α_1 : DescÉvénement α_1'$\|\alpha_2$' α' $\Rightarrow$ β, α' $\stackrel{*}{\Rightarrow}$ β	faux
estSatisfaitPar(α, e)	estAffectéPar(α, e) et $\neg$estViolépar(α, e) et $\neg$estPersistant(α)	
possèdeOperation(α, e)	α' $\Rightarrow$ β, α' $\stackrel{*}{\Rightarrow}$ β	estAffectéPar(α, e)
	α_1 : DescÉvénement $\neg\alpha$', α_1'$\|\alpha_2$'	faux
obtenirOperation(α, e) : Operation	α' $\Rightarrow$ β, α' $\stackrel{*}{\Rightarrow}$ β	β (Précondition : possèdeOperation(α, e))

Tableau 6.5: Prédicats concernant les descriptions d'événement. Chaque prédicat est défini en fonction de la forme des descriptions (colonne « Description »). La valeur du prédicat est donnée dans la colonne « Résultat ». α : Description, e : Événement.

Nous définissons ces mêmes relations sur les *engagements en action dialogique* (cf. tableau 6.6). Un engagement est persistant dès lors que son contenu l'est. En outre, un engagement *actif* est affecté, violé et satisfait de la même manière que son contenu. Un engagement qui n'est pas dans un état actif ne peut pas être affecté, violé ou satisfait. La possession et l'obtention d'une opération depuis un engagement en action dialogique *actif* sont déléguées au contenu de l'engagement. Un tel engagement possède une opération s'il est actif et si son contenu possède une opération à réaliser sur le tableau de conversation en vertu de l'occurrence d'un événement. Similairement, il est possible d'obtenir l'opération à appliquer de l'engagement actif, s'il en possède une.

Prédicat	Définition
estPersistant(c)	estPersistant(α)
estAffectéPar(c, e)	s=**Crt** et estAffectéPar(α, e)
estViolépar(c, e)	s=**Crt** et estViolépar(α, e)
estSatisfaitPar(c, e)	s=**Crt** et estSatisfaitPar(α, e)
possèdeOperation(c, e)	s=**Crt** et possèdeOperation(α, e)
obtenirOperation(c, e)	si s=**Crt** alors obtenirOperation(α, e)

Tableau 6.6: Relations entre un engagement en action dialogique et un événement. c : GameActionCommitment, c=$C_j(-,-,\alpha,s)$, e : Événement

Muni de ces relations, nous pouvons présenter l'algorithme 6.1 réalisant l'évolution du tableau de conversation pour un de ses engagements dialogiques en action suite à l'occurrence d'un événement (interne ou externe). L'engagement en action dialogique peut ne pas être affecté par l'occurrence d'un événement, il est alors laissé inchangé. Dans le cas contraire, l'engagement peut être *violé* ou non par l'événement. S'il est violé, l'opération de violation est appliquée sur l'engagement. S'il n'est pas violé, l'engagement est satisfait à condition qu'il ne soit pas persistant. Si cet engagement est porteur d'une opération à réaliser sur le tableau de conversation, celle-ci est récupérée et appliquée. L'évolution totale du tableau de conversation à la suite d'un événement est réalisée par l'application de l'algorithme 6.1 pour chaque engagement dialogique en action *initialement présent* (i.e., avant que toute modification n'intervienne). On note cette opération :

$$\text{evolution}(e : \text{Événement}, T_i : \text{TableauDeConversation})$$

Occurrence d'événements internes Chaque modification du tableau de conversation par un *événement externe* peut déclencher l'occurrence d'*événements internes*. Nous optons pour un mécanisme très simple d'événement interne. Chaque événement interne est défini par une *condition de déclenchement* sur l'état du tableau de conversation (*trigger*). Dès lors que cette condition est vérifiée, un événement interne est généré et appliqué sur le tableau de conversation. Un événement interne n'est généré qu'une fois pour un temps t donné. Par exemple, un événement interne engagementPropositionnelExistant(y, p) peut être généré dès lors que y s'engage sur la proposition p. Son *trigger* est alors la fonction qui vérifie l'appartenance de l'engagement [2] :

$$\text{is} : \text{TableauDeConversation} \mapsto \text{is} \models C(y,p,\mathbf{Crt})$$

Les événements internes sont principalement utilisés pour vérifier l'atteinte des conditions d'entrée et de sortie des jeux de dialogue.

On suppose l'existence d'une procédure permettant de gérer l'évolution du tableau de conversation à partir d'un ensemble de générateurs d'événement interne notée :

$$\text{evolutionViaInternalEvent}(\text{ensGenerator} : \text{EnsInternalEventGenerator}, \text{is} : \text{TableauDeConversation})$$

Cette procédure fait évoluer le tableau de conversation via l'ensemble d'événements générés. Elle se charge de déterminer les nouveaux événements internes pouvant être générés et fait de nouveau évoluer le tableau de conversation jusqu'à ce qu'aucun événement interne nouveau ne soit généré.

2. « is » pour « Information State »

Algorithme 6.1 Algorithme de mise à jour d'un engagement dialogique en action suite à l'occurrence d'un événement.

Nom: EvolutionEngagementDialogiqueEnAction
Role: Réalise l'évolution du tableau de conversation pour un engagement dialogique en action donné suite à l'occurrence d'un événement.
Entrée: c : GameActionCommitment, e : Événement
Sortie: $\nearrow$
Entrée/Sortie: T_i : TableauDeConversation
Déclaration: op : Operation

```
debut
  si estAffectéPar(c, e) alors
    si estViolépar(c, e) alors
      violer(c,T_i)
    sinon
      si ¬estPersistant(c) alors
        satisfaire(c,T_i)
      finsi
      si possèdeOperation(c, e) alors
        // Récupération de l'opération déclenchée par l'occurrence de l'événement
        op ← obtenirOperation(c, e)
        // Application de l'opération au tableau de conversation
        op(T_i)
      finsi
    finsi
  finsi
fin
```

Mise à jour du tableau de conversation suite à un événement externe L'algorithme 6.2 présente les opérations strictement nécessaires à la mise à jour du tableau de conversation. Cette opération est réalisée en deux étapes essentielles. La première consiste à faire évoluer l'ensemble des engagements suite à l'occurrence de l'événement externe (via l'algorithme 6.1). La seconde vise à déclencher l'ensemble des événements internes générés par le nouvel état du tableau de conversation, qui vont eux-mêmes contribuer à modifier le tableau. La copie n'est pas stricto sensu indispensable dès lors que nous limitons le tableau de conversation à une vision où il est public et strictement partagé. Néanmoins, cette étape devient indispensable dès lors que la partie publique est relativisée du point de vue de l'interlocuteur (cf. section 2.2.1). En effet, il est souvent nécessaire de conserver les tableaux précédents afin de modéliser des mécanismes de retour en arrière suite à un *grounding* trop optimiste (voir, e.g., [Larsson 2003]). Nous faisons donc apparaître cette étape qui sera nécessaire dans des évolutions futures. Cet algorithme ne fait apparaître que les éléments relatifs à l'évolution des engagements du tableau. Néanmoins, d'autres opérations complémentaires peuvent être envisagées comme le stockage de l'événement externe dans l'historique de l'interaction. Ces opérations dépendent des champs supplémentaires au *commitment store* présents dans le tableau de conversation.

Algorithme 6.2 Algorithme de mise à jour du tableau de conversation suite à l'occurrence d'un événement externe.

Nom: EvolutionTableauDeConversation
Role: Réalise l'évolution des engagements du tableau de conversation suite à l'occurrence d'un événement externe.
Entrée: T_i : TableauDeConversation, e : ÉvénementExterne, ensGenerator : EnsInternalEventGenerator
Sortie: T_{i+1} : TableauDeConversation
Entrée/Sortie: ⁄
Déclaration: ⁄
debut
// Copie du tableau de conversation courant
$T_{i+1} \leftarrow$ copie(T_i)
// Évolution des engagements à partir de l'événement externe
evolution(e, T_{i+1})
// Évolution du tableau de conversation depuis les événements internes
evolutionViaInternalEvent(ensGenerator, T_{i+1})
fin

6.1.3 Interprétation des engagements du tableau de conversation

Le tableau de conversation représente les engagements pris par les différents interlocuteurs à un moment donné de l'interaction. Ces engagements représentent des *positions sur des propositions* (engagements propositionnels) et des engagements sur la réalisation de certaines *actions* (engagements en action dialogiques ou extra-dialogiques) ou de *règles de production* (engagements en action dialogique). Ces engagements vont contraindre le comportement des interlocuteurs. En particulier, les engagements en action dialogique vont contraindre le *comportement communicatif* d'un interlocuteur. Nous définissons dans cette section les notions d'événement dialogique *attendu*, *régulier* et *interdit*. La notion d'attente est illustrée en section 7.2 dans les

exemples de fonctionnement du système que nous avons développé. Nous notons $\alpha_1|\ldots|\alpha|\ldots|\alpha_n$ le fait que l'événement α apparaisse dans une alternative.

Un événement dialogique e est *attendu* par le tableau de conversation T_i si et seulement si $\exists\alpha$ telle que :

- $T_i \models \mathrm{C_j}(-,-,\alpha,\mathbf{Crt})$ ou $T_i \models \mathrm{C_j}(-,-,\alpha_1|\ldots|\alpha|\ldots|\alpha_n,\mathbf{Crt})$, et
- correspond(e, α)

On note le prédicat qui retourne vrai lorsque l'événement e est attendu par le tableau de conversation *is*, et faux sinon :

estAttendu(e : ÉvénementDialogique, is : TableauDeConversation)

De la même façon, le prédicat estAttenduPar(e : ÉvénementDialogique, j : Jeu, is : TableauDeConversation) retourne vrai lorsque l'événement dialogique e est attendu par le jeu (de dialogue ou de communication) j dans le tableau de conversation *is*.

Un événement dialogique e *viole* un engagement du tableau de conversation T_i si et seulement si $\exists\ \alpha$ telle que :

- $T_i \models \mathrm{C_j}(-,-,\neg\alpha,\mathbf{Crt})$, et
- correspond(e, α)

On note le prédicat qui retourne vrai lorsque l'événement e viole un engagement du tableau de conversation *is*, et faux sinon :

viole(e : ÉvénementDialogique, is : TableauDeConversation)

Un événement dialogique e est *régulier* vis-à-vis du tableau de conversation T_i si et seulement si :

- il ne viole aucun engagement (i.e., $\neg$viole(e, T_i)), ou
- s'il viole un engagement c_1, il est attendu par un engagement c_2 tel que c_2 est prioritaire sur c_1.

Formellement,

($\neg$viole(e, T_i)) ou ($\forall c \in T_i$ tel que estViolePar(c, e), $\exists c'$ tel que $T_i \models c' \prec c$ et estSatisfaitPar(c', e))

On note le prédicat qui retourne vrai lorsque l'événement e est régulier par rapport au tableau de conversation *is*, et faux sinon :

estRégulier(e : ÉvénementDialogique, is : TableauDeConversation)

Il est important de remarquer qu'un événement dialogique e peut être *régulier* et *non attendu*.

Un événement dialogique e est *interdit* par rapport au tableau de conversation T_i si et seulement si il n'est pas régulier. Formellement,

estInterdit(e : ÉvénementDialogique, is : TableauDeConversation) $\Longleftrightarrow$
$\neg$estRégulier(e, is)

6.2 Jeux de dialogue et jeux de communication

Cette section définit la structure de jeux de dialogue adoptée dans notre modèle (cf. section 6.2.1) puis la structure de jeux de communication (cf. section 6.2.2). Nous abordons les combinaisons possibles entre les jeux de dialogue et l'établissement des jeux via le mécanisme de contextualisation (cf. section 6.2.3). Enfin, nous donnons une interprétation des jeux de dialogue représentés dans le tableau de conversation (cf. section 6.2.4).

6.2.1 Modèle de jeu de dialogue

Nous définissons dans un premier temps la structure de jeux de dialogue que nous envisageons. Puis nous spécifions les opérations sur les jeux de dialogue ayant un impact sur le tableau de conversation.

Définition

Un jeu de dialogue (de type JeuDeDialogue) est un couple de la forme $\langle \text{type}, \text{sujet} \rangle$ où *type* appartient à l'ensemble des types de jeux de dialogue existants (e.g., requête, interrogation) et *sujet* forme le but du jeu dans le langage d'expression du sujet du jeu. Nous notons usuellement un jeu sous la forme type(sujet) (e.g., requête(α) réfère au jeu de requête dont le but est l'action α).

Un jeu de dialogue est défini en terme d'engagements sociaux. C'est un quintuplet caractérisé pour l'*initiateur* et le *partenaire* par des :

conditions d'entrée : celles-ci spécifient l'état dans lequel doit se trouver le tableau de conversation permettant l'entrée dans le jeu de dialogue. Cet état réfère à des engagements *extra-dialogiques*. Par exemple, les conditions d'entrée du jeu de requête d'action spécifient que le partenaire ne doit pas être déjà engagé sur l'action demandée.

conditions de sortie subdivisées en :

- **conditions de succès** : celles-ci spécifient l'état du tableau de conversation qui correspond au succès. Cet état réfère à des engagements extra-dialogiques.
- **conditions d'échec** : celles-ci spécifient l'état du tableau de conversation correspondant à l'échec du jeu. Cet état réfère à des engagements extra-dialogiques.

règles : elles spécifient les enchaînements d'actes attendus ou interdits dans le contexte du jeu en terme d'engagements en action *dialogiques*.

effets : il s'agit de règles particulières qui spécifient les *effets contextualisés des actes de dialogue* en terme de production d'engagements extra-dialogiques.

Afin d'illustrer cette structure, nous présentons le jeu de dialogue de requête qui permet à l'initiateur du jeu de proposer la réalisation d'une action au partenaire qui peut accepter ou refuser (cf. tableau 6.7). Ce jeu a émergé du motif précédemment observé (cf. section 5.2.2). Le jeu est présenté sous la forme d'une *structure bilatérale* qui présente pour l'*initiateur* et le *partenaire* les conditions d'entrée, de succès et d'échec, ainsi que les règles et les effets. Le type du jeu est « requête » et son but est une action α. Les conditions sont spécifiées sous la forme d'engagements *extra-dialogiques* qui doivent tenir dans le tableau de conversation pour que les conditions soient atteintes. Les conditions d'entrée du partenaire spécifient qu'il ne doit pas être déjà engagé sur l'action α, aussi bien sur sa réalisation que sur sa non-occurrence. Les conditions

de succès sont identiques pour les deux interlocuteurs. Elles stipulent que le jeu est un succès dès lors que le partenaire est engagé sur la réalisation de l'action α. Au contraire, les conditions d'échec précisent pour les deux interlocuteurs que le jeu est un échec si la tentative d'engager le partenaire sur α échoue. Chaque règle et chaque effet est spécifié sous la forme d'un engagement en action *dialogique* que nous ne précisons pas. Ainsi, la règle « request(x, α) » de l'initiateur est équivalente à C$_{\text{j}}(x$,request(x, α),**Ina**$)$ [3]. Les règles de l'initiateur du jeu spécifient qu'il est engagé à jouer un acte *Request*. Les règles du partenaire du jeu précisent qu'il est engagé à jouer un acte *AcceptRequest* ou *DeclineRequest* à condition que l'initiateur remplisse son engagement. Enfin, les effets du partenaire indiquent que jouer un acte *AcceptRequest* a pour effet dans le *contexte de ce jeu* de l'engager sur la réalisation de α. Au contraire, jouer un acte *DeclineRequest* a pour effet dans le *contexte de ce jeu* de l'engager sur l'échec de la tentative de la création de l'engagement sur α. Ces effets produisent des engagements *extra-dialogiques* qui permettent d'atteindre les conditions de succès ou d'échec.

j=**requête**(α)		
	Initiateur (x)	Partenaire (y)
Entrée		C(y,α,**Ina**) et C(y,$\neg\alpha$,**Ina**)
Succès	C(y,α,**Crt**)	C(y,α,**Crt**)
Échec	C(y,α,**Fal**)	C(y,α,**Fal**)
Règles	request(x, α)	request(x, α) $\Rightarrow$ C$_{\text{j}}(y$,acceptRequest(y, α)|declineRequest(y, α),**Crt**$)$
Effets		acceptRequest(y, α) $\Rightarrow$ C(y,α,**Crt**)
		declineRequest(y, α) $\Rightarrow$ C(y,α,**Fal**)

Tableau 6.7: Jeu de dialogue de requête. α : Action

À chaque jeu de dialogue j est associé un ensemble d'*événements internes* relatifs aux conditions d'entrée et de sortie du jeu (présentés en détail dans le tableau B.1 de l'annexe B.1). Ces événements sont déclenchés dès lors que des conditions d'entrée, d'échec ou de succès sont atteintes par le tableau de conversation. Ces événements internes sont utilisés dans le jeu de contextualisation (cf. section 6.2.3). Par exemple, le déclenchement de l'événement atteintSuccèsInitiateur(j) engage l'initiateur à proposer la sortie du jeu.

Opérations sur les jeux de dialogue

Avant de présenter les opérations sur les jeux de dialogue, nous affinons notre définition de la relation de priorité entre les engagements. Pour deux jeux de dialogue ouverts dans le tableau de conversation T_i, on note la relation de priorité plus succinctement : $T_i \vDash j_2 \prec j_1$ (signifiant que j_2 est prioritaire sur j_1) [4]. Dans le cas particulier des engagements en action dialogique, la priorité au niveau des jeux contextualisant provoque la même priorité au niveau des engagements. Ainsi :

$$T_i \vDash j_2 \prec j_1 \Rightarrow \forall c_1 = \text{C}_{\text{j}_1}(\text{–},\text{–},\text{–},\text{–},\text{–}),\ \forall c_2 = \text{C}_{\text{j}_2}(\text{–},\text{–},\text{–},\text{–},\text{–}) \text{ alors } T_i \vDash c_2 \prec c_1$$

Cette propriété étend la relation de priorité précédemment définie (cf. section 6.1.2). En outre, un jeu de dialogue j_2 est prioritaire sur un jeu de dialogue j_1 s'il est *emboîté* dans celui-ci (cf. section 6.2.3). La relation de priorité a un impact sur la fermeture des jeux de dialogue.

3. La règle est en état inactif tant qu'elle n'a pas été effectivement créée, i.e. jusqu'au moment de l'ouverture du jeu.

4. Cette notation est équivalente à $T_i \vDash \text{C}(\text{–},j_2,\textbf{Open}) \prec \text{C}(\text{–},j_1,\textbf{Open})$.

Le tableau 6.8 présente les opérations réalisables sur les engagements conjoints sur un jeu de dialogue. Ces opérations correspondent aux transitions entre les différents états dans lesquels ce type d'engagement peut se trouver (**Ina**, **Sugg**, **Open**, **Closed**).

Avant d'être conjointement établi (état **Open**), un engagement de jeu passe par un état suggéré (**Sugg**). Dans cet état, l'engagement peut être unilatéralement annulé par un interlocuteur (e.g., celui à qui le jeu a été suggéré). Les opérations de suggestion et d'annulation de jeu permettent, respectivement, de créer l'engagement de jeu dans l'état **Sugg** et d'annuler cette création.

La création de l'engagement de jeu correspond à l'établissement *conjoint* du jeu de dialogue. Cette opération ne se résume pas à modifier l'état de l'engagement conjoint sur le jeu. En effet, elle est accompagnée de la création des *règles* et des *effets* du jeu dans le tableau de conversation. Formellement,

$$\forall\, c_j \in \text{règlesInitiateur}(j) \cup \text{règlesPartenaire}(j) \cup \text{effetsInitiateur}(j) \cup \text{effetsPartenaire}(j),\ \text{créer}(x,c_j,T_i)$$

Enfin, la fermeture d'un jeu de dialogue provoque la clôture du contexte local engendré par le jeu, modélisé par les engagements dialogiques. L'opération de fermeture conduit à changer l'état de l'engagement conjoint mais aussi à la suppression de tous les engagements contextualisés par j. Formellement,

$$\forall c \text{ tel que } c{=}\mathrm{C_j}(\text{–},\text{–},\text{–},\text{–},\text{–}) \text{ et } T_i \vDash c,\ \text{supprimer}(x,c,T_i)$$

En outre, la fermeture d'un jeu de dialogue j provoque la clôture de tous les jeux emboîtés dans celui-ci. Formellement,

$$\forall j\text{' tel que } T_i \vDash j' \prec j,\ \text{fermerJeu}(x,j',T_i)$$

Nom	Opération	Préconditions	Effet(s)
Suggestion	suggJeu(x,j,ini,par,T_i)	$T_i \vDash$ C({ini, par},j,**Ina**)	$T_i \nvDash$ C({ini, par},j,**Ina**) $T_i \vDash$ C({ini, par},j,i,**Sugg**)
Annulation	annulerSuggJeu(x,j,T_i)	$T_i \vDash$ C(–,j,**Sugg**)	$T_i \nvDash$ C(–,j,**Sugg**) $T_i \vDash$ C(–,j,i,**Ina**)
Création	créerJeu(x,j,T_i)	$T_i \vDash$ C(–,j,**Sugg**)	$T_i \nvDash$ C(–,j,**Sugg**) $T_i \vDash$ C(–,j,i,**Open**) + Création des règles + Création des effets
Fermeture	fermerJeu(x,j,T_i)	$T_i \vDash$ C(–,j,**Open**)	$T_i \nvDash$ C(–,j,**Open**) $T_i \vDash$ C(–,j,i,**Closed**) + Suppression des engagements contextualisés + Annulation des jeux emboîtés

Tableau 6.8: Opérations générales sur les engagements conjoints sur un jeu de dialogue. x représente l'interlocuteur qui réalise l'action. « ini » représente l'initiateur du jeu, « par » représente le partenaire du jeu. j : JeuDeDialogue.

6.2.2 Modèle de jeu de communication

Les *jeux de communication* que nous utilisons sont identiques à ceux de l'approche de Maudet (cf. section 3.2.3) et sont définis par un *type* (e.g., contextualisation, évaluation). Ils sont dédiés à gérer des processus généraux d'interaction (comme la compréhension mutuelle, la gestion des tours de parole, ...), et sont *toujours* activés. Ils sont spécifiés sous la forme d'engagements en action dialogiques persistants : $C_j(-,\alpha \overset{*}{\Rightarrow} \beta,\mathbf{Crt})$ (où α : DESCRIPTION et β : OPERATION). Deux jeux de communication sont considérés dans notre travail : le jeu de contextualisation (cf. section 6.2.3) et le jeu d'évaluation formalisant les motifs du corpus COGNI-CISMEF (cf. section 6.3.3).

6.2.3 Combinaisons et établissement de jeux de dialogue

Cette section décrit les combinaisons de jeux de dialogue considérées dans notre modèle, ainsi que le jeu de communication de contextualisation.

Combinaisons de jeux de dialogue

À la suite de Maudet (cf. section 3.2.3), nous considérons trois types de combinaisons observées dans le dialogue humain : le *séquencement*, le *pré-séquencement* et l'*emboîtement*. D'autres combinaisons peuvent être envisagées comme, par exemple, l'itération (répétition du jeu n fois) ou le choix [McBurney 2002].

Séquencement Le séquencement de jeux de dialogue, noté j_2 ; j_1, est une combinaison permettant de capturer la négociation de l'ouverture d'un jeu (*« Je peux te poser une question ? – Laisse moi t'en poser une avant ! – OK. »*). Une telle combinaison permet de proposer de jouer le jeu j_2 jusqu'à l'atteinte de ses conditions de succès, avant de jouer le jeu j_1. Dans notre modèle, établir une combinaison j_2 ; j_1 consiste à *ouvrir* le jeu j_2 tandis que le jeu j_1 reste *suggéré*. Une fois le jeu j_2 joué avec succès, le jeu j_1 est *ouvert*. Nous n'imposons aucune contrainte sur le séquencement de jeux de dialogue.

Pré-séquencement Le pré-séquencement informé de jeux de dialogue est un séquencement particulier (cf. section 3.2.3). De la même manière que pour le séquencement, un jeu j_1 est pré-séquencé par j_2, noté $j_2 \rightsquigarrow j_1$, si j_2 est *ouvert* alors que j_1 est *suggéré*. À la différence d'un séquencement, un pré-séquencement par un jeu j_2 sert à établir les conditions d'entrée du jeu j_1. En d'autres termes, les conditions de succès de j_2 doivent être incluses dans les conditions d'entrée de j_1 [Maudet 2001]. Les pré-séquencements valides doivent être spécifiés au niveau du domaine. Nous posons le prédicat :

$$\text{estValidePreSéquence}(j_2 : \text{JEUDEDIALOGUE}, j_1 : \text{JEUDEDIALOGUE})$$

Celui-ci retourne vrai si le pré-séquencement de j_1 par j_2 ($j_2 \rightsquigarrow j_1$) est autorisé, faux sinon.

Emboîtement L'emboîtement est une combinaison de deux jeux de dialogue *ouverts* dont l'un est le *jeu parent* et l'autre est le *jeu fils* (cf. section 3.2.3). Nous notons l'emboîtement du jeu j_2 dans j_1 : $j_2 < j_1$. Dans un contexte d'emboîtement, les engagements du jeu parent et du jeu fils restent actifs. Néanmoins, les engagements contractés dans le jeu emboîté sont *prioritaires*. Ceci est capturé dans notre modèle par la relation de priorité : $T_i \vDash j_2 \prec j_1$ (cf. section 6.2.1). La

création d'un emboîtement de j_2 dans j_1 conduit à donner la priorité aux engagements de j_2. La fermeture du jeu parent conduit à la fermeture des jeux emboîtés.

Il semble indésirable de laisser une liberté totale des emboîtements possibles [Mann 1988, Lewin 2000, Hulstijn 2000a, Maudet 2001]. Néanmoins, les sources d'emboîtements de jeux sont multiples (structure intentionnelle, contexte dialogique, etc.) et difficile à spécifier de manière générale [Maudet 2001]. Face à cette difficulté, nous requérons la spécification des emboîtements de jeux de dialogue permis au niveau du domaine via le prédicat :

$$\text{estValideEmboitement}(j_2 : \text{JeuDeDialogue}, j_1 : \text{JeuDeDialogue})$$

Celui-ci retourne vrai si l'emboîtement de j_2 dans j_1 ($j_2 < j_1$) est autorisé, faux sinon. Cette spécification peut prendre la forme de *plans de dialogue* [Larsson 2002a].

Langage de désignation de jeux de dialogue L'introduction de combinaisons de jeux de dialogue impose la spécification d'un langage de désignation des jeux et de leurs combinaisons. Les éléments de ce langage sont de type LJeux. Il est défini de la manière suivante :

- si j : JeuDeDialogue, alors j : LJeux
- si j_1 : JeuDeDialogue et j_2 : JeuDeDialogue, alors j_2 ; j_1 : LJeux
- si j_1 : JeuDeDialogue et j_2 : JeuDeDialogue, alors $j_2 \rightsquigarrow j_1$: LJeux
- si j_1 : JeuDeDialogue et j_2 : JeuDeDialogue, alors $j_2 < j_1$: LJeux

Établissement de jeux de dialogue

Nous adoptons le *jeu de communication* de contextualisation proposé par Maudet (cf. section 3.2.3) et basé sur la proposition de [Reed 1998], qui traduit le mécanisme de négociation proposé par [Mann 2002]. Ce jeu introduit des actes de niveau *contextualisation* regroupés dans le tableau 6.9. Le contenu sémantique de ces actes est de type LJeux (cf. section 6.2.3).

Dimension	Niveau	Fonctions
Dialogue Structure	Contextualisation	*prop.entrée*, *ref.entrée*, *acc.entrée*, *poursuit*, *prop.sortie*, *ref.sortie*, *acc.sortie*

Tableau 6.9: Fonctions des actes de dialogue de contextualisation

Le principe du jeu de contextualisation est présenté en figure 6.5. L'initiateur du jeu de dialogue peut proposer l'entrée du jeu (*prop.entrée*). Le jeu est alors dans l'état suggéré (**Sugg**). Cette suggestion peut être acceptée (*acc.entrée*) ou refusée (*ref.entrée*) par le partenaire. Le refus conduit à désactiver le jeu. L'acceptation conduit à établir le jeu, i.e. l'ouvrir (état **Open**). Le jeu est alors joué. Chaque acte de dialogue produit dans le cadre du jeu provoque un acte *poursuit* de niveau contextualisation. Un des interlocuteurs peut proposer la sortie du jeu (*prop.sortie*) qui peut être refusée (*ref.sortie*) ou acceptée (*acc.sortie*). L'acceptation conduit à la fermeture du jeu (état **Closed**).

La spécification complète du jeu de contextualisation est disponible en annexe B.2. Il s'agit d'une adaptation dans notre formalisme d'engagement social du jeu proposé par Maudet.

Enfin, c'est au niveau de ce jeu de contextualisation que les effets des combinaisons de jeux de dialogue sont implantés. Par exemple, c'est dans ce jeu que l'atteinte des conditions de succès [5]

5. L'atteinte des conditions de succès est détectée via le déclenchement de l'*événement interne* atteintSuccès(j).

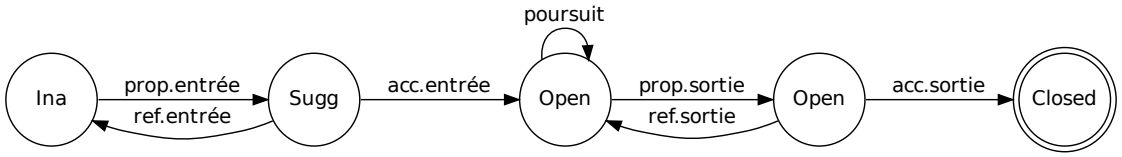

FIGURE 6.5: Principe du jeu de communication de contextualisation pour un jeu de dialogue. Chaque état de l'automate représente l'état de l'engagement conjoint sur le jeu de dialogue. Chaque transition représente l'occurrence d'un acte de dialogue de contextualisation concernant le jeu de dialogue.

d'un jeu permet d'engager un interlocuteur à jouer l'acceptation d'entrée dans le jeu suivant du séquencement.

6.2.4 Interprétation des jeux de dialogue dans le tableau de conversation

En plus de représenter les engagements en proposition et en action des interlocuteurs (cf. section 6.1.3), le tableau de conversation à un instant donné représente la structure intermédiaire du dialogue sous la forme de jeux de dialogue *suggérés*, *ouverts*, ou *fermés*[6]. L'ensemble des jeux de dialogue est partiellement ordonné par la relation de priorité. À chaque instant, il existe un ensemble de *jeux saillants*. Il s'agit des jeux de dialogue *ouverts* qui ne possèdent aucun jeu emboîté. Formellement, j est *saillant* si et seulement si :

- $T_i \models C(-, j, \textbf{Open})$, et
- $\nexists j'$ tel que $T_i \models j' \prec j$

Nous définissons alors la relation de saillance qui permet de déterminer si un jeu de dialogue j est saillant par rapport à un tableau de conversation :

$$\text{estSaillant}(j : \textsc{JeuDeDialogue}, \text{is} : \textsc{TableauDeConversation})$$

Le *jeu de dialogue courant* est le jeu saillant ouvert le plus tard. Il s'agit du jeu saillant dont le temps de passage à l'état **Open** est le plus récent.

Muni de la relation de saillance, il est alors possible de définir une relation de priorité entre les événements dialogiques attendus par le tableau de conversation. Un événement dialogique e est *prioritaire* dans le tableau de conversation T_i si :

- $\exists\ j$: JeuDeDialogue tel que estAttenduPar(e, j, T_i) et
- estSaillant(j, T_i)

En outre, nous supposons que les événements attendus dans le cadre d'un *jeu de communication* sont toujours prioritaires. On note le prédicat qui retourne vrai lorsque l'événement e est prioritaire dans le tableau de conversation *is*, et faux sinon :

$$\text{estPrioritaire}(e : \textsc{ÉvénementDialogique}, \text{is} : \textsc{TableauDeConversation})$$

6.3 Spécification empirique de jeux depuis le corpus Cogni-CISMeF

Dans cette section, nous présentons la spécification des jeux de dialogue et de communication selon le formalisme que nous venons de présenter, depuis les motifs dialogiques que nous avons

6. Ceci est illustré en section 7.2 dans les exemples de fonctionnement du système que nous avons développé.

observés dans le corpus COGNI-CISMEF (cf. section 5.2). Nous défendons tout d'abord l'intérêt pratique de restreindre notre formalisation à l'usage d'une sémantique réduite dépendante du domaine (cf. section 6.3.1). Ensuite, nous décrivons les événements dialogiques utilisés dans nos jeux qui consistent en des coups dialogiques issus de la taxonomie DIT++ (cf. section 6.3.2). Puis, nous présentons les jeux que nous avons définis depuis les motifs (cf. section 6.3.3). Pour terminer, nous étudions la couverture des jeux spécifiés en les confrontant au corpus de validation constitué durant le processus d'extraction (cf. section 6.3.4).

6.3.1 Représentation du contenu sémantique

Dans cette section, nous justifions notre choix d'une représentation sémantique simplifiée, puis nous en présentons sa spécification.

Choix d'une représentation sémantique

Deux grandes écoles divergent dans leurs objectifs vis-à-vis du modèle de dialogue, concernant le choix du formalisme de représentation du contenu sémantique des énoncés [Prévot 2004] (p. 234) [Landragin 2013]. La première école est principalement représentée par les travaux en linguistique (e.g., [Asher 2003, Ginzburg 1996, Prévot 2004]). Elle vise l'analyse, la compréhension et la modélisation fines de phénomènes linguistiques soigneusement choisis. En conséquence, ce type d'approche n'aborde pas la modélisation du dialogue dans sa globalité mais se restreint aux phénomènes linguistiques considérés. La deuxième école vise à modéliser globalement la gestion du dialogue. Elle inclut des travaux de modélisation très générale des différents aspects de la communication de la part de psychologues et de linguistes (e.g., [Allwood 1995, Clark 1996]), et des travaux visant à développer des *systèmes de dialogue* Homme-Machine (e.g., [Larsson 2002a, Jokinen 2010, Landragin 2013]). C'est dans cette école que se situent nos travaux. Les travaux de la première école peuvent être préalables aux travaux de la deuxième comme l'illustre l'utilisation des travaux de Ginzburg pour le développement de GODIS (cf. section 2.2.3).

Les études linguistiques font directement face à la problématique de la *formalisation du langage naturel*, généralement dans le cadre d'une logique (modale, temporelle, ...). Ceci est illustré par des formalismes comme la « Discourse Representation Structure » (DRT) [Kamp 1981] et la « Segmented DRT » (SDRT) [Asher 2003]. Au contraire, les travaux modélisant la gestion du dialogue dans le but de concevoir des systèmes de dialogue se limitent souvent à une sémantique simple, dépendante du domaine d'application [Larsson 2002a, Jokinen 2010, Landragin 2013]. Cela s'explique par la complexité du processus de compréhension du langage naturel qui nécessite en pratique de se restreindre à un domaine ainsi que par la volonté de manipuler une représentation tractable pour l'implémentation du système. Nous rejoignons [Larsson 2002a] en posant qu'il n'est pas nécessaire de conserver une représentation sémantique complète des énoncés pour un système de dialogue dont le domaine est limité. Nous restreignons notre modèle à l'usage d'une *représentation sémantique réduite* dont la granularité est dépendante du domaine. L'intuition de ce choix est que pour un domaine donné, la représentation complète d'un énoncé est souvent superflue. Dans le cas d'un agent assistant sur un moteur de recherche, il ne semble pas nécessaire de représenter le fait que c'est l'agent qui doit donner des informations sur le système, c'est implicitement toujours le cas. Ce choix a l'avantage de simplifier drastiquement la spécification de la sémantique pour un domaine donné.

Notre modèle repose sur la représentation sémantique réduite proposée par Larsson pour plusieurs raisons. La première est que cette sémantique est fondée sur le travail solide de Ginzburg

(cf. section 2.2.3). Ensuite, cette sémantique est *implémentée* et utilisée dans plusieurs prototypes de systèmes de dialogue [Larsson 2002a, Kronlid 2008]. Puis, cette sémantique a permis de représenter avec succès plusieurs domaines (réservation d'une agence de voyage [Larsson 2002a], lecteur multimédia [Hjelm 2005], etc.). Enfin, le dernier aspect, et non des moindres, est que l'intérêt de cette sémantique pour modéliser les questions intervenant dans le corpus Cogni-CISMeF a été défendu dans [Loisel 2008].

Néanmoins, les simplifications opérées au niveau du formalisme représentant le contenu sémantique ne sont pas sans conséquence sur les capacités du système de dialogue. L'usage d'une représentation sémantique réduite ne permet pas de prendre en compte des nombreux phénomènes linguistiques (polysémie, métonymie, etc.). En outre, elle n'offre qu'un pouvoir de représentation limité. Par exemple, notre sémantique n'intègre pas les connecteurs logiques et les quantificateurs permettant de représenter des énoncés comme *« Tous les chemins. »* ou *« Un jour, le manuscrit sera terminé. »*. Enfin, une représentation sémantique simple n'offre pas (ou peu) de mécanismes d'inférence qui permettent de représenter les sens implicites d'un énoncé.

Notons que la sémantique utilisée pour modéliser nos jeux de dialogue depuis nos motifs d'interaction n'est pas la seule disponible (voir, e.g., [Prévot 2004] pour un parcours des sémantiques sur les questions et leurs réponses). Nos motifs extraits peuvent servir de base pour de nouvelles formalisations. Notre formalisation des jeux de dialogue impose un système de dialogue supportant la sémantique que nous avons choisie.

Spécification de la sémantique

La sémantique que nous utilisons est une représentation *simple* et *réduite*, basée sur la *logique des prédicats* sans quantification, étendue avec la lambda-abstraction des propositions afin de représenter les questions.

Nous distinguons quatre grandes catégories sémantiques : les propositions (Proposition), les questions (Question), les actions (Action) et les formules sur les jeux (LJeux) (cf. section 6.2.3).

Atomes et expressions Les atomes de notre sémantique sont ceux de la logique des prédicats, à savoir :

— $\textsc{Pred}_n$ où $n \in \mathbb{N}$ dénote un prédicat d'arité n,

— Ind représente les constantes (e.g., **cœur**, **varicelle**, etc.), et

— Var regroupe les variables (e.g., x, y, Q, P, ...).

Dans notre représentation, toutes les constantes sont associées à au moins une *catégorie sémantique*. Par exemple, la constante **paludisme** peut être associée à la catégorie sémantique « mot-clé ». Ce simple mécanisme de catégorie sémantique est *dépendant du domaine*. Il permet de déterminer les propositions qui ont un sens (e.g., ajouterMotCle(**paludisme**)) de celles qui n'en ont pas (e.g., manger(**paludisme**)).

Une *expression valide* dans notre représentation est alors définie comme étant soit une proposition, soit une question, soit une action ou soit une formule du langage des jeux. Formellement,

expr : Expression ssi expr : Proposition ou expr : Question ou expr : Action
ou expr : LJeux

Dans la suite de cette section, nous présentons la sémantique des propositions et la sémantique des questions. Nous ne donnons aucune spécification de la sémantique des actions, dépendante du domaine d'application.

Propositions Les propositions sont représentées par des prédicats d'arité n dont les arguments ne comportent aucune variable. Une proposition est alors définie de la façon suivante :

expr : PROPOSITION ssi

- expr : PRED_n (ne comportant aucune variable), ou
- expr=$\neg p$ où p : PROPOSITION, ou
- expr=**fail** (q) où q : QUESTION

fail (q) est une proposition spéciale permettant de représenter le fait que la recherche d'une réponse à une question q est un échec (i.e., aucune réponse n'a pu être trouvée compte tenu de l'état d'information courant).

Le tableau 6.10 présente quelques exemples de propositions avec notre représentation sémantique.

Sémantique	Exemples
humain/machine	*« Je suis un humain. » / « Je suis une machine. »*
nom(**guillaume**)	*« Je m'appelle Guillaume. »*
definition(**eczema**)	*« Eczéma possède une définition dans la terminologie. »*
hyponyme(**coeur**, **myocarde**)	*« Un hyponyme de coeur est : myocarde. »*
traduction(**varicelle**, **anglais**, **chickenpox**)	*« Une traduction de varicelle en anglais est : chickenpox. »*

TABLEAU 6.10: Exemples de propositions dans notre représentation sémantique

Nous introduisons la relation de correction entre deux propositions :

correction (p_1, p_2) avec p_1 : PROPOSITION et p_2 : PROPOSITION

Cette relation est vraie lorsque la proposition p_2 est une correction de la proposition p_1 (e.g., **correction** (beau, pleut), **correction** (motcle(**paludisme**), motcle(**cholesterol**)), etc.), sinon elle est fausse. Cette dernière ne doit pas être confondue avec la négation de la proposition. La correction permet généralement de mettre en relation des propositions qui ne varient que par leurs arguments. Par exemple, la proposition ¬motcle(paludisme) n'est pas considérée comme une correction de motcle(paludisme). Cette relation est *dépendante du domaine.*

Questions Nous distinguons trois types de question : les questions oui/non (YNQ, symbolisées par une proposition précédée d'un « ? »), les questions à choix multiples (ALTQ, un ensemble de questions oui/non) et enfin les questions ouvertes (WHQ, lambda-abstraction de proposition dont le lambda est remplacé par « ? »). Formellement, une question est définie de la manière suivante :

expr : QUESTION ssi expr : YNQ ou expr : WHQ ou expr : ALTQ, avec :

- q : YNQ si q=?p avec p : PROPOSITION

- q : ALTQ si q=$\{q_1, \ldots, q_n\}$ avec $n > 1$, et $\forall i$, $1 \leq i \leq n$ alors q_i : YNQ
- q : WHQ si q=$?X.\text{pred}_n(\text{c}_1, \ldots, \text{c}_{n-1}, X)$ avec $n \geq 1$, où X : VAR, $\forall i$ t.q. $1 \leq i \leq n-1$ alors c_i : IND et pred_n : PRED_n (q est dite paramétrée par les constantes c_i)

Le tableau 6.11 présente quelques exemples de question.

Sémantique	Exemple
{?patient, ?etudiant, ?medecin}	*« Êtes-vous un patient, un étudiant en médecine ou un médecin ? »*
?etudiant	*« Êtes-vous un étudiant en médecine ? »*
?definition(**eczema**)	*« Est-ce que "eczéma" possède une définition dans la terminologie ? »*
$?X$.nom(X)	*« Comment vous appelez-vous ? »*
$?X$.synonyme(**paludisme**, X)	*« Quel est un synonyme de paludisme ? »*
$?X$.traduction(**varicelle**, **anglais**, X)	*« Quelle est une traduction de varicelle en anglais ? »*

TABLEAU 6.11: Exemples de question dans notre représentation sémantique

Relations entre questions et propositions Enfin, nous reprenons deux relations entre questions et propositions proposées par [Larsson 2002a] issues de la théorie QUD : la relation de résolution et la relation de pertinence (cf. section 2.2.3). Ces relations sont définies dans le tableau 6.12.

Question	**Réponses...**	
	...résolvantes	**...pertinentes non résolvantes**
$?X.\text{pred}_n(\text{c}_1, \ldots, \text{c}_{n-1}, X)$	$\text{pred}_n(\text{c}_1, \ldots, \text{c}_{n-1}, \mathbf{a})$	$\neg\text{pred}_n(\text{c}_1, \ldots, \text{c}_{n-1}, \mathbf{a})$
$?p$	p, $\neg p$	néant
$\{?p_1, \ldots, ?p_n\}$	p_i, $1 \leq i \leq n$	$\neg p_i$, $1 \leq i \leq n$

TABLEAU 6.12: Relations de résolution et de pertinence (n'inclut pas les restrictions sémantiques s'appliquant aux constantes)

La *relation de résolution* détermine si une proposition résout une question. On simplifie la définition proposée dans QUD en ne prenant pas en compte le point de vue relatif de l'agent. Cette relation est supposée *partagée* par les interlocuteurs. Nous notons cette relation :

$$\textbf{resolves}\,(p, q) \text{ avec } p : \text{PROPOSITION}, q : \text{QUESTION}$$

Elle est vraie lorsque la proposition p résout la question q, sinon elle est fausse. Cette relation est définie dans le tableau 6.12. Notons que les restrictions sémantiques jouent un rôle essentiel dans les conditions de résolution puisqu'elles déterminent les propositions correctement construites. Elles ne sont pas représentées dans le tableau. La définition de cette relation est *dépendante du domaine*.

La *relation de pertinence* est une simplification de la notion d'« à propos » proposée dans QUD. Elle permet d'identifier les propositions qui sont en rapport avec une question sans nécessairement la résoudre. Elle se présente sous la même forme que la relation de résolution :

$$\textbf{relevant}\,(p, q) \text{ avec } p : \text{PROPOSITION}, q : \text{QUESTION}$$

Elle est vraie si la proposition p est pertinente vis-à-vis de la question q, sinon elle est fausse. Notons que toutes les propositions résolvantes sont pertinentes :

$$\textbf{resolves}\,(p,q) \Rightarrow \textbf{relevant}\,(p,q)$$

Le tableau 6.12 regroupe les réponses pertinentes non résolvantes. De la même façon que la relation de résolution, cette relation est dépendante du domaine. Nous ajoutons la *relation de pertinence stricte*. Une proposition est strictement pertinente vis-à-vis d'une question si elle est pertinente sans être résolvante. Formellement,

$$\textbf{strictlyRelevant}\,(p,q) \iff \textbf{relevant}\,(p,q) \text{ et } \neg\, \textbf{resolves}\,(p,q)$$

6.3.2 Actes de dialogue

Les actes de dialogue que nous considérons sont *multidimensionnels* et *multiniveaux*. Ils proviennent de l'analyse du corpus Cogni-CISMeF (cf. chapitre 5). Les fonctions de ces actes sont regroupées dans le tableau 6.13 qui présente pour chaque fonction, sa dimension et son niveau.

Dimension	Niveau	Fonctions
Task	Standard	*Inform*, *Answer*, *Disconfirm*, *Confirm*, *Agreement*, *Disagreement*, *Correction*, *AcceptCorrection*, *DeclineCorrection* *PropositionalQuestion*, *CheckQuestion*, *PosiCheck*, *NegaCheck*, *SetQuestion*, *ChoiceQuestion* *Offer*, *AcceptOffer*, *DeclineOffer*, *Request*, *AcceptRequest*, *DeclineRequest*, *Suggestion*, *AcceptSuggestion*, *DeclineSuggestion*
Auto-feedback	Exécution	*ExecNegativeAutoFB*

Tableau 6.13: Fonctions des actes de dialogue considérés dans notre formalisation

Suite à notre restriction à la dimension *Task* dans le processus d'extraction (cf. section 5.2.1), nous nous limitons aux actes de niveau *standard* (i.e., les actes « noyaux » dans l'approche multiniveaux, cf. section 1.2.2), à l'acte *ExecNegativeAutoFB* de niveau exécution et aux actes de niveaux *contextualisation* (cf. section 6.2.3). Notre formalisation ne prend pas en compte les niveaux inférieurs (du niveau d'attention jusqu'au niveau d'évaluation) qui permettent d'établir la fonction et le contenu d'un acte, et de vérifier l'adéquation de l'occurrence de l'acte par rapport à l'état courant du dialogue. Ces processus nécessitent une étude particulière (voir, e.g., [Traum 1994a, Larsson 2003, Bunt 2007]). Ils ont vocation à s'intégrer dans notre modèle via les jeux de communication, et nécessitent des études complémentaires. Notons que les jeux de dialogue n'interviennent qu'aux niveaux où les actes ont été complètement établis (exécution et standard).

Chaque *événement dialogique* d'énonciation d'un acte de dialogue est un quadruplet ⟨dimension, fonction, locuteur, contenu sémantique⟩ (excepté *Correction*, *AcceptCorrection* et *DeclineCorrection* que nous traitons dans la suite du document). Nous omettons la dimension à ce stade du modèle puisqu'il n'existe aucune ambiguïté sur la dimension d'une fonction. Nous notons ce quadruplet : fonction(locuteur, contenu). Par exemple, inform(x, p) représente

l'événement d'énonciation par le locuteur x de l'acte de dialogue réalisé dans la dimension *Task* dont la fonction est *Inform* et le contenu sémantique est p.

Le type de contenu sémantique des actes est synthétisé dans le tableau 6.14. Celui-ci est conditionné par la catégorie de la fonction de l'acte. Les actes de transfert d'information possèdent un contenu sémantique de type PROPOSITION, les actes de discussion d'action ont un contenu de type ACTION, et l'acte *ExecNegativeAutoFB* est associé à un contenu sémantique de type QUESTION. Enfin, les actes de recherche d'information se voient attribuer un type de contenu sémantique dépendant de leur fonction. Les fonctions permettant de poser des questions polaires sont associées au type YNQ (question oui/non). L'acte *SetQuestion* possède un contenu sémantique de type WHQ (question ouverte), et l'acte *ChoiceQuestion* a un contenu sémantique de type ALTQ (question à choix multiples).

Fonction	Type de contenu sémantique
Inform, *Answer*, *Disconfirm*, *Confirm*, *Agreement*, *Disagreement*	PROPOSITION
PropositionalQuestion, *CheckQuestion*, *PosiCheck*, *NegaCheck*	YNQ
SetQuestion	WHQ
ChoiceQuestion	ALTQ
ExecNegativeAutoFB	QUESTION
Offer, *AcceptOffer*, *DeclineOffer*, *Request*, *AcceptRequest*, *DeclineRequest*, *Suggestion*, *AcceptSuggestion*, *DeclineSuggestion*	ACTION

TABLEAU 6.14: Type de contenu sémantique des actes de dialogue

L'événement provoqué par l'énonciation d'un acte de dialogue de *Correction* est un quintuplet $\langle$dimension, fonction, locuteur, contenu 1, contenu 2$\rangle$. Cet événement porte sur deux propositions (de type PROPOSITION). La première représente la proposition corrigée par la deuxième. Par exemple, correction$(x, \text{beau}, \text{pleut})$ [7] représente l'événement dialogique d'énonciation par le locuteur x de l'acte de dialogue de *Correction* de la proposition beau par pleut. Dans l'esprit des approches contextuelles, nous avons ajouté deux actes *AcceptCorrection* et *DeclineCorrection* de la même forme que *Correction*, et qui représentent respectivement, une acceptation de la correction (marquée dans le motif dialogique par une simple *Agreement*) ou un refus de la correction. Cette décision est motivée par le fait qu'affirmer un accord après une correction est selon nous différent d'affirmer un accord après un acte informatif (sans correction comme *Inform*, *Answer*, etc.). Comme ces deux acceptations provoquent des effets différents sur le tableau de conversation, elles doivent être représentées par des actes différents.

6.3.3 Jeu de communication et jeux de dialogue

Cette section présente le jeu de communication et les jeux de dialogue spécifiés depuis les motifs dialogiques observés dans le corpus COGNI-CISMEF.

7. beau est le prédicat représentant la proposition « Il fait beau. » et pleut est le prédicat représentant la proposition « Il pleut ».

Jeu de communication

Nous considérons deux jeux de communication dans notre modèle : le jeu de contextualisation précédemment évoqué (cf. section 6.2.3) et le jeu d'évaluation. Les effets d'un jeu de communication sont spécifiés sous la forme $\mathrm{C_j}(-,\alpha \overset{*}{\Rightarrow} \beta,\mathbf{Crt})$ signifiant que chaque interlocuteur est engagé sur les règles du jeu de communication j de manière persistante.

Le *jeu d'évaluation* vise à capturer les effets des actes de transfert d'information supposés valides dans toute conversation. Il est noté « ev » dans les engagements dialogiques. Ce jeu permet aux interlocuteurs de s'accorder sur l'état du monde. Il intègre deux objectifs. Premièrement, il permet de définir les *effets directs* de l'occurrence de certains actes de dialogue sur le tableau de conversation. Ces effets sont toujours valides pendant le dialogue. Les effets directs des actes de dialogue sont regroupés dans le tableau 6.15. La première ligne du tableau définit l'effet propositionnel de l'acte *Inform*. Elle indique que l'occurrence de cet acte engage son locuteur sur son contenu sémantique. Notons que la fonction *Inform* est la fonction parente de l'ensemble des fonctions de transfert d'information. Cette hiérarchie représente un cas de multifonctionnalité simultanée (cf. section 5.1.1). Ainsi, répondre à une question (*Answer*) est aussi informer de la réponse, tout comme exprimer un désaccord (*Disagreement*) permet d'informer de ce désaccord. Ces fonctions spécialisent la fonction *Inform*. Selon nous, les actes *Answer* et *Agreement* possèdent le même effet direct que l'acte *Inform*. Cependant, l'acte *Disagreement* spécialise cet effet. Il n'engage pas son interlocuteur sur la proposition formant son contenu sémantique mais sur sa négation. Précisons que les actes *Confirm* et *Disconfirm* ont un effet dépendant du jeu de dialogue dans lequel ils prennent part, et ne voient pas leurs effets définis dans le jeu d'évaluation. Notons le faible nombre d'effets directs. Ceci est tout à fait normal puisque les autres effets des actes sont définis dans le cadre des jeux de dialogue.

α	β
$\mathrm{inform}(x,p)$	$\mathrm{C}(x,p,\mathbf{Crt})$
$\mathrm{f}(x,p)$	$\mathrm{C}(x,p,\mathbf{Crt})$
$\mathrm{disagreement}(x,p)$	$\mathrm{C}(x,\neg p,\mathbf{Crt})$

$\mathrm{f} \in \{Answer, Agreement\}$

Tableau 6.15: Jeu de communication d'évaluation : effets directs

Le second objectif du jeu d'évaluation est de modéliser les motifs dialogiques d'accord et de correction observés dans notre corpus et qui permettent aux interlocuteurs de s'accorder sur l'état du monde (cf. section 5.2.2). Ces motifs sont capturés par les effets présentés dans le tableau 6.16. La correspondance entre les motifs dialogiques et la spécification est immédiate. L'occurrence d'un acte informatif (*Inform* ou *Answer*) engage l'autre interlocuteur à affirmer son accord sur ce qui a été dit (*Agreement*), son désaccord (*Disagreement*) ou à proposer une correction au locuteur (*Correction*). Nous introduisons une contrainte sémantique en spécifiant que le partenaire y s'engage à jouer un acte de *Correction* dont les contenus sémantiques sont en relation de correction.

L'effet de l'acte de correction est particulier et est défini dans le tableau 6.17. Une correction de p par s conduit à engager le locuteur sur $\neg p$ (une correction est également un désaccord sur la proposition initiale) et à engager le locuteur sur s. Le partenaire s'engage alors à accepter une correction ou à la refuser. Une acceptation de correction a pour effets d'*annuler* l'engagement sur la proposition corrigée, et de créer un engagement sur la proposition de correction. Un refus

α	β
f(x,p)	$C_{ev}(y,\text{agreement}(y,p)\|\text{disagreement}(y,p)\|\text{correction}(y,p,s),\mathbf{Crt})$ avec **correction** (p,s)

f $\in \{Inform, Answer\}$

Tableau 6.16: Jeu de communication d'évaluation

de correction amène à ne pas changer les engagements propositionnels des interlocuteurs. Ce dernier acte a été ajouté par rapport au motif dialogique initialement observé (cf. section 5.2.2) dans un souci de complétude. Il n'a cependant pas été observé dans notre corpus.

α	β
correction(x,p,s)	C(x,$\neg p$,**Crt**) C(x,s,**Crt**) $C_{ev}(y,\text{acceptCorrection}(y,p,s)\|\text{declineCorrection}(y,p,s),\mathbf{Crt})$
acceptCorrection(x,p,s)	C(x,p,**Ina**) C(x,s,**Crt**)
declineCorrection(x,p,s)	

Tableau 6.17: Jeu de communication d'évaluation : cas de la correction

Jeux de dialogue

Dans cette section, nous présentons la modélisation des motifs dialogiques de recherche d'information et de discussion d'action sous la forme de jeux de dialogue.

Vérification d'une proposition Nous avons vu que la vérification de la valeur de vérité d'une proposition se décline en quatre motifs dialogiques (vérification, vérification positive, vérification négative et question oui/non). Ces motifs ont conduit à la spécification de trois jeux de dialogue qui visent tous à demander au partenaire de déterminer la valeur de vérité d'une proposition. Ces trois jeux de dialogue possèdent les mêmes conditions d'entrée, de succès et d'échec, et diffèrent par leurs *règles* et leurs *effets*. Ces conditions sont regroupées dans le tableau 6.18. Les *conditions d'entrée* de ces jeux spécifient que le partenaire ne doit pas être déjà engagé sur la proposition vérifiée ni sur sa négation, et qu'il ne doit pas être engagé sur le fait qu'il ne connaît pas la réponse. Les *conditions de succès* sont atteintes dès lors que le partenaire est engagé sur la proposition vérifiée ou sur sa négation. Enfin, les *conditions d'échec* indiquent que le jeu est un échec lorsque le partenaire s'engage sur le fait qu'il ne peut pas trouver de réponse.

	Initiateur (x)	Partenaire (y)
Entrée		C(y,p,**Ina**) et C(y,$\neg p$,**Ina**) et C(y,**fail** (q),**Ina**)
Succès	C(y,p,**Crt**) ou C(y,$\neg p$,**Crt**)	C(y,p,**Crt**) ou C(y,$\neg p$,**Crt**)
Échec	C(y,**fail** (q),**Crt**)	C(y,**fail** (q),**Crt**)

Tableau 6.18: Conditions d'entrée, de succès et d'échec des jeux de vérification, de vérification négative et de question oui/non. p est le *but* du jeu, q=?p.

Le premier jeu de dialogue que nous présentons est celui de vérification, disponible dans le

tableau 6.19. Il regroupe les motifs dialogiques de vérification d'une information et de vérification positive dont les actes initiatifs sont liés par une relation de spécialisation. Le *but du jeu* est la proposition p. Les *règles* précisent que l'initiateur s'engage à jouer un acte *CheckQuestion* ou *PosiCheck* dont le contenu sémantique ($?p$) est de type YNQ. Une fois cet acte joué, le partenaire est engagé à jouer un acte *Confirm*, *Disconfirm* ou *ExecNegativeAutoFB*. Les *effets* stipulent que, *dans le contexte du jeu*, jouer un acte *Confirm* engage le locuteur sur son contenu sémantique, jouer un acte *Disconfirm* engage sur la négation de son contenu sémantique, et que jouer un acte *ExecNegativeAutoFB* engage sur la proposition **fail** (q). Les effets conduisent à créer des engagements extra-dialogiques permettant d'atteindre les conditions de succès ou d'échec.

j=**vérification**(p)		
	Initiateur (x)	Partenaire (y)
Règles	checkQuestion(x,q)|posiCheck(x,q)	checkQuestion(x,q)|posiCheck(x,q) $\Rightarrow$ C$_j$(y,confirm(y,p)|disconfirm(y,p)| execNegativeAutoFB(y,q),**Crt**)
Effets		confirm(y,p) $\Rightarrow$ C(y,p,**Crt**) disconfirm(y,p) $\Rightarrow$ C(y,$\neg p$,**Crt**) execNegativeAutoFB(y,q) $\Rightarrow$ C(y,**fail** (q),**Crt**)

Tableau 6.19: Jeu de dialogue de vérification. p est le *but* du jeu, q=$?p$.

Ce jeu de dialogue ne couvre pas le cas d'altération du projet conjoint observé dans le motif de vérification et qui représente 15.15% des cas observés. L'altération du projet conjoint conduit à modifier le jeu courant. Il s'agit d'une situation liée à la contextualisation du jeu et qui ne concerne pas directement la définition du jeu. Nous y revenons en section 6.3.4. La prise en compte de ce phénomène nécessite un approfondissement de la notion de contextualisation. Ce travail est conservé pour des approfondissements futurs.

Le jeu de vérification négative est présenté dans le tableau 6.20. Il formalise le motif dialogique de vérification négative qui vise à confirmer ou infirmer la négation d'une proposition (e.g., *« PAS mardi ? – Je te confirme, pas mardi ! »*). Ce jeu est similaire à celui de vérification à l'exception de l'acte initiatif (*NegaCheck*) et des effets des actes *Confirm* et *Disconfirm*. Dans le contexte de ce jeu, si le partenaire joue un acte *Confirm*, il va confirmer la négation de la proposition, et donc s'engager sur cette négation. À l'inverse, jouer un acte *Disconfirm* conduit le partenaire à s'engager sur la proposition initiale.

j=**vérificationNegative**(p)		
	Initiateur (x)	Partenaire (y)
Règles	negaCheck(x,q)	negaCheck(x,q) $\Rightarrow$ C$_j$(y,confirm(y,p)|disconfirm(y,p)| execNegativeAutoFB(y,q),**Crt**)
Effets		confirm(y,p) $\Rightarrow$ C(y,$\neg p$,**Crt**) disconfirm(y,p) $\Rightarrow$ C(y,p,**Crt**) execNegativeAutoFB(y,q) $\Rightarrow$ C(y,**fail** (q),**Crt**)

Tableau 6.20: Jeu de dialogue de vérification négative. p est le *but* du jeu, q=$?p$.

Le jeu d'interrogation polaire est présenté dans le tableau 6.21. Il formalise le motif dialogique de question oui/non. Les *règles* spécifient que l'initiateur du jeu est engagé à jouer un acte *PropositionalQuestion*. À la suite de cet acte, le partenaire est engagé à jouer un acte *Answer* confirmant ou infirmant la proposition, ou un acte *ExecNegativeAutoFB*. Ici, les effets ne définissent que celui de l'acte *ExecNegativeAutoFB* identique à ceux des jeux de dialogue précédents. L'effet de l'acte *Answer* est déjà défini dans le jeu de communication d'évaluation qui factorise les effets communs à tous les jeux de dialogue.

j=**interrogationPolaire**(p)		
	Initiateur (x)	Partenaire (y)
Règles	propositionalQuestion(x, q)	propositionalQuestion(x, q) $\Rightarrow$ C$_j$(y,answer(y, p)|answer($y, \neg p$)|execNegativeAutoFB(y, q),**Crt**)
Effets		execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)

Tableau 6.21: Jeu de dialogue d'interrogation polaire. p est le *but* du jeu, q=?p.

Interrogation ouverte et interrogation à choix multiples Les motifs de question ouverte et de question à choix multiples ont chacun donné jour à un jeu de dialogue, respectivement, d'interrogation ouverte et d'interrogation à choix multiples. Ces deux jeux possèdent une structure sous-jacente commune liée à la sémantique des conditions de résolution choisie (cf. figure 6.6). Chaque jeu débute par un acte initiatif joué par l'initiateur. À la suite de cet acte, le partenaire peut donner autant de réponses strictement pertinentes (i.e., pertinentes et non résolvantes) qu'il souhaite. Le jeu se termine avec succès dès qu'une réponse résolvante est enfin donnée, ou sur un échec si le partenaire indique qu'il ne trouve pas de réponse.

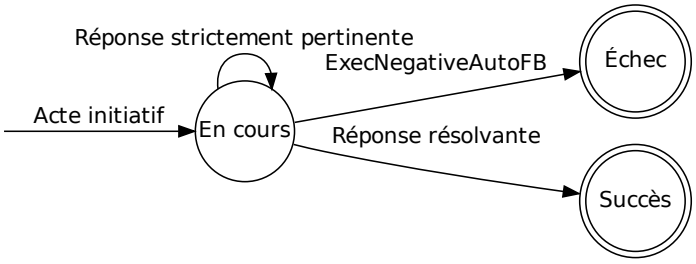

Figure 6.6: Structure sous-jacente commune aux jeux de dialogue d'interrogation ouverte et d'interrogation à choix multiples

Le jeu d'interrogation ouverte est donné dans le tableau 6.22. Le but de ce jeu est une question q telle que q : WHQ. Les *conditions d'entrée* précisent que le partenaire ne doit pas déjà être engagé sur une proposition qui résout la question, ou sur le fait qu'il ne connaît pas de réponse à la question. Les *conditions de succès* sont atteintes quand le partenaire est engagé sur une proposition qui résout la question. Les *conditions d'échec* indiquent que le jeu est un échec lorsque le partenaire s'engage sur le fait qu'il ne peut pas trouver la réponse. Les *règles* précisent que l'initiateur s'engage à jouer un acte initiatif *SetQuestion*. Ensuite, le jeu suit la structure commune précédemment décrite. Les effets stipulent que, *dans le contexte de ce jeu*, jouer un acte *ExecNegativeAutoFB* engage le partenaire sur la proposition **fail** (q).

Le jeu d'interrogation à choix multiples est donné en annexe B.3. Contrairement au jeu pré-

j=**interrogationOuverte**(q)		
	Initiateur (x)	Partenaire (y)
Entrée		C(y,p,**Ina**) avec **resolves** (p, q) et C(y,**fail** (q),**Ina**)
Succès	C(y,p,**Crt**) avec **resolves** (p, q)	C(y,p,**Crt**) avec **resolves** (p, q)
Échec	C(y,**fail** (q),**Crt**)	C(y,**fail** (q),**Crt**)
Règles	setQuestion(x, q)	setQuestion$(x, q) \Rightarrow$ C$_j$(y,answer(y, p)| execNegativeAutoFB(y, q),**Crt**) avec **relevant** (p, q) answer$(y, s) \stackrel{*}{\Rightarrow}$ C$_j$(y,answer(y, p)| execNegativeAutoFB(y, q),**Crt**) avec **strictlyRelevant** (s, q), **relevant** (p, q)
Effets		execNegativeAutoFB$(y, q) \Rightarrow$ C(y,**fail** (q),**Crt**)

Tableau 6.22: Jeu de dialogue d'interrogation ouverte. q : WHQ

cédent, le but est une question q telle que q : ALTQ. Les conditions d'entrée, de succès, d'échec et les effets sont identiques au jeu d'interrogation ouverte. La différence principale tient en l'acte initiatif qui est un acte *ChoiceQuestion*. Les règles traduisent alors le mécanisme précédemment décrit. Notons que les contraintes sémantiques sont différentes du jeu d'interrogation ouverte en raison de la différence de type de la question.

Discussion d'action Les trois motifs dialogiques d'action ont conduit à la formalisation de deux jeux de dialogue directifs (suggestion et requête) et un jeu de dialogue promissif (offre).

Le jeu de dialogue de suggestion permet à l'initiateur de proposer la réalisation d'une action au partenaire qui peut alors accepter ou refuser. Il est donné dans le tableau 6.23. Ce jeu formalise le motif de suggestion. Son but est une action α. Les *conditions* d'entrée du jeu de suggestion spécifient que le partenaire ne doit pas être déjà engagé sur l'action suggérée α, i.e., ni sur son occurrence ni sur sa non-occurrence. Les *conditions de succès* sont identiques pour les deux interlocuteurs : elles affirment que le partenaire doit s'engager à réaliser α. Symétriquement, les *conditions d'échec* sont atteintes si la tentative d'engager le partenaire sur α échoue. Les *règles* spécifient que l'initiateur du jeu est engagé à jouer un acte *Suggestion*. Le partenaire est engagé à jouer un acte *AcceptSuggestion* ou *DeclineSuggestion* si l'initiateur remplit son engagement. Enfin, les *effets* définissent que jouer un acte *AcceptSuggestion* ou *DeclineSuggestion*, dans le contexte de ce jeu, engage le partenaire sur la réalisation de α ou sur l'échec de la création de l'engagement sur la réalisation de α.

Le jeu de dialogue de requête est similaire au jeu de suggestion (les actes doivent être substitués par *Request*, *AcceptRequest* et *DeclineRequest*). Sa spécification complète a servi d'exemple afin d'illustrer la structure de jeu de dialogue (cf. section 6.2.1).

Enfin, le jeu d'offre permet à l'initiateur de proposer de réaliser une action pour le partenaire qui peut accepter ou refuser. Il est donné dans le tableau 6.24 et formalise le motif d'offre. Le but de ce jeu est une action α. Les *conditions d'entrée* du jeu d'offre spécifient que l'*initiateur* ne doit pas être engagé sur l'action offerte α, i.e., ni sur son occurrence, ni sur sa non-occurrence. Les *conditions de succès* précisent pour les deux interlocuteurs que le jeu est un succès si l'initiateur contracte l'engagement de réaliser α. À l'inverse, les *conditions d'échec* stipulent que le jeu est un

j=**suggestion**(α)		
	Initiateur (x)	Partenaire (y)
Entrée		C(y,α,**Ina**) et C(y,$\neg\alpha$,**Ina**)
Succès	C(y,α,**Crt**)	C(y,α,**Crt**)
Échec	C(y,α,**Fal**)	C(y,α,**Fal**)
Règles	suggestion(x,α)	suggestion(x,α) $\Rightarrow$ C$_j$(y,acceptSuggestion(y,α)| declineSuggestion(y,α),**Crt**)
Effets		acceptSuggestion(y,α) $\Rightarrow$ C(y,α,**Crt**) declineSuggestion(y,α) $\Rightarrow$ C(y,α,**Fal**)

Tableau 6.23: Jeu de dialogue de suggestion. α : Action

échec si la tentative de l'initiateur de s'engager sur α échoue. Les *règles* déclarent que l'initiateur est engagé à jouer un acte *Offer*. Le partenaire est engagé à accepter cette offre (*AcceptOffer*) ou à la refuser (*DeclineOffer*) dès le moment où elle est faite. Enfin, les *effets* définissent que jouer un acte *AcceptOffer* ou *DeclineOffer* dans le contexte de ce jeu engage l'initiateur sur la réalisation de α ou sur l'échec de la création de l'engagement sur la réalisation de α.

j=**offre**(α)		
	Initiateur (x)	Partenaire (y)
Entrée	C(x,α,**Ina**) et C(x,$\neg\alpha$,**Ina**)	
Succès	C(x,α,**Crt**)	C(x,α,**Crt**)
Échec	C(x,α,**Fal**)	C(x,α,**Fal**)
Règles	offer(x,α)	offer(x,α) $\Rightarrow$ C$_j$(y,acceptOffer(y,α)| declineOffer(y,α),**Crt**)
Effets	acceptOffer(y,α) $\Rightarrow$ C(x,α,**Crt**) declineOffer(y,α) $\Rightarrow$ C(x,α,**Fal**)	

Tableau 6.24: Jeu de dialogue d'offre. α : Action

Synthèse Dans cette section, nous avons spécifié un jeu de communication (celui d'évaluation) et huit jeux de dialogue collaboratifs à partir des onze motifs dialogiques observés dans le corpus Cogni-CISMeF. La finalité de ces jeux est la modification des états des engagements extra-dialogiques (propositionnel ou en action). Le tableau 6.25 présente les jeux à la disposition d'un interlocuteur x afin de modifier l'état courant d'un engagement extra-dialogique. Par exemple, si un interlocuteur souhaite s'engager envers son partenaire sur la réalisation d'une action (e.g., « Je te sers un café ? »), il peut choisir le jeu d'offre. Ce tableau montre la complétude du modèle vis-à-vis de la formalisation des engagements choisie.

Nous avons laissé de côté les états **Ful** et **Vio** pour les engagements en action car l'atteinte de ces états dépend de l'application. Nous constatons que l'*annulation* des engagements propositionnels (état **Crt** vers **Ina**) n'est que partiellement prise en compte. Il n'existe actuellement aucun jeu permettant à un interlocuteur d'annuler un de ses engagements propositionnels. Ces limites dans notre modèle proviennent du fait que les seuls cas d'annulation observés dans le corpus Cogni-CISMeF sont liés à l'acte de *Correction*. En outre, ces cas sont marginaux (cf. section 5.2.2).

Engagement	État courant	État désiré	Jeu(x)
C(x,y,α)	**Ina**	**Crt**	Offre
	Ina	**Fal**	Offre
C(y,x,α)	**Ina**	**Crt**	Requête, Suggestion
	Ina	**Fal**	Requête, Suggestion
C(x,y,p)	**Ina**	**Crt**	Jeu d'évaluation (via *Inform* [a])
	Crt	**Ina**	–
C(y,x,p)	**Ina**	**Crt**	Jeux de recherche d'information
	Crt	**Ina**	Jeu d'évaluation (via *Correction*)

a. En supposant que tout acte *Inform* est attendu par le jeu d'évaluation.

TABLEAU 6.25: Jeux utilisables par un interlocuteur afin de modifier les états des engagements extra-dialogiques. α : ACTION, p : PROPOSITION

6.3.4 Couverture du corpus par les jeux définis

Cette section présente la mise en place, la réalisation et les résultats d'une étude visant à évaluer la couverture des règles des jeux définis dans notre modèle sur le corpus de référence.

Configuration du processus

Cette étude porte sur le *corpus de référence* (un tiers du corpus COGNI-CISMEF) constitué lors du processus d'extraction et qui n'a pas été analysé (cf. section 5.2.1). Le corpus de référence a été annoté avec DIT++ sur l'ensemble des dimensions. Il représente approximativement le tiers du corpus COGNI-CISMEF en terme de tours de parole, de segments fonctionnels et de fonctions communicatives.

Deux chercheurs de notre laboratoire ont pris part à cette tâche. Ces derniers n'étaient impliqués ni dans le processus d'extraction de motifs dialogiques, ni dans le processus de formalisation. Ils étaient néanmoins familiers avec les concepts d'actes de dialogue comme modèle de l'énoncé ainsi qu'avec la théorie QUD sur laquelle se base notre sémantique des questions. Avant d'entamer ce processus, nous leur avons présenté le schéma d'annotation DIT++ sur la base de la formation opérée lors du processus d'annotation. Nous avons ensuite présenté le jeu de communication d'évaluation et les 8 jeux de dialogue que nous avons définis. Les jeux ont été exposés comme des structures capturant les engagements que les interlocuteurs contractent durant le dialogue. En ce sens, nous avons expliqué aux deux chercheurs que les règles des jeux prescrivent des séquences d'actes de dialogue conventionnellement attendues. L'aspect déclaratif des jeux a clairement facilité la compréhension de ces structures. Chacun des jeux a été accompagné d'exemples de dialogue issus du corpus d'extraction et disponibles en annexe A.4.4. Ces exemples ont illustré les enchaînements d'actes prescrits par les jeux. Enfin, la sémantique des questions et les simplifications opérées vis-à-vis de la théorie de QUD leur ont été décrites.

L'objectif de cette étude est de confronter les enchaînements de tours de parole aux règles définies dans nos jeux de manière à quantifier leur couverture. Il a été demandé aux deux chercheurs de considérer chaque *enchaînement de tours de parole* motivé par une évolution sur la dimension *Task* ou sur la dimension *Auto-feedback* via la fonction *ExecNegativeAutoFB*. Ces enchaînements forment une séquence de segments fonctionnels incluant un segment produit par chaque interlocuteur. Pour chacune de ces séquences, les volontaires avaient pour tâche de vérifier

si l'*enchaînement de fonctions* qu'elle représente était décrit par un jeu de communication ou un jeu de dialogue (indépendamment de sa contextualisation). Si l'enchaînement n'est décrit que partiellement (comme un acte initiatif suivi d'un acte inattendu dans le cadre d'un jeu) alors il est décompté comme un enchaînement ne correspondant pas au jeu. Par exemple, une question ouverte qui est posée et ignorée par le partenaire est décomptée comme un enchaînement partiel ne correspondant pas au jeu d'interrogation ouverte. Afin de détecter les séquences de tours de parole et leur adéquation avec un jeu, il a été demandé aux deux chercheurs de ne considérer que les segments fonctionnels ayant une fonction annotée dans la dimension *Task* (ou une fonction *ExecNegativeAutoFB*). Néanmoins, ils avaient accès aux annotations sur l'ensemble des dimensions leur permettant ainsi de faire la part entre les enchaînements de tours de parole motivés par la gestion de l'interaction et ceux motivés par la gestion de la tâche. En outre, il leur a été demandé de tenir compte de la forme linguistique du segment qui fournit le contenu sémantique. La cohérence sémantique spécifiée par les jeux n'a été vérifiée qu'informellement (notamment pour les jeux d'interrogation ouverte et à choix multiples). Le jeu d'évaluation présente un cas particulier via l'ajout des actes *AcceptCorrection* et *DeclineCorrection*. Ces ajouts ont été opérés lors de la formalisation, a posteriori de l'annotation du corpus de référence avec le schéma DIT++. En conséquence, les deux chercheurs ont eu pour consigne de considérer toute fonction *Agreement* ou *Disagreement* suivant un acte de *Correction* comme équivalente, respectivement, aux fonctions *AcceptCorrection* et *DeclineCorrection*.

Ce processus permet de produire un *ratio* entre les enchaînements prescrits par un jeu et ceux partiellement prescrits. Ce ratio permet de quantifier la couverture des enchaînements de fonctions communicatives prescrits par les jeux que nous avons définis avec une prise en compte informelle de la sémantique, et en faisant abstraction du processus de contextualisation (la plupart du temps implicite dans les conversations humaines [Levin 1977,Mann 1988,Mann 2002]).

Avant de répartir les entretiens du corpus de référence entre les deux chercheurs, nous leur avons demandé de réaliser tous deux le processus sur l'entretien VD03 constitué de 102 tours de parole. Cet exercice a permis d'analyser la correspondance entre les ratios produits. Ce test a été concluant puisqu'aucune différence significative concernant les ratios n'a été observée entre les chercheurs. Ils se sont ensuite répartis les entretiens. Le chercheur 1 s'est chargé des entretiens AL05, AL09 et VD09 tandis que le chercheur 2 s'est occupé des entretiens AL02 et VD08 auxquels ont été ajouté les résultats de l'entretien VD03.

Résultats du processus

356 séquences ont été mises en évidence par les deux chercheurs. Nous obtenons des résultats similaires à ceux du corpus d'extraction en terme de tours de parole impliqués dans une séquence et en terme de proportion de catégories de jeux représentées. Dans le détail, les séquences correspondant à un enchaînement prescrit par un jeu couvrent approximativement deux tiers des tours de parole du corpus de référence, tandis que les séquences ne correspondant pas à un enchaînement couvrent environ 10% des tours de parole. Les tours de parole non couverts par une séquence correspondent à une évolution sur l'axe de l'interaction (env. 25% des tours de parole). La catégorisation des séquences dont l'enchaînement est prescrit par un jeu est similaire à celle des motifs dialogiques observés dans le corpus d'extraction. C'est-à-dire qu'environ 25% des séquences ont un enchaînement prescrits par le jeu d'évaluation (catégorie de transfert d'information). Les 75% des séquences restantes se répartissent de manière équilibrée entre la catégorie de recherche d'information (jeux de dialogue de vérification, vérification négative,

interrogation polaire, ouverte et à choix multiples) et la catégorie de discussion d'action (jeux de dialogue de requête, suggestion et d'offre).

Les résultats de ce processus sont donnés dans le tableau 6.26 qui présente pour chaque jeu la proportion de séquences observées dont l'enchaînement de fonctions communicatives est prescrit. La conclusion globale est que 83.4% des séquences détectées par les volontaires dans le corpus de référence ont pu être mises en correspondance avec les règles des jeux que nous avons spécifiées. Plus précisément, nous pouvons voir que les scores sont tous au-delà de 66% indiquant une adéquation fiable entre les dialogues que nous modélisons et les jeux définis.

Évaluation	Vérification	Vérification neg.	Int. polaire	Int. ouverte	Int. choix multiples
100%	69.7%	66.7%	66.7%	88.6%	87.5%

Suggestion	Requête	Offre
84.8%	72.7%	78.1%

TABLEAU 6.26: Résultats de l'étude de la couverture des jeux. Proportion de séquences qui correspondent à un enchaînement prescrit par un jeu.

En outre, nous avons étudié les 16.6% de cas où il n'était pas possible de trouver une correspondance. Nous avons identifié deux grandes catégories : les cas liés à la structure *inter-jeu* et les cas liés à la structure *intra-jeu*.

La première catégorie est elle-même décomposable en deux sous-catégories. Dans la première, le partenaire ignore délibérément l'énoncé de l'initiateur dans 10% des cas. C'est ce que Clark appelle un *retrait du projet conjoint* [Clark 1996] et peut être illustré par : *« – si vous voulez qu'on aille dessus on peut y aller (Offer) – je rappelle que le 13 était intéressant. (Inform) »* (VD03). La prise en compte dans notre modèle de ces cas se situe au niveau de la contextualisation du jeu de dialogue. Ici, la contextualisation est implicite : l'acte initiatif du jeu est également une proposition d'entrée dans le jeu (cf. section 3.1.3). La réponse à l'acte initiatif est implicitement un refus d'entrée dans le jeu de par le fait qu'elle ne constitue pas une contribution allant dans le sens du jeu. Dans la seconde sous-catégorie, le partenaire ouvre un *jeu emboîté* qui rend obsolète le *jeu parent* dans environ 45% des cas. C'est illustré par l'exemple suivant : *« – bon est-ce qu'il y a des choses qui vous parlent par rapport à votre demande initiale ? – bah est-ce qu'il y a un mot clé prévention ? – on l'a pas rajouté. »* (VD09). La prise en compte dans notre modèle doit se faire par une spécification claire de la notion de contribution entre jeu parent et jeu emboîté. En effet, certains emboîtements semblent adresser des conditions indispensables au succès du jeu parent tandis que d'autres paraissent facultatifs (voir, e.g., GODIS qui gère des questions emboîtées indispensables et d'autres facultatives). Cette notion de contribution a alors une incidence sur la fermeture des jeux emboîtés.

La seconde grande catégorie, liée à la structure intra-jeu, peut également être décomposée en deux sous-catégories. Le premier cas prédomine (environ 35%) et apparaît lorsque le partenaire semble prendre un « raccourci » dans un motif d'interaction plus grand. Cela inclut certains cas d'*actes de dialogue indirects* et apparaît fréquemment lors du jeu de requête et lors du jeu de vérification (CheckQuestion). Par exemple : *« – vous pouvez me la formuler en une vraie phrase ou pas (CheckQuestion) – "qu'est-ce qu'on sait sur l'évolution des traitements actuels sur la migraine" (Inform) »* (VD03). Ces structures trouvent deux échos possibles dans notre modèle. La première contribution peut être analysée comme un acte de contextualisation explicite (une proposition d'entrée dans un jeu de requête). La réalisation de l'action demandée remplit alors l'engagement en action extra-dialogique qui aurait été contracté si le jeu de requête

avait été mené à son terme (par la succession des actes *Request* et *AcceptRequest* et des actes de contextualisation adéquats). Une autre façon de percevoir cette structure est de considérer que la première contribution est un acte initiant un jeu de vérification (sur la capacité du partenaire) *pré-séquençant un jeu de requête* (dans notre exemple, sur la formulation). Les deux jeux sont alors implicitement satisfaits par la réalisation de l'action demandée. D'une façon ou d'une autre, la prise en compte de ces phénomènes la plus prometteuse semble être sur l'axe de la communication implicite et de mécanismes d'accommodation. Pour une prospection de ces aspects, nous invitons le lecteur intéressé à se référer au chapitre 6 de [Maudet 2001] et à [Larsson 2002a]. Le second cas que nous avons détecté est lorsque le partenaire altère sa réponse par rapport à ce qui est attendu dans le motif pour produire une chose à laquelle il peut et souhaite réagir (approx. 10%), appelée une *altération du projet conjoint* [Clark 1996]. Cela peut être illustré par l'exemple : *« – alors vous ne connaissez pas l'équivalent de zomig (CheckQuestion) – C'est une famille de produit »* (VD03). Ces cas ont été volontairement exclus de notre étude (cf. section 6.3.3). Le partenaire n'est plus passif : il altère le jeu courant de vérification proposé par l'initiateur vers un jeu d'interrogation ouverte (*« Qu'est-ce que le "zomig" ? »*). Ce type d'altération a vocation à être modélisé via des mécanismes d'accommodation concernant la contextualisation des jeux de dialogue.

6.4 Discussion

Ce chapitre a présenté un cadre pour modéliser des motifs dialogiques observés dans un corpus, basé sur des travaux théoriques antérieurs dans le champ des jeux de dialogue. Ce cadre s'appuie sur la notion d'engagement social qui permet de donner une *sémantique publique* aux jeux de dialogue. Leurs spécifications sont indépendantes des dispositions privées des agents prenant part à l'interaction. Les jeux de dialogue sont vus comme des *projets conjoints* activés temporairement pendant le dialogue pour un *but spécifique*. Ils constituent des contextes dialogiques au sein desquels les interlocuteurs contractent des engagements qui vont guider leurs comportements communicatifs. Ils définissent des enchaînements d'actes attendus, et les effets de l'occurrence de ces actes en terme de modifications du tableau de conversation.

La formalisation des engagements sociaux et des jeux de dialogue présentée raffine celle proposée par Maudet (cf. section 3.2.3). Notre formalisation d'engagement social clarifie leur cycle de vie par l'usage d'un *état* de l'engagement. Elle distingue clairement les engagements extra-dialogiques (proposition, action et jeux) des engagements dialogiques en action qui vont contraindre le comportement communicatif des interlocuteurs. En outre, nous considérons explicitement les engagements conjoints sur les jeux de dialogue dans divers états de *contextualisation*. Le *tableau de conversation* représente les positions des interlocuteurs sur des propositions, les engagements sur la réalisation d'actions et la structure intermédiaire du dialogue sous la forme de jeux suggérés, ouverts ou fermés. Ce dernier évolue sous l'impulsion d'*événements externes* (dialogiques ou non) et d'*événements internes* déclenchés par l'atteinte de certains états du tableau. La formalisation de jeux de dialogue que nous utilisons ajoute des conditions d'échecs définies en terme d'état du tableau de conversation, la définition des effets contextualisés des actes de dialogue dans le cadre d'un jeu de dialogue, et la possibilité de spécifier des contraintes de cohérence sémantique dans les enchaînements d'actes. Le cadre que nous proposons cantonne l'intervention du domaine d'application à trois grands aspects. Le premier est celui des actions *extra-dialogiques* dont il est nécessaire de spécifier l'atteinte des états de succès (**Ful**) et d'échec (**Vio**). Nous avons noté que la solution communément admise consiste en un jeu de décharge

aisément intégrable à notre approche. Le deuxième est celui de la sémantique des énoncés dont nous avons vu qu'il est nécessaire en pratique de restreindre au domaine d'application. Enfin, le dernier aspect est la spécification des combinaisons de jeux de dialogue possibles. Cette spécification peut dans un premier temps se baser sur des *plans de dialogue* [Larsson 2002a]. Enfin, notre formalisation bénéficie d'un ensemble d'avantages que nous avons déjà présentés en section 3.3.2.

Nous avons illustré la spécification des jeux de dialogue depuis des motifs dialogiques en mettant en œuvre l'ensemble des étapes de notre méthodologie sur un corpus orienté vers une tâche de recherche d'information dans le domaine médical. Nos jeux spécifient des enchaînements d'actes conventionnellement attendus. Cette spécification tire partie d'une sémantique précédemment utilisée sur le corpus COGNI-CISMEF et inspirée de GODIS. Une étude de la couverture des règles des jeux (de communication et de dialogue) a été réalisée par deux chercheurs sur le corpus de référence constitué lors du processus d'extraction. Elle a permis de montrer qu'une large majorité des enchaînements de fonctions communicatives dans la dimension *Task* apparaissant dans ce corpus est prescrite par les jeux que nous avons définis.

En outre, cette étude a permis de pointer les limites de notre modèle.

Notre modèle d'engagement social différencie engagement propositionnel (dirigés vers le présent) et engagement en action (dirigés vers le futur). Il est tout à fait possible de contracter un engagement propositionnel dirigé vers le futur (« Les documents seront triés demain. ») ou un engagement vers le passé (« Hier, j'ai recherché ton livre toute la soirée »). Il existe une volonté théorique d'uniformisation du traitement des engagements propositionnels et en action [Singh 2008]. Ce problème semble étroitement lié au langage de contenu des engagements [Pasquier 2005]. Des avancées théoriques dans cette direction peuvent bénéficier directement à notre modèle de l'engagement social.

Notre modèle ne considère qu'un nombre réduit d'états pour les engagements sociaux. Nous avons vu que le formalisme d'engagement proposé constitue un minimum vital qui nous a permis de donner une sémantique publique satisfaisante aux motifs dialogiques observés dans le corpus. C'est une première approche simplifiée qui a l'avantage d'être *extensible*. Il est possible d'envisager d'autres états comme l'annulation, la mise en attente, l'expiration, etc. [Chaib-Draa 2006, Telang 2012]. Dans la perspective d'uniformisation des engagements propositionnels et en action, il semble raisonnable d'étendre ces états vers ceux des engagements en action.

Notre modèle de jeu de dialogue requiert la spécification manuelle des combinaisons possibles de jeux de dialogue. Ces combinaisons peuvent être motivées *intentionnellement* ou *dialogiquement* (cf. section 3.3.1). Il est nécessaire d'approfondir les raisons conduisant à certaines combinaisons et pas à d'autres (notamment dans le cas de l'emboîtement).

Suite à la restriction à la dimension *Task* opérée dans le processus d'extraction, *notre modèle se limite à la modélisation de la communication au niveau de la tâche.* Nous avons laissé de côté tous les problèmes de communication pouvant intervenir aux niveaux inférieurs (attention, perception, interprétation, etc.). En conséquence, le tableau de conversation modélisé représente une partie publique *strictement partagée* par les interlocuteurs. La considération des autres niveaux de la communication va éventuellement nécessiter la relativisation du point de vue de l'agent du tableau de conversation. Ces processus de gestion de l'interaction nécessitent une étude particulière, et ont vocation à s'intégrer dans notre modèle via les jeux de communication.

Enfin, *notre modèle se limite à la modélisation de la communication explicite.* L'étude de la couverture des règles des jeux de dialogue permet de pointer cette limitation. Les cas non-couverts par notre modèle sont majoritairement liés au fait que nous nous sommes limités aux phénomènes explicites. Cependant, la communication humaine est souvent implicite. Ceci trans-

paraît de manière criante au niveau de la contextualisation des jeux de dialogue. En effet, notre étude a permis de confirmer la quasi absence des actes de contextualisation : ce mécanisme est réalisé de manière *implicite* dans les dialogues naturels. Cela conduit à des propositions d'entrée et des refus implicites (cf. section 3.1.3). L'extension du mécanisme de contextualisation à la communication implicite a vocation à prendre en compte des cas de retrait et d'altération du projet conjoint. Une piste de recherche s'ouvre au niveau de la prise en compte des phénomènes implicites et de mécanismes d'accommodation.

Chapitre 7

Mise en œuvre du modèle

Sommaire

7.1 Dogma : « DialOgue Game MAnager » 183
7.1.1 Architecture d'un système de dialogue Homme-Machine 184
7.1.2 Exploitation des jeux de dialogue par un gestionnaire de dialogue 184
7.1.3 Présentation de Dogma 185
7.1.4 Implémentation 188
7.2 Traces de fonctionnement de Dogma 189
7.2.1 Tableau de conversation 189
7.2.2 Exemple de jeu de requête 189
7.2.3 Exemple de jeu de question à choix multiples 194
7.2.4 Exemple d'emboîtement 201
7.3 Vers une validation de Dogma 205
7.3.1 Test « à la Turing » 206
7.3.2 Développement de systèmes de dialogue 207
7.4 Discussion : vers un agent dialogique utilisant les jeux 207

Ce chapitre montre l'apport des jeux de dialogue vus comme une structure capturant les engagements pour la gestion du comportement communicatif conventionnel d'un agent interactif. La section 7.1 décrit l'intérêt *interprétatif* et *génératif* des jeux de dialogue pour un agent interactif. Elle présente Dogma, un module normatif de gestion de l'interaction conventionnelle qui tire partie des jeux de dialogue pour fonder le comportement communicatif conventionnel d'un agent interactif. La section 7.2 illustre le fonctionnement de Dogma à travers trois exemples mêlant jeux de dialogue, jeux de communication et combinaisons de jeux. La section 7.3 envisage plusieurs scénarios d'évaluation pour Dogma à moyen et long terme. Enfin, la section 7.4 synthétise ce chapitre et donne quelques pistes pour l'intégration des jeux de dialogue dans le processus délibératif d'un agent interactif.

7.1 Dogma : « DialOgue Game MAnager »

Cette section s'intéresse à l'exploitation des jeux de dialogue par le gestionnaire de dialogue d'un agent interactif. La section 7.1.1 synthétise brièvement l'architecture d'un système de dialogue Homme-Machine et pointe l'importance du gestionnaire de dialogue. La section 7.1.2 présente l'intérêt computationnel des jeux de dialogue fondés sur les engagements. La section 7.1.3 présente le module de gestion de l'interaction développé dans cette thèse, et décrit son architecture haut niveau. La section 7.1.4 fournit certains détails de sa mise en œuvre.

7.1.1 Architecture d'un système de dialogue Homme-Machine

L'architecture classique d'un système de dialogue Homme-Machine consiste en un ensemble de modules communiquant les uns avec les autres permettant de produire le comportement du système. Comme noté par [Landragin 2013], il n'existe pas d'architecture de référence fiable et générique pouvant être appliquée à n'importe quel système de dialogue. Ceci est principalement dû au fait que chaque système possède des capacités différentes en terme d'entrée (i.e., d'interprétation de la contribution de l'utilisateur en fonction des modalités d'expression) et de sortie (i.e., de réalisation de la contribution du système). En conséquence, chaque système admet une architecture qui lui est propre. Néanmoins, un certain nombre de composants sont récurrents et peuvent être identifiés.

L'architecture d'un système de dialogue s'articule autour de trois grandes tâches [Jokinen 2010, Landragin 2013] : l'acquisition et l'interprétation du comportement communicatif de l'utilisateur, la gestion du comportement dialogique du système, et enfin, la génération effective du comportement du système. Elle intègre un ensemble de *modules d'interprétation* permettant d'acquérir la contribution de l'utilisateur (reconnaissance vocale, caméra, clavier, etc.) et de la transformer en une *représentation computationnelle* utilisable par le *gestionnaire de dialogue*. Ce dernier permet d'intégrer la contribution de l'utilisateur et de générer celle du système. Celle-ci est réalisée par un ensemble de *modules de génération* (synthèse vocale, animation de la représentation graphique d'un agent animé, etc.).

Une grande constante de ces architectures est la centralité du *gestionnaire de dialogue*. Ce module est le cœur du système [Jokinen 2010, Landragin 2013]. Schématiquement, il inclut deux grands composants [Jokinen 2010]. Le premier est l'*état d'information* (ou *modèle du contexte du dialogue*). Ce composant enregistre les éléments pertinents du contexte dialogique (cf. section 1.2.2) nécessaires à l'interprétation du comportement communicatif de l'utilisateur et à la génération des contributions dialogiques du système. Ce modèle ne se limite pas à la représentation de l'avancement de la tâche sous-jacente mais inclut des éléments des contextes linguistique, sémantique, physique et perceptuel, et social [Bunt 2011a]. Le second est le module de *contrôle du dialogue*. Ce module est responsable du comportement communicatif du système : sur la base de l'état d'information, il détermine ce que le système doit faire. Le rôle du gestionnaire de dialogue est donc double. Il est responsable de la mise à jour de l'*état d'information* suite à l'intervention d'événements dialogiques provenant de l'utilisateur ou du système (rôle interprétatif), et régit le comportement communicatif du système (rôle génératif).

7.1.2 Exploitation des jeux de dialogue par un gestionnaire de dialogue

Les systèmes de dialogue Homme-Machine inspirés des systèmes dialectiques présentent un intérêt pour la modélisation de l'interaction Homme-Machine [Yuan 2011]. Leur principal atout réside dans leur nature *normative* : les engagements contractés pendant la communication contraignent le comportement communicatif des interlocuteurs. D'un point de vue computationnel, ces modèles présentent deux grands intérêts pour un agent interactif. Ils ont un *intérêt interprétatif* qui va nous permettre de déterminer la *légalité* d'un coup provenant de l'utilisateur. De là, un agent peut adapter sa réaction en fonction d'un coup *légal* ou non. Ces modèles ont également un *intérêt génératif* : ils permettent d'assister le système pour la production d'une réponse en limitant l'attention à l'ensemble des coups *légaux*. Cela conduit typiquement à réduire l'espace des réponses possibles pour l'agent et facilite le processus de délibération sur la réponse à produire.

Un gestionnaire de dialogue peut faire un usage avantageux des relations définies entre un événement dialogique et le tableau de conversation (cf. sections 6.1.3 et 6.2.4). D'un point de vue interprétatif, un modèle fondé sur les engagements sociaux envisage la légalité d'un coup dialogique en fonction de l'état du dialogue selon plusieurs angles [Maudet 2001] :

interdiction un coup peut être interdit par les engagements contractés durant l'interaction (relation estInterdit)

attente un coup peut être attendu par un engagement contracté durant l'interaction (relation estAttendu)

priorité un coup attendu peut être prioritaire sur les autres coups (relation estPrioritaire)

Le rôle interprétatif d'un gestionnaire de dialogue basé sur les jeux de dialogue tels que nous les avons présentés consiste à classer un événement dialogique selon ces critères, et à réagir en conséquence. Ce rôle est illustré par l'algorithme 7.1 qui présente la structure générale du comportement du gestionnaire de dialogue. Un événement dialogique autorisé, attendu et prioritaire conduit à la mise à jour du tableau de conversation via l'algorithme 6.2 précédemment décrit. La gestion des interdictions, des inattendus et des événements non-prioritaires dépend de l'agent interactif modélisé. Elle est déléguée à la conception de l'agent. Une grande variété de comportements peut être mise en place. Par exemple, GoDiS inclut une règle « irrelevantFollowup » [Larsson 2002a] (p. 141) permettant d'interpréter un acte *inattendu* après une question du système comme un refus d'aborder la question. D'autres systèmes d'interaction Homme-Machine pour le débat interdisent purement et simplement les coups non légaux [Yuan 2008].

Notre modèle permet de déterminer les coups conventionnellement attendus en fonction du tableau de conversation. Cette propriété présente un intérêt aussi bien du point de vue interprétatif que génératif. Pour ce dernier point, nous proposons de classer les événements dialogiques attendus en fonction des jeux dans lesquels ils prennent part, puis par précédence conversationnelle. L'idée, déjà évoquée en section 6.2.4, est qu'il existe à un moment donné un ensemble de jeux de dialogue ouverts et saillants, ordonnés par leurs temps de passage à l'état **Open** (du plus récent au plus ancien). Chaque jeu entretient un ensemble d'événements dialogiques attendus dans son cadre. Cet ensemble est lui-même ordonné par le temps de passage des engagements à l'état actif (du plus récent au plus ancien). Nous partons du principe que les jeux de communication sont moins saillants que les jeux de dialogue. La structure est alors complétée par l'ensemble ordonné des événements attendus dans le cadre des jeux de communication. De cette manière, nous obtenons un ensemble ordonné d'événements dialogiques *prioritaires* qui peut être exploité par le processus délibératif d'un agent. Cette structure est illustrée dans les exemples de fonctionnement du système que nous avons développés, et qui sont présentés en section 7.2.

7.1.3 Présentation de Dogma

Dogma (pour « DialOgue Game MAnager ») est un module normatif de gestion de l'interaction conventionnelle basé sur les jeux de dialogue, utilisable par un agent interactif lors d'un dialogue impliquant deux interlocuteurs. Dogma adopte l'approche par état d'information (ISU) [Larsson 2000b, Traum 2003] et est centré autour de la mise à jour et de l'exploitation du *commitment store* localisé dans la *partie publique* de l'état d'information du gestionnaire de dialogue. Dogma met en œuvre les spécifications présentées dans les sections 6.1 et 6.2. Il implémente notamment les algorithmes précédemment présentés et les *opérations* réalisables sur le tableau de conversation.

Algorithme 7.1 Structure de l'algorithme de gestion de dialogue basé sur les jeux de dialogue.

Nom: GestionnaireDeDialogue
Role: Met à jour le tableau de conversation suite à l'occurrence d'un événement externe *dialogique*
Entrée: T_i : TableauDeConversation , e : ÉvénementDialogique, ensGenerator : EnsInternalEventGenerator
Sortie: T_{i+1} : TableauDeConversation
Entrée/Sortie: t : Temps
Déclaration: /
debut
  // Progression du temps
  $t \leftarrow$ incrementeTemps(t)
  // Classement du coup dialogique dans différentes catégories
  si estInterdit(e, T_i) **alors**
    // Gestion de l'événement dialogique interdit
    ... à définir...
  sinon
    si ¬estAttendu(e, T_i) **alors**
      // Gestion d'un événement dialogique inattendu
      ... à définir...
    sinon
      si ¬estPrioritaire(e, T_i) **alors**
        // Gestion d'un événement dialogique non prioritaire
        ... à définir...
      sinon
        // Le coup est autorisé, attendu et prioritaire : c'est un coup pertinent
        // Mise à jour du tableau de conversation
        $T_{i+1} \leftarrow$ evolutionTableauDeConversation(e, ensGenerator, T_i)
      finsi
    finsi
  finsi
fin

L'architecture dans laquelle s'intègre Dogma est présentée en figure 7.1. L'élément central de cette architecture est l'*état d'information* qui est classiquement subdivisé en une partie publique (le tableau de conversation) et une partie privée. C'est dans cette partie publique que s'intègre le composant central de Dogma : le *commitment store*. Ce dernier est constitué d'un ensemble d'engagements contractés par les interlocuteurs, partiellement ordonné par la relation de priorité. Il contient *pour chaque interlocuteur* les engagements propositionnels (extra-dialogiques) et en action (extra-dialogiques et dialogiques) dont il est débiteur. Il inclut également l'ensemble des jeux de dialogue sur lesquels les interlocuteurs sont engagés dans différents états de contextualisation. Le *commitment store* est maintenu à jour suite à l'occurrence d'événements externes par le *gestionnaire d'engagements* qui a accès en lecture et en écriture à cette partie du tableau de conversation. Ce module met en œuvre l'algorithme 6.2, et se charge de faire évoluer l'ensemble des engagements suite à l'occurrence d'un événement externe et des événements internes

subséquents. Ce composant est sollicité par le module de *contrôle* du gestionnaire de dialogue.

Dogma fournit deux modules ayant accès en lecture au *commitment store* et permettant de l'exploiter : le module *arbitre* et le module de *gestionnaire du comportement conventionnel*. Le module *arbitre* permet de juger de la légalité d'un événement dialogique. Il accède au *commitment store* afin de déterminer si l'événement est interdit, attendu ou prioritaire. Ce composant est sollicité par l'algorithme de *contrôle* du gestionnaire de dialogue dans sa *phase interprétative* (dont la structure suit l'algorithme 7.1). Le module de *gestion du comportement conventionnel* exploite le *commitment store* afin d'identifier les descriptions d'événements dialogiques attendus de la part de chaque interlocuteur. Il permet d'ordonner les événements dialogiques *prioritaires* en fonction des jeux de dialogue en cours comme précisé en section 7.1.2. Ce module donne accès aux jeux de dialogue en cours ou passé (i.e., tous les jeux dont l'état n'est pas **Ina**). Il permet de déterminer les jeux saillants, le jeu de dialogue courant et les événements dialogiques attendus dans le cadre de chaque jeu (de communication ou de dialogue). Ce module donne des *motivations conventionnelles* à la production de certains actes en fonction des engagements en action dialogiques contractés et des jeux en cours. Ces éléments peuvent entrer en ligne de compte dans le processus délibératif de l'agent interactif (cf. section 7.4).

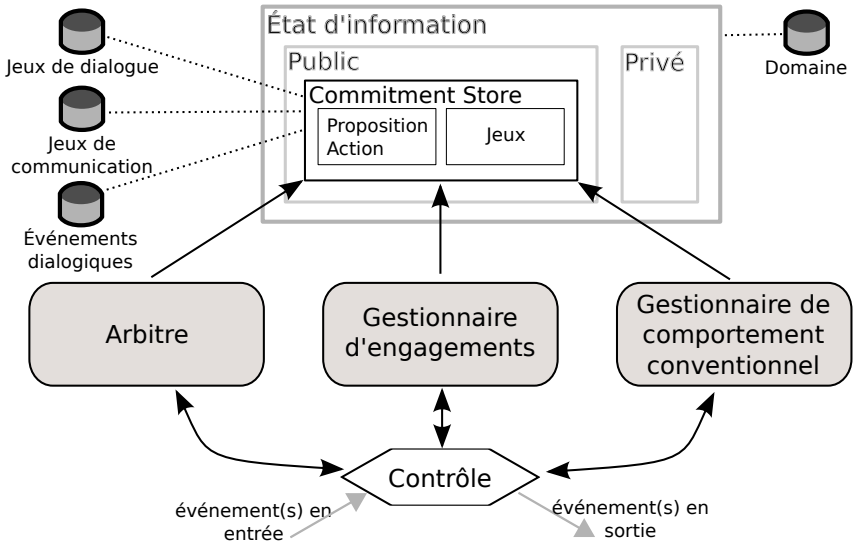

Figure 7.1: Architecture de Dogma

Les trois principales ressources configurant Dogma sont : la librairie des *événements dialogiques considérés*, la librairie des *jeux de communication* et enfin la librairie des *jeux de dialogue*. Ces trois ressources permettent de définir les coups dialogiques utilisables par les interlocuteurs ainsi que les jeux sur lesquels les utilisateurs vont pouvoir s'engager durant l'interaction. Ces ressources spécifient un *système normatif d'interaction* déclinable sur plusieurs domaines d'application. Une instantiation possible de ces ressources est celle que nous avons spécifiée en section 6.3. Enfin, la ressource *domaine* fournit les éléments dépendants du domaine d'application de l'agent interactif. Cette ressource spécifie notamment les éléments relatifs au domaine de la sémantique des énoncés, les événements externes, et les combinaisons autorisées entre les jeux de dialogue.

7.1.4 Implémentation

Plusieurs frameworks de développement de systèmes de dialogue basés sur un état d'information existent parmi lesquels nous pouvons citer TRINDIKIT [Larsson 2002b], DIPPER [Bos 2003] (basé sur TRINDIKIT), et TRINDIKIT.PY [Ljunglöf 2009]. TRINDIKIT est très certainement le framework le plus populaire dans le milieu académique. Il a permis de développer plusieurs prototypes dont GODIS (cf. section 2.2.3), et est à la base de DIPPER. TRINDIKIT est un *projet libre*[1], développé dans un dialecte de Prolog appelé Sicstus Prolog. Malheureusement, cette implémentation pose des problèmes pour la mise en œuvre d'un système de dialogue [Ljunglöf 2009]. Bien qu'étant un projet libre, TRINDIKIT est basé sur une implémentation propriétaire de Prolog. En outre, l'architecture modulaire de ce framework est fondée sur l'« Open Agent Architecture » (OAA) [Martin 1999] qui n'est plus en développement actif depuis 2007, laissant envisager des problèmes de compatibilité. TRINDIKIT.PY est né en 2009 afin de surmonter ces problèmes. Son objectif est de proposer une alternative entièrement libre de TRINDIKIT basée sur le langage Python[2]. Cependant, à l'heure où nous écrivons, l'activité de ce projet semble être au point mort (dernière activité en septembre 2010). De plus, les développeurs signalent explicitement que le code comporte actuellement des erreurs et n'est pas documenté.

Le développement de notre prototype DOGMA est parti du constat qu'aucune alternative existante n'est satisfaisante. Nous nous sommes tournés vers le langage SCALA [Odersky 2008][3] afin de développer DOGMA. Le code source de DOGMA est sous license GPLv3 et est actuellement hébergé à l'URL `https://labanquise.insa-rouen.fr/projects/dogma/`. L'annexe C.1 présente des exemples de code tiré de DOGMA et montre l'implémentation du jeu de dialogue de requête et du jeu de communication d'évaluation.

DOGMA a vocation à s'intégrer dans le module de gestion de dialogue de la plateforme AgentSlang [Serban 2013] développée dans l'équipe de recherche Modélisation, Interation et Usage (MIU) du LITIS. Cette plateforme fournit une architecture distribuée pour le développement d'agent interactif.

DOGMA a été utilisé avec succès afin d'implémenter deux *systèmes normatifs d'interaction*. Le premier système est celui spécifié dans cette thèse à partir de l'étude du corpus COGNI-CISMEF (cf. section 6.3). C'est ce système que nous utilisons afin de produire des traces de fonctionnement en section 7.2. Le second système a été conçu et implémenté par Coralie Farges lors de son stage de Master [Farges 2013]. Cette étude a consisté en l'application de notre méthodologie sur un corpus de dialogues Homme-Homme dans le contexte du jeu de société Négoces[4]. Elle a conduit à l'annotation du corpus en terme d'actes de dialogue suivie de la détection de motifs d'interaction récurrents. Ces motifs ont été formalisés dans le cadre théorique proposé au chapitre 6. Cette formalisation a conduit à l'implémentation des jeux de dialogue via DOGMA afin de produire un système normatif de l'interaction dans le cadre du jeu Négoces.

1. Le code source de TRINDIKIT est disponible à l'URL : `http://sourceforge.net/projects/trindikit` (dernière mise à jour : 15/04/2013)
2. Le code source de TRINDIKIT.PY est disponible à l'URL : `https://code.google.com/p/py-trindikit` (dernière mise à jour : 17/09/2010)
3. SCALA est développé à l'École Polytechnique Fédérale de Lausanne (EPFL), `http://www.scala-lang.org/` (dernière mise à jour : 03/12/2013)
4. Le jeu Négoces a été conçu par François Tréca (`http://www.negoces.fr`, dernière consultation le 20/12/2013).

7.2 Traces de fonctionnement de Dogma

Dans cette section, nous présentons trois exemples de fonctionnement qui illustrent les caractéristiques du modèle et les capacités de Dogma. La section 7.2.1 introduit le tableau de conversation utilisé lors des exemples. La section 7.2.2 décrit le déroulement complet d'un jeu de dialogue requête, de son établissement à sa clôture. Ensuite, la section 7.2.3 illustre une séquence d'actes dépassant le simple cadre de la paire adjacente et entremêlant jeu de dialogue et jeu de communication d'évaluation. Enfin, la section 7.2.4 introduit un exemple d'emboîtement de jeux. Trois traces de fonctionnement supplémentaires sont disponibles en annexe C.2. Elles traitent des cas marginaux d'échec et de correction.

7.2.1 Tableau de conversation

Ces exemples impliquent un tableau de conversation minimal. Il inclut l'*historique du dialogue* en terme d'événements dialogiques (champ « Tours »), les *engagements propositionnels* et *en action* contractés par les interlocuteurs (colonnes C_x et C_y) et les *engagements conjoints* des interlocuteurs x et y sur les jeux de dialogue (champ C_{xy}^{Jeu}). En outre, le *jeu de dialogue saillant* est précisé s'il existe. Pour chaque interlocuteur, nous précisons les *propositions* sur lesquelles il est engagé (ligne C_p), les actions *extra-dialogiques* dans les états **Crt** (ligne $C_a(\textbf{Crt})$) et **Fal** (ligne $C_a(\textbf{Fal})$), et les engagements en action *dialogiques* (ligne C_{dA}). Les engagements affichés sont les engagements *actifs*.

Pour chaque tableau de conversation, Dogma présente son exploitation en terme d'*actes attendus* et d'*actes interdits*. Nous n'affichons que les *actes attendus* puisque nos jeux ne présentent aucune règle d'interdiction. Les actes attendus sont ordonnés en fonction de la saillance des jeux et de l'instant auquel les engagements ont été contractés. Les jeux de communication sont identifiés par `ctx` (contextualisation) et `ev` (évaluation).

Dans Dogma, les jeux de dialogue et les jeux de communication sont gérés uniformément. Les règles de ces jeux sont chargées lorsque le jeu est ouvert. Pour les jeux de communication, les règles sont chargées au lancement du système puisqu'ils sont toujours actifs. Les règles des jeux de dialogue sont chargées à l'ouverture du jeu et déchargées à leur fermeture. Afin d'améliorer la lisibilité du tableau de conversation, nous ne faisons pas apparaître les règles des jeux de communication. Nous exhibons les traces du système qui démontrent leur application. Pour éviter de surcharger le tableau de conversation, l'acte de contextualisation *poursuit* a été évincé des exemples. Cet acte est joué chaque fois qu'un acte de niveau inférieur attendu est appliqué au tableau.

Les traces sont générées par Dogma au format LaTeX. Ces traces sont *post-traitées* de manière à accroître la lisibilité du tableau de conversation : le contenu des actes de contextualisation est factorisé par l'identifiant du jeu (précisé dans la partie C_{xy}^{Jeu}), le but du jeu est factorisé par une expression plus générique (α, p, ...), et certaines propositions sont également factorisées.

7.2.2 Exemple de jeu de requête

Ce premier exemple présente l'*établissement* et le *déroulement* d'un jeu de dialogue. Il illustre le jeu de *contextualisation* qui capture explicitement l'établissement par les interlocuteurs du projet conjoint formé par le jeu. Il montre la façon dont les règles du jeu évoluent sous l'impulsion d'événements dialogiques produits par les interlocuteurs. En particulier, il souligne l'application des règles de production qui permettent de spécifier des enchaînements d'actes. Cet exemple

illustre également l'intervention d'*événements internes* relatifs à l'état du tableau de conversation. Il présente la création d'engagements extra-dialogiques qui vont persister à l'issue du déroulement du jeu de dialogue. Enfin, il montre à chaque étape l'*exploitation du tableau de conversation* qui génère des motivations conventionnelles à la production de certains actes.

Ce premier exemple présente un jeu de dialogue de requête sur l'action « ajouterMotCle(**grippe**) » qui forme le but du jeu. Il débute par un acte de contextualisation de proposition d'entrée dans ce jeu (*« Je peux vous demander de réaliser une action ? »*). De tels actes de proposition d'entrée sont *toujours attendus*. Un extrait de traces est présenté ci-dessous :

```
Regular act prop.entrée(x, Request(ajouterMotCle(grippe)))
...
1) Fulfilling commitment: C(x, y, prop.entrée(x,Request(ajouterMotCle(grippe)))=>create(x, C(y,
x, acc.entrée(y,Request(ajouterMotCle(grippe)))|ref.entrée(y,Request(ajouterMotCle(grippe)))|
prop.entrée(y,Ys~>Request(ajouterMotCle(grippe)))|prop.entrée(y,Ys;Request(ajouterMotCle(grippe))),
Ina):ctx),Crt):ctx
2) Adding a commitment: C(y, x, acc.entrée(y, Request(ajouterMotCle(grippe)))|ref.entrée(y,
Request(ajouterMotCle(grippe)))|prop.entrée(y, Ys~>Request(ajouterMotCle(grippe)))|prop.entrée(y,
Ys;Request(ajouterMotCle(grippe))), Crt):ctx
...
3) Adding a suggested game: Request(ajouterMotCle(grippe))_g0
4) Executing internal events:
EntryConditionsReached(g0, x), EntryConditionsReached(g0, y), FullEntryConditionsReached(g0)
```

L'événement dialogique conduit à *satisfaire* un engagement pris dans le cadre du jeu de contextualisation (ligne 1). Cette satisfaction engage l'autre interlocuteur à accepter ou à refuser cette proposition d'entrée, ou bien à proposer un pré-séquencement ou un séquencement de jeux de dialogue (ligne 2). L'engagement conjoint sur le jeu est établi en état **Sugg** (ligne 3). En outre, des événements internes sur l'atteinte des conditions d'entrée sont générés (ligne 4). Ceux-ci n'ont aucun effet sur le tableau courant compte-tenu de son état. Le tableau de conversation et son exploitation sont alors les suivants :

Tours: (prop.entrée(x, g0))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\emptyset$

C_{xy}^{Jeu} :

Sugg.	:	Request(ajouterMotCle(**grippe**))=g0
Ouv.	:	
Ferm.	:	
Saillants	:	$\emptyset$

Actes attendus :
ctx, ev:
acc.entrée(y, g0) | ref.entrée(y, g0) | prop.entrée(y, Ys⇝g0) | prop.entrée(y, Ys ; g0)

L'exploitation du tableau de conversation spécifie que les seuls actes attendus sont de la part de l'interlocuteur y. Il s'agit pour lui d'accepter ou de refuser l'entrée dans le jeu, ou bien de

proposer une combinaison de jeu. Le terme `Ys` représente une variable devant être remplacée par une proposition de jeu valide.

L'interlocuteur y accepte alors l'entrée dans le jeu via l'acte adéquat attendu (acc.entrée(y, g0)). Le cas du *refus* est présenté en annexe C.2.1.

```
Regular act acc.entrée(y, Request(ajouterMotCle(grippe)))
...
1) Adding an opened game: Request(ajouterMotCle(grippe))_g0
2) Loading a game: Request(ajouterMotCle(grippe))_g0
...
3) Adding a commitment: C(x, y, FailureConditionsReached(Request(ajouterMotCle(grippe))_g0, x)|
SuccessConditionsReached(Request(ajouterMotCle(grippe))_g0,x)=>create(x, C(x, y,
prop.sortie(x, Request(ajouterMotCle(grippe))), Ina):ctx), Crt):ctx
4) Adding a commitment: C(y, x, FailureConditionsReached(Request(ajouterMotCle(grippe))_g0, y)|
SuccessConditionsReached(Request(ajouterMotCle(grippe))_g0,y)=>create(y, C(y, x,
prop.sortie(y, Request(ajouterMotCle(grippe))), Ina):ctx), Crt):ctx
5) Adding a commitment: C(x, y, prop.entrée(x, Ys<Request(ajouterMotCle(grippe))), Crt):ctx
6) Adding a commitment: C(y, x, prop.entrée(y, Ys<Request(ajouterMotCle(grippe))), Crt):ctx
```

L'occurrence de cet événement conduit à ouvrir le jeu de dialogue de requête (ligne 1) et à charger ses règles (ligne 2). En outre, les interlocuteurs contractent chacun un engagement les autorisant à proposer l'entrée dans un jeu de dialogue emboîté (dans le cadre du jeu de contextualisation) (lignes 5 et 6). Toujours dans ce jeu, les interlocuteurs sont conditionnellement engagés à proposer la sortie du jeu si les conditions d'échec ou de succès du jeu sont atteintes (lignes 3 et 4). Le tableau de conversation et son exploitation sont alors les suivants :

Tours: (acc.entrée(y, g0), prop.entrée(x, g0))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\lfloor$request(x, α)$\rfloor_{:g0}$	$\lfloor$request(x, α) $\Rightarrow$ C_{g0}(y,acceptRequest(y, α) | declineRequest(y, α),**Crt**)$\rfloor_{:g0}$, $\lfloor$declineRequest(y, α) $\Rightarrow$ C(y,α,**Fal**)$\rfloor_{:g0}$, $\lfloor$acceptRequest(y, α) $\Rightarrow$ C(y,α,**Crt**)$\rfloor_{:g0}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Request(ajouterMotCle(**grippe**))=g0
Ferm.	:	
Saillants	:	g0

α=ajouterMotCle(**grippe**)

Actes attendus :
g0:
request(x, α)
ctx, ev:
prop.entrée(x, Ys<g0), prop.entrée(y, Ys<g0)

L'interlocuteur x est engagé sur les règles du jeu de dialogue en tant qu'initiateur. Le partenaire est engagé sur les règles et les effets de ce jeu qui prennent la forme de règles de production. Le jeu de dialogue saillant est le jeu de requête, le seul jeu de dialogue ouvert. Les actes attendus dans le cadre de ce jeu sont alors l'acte *Request* de la part de l'initiateur x, et les actes de proposition d'entrée dans un jeu emboîté.

L'interlocuteur x joue son rôle dans le jeu de dialogue et produit l'acte *Request* (*« Essaye d'ajouter le mot-clé grippe »*).

Tours: (request(x, α), acc.entrée(y, g0), prop.entrée(x, g0))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$acceptRequest(y, α) | declineRequest(y, α)$\rfloor_{:g0}$, $\lfloor$declineRequest(y, α) $\Rightarrow$ C(y,α,**Fal**)$\rfloor_{:g0}$, $\lfloor$acceptRequest(y, α) $\Rightarrow$ C(y,α,**Crt**)$\rfloor_{:g0}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Request(ajouterMotCle(**grippe**))=g0
Ferm.	:	
Saillants	:	g0

α=ajouterMotCle(**grippe**)

Actes attendus :
g0:
acceptRequest(y, α) | declineRequest(y, α)
ctx, ev:
prop.entrée(x, Ys<g0), prop.entrée(y, Ys<g0)

Cela conduit à satisfaire l'engagement de x et à développer la règle de production sur laquelle y était engagée. L'acte *Request* précédemment attendu est alors substitué par les actes *AcceptRequest* et *DeclineRequest* attendus de la part de y.

L'interlocuteur y joue alors un acte *AcceptRequest* (*« D'accord »*).

```
Regular act acceptRequest(y, ajouterMotCle(grippe))
...
Adding a commitment: C(y, x, ajouterMotCle(grippe), Crt)
Executing internal events:
SuccessConditionsReached(g0, x), SuccessConditionsReached(g0, y), FullSuccessConditionsReached(g0),
ExitConditionsReached(g0)
...
Adding a commitment: C(x, y, prop.sortie(x, Request(ajouterMotCle(grippe))), Crt):ctx
Adding a commitment: C(y, x, prop.sortie(y, Request(ajouterMotCle(grippe))), Crt):ctx
```

Cet événement contribue à satisfaire l'un des effets du jeu et conduit à engager y sur la réalisation de l'action α. Les conditions de succès du jeu sont alors atteintes, générant les événements internes liés. L'occurrence de ces événements internes engagent les interlocuteurs à proposer la sortie du jeu de dialogue dans le cadre du jeu de contextualisation.

Tours: (acceptRequest(y, α), request(x, α), acc.entrée(y, g0), prop.entrée(x, g0))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	α
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$declineRequest(y, α) $\Rightarrow$ C(y,α,**Fal**)$\rfloor_{:g0}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Request(ajouterMotCle(**grippe**))=g0
Ferm.	:	
Saillants	:	g0

α=ajouterMotCle(**grippe**)

Actes attendus :
ctx, ev:
prop.sortie(x, g0), prop.sortie(y, g0)
prop.entrée(x, Ys<g0), prop.entrée(y, Ys<g0)

Plus aucun acte n'est alors attendu dans le jeu de requête. Les seuls actes attendus sont des actes de contextualisation de proposition de sortie ou d'emboîtement de jeux. L'interlocuteur x propose la sortie du jeu.

Tours: (prop.sortie(x, g0), acceptRequest(y, α), request(x, α), acc.entrée(y, g0), prop.entrée(x, g0))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	α
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$declineRequest(y, α) $\Rightarrow$ C(y,α,**Fal**)$\rfloor_{:g0}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Request(ajouterMotCle(**grippe**))=g0
Ferm.	:	
Saillants	:	g0

α=ajouterMotCle(**grippe**)

Actes attendus :
ctx, ev:
acc.sortie(y, g0) | ref.sortie(y, g0)
prop.sortie(y, g0)
prop.entrée(x, Ys<g0), prop.entrée(y, Ys<g0)

Par précédence conversationnelle, une acceptation ou un refus de la proposition sont attendus

en priorité de la part de y. Le partenaire accepte la sortie du jeu.

$$\left[\begin{array}{l}\text{Tours: } \left(\begin{array}{l}\text{acc.sortie(y, g0)}, \text{prop.sortie(x, g0)}, \text{acceptRequest(y, }\alpha\text{)}, \text{request(x, }\alpha\text{)}, \text{acc.entrée(y, g0)},\\ \text{prop.entrée(x, g0)}\end{array}\right)\\ \\ \begin{array}{|l|c|c|}\hline & \mathrm{C_x} & \mathrm{C_y}\\ \hline \mathrm{C_p} & \emptyset & \emptyset\\ \hline \mathrm{C_a}(\textbf{Crt}) & \emptyset & \alpha\\ \mathrm{C_a}(\textbf{Fal}) & \emptyset & \emptyset\\ \hline \mathrm{C_{dA}} & \emptyset & \emptyset\\ \hline\end{array}\\ \\ \mathrm{C}_{\mathrm{xy}}^{\mathrm{Jeu}}: \left[\begin{array}{rcl}\text{Sugg.} & : & \\ \text{Ouv.} & : & \\ \text{Ferm.} & : & \text{Request(ajouterMotCle(}\textbf{grippe}\text{))=g0}\\ \hline \text{Saillants} & : & \emptyset\end{array}\right]\end{array}\right]$$

α=ajouterMotCle(**grippe**)

Cette acceptation conduit à supprimer l'ensemble des engagements contextualisés dans le jeu de requête ainsi que les engagements du jeu de contextualisation liés. Le jeu de dialogue est alors fermé et il n'y a plus d'acte attendu (excepté les propositions d'entrée dans un jeu). Seul persiste l'engagement en action extra-dialogique contracté par y.

7.2.3 Exemple de jeu de question à choix multiples

Ce deuxième exemple illustre l'entremêlement possible entre jeu de communication et jeu de dialogue. Il montre que ce modèle ne se résume pas à la description de paires adjacentes. La prise en compte du *tableau de conversation* permet la production de séquences d'actes plus importantes.

Cet exemple présente un jeu de question à choix multiples sur la question *« Êtes-vous un patient, un étudiant en médecine ou un médecin ? »*. Il entremêle jeu de dialogue et jeu de communication d'évaluation. Nous ne présentons ni les étapes d'établissement du jeu, ni la sortie du jeu, similaires à celle du premier exemple.

L'interlocuteur x est l'initiateur du jeu et y le partenaire. À la suite de l'établissement du jeu de question à choix multiples, le tableau est le suivant :

Tours: (acc.entrée(y, g1), prop.entrée(x, g1))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\lfloor$choiceQuestion(x, q)$\rfloor_{:g1}$	$\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g1}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g1}$, $\lfloor$choiceQuestion(x, q) $\Rightarrow$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g1}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Choice({?patient, ?etudiant, ?medecin})=g1
Ferm.	:	
Saillants	:	g1

q={?patient, ?etudiant, ?medecin}

Actes attendus :
g1:
choiceQuestion(x, q)
ctx, ev:
prop.entrée(y, Ys<g1), prop.entrée(x, Ys<g1)

Le jeu saillant est le seul ouvert, i.e. le jeu de question à choix multiples. Les interlocuteurs x et y sont engagés sur les règles de ce jeu de dialogue dont le but est la question {?patient, ?etudiant, ?medecin}. L'acte attendu dans le cadre du jeu de dialogue est un acte *ChoiceQuestion* de l'initiateur. De la même manière que pour le premier exemple, les interlocuteurs peuvent proposer des emboîtements de jeu.

L'interlocuteur x joue alors l'acte de question *« Êtes-vous un patient, un étudiant en médecine ou un médecin ? »*. L'engagement de x

est alors satisfait et les règles de production sur lesquelles y est engagé sont développées.

Tours: (choiceQuestion(x, q), acc.entrée(y, g1), prop.entrée(x, g1))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)$\rfloor_{:g1}$, $\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g1}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g1}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Choice({?patient, ?etudiant, ?medecin})=g1
Ferm.	:	
Saillants	:	g1

q={?patient, ?etudiant, ?medecin}

Actes attendus :

g1:

answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)

ctx, ev:

prop.entrée(y, Ys<g1), prop.entrée(x, Ys<g1)

Le partenaire est alors engagé à produire une réponse *pertinente* (ce qui inclut les réponses résolvantes) ou à abandonner la question.

Le partenaire produit une réponse pertinente mais non résolvante (*« Je ne suis pas médecin. »*).

Tours: (answer(y, ¬medecin), choiceQuestion(x, q), acc.entrée(y, g1), prop.entrée(x, g1))

	C_x	C_y
C_p	$\emptyset$	¬medecin
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)$\rfloor_{:g1}$, $\lfloor$execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)$\rfloor_{:g1}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g1}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Choice({?patient, ?etudiant, ?medecin})=g1
Ferm.	:	
Saillants	:	g1

q={?patient, ?etudiant, ?medecin}

Actes attendus :
g1:
answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)
ctx, ev:
agreement(x, ¬medecin) | disagreement(x, ¬medecin) | correction(x, ¬medecin, P2) avec **correct** (¬medecin, P2)
prop.entrée(y, Ys<g1), prop.entrée(x, Ys<g1)

La réponse du partenaire conduit à créer l'*engagement propositionnel* sur « ¬medecin » dans le cadre du *jeu d'évaluation*. En outre, l'interlocuteur x contracte l'engagement d'affirmer son accord ou son désaccord envers cette proposition via le jeu d'évaluation. L'interlocuteur y est engagé sur la production d'une nouvelle réponse puisqu'il a produit une réponse strictement pertinente (i.e., pertinente et non résolvante).

L'initiateur affirme son accord (*« Tout à fait ! »*) en produisant un acte *Agreement*. Cela le conduit à s'engager sur la proposition

affirmant que y n'est pas médecin.

Tours: (agreement(x, ¬medecin), answer(y, ¬medecin), choiceQuestion(x, q), acc.entrée(y, g1), prop.entrée(x, g1))

	C_x	C_y
C_p	¬medecin	¬medecin
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)$\rfloor_{:g1}$, $\lfloor$execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)$\rfloor_{:g1}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\overset{*}{\Rightarrow}$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g1}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Choice({?patient, ?etudiant, ?medecin})=g1
Ferm.	:	
Saillants	:	g1

q={?patient, ?etudiant, ?medecin}

Actes attendus :

g1:
answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)

ctx, ev:
prop.entrée(y, Ys<g1), prop.entrée(x, Ys<g1)

Le partenaire fournit une réponse résolvante (*« Je suis un patient »*).

Tours: (answer(y, patient), agreement(x, ¬medecin), answer(y, ¬medecin), choiceQuestion(x, q), acc.entrée(y, g1), prop.entrée(x, g1))

	C_x	C_y
C_p	¬medecin	patient, ¬medecin
C_a(**Crt**)	∅	∅
C_a(**Fal**)	∅	∅
C_{dA}	∅	⌊execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)⌋$_{:g1}$, ⌊answer(y, P) avec **strictRel** (P, q) $\overset{*}{\Rightarrow}$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)⌋$_{:g1}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Choice({?patient, ?etudiant, ?medecin})=g1
Ferm.	:	
Saillants	:	g1

q={?patient, ?etudiant, ?medecin}

Actes attendus :
ctx, ev:
agreement(x, patient) | disagreement(x, patient) | correction(x, patient, P2) avec **correct** (patient, P2)
prop.sortie(y, g1), prop.sortie(x, g1)
prop.entrée(y, Ys<g1), prop.entrée(x, Ys<g1)

Cette réponse a les mêmes effets que précédemment mais sur la proposition « patient ». En outre, cet événement permet d'atteindre les *conditions de succès* du jeu de dialogue. Les interlocuteurs sont alors engagés à proposer la sortie du jeu suite à l'occurrence des événements internes.

Avant de clore le jeu, l'initiateur affirme son accord sur le fait que l'interlocuteur y est un patient. Cela conduit à l'engager sur cette

proposition.

Tours: (agreement(x, patient), answer(y, patient), agreement(x, ¬medecin), answer(y, ¬medecin), choiceQuestion(x, q), acc.entrée(y, g1), prop.entrée(x, g1))

	C_x	C_y
C_p	patient, ¬medecin	patient, ¬medecin
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g1}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g1}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g1}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Choice({?patient, ?etudiant, ?medecin})=g1
Ferm.	:	
Saillants	:	g1

q={?patient, ?etudiant, ?medecin}

Les interlocuteurs ferment ensuite le jeu de dialogue. Les seuls engagements qui persistent sont les engagements propositionnels (extra-dialogiques).

7.2.4 Exemple d'emboîtement

Ce dernier exemple illustre la représentation de la *structure intermédiaire du dialogue* par le tableau de conversation via une combinaison d'emboîtement entre deux jeux de dialogue. Il présente un exemple de *priorité* pouvant intervenir entre les règles de jeux de dialogue.

Ce troisième exemple implique l'emboîtement d'un jeu de vérification dans un jeu de question polaire. L'initiateur x et le partenaire y sont engagés sur le jeu de question polaire dont le but est la question *« Est-ce que le terme "eczéma" possède une définition dans la terminologie ? »* (?definition(**eczema**)). y est alors engagé à répondre :

Tours: (propositionalQuestion(x, q), acc.entrée(y, g2), prop.entrée(x, g2))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$answer(y, p) | answer(y, $\neg p$) | execNegativeAutoFB(y, q)$\rfloor_{:g2}$, $\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g2}$

C^{Jeu}_{xy} :

Sugg.	:	
Ouv.	:	YNInterrogation(definition(**eczema**))=g2
Ferm.	:	
Saillants	:	g2

q=?p, p=definition(**eczema**)

Actes attendus :
g2:
answer(y, p) | answer(y, $\neg p$) | execNegativeAutoFB(y, q)
ctx, ev:
prop.entrée(x, Ys<g2), prop.entrée(y, Ys<g2)

L'interlocuteur y propose alors l'ouverture d'un jeu de dialogue de vérification emboîté dans le jeu de question polaire. Ce jeu porte sur la question « Vous êtes médecin ? » (?medecin). Nous supposons dans cet exemple que le fait que la terminologie possède une définition d'eczéma dépend du statut de x (médecin ou non). Le jeu de vérification est alors établi via la séquence d'actes prop.entrée(y, g3<g2) et acc.entrée(x, g3<g2). Une fois ce jeu établi, le tableau de conversation est le suivant :

Tours: (acc.entrée(x, g3<g2), prop.entrée(y, g3<g2), propositionalQuestion(x, q), acc.entrée(y, g2), prop.entrée(x, g2))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	⌊checkQuestion(y, q') | posiCheck(y, q') ⇒ C_{g3}(x, confirm(x, medecin) | disconfirm(x, medecin) | execNegativeAutoFB(x, q') ,**Crt**) ⌋:g3 ⌊disconfirm(x, medecin) ⇒ C(x,¬medecin,**Crt**)⌋:g3, ⌊confirm(x, medecin) ⇒ C(x,medecin,**Crt**)⌋:g3, ⌊execNegativeAutoFB(x, q') ⇒ C(x,**fail** (q'),**Crt**)⌋:g3	⌊checkQuestion(y, q') | posiCheck(y, q')⌋:g3, ⌊answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)⌋:g2, ⌊execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)⌋:g2

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Verification(medecin)=g3, YNInterrogation(definition(**eczema**))=g2
Ferm.	:	
Saillants	:	g3

q=?p, p=definition(**eczema**), q'=?medecin

Actes attendus :
g3:
checkQuestion(y, q') | posiCheck(y, q')
g2:
answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)
ctx, ev:
prop.entrée(x, Ys<g3), prop.entrée(y, Ys<g3)
prop.entrée(y, Ys<g2), prop.entrée(x, Ys<g2)

Les deux jeux de dialogue sont *ouverts*. Le jeu de vérification est *prioritaire* sur le jeu de question polaire. En conséquence, le seul jeu saillant est le jeu de vérification, et les engagements de ce jeu sont prioritaires sur ceux du jeu de question polaire.

L'interlocuteur y joue la question de vérification (*« Vous êtes médecin ? »*). Cet événement dialogique provoque l'évolution des enga-

gements dialogiques du jeu de vérification :

Tours: (checkQuestion(y, q'), acc.entrée(x, g3<g2), prop.entrée(y, g3<g2), propositionalQuestion(x, q), acc.entrée(y, g2), prop.entrée(x, g2))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\lfloor$confirm(x, medecin) | disconfirm(x, medecin) | execNegativeAutoFB(x, q')$\rfloor_{:g3}$, $\lfloor$disconfirm(x, medecin) ⇒ C(x,¬medecin,**Crt**)$\rfloor_{:g3}$, $\lfloor$confirm(x, medecin) ⇒ C(x,medecin,**Crt**)$\rfloor_{:g3}$, $\lfloor$execNegativeAutoFB(x, q') ⇒ C(x,**fail** (q'),**Crt**)$\rfloor_{:g3}$	$\lfloor$answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)$\rfloor_{:g2}$, $\lfloor$execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)$\rfloor_{:g2}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Verification(medecin)=g3, YNInterrogation(definition(**eczema**))=g2
Ferm.	:	
Saillants	:	g3

q=?p, p=definition(**eczema**), q'=?medecin

Actes attendus :

g3:
confirm(x, medecin) | disconfirm(x, medecin) | execNegativeAutoFB(x, q')

g2:
answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)

ctx, ev:
prop.entrée(x, Ys<g3), prop.entrée(y, Ys<g3)
prop.entrée(y, Ys<g2), prop.entrée(x, Ys<g2)

L'interlocuteur x confirme qu'il est médecin (*« Oui tout à fait »*). Le jeu de vérification est alors un succès :

Tours: (confirm(x, medecin), checkQuestion(y, q'), acc.entrée(x, g3<g2), prop.entrée(y, g3<g2), propositionalQuestion(x, q), acc.entrée(y, g2), prop.entrée(x, g2))

	C_x	C_y
C_p	medecin	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\lfloor$disconfirm(x, medecin) $\Rightarrow$ C(x,¬medecin,**Crt**)$\rfloor_{:g3}$, $\lfloor$execNegativeAutoFB(x, q') $\Rightarrow$ C(x,**fail** (q'),**Crt**)$\rfloor_{:g3}$	$\lfloor$answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)$\rfloor_{:g2}$, $\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g2}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	Verification(medecin)=g3, YNInterrogation(definition(**eczema**))=g2
Ferm.	:	
Saillants	:	g3

q=?p, p=definition(**eczema**), q'=?medecin

Actes attendus :

g2:

answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)

ctx, ev:

prop.sortie(y, g3), prop.sortie(x, g3)
prop.entrée(x, Ys<g3), prop.entrée(y, Ys<g3)
prop.entrée(y, Ys<g2), prop.entrée(x, Ys<g2)

Le jeu de vérification est clos via la séquence d'actes prop.sortie(x, g3), acc.sortie(x, g3). Le jeu d'interrogation polaire redevient alors le jeu saillant. Les interlocuteurs peuvent reprendre le cours de ce jeu.

Tours: (acc.sortie(x, g3), prop.sortie(y, g3), confirm(x, medecin), checkQuestion(y, q'), acc.entrée(x, g3<g2), prop.entrée(y, g3<g2), propositionalQuestion(x, q), acc.entrée(y, g2), prop.entrée(x, g2))

	C_x	C_y
C_p	medecin	∅
C_a(**Crt**)	∅	∅
C_a(**Fal**)	∅	∅
C_{dA}	∅	⌊answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)⌋:g2, ⌊execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)⌋:g2

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	YNInterrogation(definition(**eczema**))=g2
Ferm.	:	Verification(medecin)=g3
Saillants	:	g2

q=?p, p=definition(**eczema**), q'=?medecin

Actes attendus :
g2:
answer(y, p) | answer(y, ¬p) | execNegativeAutoFB(y, q)
ctx, ev:
prop.entrée(y, Ys<g2), prop.entrée(x, Ys<g2)

L'interlocuteur y affirme alors que la terminologie possède une définition d'eczéma.

Tours: (answer(y, p), acc.sortie(x, g3), prop.sortie(y, g3), confirm(x, medecin), checkQuestion(y, q'), acc.entrée(x, g3<g2), prop.entrée(y, g3<g2), propositionalQuestion(x, q), acc.entrée(y, g2), prop.entrée(x, g2))

	C_x	C_y
C_p	medecin	definition(**eczema**)
C_a(**Crt**)	∅	∅
C_a(**Fal**)	∅	∅
C_{dA}	∅	⌊execNegativeAutoFB(y, q) ⇒ C(y,**fail** (q),**Crt**)⌋:g2

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	YNInterrogation(definition(**eczema**))=g2
Ferm.	:	Verification(medecin)=g3
Saillants	:	g2

q=?p, p=definition(**eczema**), q'=?medecin

L'état de succès du jeu est atteint. Le jeu peut alors être fermé par les interlocuteurs.

7.3 Vers une validation de Dogma

Dans cette section, nous envisageons plusieurs scénarios d'évaluation pour Dogma qui s'ajoutent à la validation de la couverture des jeux (cf. section 6.3.4). La section 7.3.1 présente un test « à la Turing » permettant d'évaluer la capacité de Dogma à reproduire des motifs d'interaction dialogiques. La section 7.3.2 évoque une évaluation de Dogma réalisable sur le long terme du point de vue du développement d'un système d'interaction Homme-Machine.

7.3.1 Test « à la Turing »

La validation de DOGMA consiste à montrer qu'il est possible de modéliser de façon implémentable les conventions d'interaction dialogiques observées dans notre corpus. Il s'agit de montrer que la structure de jeux de dialogue associée au tableau de conversation proposés dans DOGMA permet à un agent interactif de réguler son comportement communicatif conventionnel de bas niveau, tout en étant intégrable à un processus délibératif de haut niveau. Notre démarche de modélisation cognitive propose une validation sous la forme d'un test « à la Turing » qui consiste à comparer un ensemble de traces réelles obtenues par expérimentation à un ensemble de traces générées par notre système [Chaignaud 1996, Pauchet 2006]. Cette variante implique des experts qui ont pour tâche d'analyser un échantillon de traces (mélangeant traces réelles et traces générées) dans le but de déterminer s'il est possible de séparer aisément les deux ensembles.

Ce test implique la confrontation de *motifs dialogiques générés automatiquement* à des *motifs dialogiques réels* tirés de notre corpus de référence. Il nécessite un travail important décomposable en plusieurs opérations. La première est l'établissement d'une couverture sémantique *complète* des questions, propositions et actions intervenant dans les motifs dialogiques du corpus d'interaction. Cette étape vise à compléter le travail initié dans [Loisel 2008]. La spécification de cette sémantique dépendante du domaine s'appuie sur la formalisation présentée en section 6.3.1. Cette étape permet d'établir une base sémantique depuis laquelle il est possible de générer des traces de motifs dialogiques, et de faire la correspondance entre représentation sémantique et langage naturel. La deuxième étape consiste à extraire les motifs dialogiques du *corpus de référence* en opérant un processus d'extraction. Ces motifs constituent une première bibliothèque de traces (les *traces réelles*).

L'étape suivante consiste à générer automatiquement un certain nombre de traces en mettant deux agents dialogiques face à face. Nous pouvons envisager deux modes de génération automatique : un mode *complètement aléatoire* et un mode *basé sur l'usage de* DOGMA. Le mode complètement aléatoire implique deux agents dialogiques réalisant des choix arbitraires de coups à produire. Le tour de parole est assigné aléatoirement à un des deux agents. Celui-ci choisit alors arbitrairement un acte de dialogue à générer en complétant son contenu sémantique via la base de connaissances sémantiques préalablement constituée. Le nombre de tours de parole générés par confrontation peut se baser sur la taille moyenne observée dans le corpus (i.e., approximativement deux tours de parole). De même, la répartition des fonctions communicatives observée lors du processus d'extraction peut être mise à profit. Cette expérience permet de constituer une deuxième bibliothèque de traces (traces générées). Le mode basé sur l'usage de DOGMA consiste à fonder la production de contributions dialogiques des agents sur le tableau de conversation. Les motifs dialogiques sont alors générés sur la base des jeux de dialogue définis, de la base sémantique mise en place et d'un processus délibératif minimal. La génération automatique de traces impliquent trois grandes étapes. Premièrement, un jeu de dialogue et un but du jeu sont choisis aléatoirement depuis la bibliothèque de jeux de dialogue et de la ressource sémantique précédemment constituée. Le choix du jeu à initier peut se baser sur les proportions observées dans le corpus COGNI-CISMEF. Ensuite, un agent est désigné *initiateur* du jeu tandis que l'autre est désigné *partenaire*. L'initiateur joue alors une proposition d'entrée dans le jeu de dialogue. Puis, la génération automatique prend effectivement place. Sur la base du tableau de conversation et de la ressource sémantique, un coup dialogique de l'initiateur ou du partenaire est aléatoirement choisi. Le choix du coup dialogique peut être influencé par l'*historique du dialogue*

(pour que les agents ne se répètent pas), la priorité et l'ordre d'attente des coups dialogiques tirés du tableau de conversation, et la répartition observée lors du processus d'extraction des secondes paires de motifs dialogiques. La génération continue tant qu'il est possible de produire des actes, et que le jeu de dialogue initial n'est pas fermé. Cette deuxième expérience permet la constitution d'une troisième bibliothèque de traces (traces générées par DOGMA). Chaque trace automatiquement générée peut se voir attribuer une forme en langue naturelle sur la base de la ressource sémantique constituée depuis le corpus.

Une *évaluation qualitative* des traces obtenues par ces trois moyens peut alors être envisagée. Elle consiste à demander aux experts pour chaque trace si celle-ci semble *cohérente* (au sens de [Mann 2002], i.e., si chaque intervention contribue au reste et qu'il n'y a pas de contribution dialogique dont la présence n'est pas facilement explicable), et si celle-ci semble *naturelle* (i.e., produite par une interaction entre deux humains). Les résultats de cette évaluation permettront de situer les traces générées par DOGMA en terme de cohérence et de « naturel » entre un système complètement aléatoire et des motifs dialogiques produits par des humains.

7.3.2 Développement de systèmes de dialogue

En permettant la mise en œuvre de systèmes normatifs de l'interaction conventionnelle en partie déconnectés du domaine d'application, DOGMA présente un intérêt du point de vue des développeurs de systèmes de dialogue (en tant que module d'un système de dialogue orienté vers la réalisation d'une tâche). Néanmoins, l'évaluation de systèmes de dialogue combinant plusieurs modules avancés est un sujet difficile [Jokinen 2010] qui amène certains chercheurs à s'interroger sur sa faisabilité [Landragin 2013] (chapitre 10). L'inconvénient de ce type d'évaluation est que le système est évalué comme un tout : l'évaluation ne rend pas compte de l'apport d'un module isolé. Ainsi, le système peut être pénalisé par un module trop faible (e.g., une couverture lexicale insuffisante) ou, au contraire, avantagé par un composant du système compensant les autres. En outre, les systèmes de dialogue actuellement développés se cantonnent au stade de prototypes de recherche limités à des domaines d'application très spécifiques [Landragin 2013]. Leur évaluation est fortement liée au système rendant mal aisée l'émergence d'une méthodologie d'évaluation applicable à tout système de dialogue. Cela rend également délicat les études comparatives entre systèmes de dialogue.

Malgré les difficultés posées par l'évaluation d'un système de dialogue, une perspective intéressante dans cette direction est une *étude de la réutilisabilité de* DOGMA à travers l'implémentation de systèmes dans différents domaines et pour différentes tâches.

7.4 Discussion : vers un agent dialogique utilisant les jeux

Ce chapitre a souligné l'intérêt *interprétatif* et *génératif* des jeux de dialogue comme structure d'engagements pour un agent interactif. Ce dernier se manifeste autour du *tableau de conversation* qui représente l'état du dialogue à un instant donné. D'un point de vue interprétatif, le tableau permet de déterminer la *légalité* d'un coup dialogique en terme d'attente, de priorité ou d'interdiction. L'agent interactif peut alors adapter son comportement communicatif autour de ces différentes situations. D'un point de vue génératif, le tableau de conversation donne des motivations conventionnelles à l'agent pour la production de coups dialogiques attendus de sa part dans le cadre des jeux ouverts. Il permet de limiter l'attention de l'agent aux coups conventionnellement attendus, et de pointer les coups prioritaires. Dans cette direction,

nous avons présenté DOGMA, un module normatif de gestion de l'interaction conventionnelle basé sur les jeux de dialogue exploitable par un agent interactif. DOGMA est paramétré par trois ressources principales : la bibliothèque des événements dialogiques autorisés, la bibliothèque des jeux de dialogue et celles des jeux de communication. Nous avons illustré le fonctionnement de DOGMA sur des exemples présentant l'évolution et l'exploitation du tableau de conversation. Ceux-ci ont montré des cas typiques de déroulement complet du jeu de dialogue, d'entremêlement de jeu de dialogue et de jeu de communication et de combinaisons de jeux de dialogue. À moyen terme, nous envisageons la réalisation d'une étude qualitative confrontant en terme de *cohérence* des motifs dialogiques générés automatiquement à des motifs dialogiques réels tirés du corpus COGNI-CISMEF. À long terme, nous pensons étudier la *réutilisabilité* de DOGMA à travers son usage dans différents systèmes d'interaction Homme-Machine.

L'intégration effective des jeux de dialogue dans le processus délibératif d'un agent interactif admet plusieurs perspectives intéressantes. Comme précédemment évoqué (cf. section 3.3), le niveau public dégagé par le tableau de conversation ne dispense pas de la nécessité d'un niveau privé. Un agent interactif utilisant les jeux doit donc raisonner en prenant en compte la structure des jeux et ce niveau public. Deux points de vue semblent se distinguer par la granularité de prise en compte de la structure des jeux de dialogue. La question est de savoir si l'agent doit planifier jusqu'aux engagements extra-dialogiques produits par l'exécution d'un jeu (via les conditions de succès et d'échec), ou s'il possède une *structure délibérative* de plus haut niveau intégrant les jeux.

La première vision appréhende les jeux de dialogue comme des structures régulant la manipulation des engagements sociaux par les interlocuteurs (voir, e.g., [Pasquier 2005, Chaib-Draa 2006]). La spécification du comportement dialogique de l'agent peut reposer sur les engagements sociaux et leurs interdépendances en cas de modification de leurs états. [Chaib-Draa 2006] propose par exemple la notion de *réseau d'engagements* qui permet de spécifier le comportement dialogique d'un agent pour une tâche donnée. Un réseau d'engagements décrit les liens de dépendance existant entre les états des engagements extra-dialogiques liés à une tâche. Cette structure énumère les conséquences du changement d'état d'un engagement sur les autres engagements. Les changements d'états sont réalisés dialogiquement via les jeux de dialogue qui conviennent. Dans le cas d'un système d'aide pour CISMEF, un tel réseau pourrait par exemple spécifier que la création de l'engagement propositionnel « Je suis un utilisateur débutant » (via un acte *Inform*) conduit à la tentative de création par l'agent dialogique d'un engagement sur l'affichage d'une vidéo explicative de la terminologie CISMEF (via un jeu d'offre). Le succès du jeu d'offre oblige l'agent à afficher la vidéo, tandis qu'un échec peut par effet de cascade déclencher d'autres tentatives d'engagements (assistance à la construction de la première requête, désactivation de l'agent assistant, etc.). Une telle approche semble adaptée pour la spécification du comportement dialogique simple sur un domaine restreint. Son principal désavantage est de nécessiter l'explicitation de l'ensemble des conséquences possibles d'une modification d'un engagement social. Par ailleurs, de récents travaux cherchent à lier les cycles de vie des engagements extra-dialogiques à ceux des buts des agents (e.g., [Telang 2012]). Le rapprochement de ces deux cycles conduit les agents à contracter des engagements sociaux afin de réaliser leurs buts, ou à considérer de nouveaux buts afin de remplir leurs engagements en cours. [Telang 2012] propose par exemple des règles qui amènent les agents à raisonner en terme d'engagements extra-dialogiques depuis les buts qu'ils entretiennent. L'établissement effectif de ces engagements peut alors se baser sur les jeux de dialogue, vus comme structures permettant de manipuler les états des engagements sociaux.

La seconde vision envisage le lien entre processus de délibération et jeux de dialogue via une *structure délibérative* liant but discursif et jeu de dialogue [Mann 1988, Hulstijn 2000b, Hulstijn 2000a, Maudet 2001, Mann 2002]. Cette structure conventionnelle inclut le but de l'initiateur, le but du partenaire et le jeu de dialogue (cf. section 3.1.3). Un agent dialogique délibératif doit alors prendre en compte des éléments *subjectifs* issus de cette structure et des éléments *objectifs* relatifs aux jeux de dialogue et à l'état du tableau de conversation. Un des défis se situe dans la détermination de l'ensemble des éléments composant cette structure délibérative et à leur prise en compte au cours du dialogue.

Cette dernière vision met en évidence la potentielle complexité sous-jacente à la définition du comportement de l'agent qui doit mêler *réactivité* et *délibération*. Les deux points de vue précédents se rejoignent sur la vision schématique du dialogue comme impliquant un haut niveau délibératif à propos des jeux et un bas niveau régulé par les règles des jeux en cours. La dichotomie délibération/réactivité n'est cependant pas aussi claire. Le comportement *réactif* de l'agent peut se baser sur les *éléments objectifs* issus du tableau de conversation et de la définition des jeux de dialogue. Sur la base de ces éléments, un agent peut réactivement accepter l'entrée dans un jeu (en vérifiant l'atteinte des conditions d'entrée), proposer et accepter la sortie d'un jeu (sur la base des conditions de sortie) et fonder son comportement interactif (sur la base des engagements contractés). Le comportement *délibératif* intervient aussi bien pour la proposition et la considération de l'entrée dans un jeu que pour la sortie, et fait intervenir des états mentaux privés. La question est alors de savoir comment mêler ces deux comportements qui font intervenir des motivations sociales (engagements) et privées (e.g., intentions, désirs et croyances). Le cas général (à notre connaissance non résolu) invite à proposer des architectures d'agents intégrant des notions sociales dans le processus délibératif « classique » (e.g., [Castelfranchi 2000]). Face à cette difficulté théorique, il semble possible de se limiter en pratique à une spécification procédurale entre comportement réactif et comportement délibératif [Traum 1997, Larsson 2002a, Orkin 2013]. C'est cette dernière option que nous favorisons dans l'objectif de mise en œuvre de systèmes d'interaction Homme-Machine pour une tâche donnée basés sur DOGMA.

Conclusion

Le travail présenté dans ce document vise à améliorer les capacités communicatives des agents logiciels en interaction avec des humains. Dans ce but, nous avons suivi une méthodologie permettant d'extraire des régularités d'un corpus d'interactions Homme-Homme orientées vers la réalisation d'une tâche. Nous nous sommes concentrés tout particulièrement sur les motifs d'interaction, c'est-à-dire les régularités observées au niveau de l'interaction dialogique. Nous avons étudié les structures de haut niveau spécifiables à partir de ces motifs dans le but de produire un modèle d'interaction pour un agent interactif, capable de s'intégrer dans un processus délibératif. Nous avons considéré la métaphore des jeux de dialogue qui permet de lier deux précédents points de vue initialement opposés sur les motifs d'interaction (preuve d'un plan et conventions). Enfin, nous avons souligné l'intérêt de cette structure aussi bien pour la formalisation des motifs d'interaction pour un agent interactif que pour le fondement de son comportement communicatif conventionnel. L'ensemble de notre démarche a été appliqué au corpus Cogni-CISMeF [5].

Contributions

Les contributions de ce travail se situent à plusieurs niveaux. Du point de vue *méthodologique*, nous avons illustré l'application de la méthodologie sur le corpus Cogni-CISMeF, de l'étape d'annotation jusqu'à celle de formalisation des motifs d'interaction. Cela nous a amené à analyser la structure haut niveau du corpus Cogni-CISMeF. Nous avons caractérisé la tâche sous-jacente aux entretiens qui consiste en la réalisation collaborative d'une recherche d'information sur CISMeF entre un expert et un utilisateur. Cette caractérisation s'est appuyée sur l'étude du processus de recherche d'information dans le cas individuel. Nous avons souligné le fait que la tâche de recherche d'information collaborative met en jeu un processus itératif, opportuniste, stratégique et interactif entre les participants.

Nous avons réalisé l'annotation multidimensionnelle du corpus de dialogues Homme-Homme Cogni-CISMeF avec le schéma DIT++. L'usage de ce schéma a permis de prendre en compte la richesse qui fait la complexité du dialogue. Le corpus a été transformé en une séquence d'unités multifonctionnelles produites par les interlocuteurs, qui traduit la participation de ces unités dans les différentes activités parallèles sous-jacentes au dialogue. Entre autres résultats, nous avons confirmé la multifonctionnalité des énoncés et observé que le dialogue est utilisé pour faire progresser la tâche sous-jacente et pour gérer l'interaction elle-même.

Nous avons extrait des motifs dialogiques à partir du corpus annoté. Ces motifs récurrents prennent majoritairement la forme de paires adjacentes avec une seconde partie de paire préférée ou non. Ils se répartissent dans trois catégories : recherche d'information, transfert d'information et discussion d'actions. Ils forment une bibliothèque initiale de fragments d'interaction conventionnelle que l'on souhaite voir reproduit par un agent interactif.

5. « Catalogue et Index des Sites Médicaux de langue Française », disponible à l'URL www.cismef.org

À la lumière de notre étude bibliographique, nous avons interprété les motifs d'interaction observés dans le corpus comme des manifestations de conventions, nécessaires à la coordination du dialogue vu comme une activité conjointe, opportuniste et sociale. Nous avons exploité les jeux de dialogue dont l'objectif est de répertorier les conventions dialogiques dans une structure pouvant être intégrée au processus délibératif d'un agent interactif. Ces jeux permettent de lier ingénieusement délibération et réactivité du comportement dialogique en proposant une vision schématique du dialogue comme étant planifié au niveau des jeux et régulé à bas niveau par les règles des jeux. Nous avons rejoint la vision des jeux de dialogue comme structure capturant les engagements des interlocuteurs durant le dialogue. Sur la base de ces travaux, nous avons établi un cadre permettant de spécifier des jeux s'appuyant sur une sémantique publique via les notions d'engagement social et de tableau de conversation. La spécification des jeux est indépendante des dispositions privées des agents prenant part à l'interaction.

Nous avons illustré la spécification des jeux de dialogue depuis des motifs dialogiques en mettant en œuvre l'ensemble des étapes de notre méthodologie sur le corpus Cogni-CISMeF. Ces jeux spécifient des enchaînements d'actes conventionnellement attendus. Une étude de la couverture des règles des jeux réalisée par deux chercheurs a montré que les jeux définis couvrent adéquatement les enchaînements de fonctions communicatives de la dimension *Task* observés dans le corpus. Les motifs d'interaction ainsi formalisés entament la constitution d'une bibliothèque de jeux de dialogue et de communication pour modéliser et construire des systèmes d'interaction Homme-Machine.

Enfin, nous nous sommes intéressé à la façon dont un agent interactif peut bénéficier des jeux de dialogue afin de fonder son comportement communicatif conventionnel. Une formalisation des jeux de dialogue comme structure d'engagements permet de discerner un intérêt interprétatif (en terme de légalité de coups dialogiques et d'attente) et un intérêt génératif (en terme de motivations conventionnelles pour la production de coups attendus). Dans ce sens, nous avons développé Dogma, un module normatif de gestion de l'interaction conventionnelle basé sur les jeux de dialogue, exploitable par un agent interactif afin de réguler son comportement communicatif conventionnel dans un dialogue impliquant deux interlocuteurs. De plus, des scénarios d'évaluation de Dogma ont été proposés.

Perspectives

Un certain nombre de perspectives peuvent être envisagées à l'issue de ce travail. Nous cernons ici les principales.

Dans un futur proche, l'évaluation de Dogma doit être menée afin de cerner son apport pour un agent interactif. Nous nous dirigeons vers la réalisation du test « à la Turing » précédemment décrit. Ce dernier vise à comparer en terme de cohérence des motifs dialogiques générés automatiquement à des motifs dialogiques réels tirés du corpus Cogni-CISMeF.

Sur le plan méthodologique, notre travail souligne l'intérêt d'aller vers une semi-automatisation des phases d'annotation et d'extraction de régularités. L'exploitation manuelle du corpus est une tâche coûteuse aussi bien en temps qu'en effort à produire. Il paraît nécessaire de réduire le temps passé dans ces étapes préliminaires afin de se concentrer sur l'exploitation des régularités pour la modélisation d'un agent interactif. La solution la plus réaliste consiste à envisager ces étapes sous la forme d'un processus collaboratif entre l'humain et la machine. Un équilibre très intéressant vise à confier à l'humain la responsabilité d'annoter les régularités, et de sélectionner les fragments que l'agent doit reproduire. Dans cette direction, un ensemble

d'outils d'annotation peut être conçus et mis à la disposition de l'humain afin de faciliter sa tâche. Le processus délibératif de l'agent doit alors mettre en œuvre un ensemble de mécanismes permettant d'exploiter les fragments de comportement spécifiés par l'humain.

Une perspective ambitieuse pour notre méthodologie consiste à aller vers une prise en compte formelle de l'aspect sémantique des énoncés. Cet aspect a été très largement sous-exploité dans ce document, où nous nous sommes principalement concentrés sur des aspects pragmatiques. Des travaux dans cette direction vont conduire à affronter le problème de la formalisation du langage naturel, et à attaquer de front le rapprochement entre sémantique et pragmatique. Cette perspective non triviale ouvre la porte à des systèmes interactifs ouverts, non limités à une tâche unique.

La formalisation des motifs d'interaction par jeux de dialogue développée dans ce document à vocation à être étendue dans plusieurs directions. Le modèle proposé peut être enrichi par l'observation de nouveaux motifs dialogiques et de nouveaux actes de dialogue. L'extension du modèle peut passer par la considération de nouveaux états d'engagements associés à de nouvelles opérations de modification du tableau de conversation (e.g., annulation, délégation, mise en attente). La direction multidimensionnelle pousse à étudier les motifs dialogiques en considérant des dimensions supplémentaires à la dimension *Task*. Nos résultats sur la co-occurrence des dimensions dans le corpus Cogni-CISMeF peuvent être exploités afin de déterminer les dimensions prioritaires. Nous avons constaté qu'une fonction de la dimension *Task* co-occurre principalement avec des fonctions des dimensions permettant de gérer l'élocution du locuteur, de produire des feedbacks ou de gérer les tours de parole. Ces futures études ouvrent la porte à la communication multidimensionnelle permettant la production d'énoncés multifonctionnels. Celle-ci va de pair avec la direction de la communication implicite qui pousse à l'étude et la mise en place de mécanismes permettant de gérer des phénomènes implicites (comme des « raccourcis » pris par les interlocuteurs dans des motifs dialogiques). Enfin, une extension non triviale de la métaphore des jeux de dialogue consiste à l'étendre à l'interaction multipartite.

Pour terminer, les perspectives de ce travail incluent l'intégration effective des jeux de dialogue dans un agent interactif. Il s'agit de proposer des modèles délibératifs intégrant pleinement les jeux de dialogue. Ces modèles doivent faire face à la problématique plus globale de l'intégration de notions sociales dans le processus délibératif d'un agent interactif. D'un point de vue pratique, nous penchons dans un premier temps vers une spécification procédurale des priorités entre les notions sociales et privées pour la production de systèmes d'interaction limités à une tâche simple.

Annexes

Annexe A

Annexes de l'analyse du corpus Cogni-CISMeF

A.1 Le schéma XML d'annotation pour un acte de dialogue dans Gate

Le listing A.1 présente le *schéma d'annotation* sous Gate qui permet de standardiser les annotations manuelles réalisées dans les dialogues. Il définit le *type* d'annotation « acte de dialogue » comme étant constitué de trois *features* : (i) le *locuteur* qui peut prendre la valeur « Expert » ou « Enquêté », (ii) la *dimension* qui peut prendre comme valeur l'une des dimensions de DIT++, et (iii) la *fonction communicative* qui peut prendre comme valeur une des fonctions communicatives de DIT++. L'annotation d'un segment fonctionnel consiste en l'annotation de plusieurs « actes de dialogue » sur la même zone de texte (en respectant la contrainte d'au plus une fonction communicative par dimension).

```
<?xml version="1.0"?>
<schema
    xmlns="http://www.w3.org/2000/10/XMLSchema">
  <!-- XSchema definition for DialogueAct-->

  <element name="DialogueAct">
    <complexType>
      <attribute name="Speaker" use="required">
     <simpleType>
       <restriction base="string">
         <enumeration value="Expert"/>
         <enumeration value="Enquete"/>
       </restriction>
     </simpleType>
      </attribute>

      <attribute name="Dimension" use="required">
     <simpleType>
       <restriction base="string">
         <enumeration value="Task"/>
         <enumeration value="Auto-Feedback"/>
         <enumeration value="Allo-Feedback"/>
         <enumeration value="Contact Management"/>
         <enumeration value="Turn Management"/>
         <enumeration value="Time Management"/>
         <enumeration value="Discourse Structuring"/>
```

```
        <enumeration value="Own Communication Management"/>
        <enumeration value="Partner Communication Management"/>
        <enumeration value="Social Obligations Management"/>
      </restriction>
    </simpleType>
     </attribute>

     <attribute name="CommunicativeFunction" use="required">
    <simpleType>
      <restriction base="string">
        <enumeration value="Set Question"/>
        <enumeration value="Propositional Question"/>
        <enumeration value="Check Question"/>
        <enumeration value="Posi-check"/>
        <enumeration value="Nega-check"/>
        <enumeration value="Choice Question"/>

        <enumeration value="Inform"/>
        <enumeration value="Agreement"/>
        <enumeration value="Disagreement"/>
        <enumeration value="Correction"/>
        <enumeration value="Answer"/>
        <enumeration value="Confirm"/>
        <enumeration value="Disconfirm"/>

        <enumeration value="Offer"/>
        <enumeration value="Promise"/>
        <enumeration value="Threat"/>
        <enumeration value="Address Request"/>
        <enumeration value="AcceptRequest"/>
        <enumeration value="DeclineRequest"/>
        <enumeration value="Address Suggestion"/>
        <enumeration value="AcceptSuggestion"/>
        <enumeration value="DeclineSuggestion"/>

        <enumeration value="Request"/>
        <enumeration value="Instruct"/>
        <enumeration value="Address Offer"/>
        <enumeration value="AcceptOffer"/>
        <enumeration value="DeclineOffer"/>
        <enumeration value="Suggestion"/>
        <enumeration value="Indirect Request"/>

        <enumeration value="AutoPositive"/>
        <enumeration value="AttentPositiveAutoFeedback"/>
        <enumeration value="PerceptPositiveAutoFeedback"/>
        <enumeration value="InterprPositiveAutoFeedback"/>
        <enumeration value="EvalPositiveAutoFeedback"/>
        <enumeration value="ExecPositiveAutoFeedback"/>

        <enumeration value="AutoNegative"/>
        <enumeration value="ExecNegativeAutoFeedback"/>
```

```
        <enumeration value="EvalNegativeAutoFeedback"/>
        <enumeration value="InterprNegativeAutoFeedback"/>
        <enumeration value="PerceptNegativeAutoFeedback"/>
        <enumeration value="AttentNegativeAutoFeedback"/>

        <enumeration value="AlloPositive"/>
        <enumeration value="AttentPositiveAlloFeedback"/>
        <enumeration value="PerceptPositiveAlloFeedback"/>
        <enumeration value="InterprPositiveAlloFeedback"/>
        <enumeration value="EvalPositiveAlloFeedback"/>
        <enumeration value="ExecPositiveAlloFeedback"/>

        <enumeration value="AlloNegative"/>
        <enumeration value="ExecNegativeAlloFeedback"/>
        <enumeration value="EvalNegativeAlloFeedback"/>
        <enumeration value="InterprNegativeAlloFeedback"/>
        <enumeration value="PerceptNegativeAlloFeedback"/>
        <enumeration value="AttentNegativeAlloFeedback"/>

        <enumeration value="AttentionFeedbackElicitation"/>
        <enumeration value="PerceptionFeedbackElicitation"/>
        <enumeration value="InterpretationFeedbackElicitation"/>
        <enumeration value="EvaluationFeedbackElicitation"/>
        <enumeration value="ExecutionFeedbackElicitation"/>

        <enumeration value="Turn Take"/>
        <enumeration value="Turn Accept"/>
        <enumeration value="Turn Grab"/>

        <enumeration value="Turn Keep"/>
        <enumeration value="Turn Assign"/>
        <enumeration value="Turn Release"/>

        <enumeration value="Stalling"/>
        <enumeration value="Pausing"/>

        <enumeration value="Contact Check"/>
        <enumeration value="Contact Indication"/>

        <enumeration value="Error signaling"/>
        <enumeration value="Retraction"/>
        <enumeration value="Self-correction"/>

        <enumeration value="Completion"/>
        <enumeration value="Correct-misspeaking"/>

        <enumeration value="Opening"/>
        <enumeration value="Preclosing"/>
        <enumeration value="Topic introduction"/>
        <enumeration value="Topic shift annoucement"/>
        <enumeration value="Topic shift"/>
```

```
        <enumeration value="Initial greeting"/>
        <enumeration value="Return greeting"/>

        <enumeration value="Initial self-introduction"/>
        <enumeration value="Return self-introduction"/>

        <enumeration value="Apology"/>
        <enumeration value="Apology-downplay"/>

        <enumeration value="Thanking"/>
        <enumeration value="Thanking-downplay"/>
        <enumeration value="Initialgoodbye"/>
        <enumeration value="Return goodbye"/>

      </restriction>
    </simpleType>
     </attribute>

     <attribute name="Certitude" use="optional">
    <simpleType>
      <restriction base="string">
        <enumeration value="01_Sur et certain"/>
        <enumeration value="02_Plutot sur"/>
        <enumeration value="03_NSP"/>
        <enumeration value="04_Hesistant"/>
        <enumeration value="05_Tres hesitant"/>
      </restriction>
    </simpleType>
     </attribute>


   </complexType>
  </element>

</schema>
```

Listing A.1: Schéma d'annotation pour un acte de dialogue dans le processus d'annotation.

A.2 Alignement entre la taxonomie d'actes de dialogue de Loisel et DIT++

Le tableau A.1 présente l'alignement entre la taxonomie proposée par Loisel [Loisel 2008] et DIT++. Nous pouvons constater que les actes de dialogue proposés par Loisel possèdent tous un équivalent dans DIT++. En outre, la taxonomie proposée par Loisel ne couvre que les trois dimensions sur les dix proposées par DIT++ : *Task*, *Auto-feedback* et *Social Obligation Management*.

Acte	Équivalent DIT++	Acte	Équivalent DIT++	Acte	Équivalent DIT++
`Inform`	*Inform*	`RequestInfo`	Hiérarchie *Question*	`Offer`	*Offer*
`RequestDirective`	*Request*	`Direct`	*Instruct*	`Suggest`	*Suggestion*
`Answer`	*Answer*	`ShortAnswer`	*Answer*	`Accept`	*AcceptOffer*
`Refuse`	*DeclineOffer*	`Acknowledge`	*ExecPositiveAutoFB*	`icm:con*neg`	*AttentNegativeAutoFB*
`icm:con*pos`	*AttentPositiveAutoFB*	`icm:per*neg`	*PerceptNegativeAutoFB*	`icm:per*pos`	*PerceptPositiveAutoFB*
`icm:sem*pos`	*InterprPositiveAutoFB*	`icm:sem*neg`	*InterprNegativeAutoFB*	`icm:und*neg`	*EvalNegativeAutoFB*
`icm:und*pos`	*EvalPositiveAutoFB*	`icm:und*chk`	*CheckQuestion*(AF)	`icm:acc*pos`	*ExecPositiveAutoFB*
`icm:acc*neg`	*ExecNegativeAutoFB*	`icm:acc*chk`	*CheckQuestion*(AF)	`Greet`	*Initial Greeting*
`Thanks`	*Thanking*	`Thanks-return`	*Thanking-downplay*	`InformIntent`	*Promise*
`Bye`	*Initial Goodbye*				

TABLEAU A.1: Alignement entre la taxonomie de Loisel [Loisel 2008] et DIT++. Les fonctions générales sont appliquées dans la dimension *Task* sauf si précisé autrement. AF = *Auto-feedback*.

A.3 Annexes des résultats du processus d'annotation

Ces annexes présentent des données et résultats supplémentaires du processus d'annotation.

L'annexe A.3.1 présente des dialogues annotés issus du corpus Cogni-CISMeF.

L'annexe A.3.2 fournit le détail des tours de parole (total, annoté, exclu) pour chaque entretien du corpus.

L'annexe A.3.3 présente de plus amples détails sur l'accord inter-annotateur. Elle approfondit la définition de précision, rappel et F-mesure. Elle affiche des résultats supplémentaires de calcul d'accord.

L'annexe A.3.4 présente le détail de l'analyse au niveau des segments fonctionnels par corpus (AL et VD), par entretien et par annotateur.

L'annexe A.3.5 fournit le détail de l'analyse des dimensions par corpus (AL et VD) et par annotateur.

L'annexe A.3.6 présente la matrice de co-occurrence de dimension du corpus Cogni-CISMeF. Elle décrit également des résultats de co-occurrence disponibles dans la littérature.

Enfin, l'annexe A.3.7 fournit l'analyse des fonctions communicatives par dimension.

A.3.1 Exemples de dialogues annotés du corpus Cogni-CISMeF

Cette section présente cinq exemples de dialogue. Précisons que ces exemples présentent des annotations *possibles*. La tâche étant interprétative, d'autres solutions peuvent s'avérer valables. La section A.3.1 présente deux exemples de dialogue évoluant principalement sur la dimension *Task*. Le premier exemple propose un court dialogue de suggestions d'ajout de mot-clé. Il illustre les dimensions *Task* et *Auto-feedback*. Le deuxième exemple implique les dimensions *Task*, *Auto-feedback* et *Time Management*. Les trois autres exemples sont situés dans la section A.3.1. Ils illustrent des dialogues mêlant gestion de la tâche et gestion de l'interaction. Les deux premiers dialogues impliquent les dimensions *Task*, *Auto-feedback* et *Allo-feedback*. Enfin, le troisième implique cinq dimensions : *Task*, *Auto-feedback*, *Allo-feedback*, *Turn Management* et *Partner Communication Management*.

Exemple de dialogue progressant sur la tâche

Le dialogue A.1 présente un exemple de dialogue évoluant sur l'avancement de la tâche et illustrant l'enchaînement de deux suggestions. Cet extrait est annoté sur deux dimensions : *Task* et *Auto-feedback*.

	Task	*AutoFB*
S1 : on peut enlever analyse	*Suggestion*	
A2 : alors enlevons analyse	*AcceptSuggestion*	*AutoPositive*
S3 : et diagnostic	*Suggestion*	
A4 : oui	*AcceptSuggestion*	

Dialogue A.1 – Exemple de dialogue annoté (entretien VD06) : une succession de deux suggestions. Cet extrait est annoté sur deux dimensions : *Task* et la dimension *Auto-feedback*. *AutoFB* = *Auto-feedback*

Dans le premier tour de parole (**S**1), l'interlocuteur S propose d'enlever le mot-clé « analyse »

de la requête, ce qui a été annoté comme une fonction communicative *Suggestion* dans la dimension *Task*. Dans le deuxième tour (**A**2), l'interlocuteur A accepte cette suggestion. Notons qu'il fournit également un feedback positif en répétant une partie de la suggestion « enlevons analyse ». Il s'agit d'un segment fonctionnel annoté sur deux dimensions : avec la fonction communicative *Suggestion* sur la dimension *Task*, et la fonction communicative *AutoPositive* spécifique à la dimension *Auto-feedback*. Il s'ensuit une seconde suggestion qui consiste à supprimer le mot-clé « diagnostic » de la requête, annotée de la même manière que la première suggestion (tour **S**3). Cette seconde suggestion est acceptée au quatrième tour de parole mais cette fois sans feedback explicite ce qui conduit à une annotation avec la fonction *AcceptSuggestion* sur la dimension *Task*.

Le dialogue A.2 présente un exemple de dialogue évoluant principalement sur l'avancement de la tâche. C'est un dialogue qui intervient entre l'expert et le demandeur sur la sélection de qualificatifs lors de la construction de la première requête. Cet exemple implique les dimensions *Task*, *Auto-feedback* et *Time Management*. Dans le tour **S**1, l'expert définit au demandeur ce qu'est un qualificatif. Il lui décrit la procédure de sélection de qualificatif. Puis, il demande au demandeur de l'informer des qualificatifs intéressants. Le demandeur informe l'expert d'un qualificatif intéressant (« chimiothérapie ») dans le tour **A**2. L'expert produit alors un signal de retour et demande à l'utilisateur s'il souhaite continuer la sélection de qualificatif (tour **S**3). Celui-ci accepte (tour **A**4). L'expert confirme la continuation de l'activité de sélection de qualificatif bien qu'il réalise un lapsus entre « accès thématique » et « qualificatif » (tour **S**5). La sélection de qualificatif continue jusqu'au tour **S**11. Les partenaires sélectionnent les qualificatifs « diagnostic », « effets indésirables » et « sang ». Notons l'utilisation par l'expert de la fonction communicative *Pausing* dans la dimension *Time Management* dans les tours **S**5, **S**7 et **S**9. À chacun de ces tours, l'expert suspend le dialogue afin d'inscrire le qualificatif choisi par le demandeur. L'expert interrompt alors la sélection de qualificatif. Il offre la possibilité au demandeur de regarder les types de ressources spécifiques pour les patients. L'utilisateur accepte cette offre (tour **A**12). Dans le tour **S**13, l'expert produit un feedback positif de l'acceptation de l'offre par l'utilisateur. Il l'informe qu'il ajoute le type de ressources « patient » à la requête. Il suggère ensuite de lancer la requête.

	Task	*AutoFB*	*Time M*
S1 : et le qualificatif permet de préciser un tout petit peu	*Inform*		
ce mot clé / donc par rapport à votre question			
c'est de la même façon	*Inform*		
je vais faire dérouler le menu	*Inform*		
et vous allez me dire si vous voyez des choses	*Instruct*		
A2 : chimiothérapie	*Inform*		
S3 : alors chimiothérapie ok		*AutoPositive*	
alors vous souhaitez mettre d'autres qualificatifs	*CheckQuestion*		
A4 : oui	*Confirm*		
S5 : D'autres précisions			
alors on va aller chercher / accès thématique	*Inform*		
non c'est pas accès thématique c'est qualificatif			
voilà donc je mets en fait / j'inscris qualificatif			*Pausing*
[...]			
A6 : diagnostic	*Inform*		
S7 : diagnostic ok		*AutoPositive*	
donc je vais aller l'écrire diagnostic			*Pausing*
ok je continue à faire dérouler le menu de qualificatifs	*Inform*		
A8 : effets indésirables	*Inform*		
S9 : effets indésirables / ok		*AutoPositive*	
alors effets indésirables			*Pausing*
donc je continue à faire dérouler les types de ressources	*Inform*		
/ donc alors le menu déroulant			
A10 : sang	*Inform*		
S11 : allons-y / sang		*AutoPositive*	
et de toute façon là je pense que ça va être à peu près	*Inform*		
tout			
[...]			
alors souhaitez-vous qu'on aille regarder dans un type	*Offer*		
de ressource particulière relative aux associations /			
base de données / bibliothèque médicale / etc / ou			
quelque chose qui soit spécifique pour les patients			
A12 : ouais	*AcceptOffer*		
S13 : oui		*AutoPositive*	
alors je mets type de ressource : patient	*Inform*	*AutoPositive*	
bon on va le lancer comme ça	*Suggestion*		

Dialogue A.2 – Exemple de dialogue annoté (entretien VD02) : sélection de qualificatif dans la construction de la première requête. La dimension *Own Communication Management* a été volontairement exclue pour améliorer la lisibilité. *AutoFB = Auto-feedback*, *Time M = Time Management*.

Exemples de dialogues progressant sur l'axe de la tâche et de l'interaction

Le dialogue A.3 présente un exemple de demande de verbalisation du besoin d'information du demandeur. Cet exemple implique les dimensions *Task*, *Auto-feedback* et *Allo-feedback*.

	Task	*AutoFB*	*AlloFB*
S1 : alors formulez-moi une de vos demandes	*Request*		
alors comme en langage naturel j'allais dire			
A2 : vous êtes sociologue	*Inform*		
c'est ça			*EvalFBElicitation*
S3 : moi je suis linguiste	*Correction*		
A4 : linguistique / psycho-linguistique / sociolinguistique	*ChoiceQuestion*		
S5 : non sociolingue	*Answer*		
A6 : sociolinguistique		*CheckQuestion*	
S7 : sociolinguistique			*Confirm*
A8 : D'accord		*AutoPositive*	
alors comment je vais formuler ma demande	*AcceptRequest*	*AutoPositive*	
oui je dois simplement poser une question	*CheckQuestion*		
S9 : oui vous me posez une question	*Confirm*	*AutoPositive*	

Dialogue A.3 – Exemple de dialogue annoté (entretien VD05) : demande de verbalisation du besoin d'information. *AutoFB* = *Auto-feedback*, *AlloFB* = *Allo-feedback*.

L'expert demande à l'utilisateur de verbaliser sa demande (tour **S**1). Le demandeur ne considère pas cette requête avant le tour **A**8. Avant cela, il affirme la profession de l'expert et lui demande d'évaluer son assertion (tour **A**2). Une analyse alternative peut considérer ce tour comme étant une *CheckQuestion*. L'expert corrige l'utilisateur en lui précisant qu'il est linguiste (tour **S**3). Le demandeur surenchérit en interrogeant l'expert sur son champ d'étude (tour **A**4). L'expert lui répond qu'il est « sociolingue » (tour **S**5). S'ensuit un sous-dialogue de clarification (tours **A**6, **S**7 et **A**8) où l'utilisateur demande à l'expert de confirmer qu'il appartient au champ de la sociolinguistique (tour **A**6). Ce que l'expert fait (tour **S**7). Le demandeur produit un signal de retour positif (tour **A**8). Il considère dans le même tour la requête initiale. Il demande alors à l'expert de lui confirmer s'il doit énoncer son besoin d'information sous forme de question. Ce dernier lui confirme (tour **S**9).

Le dialogue A.4 implique les dimensions *Task*, *Auto-feedback* et *Allo-feedback*.

	Task	*AutoFB*	*AlloFB*
S1 : qu-est-ce que vous entendez par problèmes avec la nourriture ?	*SetQuestion*		
A2 : euh : la relation qu'une personne peut entretenir vis-à-vis du fait d'ingérer des aliments	*Answer*		
donc typiquement j'ai pensé à l'anorexie mais bon	*Inform*		
S3 : d'accord		*AutoPositive*	
A4 : mais peut-être que c'est euh trop spécifique ?	*CheckQuestion*		
S5 : donc l'anorexie et les problèmes du même genre ?		*CheckQuestion*	
A6 : oui			*Confirm*
S7 : donc on va essayer de trouver euh : en tapant anorexie déjà	*Suggestion*		

Dialogue A.4 – Exemple de dialogue annoté (entretien AL07) : question/réponse suivie de clarifications. La dimension *Time Management* a été volontairement exclue pour améliorer la lisibilité. *AutoFB* = *Auto-feedback*, *AlloFB* = *Allo-feedback*.

Les trois premiers tours présentent un exemple d'un échange question/réponse (tour **S**1 et début du tour **A**2) suivi d'un don d'information spontané (fin du tour **A**2). Dans le tour **A**4, l'utilisateur pose une question à l'expert qui ne sera jamais considérée. L'expert demande en contrepartie une confirmation de l'information donnée par le demandeur en **A**2 (tour **S**5). Ce que l'utilisateur lui fournit (tour **A**6). L'expert suggère alors de lancer la requête avec le mot-clé « anorexie » (tour **S**7).

Le dialogue A.5 propose l'annotation du dialogue 5.1 que nous avons déjà rencontré en section 5.1.1. Il implique les dimensions *Task*, *Auto-feedback*, *Allo-feedback*, *Turn Management* et *Partner Communication Management*. Dans le premier tour de parole de ce dialogue, l'utilisateur (S) demande à l'expert (A) de garder les mots-clés « membres inférieurs » et « néphrite ». Les interlocuteurs sont interrompus par un problème technique. Ce problème est verbalisé par le début du tour **A**2 de l'expert. Dans la suite du dialogue, l'expert va accepter la requête du demandeur en la répétant (tours **A**2, **A**4, **A**6). Ces trois tours forment le même segment fonctionnel. Ils sont tous annotés avec la fonction *AcceptRequest* sur la dimension *Task*, et la fonction *AutoPositive* sur la dimension *Auto-feedback* (puisqu'en répétant la requête, l'interlocuteur donne un feedback positif). Le segment fonctionnel est interrompu par l'autre interlocuteur dans le tour **S**3 qui l'informe de faire attention. Ce tour est annoté par une fonction *Inform* sur la dimension *Task* associée à une prise de tour de parole via la fonction *Turn Grab* de la dimension *Turn Management*. Le demandeur interrompt de nouveau l'expert dans le tour **S**5. Dans celui-ci, il valide la compréhension de la requête par l'expert, i.e. il donne son avis sur les processus de l'expert engagés dans la compréhension du tour **S**1. Ce retour positif est annoté comme une fonction *AlloPositive* dans la dimension *Allo-feedback*. Ce retour est associé à une prise de parole inattendue (*Turn Grab*). Le demandeur réalise la même opération au tour **S**7 à l'exception de la prise de parole qui est implicitée par une complétion (annotée par la fonction *Completion* dans la dimension *Partner Communication Management*). Pour finir, l'expert produit un retour positif de la complétion du demandeur (tour **A**8). Il suggère alors de lancer la requête.

	Task	*AutoFB*	*AlloFB*	*Turn M*	*PCM*
S1 : [. . .] on va garder membres inférieurs / on va garder néphrite et	*Request*				
A2 : *pourquoi / bon c'est pas grave je vais revenir en disant que ça m'est déjà arrivée / dans ce cas là je vais vous noter en 009bis et je serais que c'est vous /*					
donc on a dit qu'on mettait	*AcceptRequest*	*AutoPositive*			
S3 : attention	*Inform*			*Turn Grab*	
A4 : néphrite	*AcceptRequest*	*AutoPositive*			
S5 : ouais			*AlloPositive*	*Turn Grab*	
A6 : et qu'on gardait	*AcceptRequest*	*AutoPositive*			
S7 : membres inférieurs			*AlloPositive*		*Completion*
A8 : D'accord		*AutoPositive*			
alors allons-y	*Suggestion*				

Dialogue A.5 – Exemple de dialogue annoté (VD08) : segment fonctionnel sur plusieurs tours de parole. En italique est représentée la verbalisation du problème technique survenu. Le texte en bleu représente le segment fonctionnel sur plusieurs tours. *AutoFB = Auto-feedback*, *AlloFB = Allo-feedback*, *Turn M = Turn Management*, *PCM = Partner Communication Management*.

A.3.2 Proportion des tours de parole annotés

Cette annexe fournit le détail des tours de parole (total, annoté, exclu) pour chaque entretien du corpus. Le tableau A.2 présente l'analyse pour le corpus AL tandis que le tableau A.3 présente les résultats pour le corpus VD. Les tours de parole peuvent être exclus parce qu'ils font partie d'une discussion libre hors du cadre CISMeF (ligne « Hors CISMeF ») ou parce qu'ils ne sont pas interprétables compte tenu des indices (ligne « Incompréhensible »). On peut remarquer que les tours « Hors CISMeF » et « incompréhensibles » interviennent principalement dans le corpus VD.

	AL02	AL03	AL04	AL05	AL06	AL07	AL09	AL10	AL11	AL12	
Tours	11	19	22	32	23	36	43	43	12	30	271
Annoté	11	19	22	32	23	34	43	43	12	30	269 (99%)
Hors CISMeF						0					0
Incompréhensible						2					2 (1%)

Tableau A.2: Proportion des tours de parole annotés sur le corpus AL

	VD02	VD03	VD04	VD05	VD06	VD07	VD08	VD09	
Tours	70	149	91	253	189	107	196	119	1174
Annoté	42	102	56	184	131	81	116	75	787 (67%)
Hors CISMeF	0	0	23	65	56	14	80	28	266 (23%)
Incompréhensible	28	47	12	4	2	12	0	16	121 (10%)

Tableau A.3: Proportion des tours de parole annotés sur le corpus VD

A.3.3 Accords inter-annotateur

Calcul de l'accord inter-annotateur en terme de précision, rappel et F-measure

L'accord inter-annotateur en terme de précision, rappel et F-measure permet de comparer deux ensembles d'annotation : l'*ensemble de référence* et l'*ensemble de comparaison*. Une annotation de l'ensemble de comparaison est dite correcte si elle est présente dans l'ensemble de référence. La *précision* mesure le nombre d'annotation correcte identifiée dans l'ensemble de comparaison proportionnellement au nombre total d'annotation de cet ensemble. Une précision élevée indique une inclusion forte des annotations de l'ensemble de comparaison dans l'ensemble des annotations de référence. Le *rappel* mesure le nombre d'annotation correcte identifiée dans l'ensemble de comparaison proportionnellement au nombre total d'annotation de l'ensemble de référence. Un rappel élevé indique qu'un nombre important d'annotation de l'ensemble de référence appartient également à l'ensemble de comparaison. Il est nécessaire d'associer une pondération entre *précision* et *rappel*. En effet, il est facile d'obtenir une précision de 100% (il suffit de ne rien identifier) ou un rappel maximum (il suffit d'annoter l'ensemble des annotations possibles). La *F-mesure* est généralement utilisée afin de pondérer précision et rappel. Précision, rappel et F-mesure sont calculés de la manière suivante :

$$\text{Precision} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Spurious} + \text{Partial}}$$

$$\text{Recall} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Missing} + \text{Partial}}$$

$$\text{F-measure} = \frac{(\beta^2 + 1)P \times R}{(\beta^2 P) + R}$$

« Correct » représente le nombre d'annotation appartenant aux deux ensembles (référence et comparaison). « Partial » représente le nombre d'annotation se chevauchant entre l'ensemble de référence et l'ensemble de comparaison (voir ci-après). « Missing » représente le nombre d'annotation appartenant à l'ensemble de référence et n'appartenant pas à l'ensemble de comparaison. « Spurious » représente le nombre d'annotation appartenant à l'ensemble de comparaison et n'appartenant pas à l'ensemble de référence.

Le paramètre β dans le calcul de la F-measure peut être fixé, par exemple, de la manière suivante :

— $\beta = 1$, la précision et le rappel sont pondérés de façon égale (F_1 score)

— $\beta = 0.5$, la précision pèse deux fois plus que le rappel

— $\beta = 2$, le rappel pèse deux fois plus que la précision

Notons que les annotations partielles peuvent être considérées selon trois catégories : *strict*, *lenient* et *average*. Celles-ci varient par leur façon de considérer dans le calcul de la précision et du rappel les annotations qui se chevauchent sans se confondre. Le mode *strict* considère les annotations qui se chevauchent comme incorrectes alors que le mode *lenient* les considère correctes. Le mode *average* est une moyenne des deux modes précédents.

Scores	Accord
$\geq 0,85$	Accord fort
Entre $0,70$ et $0,85$	Accord fiable
Entre $0,5$ et $0,7$	Accord modéré
$< 0,5$	Accord faible

Tableau A.4: Relation entre accord et valeur de l'IAA

Le tableau A.4 présente le lien entre accord et valeur des scores de précision, rappel et F-mesure que nous avons fixé.

Accord inter-annotateur sur la segmentation par corpus

Le tableau A.5 fournit le calcul de l'IAA pour le corpus AL tandis que le tableau A.6 fournit le calcul de l'IAA pour le corpus VD. On peut noter que les scores du corpus VD sont légèrement plus élevés que ceux du corpus AL.

Strict			Lenient			Average		
R	P	F	R	P	F	R	P	F
0,91	0,91	0,91	0,97	0,97	0,97	0,94	0,94	0,94

Tableau A.5: Accord inter-annotateur pour la tâche de segmentation sur le corpus AL. R = Rappel, P = Précision, F = F-measure (F_1 score).

Strict			Lenient			Average		
R	P	F	R	P	F	R	P	F
0,95	0,96	0,95	0,97	0,99	0,98	0,96	0,98	0,97

Tableau A.6: Accord inter-annotateur pour la tâche de segmentation sur le corpus VD. R = Rappel, P = Précision, F = F-measure (F_1 score).

Accord inter-annotateur sur la segmentation et l'étiquetage par corpus

Le tableau A.7 présente l'accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension sur le corpus AL. En comparaison des résultats globaux (cf. section 5.1.3), le corpus démontre des scores nettement plus faibles dans les dimensions *Turn Management* et *SOM*. Les scores restent globalement supérieurs à 0.72, dénotant un accord fiable.

	Strict			Lenient			Average			
Dimension	R	P	F	R	P	F	R	P	F	Prop.
Task	0,77	0,78	0,78	0,82	0,83	0,82	0,8	0,8	0,8	65,93%
Time Management	0,78	0,8	0,79	0,87	0,9	0,88	0,82	0,85	0,84	17,29%
Auto-Feedback	0,72	0,71	0,72	0,76	0,75	0,76	0,74	0,73	0,74	6,07%
OCM	0,36	0,39	0,38	0,44	0,47	0,45	0,4	0,43	0,42	5,22%
Turn Management	0,22	0,59	0,32	0,24	0,65	0,35	0,23	0,62	0,34	2,86%
PCM	0,7	0,78	0,74	0,8	0,89	0,84	0,75	0,83	0,79	0,84%
Discourse Structuring	0,57	0,5	0,53	0,71	0,62	0,67	0,64	0,56	0,6	0,57%
Allo-Feedback	0,43	0,6	0,5	0,43	0,6	0,5	0,43	0,6	0,5	0,49%
Contact Management	0,6	0,6	0,6	1	1	1	0,8	0,8	0,8	0,42%
SOM	0	0	0	0,25	1	0,4	0,12	0,5	0,2	0,31%
Global	0,72	0,75	0,74	0,78	0,82	0,8	0,75	0,78	0,77	

Tableau A.7: Accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension sur le corpus AL. R = Rappel, P = Précision, F = F-measure (F_1 score). OCM = Own Communication Management, PCM = Partner Communication Management, SOM = Social Obligation Management.

Le tableau A.8 présente l'accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension sur le corpus VD. Les résultats sont comparables aux résultats globaux (cf. section 5.1.3). Les scores restent globalement supérieurs à 0.8, dénotant un accord fiable.

	Strict			Lenient			Average			
Dimension	R	P	F	R	P	F	R	P	F	Prop.
Task	0,86	0,87	0,87	0,88	0,89	0,88	0,87	0,88	0,88	69,98%
Auto-Feedback	0,79	0,84	0,81	0,81	0,86	0,83	0,8	0,85	0,82	11,83%
OCM	0,41	0,52	0,46	0,45	0,57	0,5	0,43	0,55	0,48	5,38%
Time Management	0,7	0,75	0,73	0,71	0,76	0,74	0,71	0,76	0,73	4,74%
Turn Management	0,46	0,78	0,58	0,49	0,84	0,62	0,48	0,81	0,6	2,68%
SOM	0,48	0,7	0,57	0,5	0,74	0,6	0,49	0,72	0,58	1,80%
Allo-Feedback	0,64	0,51	0,57	0,68	0,54	0,6	0,66	0,53	0,59	1,69%
PCM	0,88	0,96	0,92	0,88	0,96	0,92	0,88	0,96	0,92	1,28%
Contact Management	1	0,38	0,55	1	0,38	0,55	1	0,38	0,55	0,30%
Discourse Structuring	0,8	0,67	0,73	0,8	0,67	0,73	0,8	0,67	0,73	0,29%
Global	0,8	0,83	0,81	0,81	0,85	0,83	0,8	0,84	0,82	

Tableau A.8: Accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension sur le corpus VD. R = Rappel, P = Précision, F = F-measure (F_1 score). OCM = Own Communication Management, PCM = Partner Communication Management, SOM = Social Obligation Management.

Accord inter-annotateur sans impact hiérarchique

Le calcul de l'IAA ne prend pas en compte les propriétés taxonomiques de DIT++. Pour étudier l'impact de ces propriétés, nous avons re-calculé l'IAA en considérant que toutes les fonctions partageant la même racine sont équivalentes (cf. figures 1.1 et 1.2, p. 25). Par exemple, une `Check Question` et une `Propositional Question` sont vues comme équivalentes. Notons que cette méthode réalise des équivalences abusives. Une fonction `Correction` est ainsi équivalente à un `Answer`. Les résultats de ce nouveau calcul sont présentés dans le tableau A.9. Trois dimensions voient leur score augmenter (les autres restant inchangées). La dimension *Task* et *Allo-feedback* augmentent légèrement leur F-mesure de 0.02 point (en mode *average*). La dimension *OCM* voit ses scores considérablement croître. En mode *average*, sa précision passe à 0.62, son rappel à 0.72 et sa F-mesure à 0.67 atteignant ainsi un accord *modéré haut*.

	Strict			Lenient			Average		
Dimension	R	P	F	R	P	F	R	P	F
Task	0,85	0,86	0,86	0,89	0,9	0,89	**0,87**	**0,88**	**0,88**
Time Management	0,75	0,8	0,77	0,81	0,86	0,83	0,78	0,83	0,8
Auto-Feedback	0,79	0,82	0,8	0,81	0,84	0,83	0,8	0,83	0,81
OCM	0,59	0,68	0,63	0,66	0,76	0,71	**0,62**	**0,72**	**0,67**
Turn Management	0,37	0,73	0,49	0,41	0,8	0,54	0,39	0,76	0,51
Allo-Feedback	0,63	0,54	0,58	0,69	0,59	0,63	**0,66**	**0,56**	**0,61**
SOM	0,43	0,69	0,53	0,48	0,76	0,59	0,46	0,72	0,56
PCM	0,86	0,91	0,89	0,86	0,91	0,89	0,86	0,91	0,89
Discourse Structuring	0,67	0,57	0,62	0,75	0,64	0,69	0,71	0,61	0,65
Contact Management	0,67	0,46	0,55	0,89	0,62	0,73	0,78	0,54	0,64
Global	0,8	0,83	0,81	0,83	0,87	0,85	0,81	0,85	0,83

Tableau A.9: Accord inter-annotateur pour les tâches de segmentation et d'étiquetage par dimension sans impact hiérarchique. R = Rappel, P = Précision, F = F-measure (F_1 score). OCM = Own Communication Management, PCM = Partner Communication Management, SOM = Social Obligation Management.

A.3.4 Analyse au niveau des segments fonctionnels

Cette annexe présente le détail de l'analyse au niveau des segments fonctionnels par corpus (AL et VD), par entretien et par annotateur. Chaque tableau présente le nombre de fonctions annotées, le nombre de segments et le ratio du nombre de fonction par segment. Le tableau A.10 présente les résultats pour l'annotation du corpus AL par l'annotateur 1. Le tableau A.11 présente les résultats pour l'annotation du corpus AL par l'annotateur 2. Le tableau A.12 présente les résultats pour l'annotation du corpus VD par l'annotateur 1. Le tableau A.13 présente les résultats pour l'annotation du corpus VD par l'annotateur 2.

	AL02	AL03	AL04	AL05	AL06	AL07	AL09	AL10	AL11	AL12	Total
Fonctions	83	51	154	152	167	128	179	184	93	149	1340
Segments	68	45	134	127	134	112	148	156	69	134	1127
Fonctions / Segment	1,22	1,13	1,15	1,20	1,25	1,14	1,21	1,18	1,35	1,11	1,19

Tableau A.10: Nombre de fonctions communicatives et de segments fonctionnels pour chaque entretien du corpus AL pour l'annotateur 1

	AL02	AL03	AL04	AL05	AL06	AL07	AL09	AL10	AL11	AL12	Total
Fonctions	70	54	143	156	167	115	175	171	100	134	1285
Segments	67	49	136	128	137	111	147	150	72	130	1127
Fonctions / Segment	1,05	1,10	1,05	1,22	1,22	1,04	1,19	1,14	1,39	1,03	1,14

Tableau A.11: Nombre de fonctions communicatives et de segments fonctionnels pour chaque entretien du corpus AL pour l'annotateur 2

	VD02	VD03	VD04	VD05	VD06	VD07	VD08	VD09	Total
Fonctions	154	251	176	365	318	194	247	199	1904
Segments	134	235	159	315	248	178	202	160	1631
Fonctions / Segment	1,15	1,07	1,11	1,16	1,28	1,09	1,22	1,24	1,17

Tableau A.12: Nombre de fonctions communicatives et de segments fonctionnels pour chaque entretien du corpus VD pour l'annotateur 1

	VD02	VD03	VD04	VD05	VD06	VD07	VD08	VD09	Total
Fonctions	143	229	172	345	297	192	243	193	1814
Segments	133	227	160	310	242	178	192	159	1601
Fonctions / Segment	1,08	1,01	1,08	1,11	1,23	1,08	1,27	1,21	1,13

Tableau A.13: Nombre de fonctions communicatives et de segments fonctionnels pour chaque entretien du corpus VD pour l'annotateur 2

A.3.5 Analyse au niveau des dimensions

Cette annexe fournit le détail de l'analyse des dimensions par corpus (AL et VD) et par annotateur. Les tableaux A.14 présentent la proportion de fonctions annotées dans chaque dimension et pour chaque annotateur pour le corpus AL. Ils font également apparaître la synthèse pour le corpus AL. Les tableaux A.15 présentent la proportion de fonctions annotées dans chaque dimension et pour chaque annotateur pour le corpus VD. Ils font également apparaître la synthèse pour le corpus VD.

De manière intéressante, les corpus AL et VD font ressortir les quatre mêmes dimensions majoritaires sans variation significative entre les annotateurs. Une différence notable entre les deux corpus concerne la dimension *Time Management*. Cette dernière est nettement plus présente dans le corpus AL (17.29%) que dans le corpus VD (4.74%). Ce phénomène s'explique par les longs tours de parole de l'expert AL qui impose l'usage de segments pour combler les temps d'attente (e.g., *« euh »*, *« alors »*). En outre, le corpus VD possède proportionnellement plus de fonctions dans la dimension *Auto-feedback* (11.83%) que le corpus AL (6.07%). Cela s'explique par les interventions plus fréquentes du demandeur dans ce corpus qui nécessitent l'usage de feedbacks afin d'assurer la compréhension mutuelle. Les deux autres dimensions principales (*Task* et *OCM*) interviennent dans les mêmes proportions.

(a) Détail par annotateur pour le corpus AL

	Annotateur 1	Annotateur 2	Total
Allo-Feedback	0.52%	0.47%	0.49%
Auto-Feedback	5.83%	6.32%	6.07%
Contact Management	0.44%	0.39%	0.42%
Discourse Structuring	0.52%	0.62%	0.57%
Own Communication Management	5.15%	5.29%	5.22%
Partner Communication Management	0.83%	0.84%	0.84%
Social Obligations Management	0.46%	0.15%	0.31%
Task	64.84%	67.08%	65.93%
Time Management	17.46%	17.11%	17.29%
Turn Management	3.95%	1.72%	2.86%

(b) Synthèse pour le corpus AL

Dimension	Proportion
Task	65.93%
Time Management	17.29%
Auto-Feedback	6.07%
OCM	5.22%
Turn Management	2.86%
PCM	0.84%
Discourse Structuring	0.57%
Allo-Feedback	0.49%
Contact Management	0.42%
SOM	0.31%

Tableau A.14: Proportions de fonctions par dimension dans le corpus AL par annotateur et au total

(a) Détail par annotateur pour le corpus VD

	Annotateur 1	Annotateur 2	Total
Allo-Feedback	1,45%	1,93%	1,69%
Auto-Feedback	11,92%	11,73%	11,83%
Contact Management	0,15%	0,45%	0,30%
Discourse Structuring	0,26%	0,33%	0,29%
Own Communication Management	5,89%	4,85%	5,38%
Partner Communication Management	1,29%	1,26%	1,28%
Social Obligations Management	2,10%	1,49%	1,80%
Task	68,79%	71,22%	69,98%
Time Management	4,78%	4,69%	4,74%
Turn Management	3,30%	2,04%	2,68%

(b) Synthèse pour le corpus VD

Dimension	Proportion
Task	69.98%
Auto-Feedback	11.83%
OCM	5.38%
Time Management	4.74%
Turn Management	2.68%
SOM	1.80%
Allo-Feedback	1.69%
PCM	1.28%
Contact Management	0.30%
Discourse Structuring	0.29%

Tableau A.15: Proportions de fonctions par dimension dans le corpus VD par annotateur et au total

A.3.6 Co-occurrence des dimensions

Le tableau A.16 présente la matrice de co-occurrence de dimension pour le corpus Cogni-CISMeF. Chaque ligne affiche le nombre relatif de fois qu'un segment fonctionnel contenant une fonction dans la dimension en tête de ligne possède également une fonction dans la dimension en colonne.

Nous obtenons les co-occurrences suivantes pour les cinq dimensions principales (ordonnées par proportion d'occurrence décroissante) :

Task : *Own Communication Management*, *Time Management*, *Auto-feedback*, *Turn Management*.

Auto-feedback : *Task*, *Partner Communication Management*, *Turn Management*, *Time Management*.

Time Management : *Task*, *Turn Management*, *Own Communication Management*, *Auto-feedback*.

Own Communication Management : *Task*, *Time Management*, *Turn Management*.

Turn Management : *Task*, *Time Management*, *Own Communication Management*, *Partner Communication Management*.

Le corpus AMI obtient les co-occurrences de dimensions suivantes (ordonnées par proportion d'occurrence décroissante) [Bunt 2009] [1] :

Task : ***Turn Management***, ***Own Communication Management***, *Social Obligation Management*, ***Auto-feedback***

Auto-feedback : ***Turn Management***, ***Task***, ***Time Management***, *Discourse Structuring*

Time Management : ***Task***, ***Turn Management***, ***Auto-feedback***

Own Communication Management : ***Task***, ***Turn Management***

Turn Management : ***Time Management***, ***Task***, *Auto-feedback*, ***Own Communication Management***, *Allo-feedback*

De manière intéressante, nous pouvons constater que 14 dimensions sur 18 se recoupent avec nos résultats de co-occurrence (soit environ 78%).

A.3.7 Analyse au niveau des fonctions communicatives

Cette annexe fournit l'analyse des fonctions communicatives qui interviennent dans les 10 dimensions de DIT++ (i.e. *Task*, *Time Management*, *Own Communication Management*, *Auto-feedback*, *Turn Management*, *Allo-feedback*, *Social Obligation Management*, *Partner Communication Management*, *Discourse Structuring* et *Contact Management*).

Dimension *Task*

Les tableaux A.17 présentent la proportion de fonctions annotées observée dans la dimension *Task* pour l'expert et le demandeur. Le tableau A.18 informe de la proportion de fonctions annotées observée dans la dimension *Task* de manière globale.

Le tableau A.19 présente la considération verbale des fonctions d'initiative d'action (suggestion, requête, offre) produites par l'expert ou par le demandeur. Il est intéressant de noter que la considération verbale des initiatives de l'expert est rare (14% pour les suggestions, 29% pour les requêtes et 30% pour les offres). Ces faibles chiffres peuvent s'expliquer par (i) l'absence des indices non verbaux (e.g., l'initiative est acceptée ou refusée par un mouvement de la tête) (ii) la position dominante de l'expert dans l'expérimentation qui suggère une action et la réalise en supposant l'accord du demandeur. Au contraire, il est essentiel de remarquer que la considération verbale des initiatives du demandeur est élevée (73% pour les suggestions, 79% pour les requêtes et 100% pour les offres).

1. En gras, nous représentons les co-occurrences également présentes dans notre corpus

	Task	Auto-F.	Allo-F.	Turn M.	Time M.	DS	Contact M.	OCM	PCM	SOM
Task		2.63%	0.11%	2.38%	5.13%	0.14%	0.09%	6.20%	0.66%	0.11%
Auto-Feedback	18.83%		0.00%	1.79%	1.46%	0.00%	0.00%	0.65%	1.95%	0.49%
Allo-Feedback	6.58%	0.00%		3.95%	1.32%	0.00%	0.00%	5.26%	6.58%	0.00%
Turn Management	53.57%	5.61%	1.53%		19.39%	0.51%	2.04%	3.57%	3.57%	1.53%
Time Management	33.28%	1.33%	0.15%	5.60%		0.00%	0.00%	5.45%	0.15%	0.29%
Discourse Structuring	22.22%	0.00%	0.00%	3.70%	0.00%		11.11%	0.00%	0.00%	7.41%
Contact Management	16.67%	0.00%	0.00%	16.67%	0.00%	12.50%		8.33%	0.00%	0.00%
OCM	70.18%	1.03%	1.03%	1.80%	9.51%	0.00%	0.51%		0.00%	1.29%
PCM	37.18%	15.38%	6.41%	8.97%	1.28%	0.00%	0.00%	0.00%		0.00%
SOM	6.25%	3.75%	0.00%	3.75%	2.50%	2.50%	0.00%	6.25%	0.00%	

Tableau A.16: Co-occurrence des dimensions dans le corpus Cogni-CISMeF. Chaque ligne affiche le nombre relatif de fois qu'un segment fonctionnel contenant une fonction dans la dimension en tête de ligne possède également une fonction dans la dimension en colonne. DS = Discourse Structuring, OCM = Own Communication Management, PCM = Partner Communication Management, SOM = Social Obligation Management.

(a) Expert

Fonction	Prop. globale
Inform	38.20%
Suggestion	13.40%
Offer	3.00%
Check Question	2.40%
Agreement	1.60%
AcceptSuggestion	1.50%
Set Question	1.26%
Request	1.10%
AcceptRequest	0.90%
Answer	0.90%
Instruct	0.80%
Promise	0.70%
Choice Question	0.40%
Confirm	0.40%
Disconfirm	0.40%
Nega-check	0.30%
Propositional Question	0.30%
Correction	0.20%
DeclineSuggestion	0.20%
Posi-check	0.20%
AcceptOffer	0.00%
Address Request	0.00%
DeclineRequest	0.00%
Disagreement	0.00%

(b) Demandeur

Fonction	Prop. globale
Inform	29.80%
Suggestion	6.40%
Agreement	5.10%
AcceptSuggestion	4.60%
Answer	4.60%
Confirm	3.50%
Request	3.40%
Check Question	2.90%
Set Question	1.55%
AcceptOffer	1.50%
Disconfirm	1.50%
DeclineOffer	0.90%
AcceptRequest	0.60%
Correction	0.40%
DeclineSuggestion	0.40%
Disagreement	0.40%
Address Offer	0.20%
Address Request	0.20%
Address Suggestion	0.20%
Choice Question	0.20%
Propositional Question	0.20%
DeclineRequest	0.10%
Instruct	0.10%
Offer	0.10%
Posi-check	0.10%

Tableau A.17: Répartition des fonctions dans la dimension *Task*

Fonction	Prop. globale
Inform	36.00%
Suggestion	11.50%
Agreement	2.50%
Check Question	2.50%
AcceptSuggestion	2.30%
Offer	2.30%
Answer	1.90%
Request	1.70%
Set Question	1.34%
Confirm	1.20%
AcceptRequest	0.80%
Disconfirm	0.70%
Instruct	0.60%
Promise	0.50%
AcceptOffer	0.40%
Choice Question	0.40%
Correction	0.30%
Propositional Question	0.30%
DeclineOffer	0.20%
DeclineSuggestion	0.20%
Nega-check	0.20%
Posi-check	0.20%
Address Request	0.10%
Address Suggestion	0.10%
Disagreement	0.10%
Address Offer	0.00%
DeclineRequest	0.00%

Tableau A.18: Répartition globale des fonctions dans la dimension *Task*

(a) Initiative par l'expert, considération par le demandeur

Fonction	Total	Considérée
Suggestion	623	88
Request	49	14
Offer	141	43

(b) Initiative par le demandeur, considération par l'expert

Fonction	Total	Considérée
Suggestion	107	78
Request	57	45
Offer	2	2

Tableau A.19: Considération verbale des fonctions initiatives d'action

Dimension *Time Management*

Les tableaux A.20 présentent la proportion de fonctions annotées observée dans la dimension *Time Management* pour l'expert et le demandeur et de manière globale. Cette dimension s'illustre par une majorité écrasante de *Stalling* aussi bien pour l'expert que le demandeur.

(a) Expert

Fonction	Prop. globale
Stalling	10.20%
Pausing	1.70%
Inform	0.10%

(b) Demandeur

Fonction	Prop. globale
Stalling	4.10%
Pausing	0.20%
Inform	0.10%

(c) Global

Fonction	Prop. globale
Stalling	8.60%
Pausing	1.30%
Inform	0.10%

Tableau A.20: Répartition des fonctions dans la dimension *Time Management*

Dimension *Own Communication Management*

Les tableaux A.21 présentent la proportion de fonctions annotées observée dans la dimension *Own Communication Management* pour l'expert et le demandeur et de manière globale. Cette dimension s'illustre par l'usage de deux fonctions *Self-correction* et *Retraction* de la part de l'expert et du demandeur.

(a) Expert

Fonction	Prop. globale
Self-correction	2.90%
Retraction	2.30%
Error signaling	0.00%

(b) Demandeur

Fonction	Prop. globale
Self-correction	3.60%
Retraction	2.00%

(c) Global

Fonction	Prop. globale
Self-correction	3.10%
Retraction	2.20%
Error signaling	0.00%

Tableau A.21: Répartition des fonctions dans la dimension *Own Communication Management*

Dimension *Auto-feedback*

Les tableaux A.22 présentent la proportion de fonctions annotées observée dans la dimension *Auto-feedback* pour l'expert et le demandeur et de manière globale. Cette dimension s'illustre par une majorité écrasante de feedback positif sous-spécifié (*AutoPositive*).

(a) Expert

Fonction	Prop. globale
AutoPositive	8.70%
Check Question	0.30%
EvalPositiveAutoFeedback	0.10%
ExecNegativeAutoFeedback	0.10%
Inform	0.10%
PerceptNegativeAutoFeedback	0.10%
Posi-check	0.10%
Set Question	0.10%
AutoNegative	0.00%
EvalPositiveAlloFeedback	0.00%
InterprPositiveAutoFeedback	0.00%
PerceptPositiveAutoFeedback	0.00%

(b) Demandeur

Fonction	Prop. globale
AutoPositive	7.40%
Check Question	0.30%
EvalPositiveAutoFeedback	0.20%
AutoNegative	0.10%
ExecNegativeAutoFeedback	0.10%
InterprNegativeAutoFeedback	0.10%
PerceptNegativeAutoFeedback	0.10%
Posi-check	0.10%

(c) Global

Fonction	Prop. globale
AutoPositive	8.40%
Check Question	0.30%
EvalPositiveAutoFeedback	0.10%
ExecNegativeAutoFeedback	0.10%
Inform	0.10%
PerceptNegativeAutoFeedback	0.10%
Posi-check	0.10%
Set Question	0.10%
AutoNegative	0.00%
InterprNegativeAutoFeedback	0.00%
InterprPositiveAutoFeedback	0.00%
PerceptPositiveAutoFeedback	0.00%

Tableau A.22: Répartition des fonctions dans la dimension *Auto-feedback*

Dimension *Turn Management*

Les tableaux A.23 présentent la proportion de fonctions annotées observée dans la dimension *Turn Management* pour l'expert et le demandeur et de manière globale. Les fonctions apparaissant dans cette dimension sont majoritairement des fonctions spécifiques. Notons que le demandeur réalise la majorité des prises de parole intempestives (*Turn Grab*). Ceci s'explique par le fait que l'expert parle beaucoup. Il doit en effet mener la recherche en tant que spécialiste de CISMeF et également verbaliser ce qu'il fait pour l'expérimentation. Le demandeur est donc amené à interrompre l'expert pour s'exprimer. Cette remarque se vérifie également par l'usage de fonctions dans la dimension *Partner Communication Management*.

(a) Expert

Fonction	Prop. globale
Turn Keep	0.80%
Turn Grab	0.60%
Turn Assign	0.20%
Turn Release	0.10%
Agreement	0.00%
Turn Take	0.00%

(b) Demandeur

Fonction	Prop. globale
Turn Grab	4.50%
Turn Release	0.30%
Turn Assign	0.20%
Turn Take	0.20%
Turn Keep	0.10%

(c) Global

Fonction	Prop. globale
Turn Grab	1.70%
Turn Keep	0.60%
Turn Assign	0.20%
Turn Release	0.10%
Turn Take	0.10%
Agreement	0.00%

TABLEAU A.23: Répartition des fonctions dans la dimension *Turn Management*

Dimension *Allo-feedback*

Les tableaux A.24 présentent la proportion de fonctions annotées observée dans la dimension *Allo-feedback* pour l'expert et le demandeur et de manière globale. Cette dimension est faiblement représentée dans notre corpus. Elle va de pair avec la dimension *Auto-feedback* lors de mauvaises compréhension. Globalement, c'est l'usage de signaux de retour positif sous-spécifiés (*AlloPositive*) et de confirmation aux questions de vérification dans la dimension *Auto-feedback* qui est majoritaire. Notons la présence de fonctions permettant la demande de retours sur ce qui a été dit (*EvalFBElicitation*) illustrées par l'exemple suivant : *« C'est quand même une belle maladie ça, non ? »* (VD04).

(a) Expert

Fonction	Prop. globale
AlloPositive	0.20%
Confirm	0.10%
EvaluationFeedbackElicitation	0.10%
Request	0.10%
AlloNegative	0.00%
Answer	0.00%
Set Question	0.00%

(b) Demandeur

Fonction	Prop. globale
Confirm	0.80%
AlloPositive	0.70%
Answer	0.40%
EvaluationFeedbackElicitation	0.40%
Disconfirm	0.20%
Inform	0.20%
AttentionFeedbackElicitation	0.10%
Check Question	0.10%
InterprNegativeAlloFeedback	0.10%

(c) Global

Fonction	Prop. globale
AlloPositive	0.30%
Confirm	0.30%
EvaluationFeedbackElicitation	0.20%
Answer	0.10%
Disconfirm	0.10%
Request	0.10%
AlloNegative	0.00%
AttentionFeedbackElicitation	0.00%
Check Question	0.00%
Inform	0.00%
InterprNegativeAlloFeedback	0.00%
Set Question	0.00%

Tableau A.24: Répartition des fonctions dans la dimension *Allo-feedback*

Dimension *Social Obligation Management*

Les tableaux A.25 présentent la proportion de fonctions annotées observée dans la dimension *Social Obligation Management* pour l'expert et le demandeur et de manière globale. Cette dimension est faiblement représentée dans notre corpus. Notons la présence des fonctions spécifiques de la dimension (*Thanking*, *Thanking-downplay*, *Apology*, *Apology-downplay*).

(a) Expert

Fonction	Prop. globale
Apology	0.40%
Agreement	0.20%
Inform	0.20%
Thanking	0.10%
Answer	0.00%
Apology-downplay	0.00%
Check Question	0.00%
Disconfirm	0.00%
Offer	0.00%
Request	0.00%

(b) Demandeur

Fonction	Prop. globale
Inform	0.50%
Apology	0.20%
Apology-downplay	0.10%
Check Question	0.10%
Choice Question	0.10%
DeclineOffer	0.10%
Initial greeting	0.10%
Offer	0.10%
Posi-check	0.10%
Thanking	0.10%
Thanking-downplay	0.10%

(c) Global

Fonction	Prop. globale
Apology	0.40%
Inform	0.30%
Agreement	0.10%
Apology-downplay	0.10%
Thanking	0.10%
Answer	0.00%
Check Question	0.00%
Choice Question	0.00%
DeclineOffer	0.00%
Disconfirm	0.00%
Initial greeting	0.00%
Offer	0.00%
Posi-check	0.00%
Request	0.00%
Thanking-downplay	0.00%

Tableau A.25: Répartition des fonctions dans la dimension *SOM*

Dimension *Partner Communication Management*

Les tableaux A.26 présentent la proportion de fonctions annotées observée dans la dimension *Partner Communication Management* pour l'expert et le demandeur et de manière globale. Cette dimension s'illustre par l'usage de la fonction *Completion*. Notons que celle-ci est majoritairement utilisée par le demandeur afin de récupérer le tour de parole (cf. annexe A.3.7 sur la dimension *Turn Management*).

(a) Expert

Fonction	Prop. globale
Completion	0.50%

(b) Demandeur

Fonction	Prop. globale
Completion	2.70%

(c) Global

Fonction	Prop. globale
Completion	1.10%

Tableau A.26: Répartition des fonctions dans la dimension *PCM*

Dimension *Discourse Structuring*

Les tableaux A.27 présentent la proportion de fonctions annotées observée dans la dimension *Discourse Structuring* pour l'expert et le demandeur et de manière globale. Cette dimension est faiblement représentée dans notre corpus.

(a) Expert

Fonction	Prop. globale
Opening	0.20%
Preclosing	0.10%
Inform	0.00%
Topic introduction	0.00%
Topic shift	0.00%
Topic shift annoucement	0.00%

(b) Demandeur

Fonction	Prop. globale
Inform	0.10%
Preclosing	0.10%
Topic shift	0.10%

(c) Global

Fonction	Prop. globale
Opening	0.20%
Inform	0.10%
Preclosing	0.10%
Topic introduction	0.00%
Topic shift	0.00%
Topic shift annoucement	0.00%

Tableau A.27: Répartition des fonctions dans la dimension *Discourse Structuring*

Dimension *Contact Management*

Les tableaux A.28 présentent la proportion de fonctions annotées observée dans la dimension *Contact Management* pour l'expert et le demandeur et de manière globale. Cette dimension est faiblement représentée dans notre corpus.

(a) Expert

Fonction	Prop. globale
Contact Indication	0.30%
Inform	0.00%

(b) Demandeur

Fonction	Prop. globale
Contact Indication	0.20%
Inform	0.10%
Request	0.10%

(c) Global

Fonction	Prop. globale
Contact Indication	0.30%
Inform	0.00%
Request	0.00%

Tableau A.28: Répartition des fonctions dans la dimension *Contact Management*

A.4 Annexes du processus d'extraction

Ces annexes présentent des données et résultats supplémentaires du processus d'extraction.

L'annexe A.4.1 présente les données détaillées de répartition entre le corpus d'extraction et le corpus de référence.

L'annexe A.4.2 fournit des précisions sur les fonctions qui précèdent une fonction *ExecNegativeAutoFB*.

L'annexe A.4.3 présente le détail de l'intervention d'instances de motifs dialogiques dans les entretiens du corpus d'extraction.

Enfin, l'annexe A.4.4 fournit des exemples supplémentaires de dialogue tirés du corpus d'extraction illustrant chacun des motifs dialogiques extraits.

A.4.1 Répartition entre le corpus d'extraction et le corpus de référence

Cette annexe présente la constitution des corpus de référence et des corpus d'extraction. Le tableau A.29 rappelle la distribution des entretiens du corpus Cogni-CISMeF entre les corpus de référence et d'extraction. Le tableau A.30 présente la répartition des tours de parole, des segments fonctionnels et des fonctions communicatives entre le corpus d'extraction et le corpus de référence (en proportion et en chiffres).

	AL02	AL03	AL04	AL05	AL06	AL07	AL09	AL10	AL11	AL12
Sélectionné		×	×		×	×		×	×	×

	VD02	VD03	VD04	VD05	VD06	VD07	VD08	VD09
Sélectionné	×		×	×	×	×		

Tableau A.29: Dialogues sélectionnés pour l'extraction manuelle des motifs d'interaction

	Tour	Segment	Fonction
Corpus d'extraction	1354 (64.11%)	3626 (66.10%)	4166 (65.68%)
Corpus de référence	758 (35.89%)	1860 (33.90%)	2177 (34.32%)
Total	2112 (1056 × 2)	5486	6343

Tableau A.30: Répartition des tours de parole, des segments fonctionnels et des fonctions communicatives entre le corpus d'extraction et le corpus de référence

A.4.2 Cas de la fonction *ExecNegativeAutoFB*

La figure A.1 présente les fonctions qui précèdent la fonction *ExecNegativeAutoFB* observées dans le corpus Cogni-CISMeF. On constate que la fonction *ExecNegativeAutoFB* a été exclusivement observée en réaction à des fonctions représentant une question (*CheckQuestion*, *SetQuestion*, *ChoiceQuestion* et *PropositionalQuestion*).

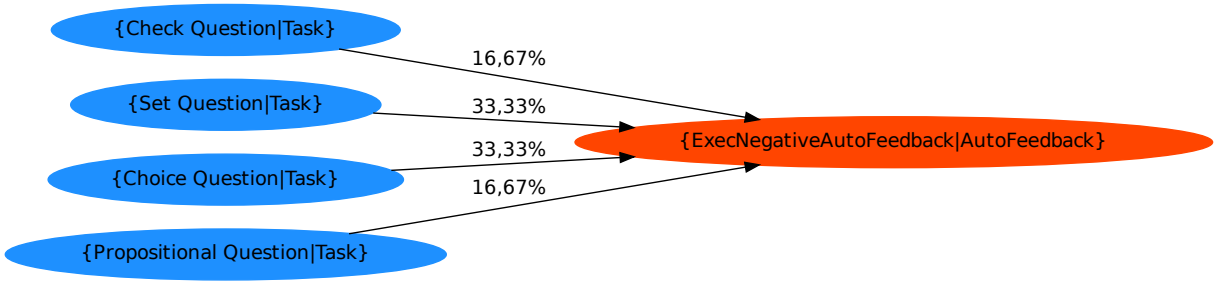

Figure A.1: Occurrence de la fonction *ExecNegativeAutoFB*

A.4.3 Présence des motifs dans les entretiens du corpus d'extraction

Cette annexe présente la présence des motifs dialogiques dans les entretiens du corpus d'extraction. Un motif dialogique est présent dans un entretien s'il possède une instance dans l'entretien annoté par l'annotateur 1 *ou* dans l'entretien annoté par l'annotateur 2. Le tableau A.31 présente la présence des motifs dans le corpus AL tandis que le tableau A.32 présente la présence des motifs dans le corpus VD. Le tableau A.33 présente la synthèse de la présence des motifs dialogiques dans les entretiens du corpus d'extraction.

AL	3	4	6	7	10	11	12	Bilan
Question oui/non	×				×			2/7
Vérification	×	×	×	×	×	×	×	7/7
Vérification positive	×		×					2/7
Vérification négative		×						1/7
Question ouverte	×		×	×	×		×	5/7
Question à choix multiples			×			×		2/7
Suggestion		×	×	×	×	×	×	6/7
Requête				×		×		2/7
Offre		×		×	×			3/7
Accord		×	×	×	×	×	×	6/7
Correction					×			1/7

Tableau A.31: Présence des motifs dans les entretiens du corpus AL

VD	2	4	5	6	7	Bilan
Question oui/non		×		×		2/5
Vérification	×	×	×	×	×	5/5
Vérification positive			×		×	2/5
Vérification négative	×	×			×	3/5
Question ouverte	×	×	×	×		4/5
Question à choix multiples			×	×	×	3/5
Suggestion	×	×	×	×	×	5/5
Requête		×	×	×	×	4/5
Offre	×	×	×	×	×	5/5
Accord	×	×	×	×	×	5/5
Correction	×	×	×			3/5

Tableau A.32: Présence des motifs dans les entretiens du corpus VD

Motif	Bilan
Question oui/non	*4/12*
Vérification	**12/12**
Vérification positive	*4/12*
Vérification négative	*4/12*
Question ouverte	9/12
Question à choix multiples	5/12
Suggestion	**11/12**
Requête	6/12
Offre	8/12
Accord	**11/12**
Correction	*4/12*

Tableau A.33: Bilan de la présence des motifs dans les entretiens

A.4.4 Exemples supplémentaires d'instance de motif

Cette annexe présente des exemples supplémentaires d'instances de motifs dialogiques facilement compréhensibles hors du contexte du dialogue. Pour chaque motif dialogique, un graphe représentant le motif en question est fourni. Chaque nœud du graphe représente l'occurrence d'une fonction communicative dans une dimension. Chaque arc indique une suite possible réalisée par un autre interlocuteur. Les étiquettes sur les arcs informent de la proportion observée dans le corpus d'extraction.

Motif de question oui/non

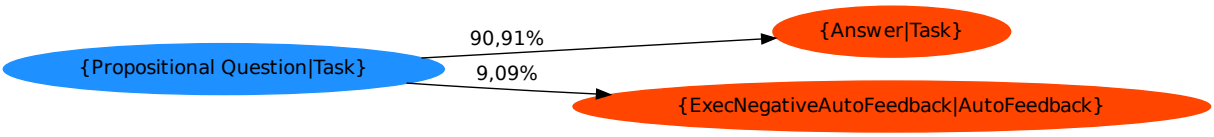

FIGURE A.2: Motif dialogique d'interrogation oui/non

	Task
S1 : alors "pathologie de la xxx xxx" / est-ce que ça vous dit quelque chose ?	*PropositionalQuestion*
A2 : ça se pourrait bien / oui	*Answer*

Dialogue A.6 – Exemple d'un motif de type question oui/non avec seconde partie de paire préférée (entretien AL03)

Motif de vérification

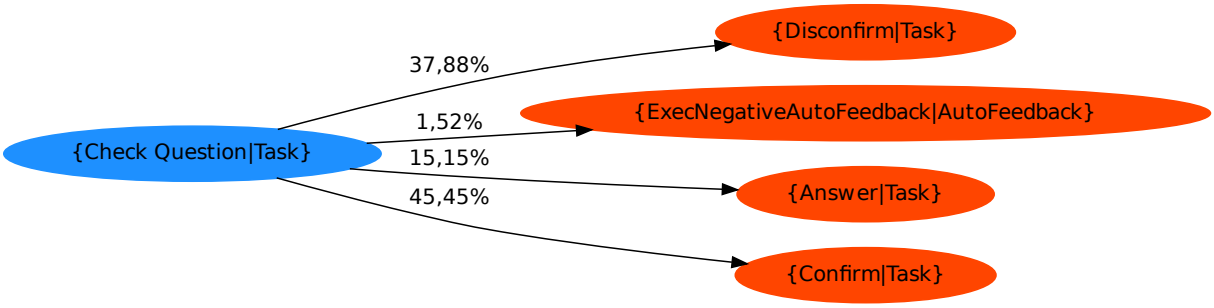

FIGURE A.3: Motif dialogique de vérification d'une information

	Task
S1 : donc vous voulez des informations sur la périarthrite ?	*CheckQuestion*
A2 : voilà	*Confirm*

Dialogue A.7 – Exemple d'un motif de type vérification avec seconde partie de paire préférée (entretien AL3)

	Task
S1 : alors vous souhaitez mettre d'autres qualificatifs finalement ?	*CheckQuestion*
A2 : oui	*Confirm*

Dialogue A.8 – Exemple d'un motif de type vérification avec seconde partie de paire préférée (entretien VD2)

	Task
S1 : C'est tout ce qu'il y a ?	*CheckQuestion*
A2 : non	*Disconfirm*

Dialogue A.9 – Exemple d'un motif de type vérification avec seconde partie de paire non-préférée (entretien VD04)

	Task
S1 : mais il était vraiment enlevé ?	*CheckQuestion*
A2 : je vais vérifier	*Inform*
il n'était pas enlevé	*Disconfirm*

Dialogue A.10 – Exemple d'un motif de type vérification avec seconde partie de paire non-préférée (entretien VD05)

Motif de vérification positive

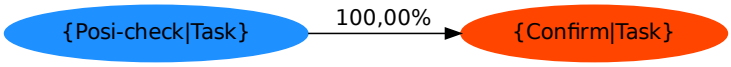

FIGURE A.4: Motif dialogique de vérification positive d'une information

	Task
S1 : alors plus précisémment ce serait le genou alors ?	*PosiCheck*
A2 : oui	*Confirm*

Dialogue A.11 – Exemple d'un motif de type vérification positive (entretien AL06)

	Task
S1 : alors vu la présentation de la fenêtre je suppose qu'il y en a une bonne quantité derrière ?	*PosiCheck*
A2 : oui tout à fait	*Confirm*

Dialogue A.12 – Exemple d'un motif de type vérification positive (entretien VD05)

	Task
S1 : ca vous va ?	*PosiCheck*
A2 : ouais	*Confirm*

Dialogue A.13 – Exemple d'un motif de type vérification positive (entretien VD07)

Motif de vérification négative

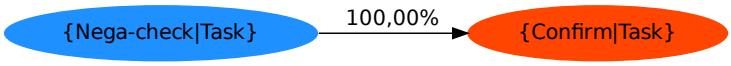

FIGURE A.5: Motif dialogique de vérification négative d'une information

	Task
S1 : on n'avait pas le mot diagnostic ?	*NegaCheck*
A2 : si	*Confirm*

Dialogue A.14 – Exemple d'un motif de type vérification negative (entretien VD02)

	Task
S1 : vous ne voyez pas grand chose non plus ?	*NegaCheck*
A2 : bah j'essaie / j'essaie je regarde	*Inform*
rien / je vois rien	*Confirm*

Dialogue A.15 – Exemple d'un motif de type vérification negative (entretien VD04)

Motif de question ouverte

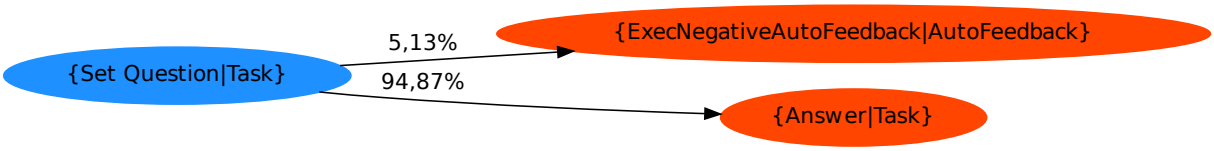

FIGURE A.6: Motif dialogique de demande d'information ouverte

	Task
S1 : donc c'est des douleurs articulaires à quel niveau ?	*SetQuestion*
A2 : au niveau des genoux et autres	*Answer*

Dialogue A.16 – Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien AL06)

	Task
S1 : ça serait dans quelle discipline de sport ?	*SetQuestion*
A2 : le rugby pour le rugby	*Answer*

Dialogue A.17 – Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien AL06)

	Task
S1 : donc document c'est / c'est des documents qui / c'est quel type de document parce que je ne sais pas où se fait la recherche en fait ?	*SetQuestion*
A2 : alors la recherche ça se fait / par un ensemble de documents médicaux qui sont dans un annuaire / c'est-à-dire qui ont été sélectionnés par un équipe de documentalistes spécialisés en médecine	*Answer*
S3 : et ces documents c'est quoi c'est des rapports ? c'est des articles de journaux c'est	*SetQuestion*
A4 : des documents / il y a tout ce qui va s'adresser aux médecins euh : des cours pour les étudiants en médecine	*Answer*

Dialogue A.18 – Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien AL10)

Motif de question à choix multiples

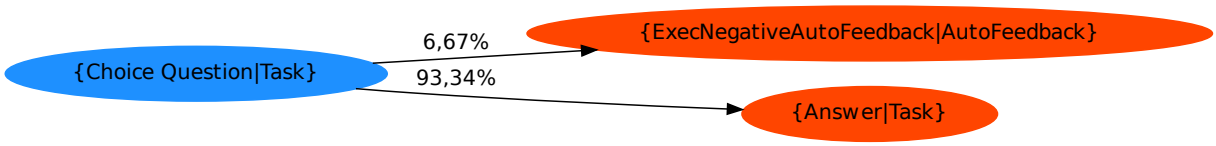

FIGURE A.7: Motif dialogique de demande d'information à choix multiples

	Task
S1 : dans les types de ressources est-ce qu'on va plutôt chercher des choses dans les associations / dans les bases de données / bibliothèque médicale / étude d'évaluation / forum / hôpital / image / des choses sur des lignes de consensus de professionnels / des matériels de l'enseignement / des documents pour les patients / des périodiques / des rapports techniques / des recommandations ou un service hospitalier	*ChoiceQuestion*
A2 : oh service hospitalier pour essayer	*Answer*

Dialogue A.19 – Exemple de motif de type question à choix multiples avec seconde partie de paire préférée (entretien VD07)

	Task	*AutoFB*
S1 : alors on a créatinine quinaz ou créatinine	*ChoiceQuestion*	
A2 : moi je mettrais que créatinine	*Answer*	
S3 : on met que créatinine		*AutoPositive*

Dialogue A.20 – Exemple de motif de type question à choix multiples avec seconde partie de paire préférée (entretien VD07)

Motif de suggestion

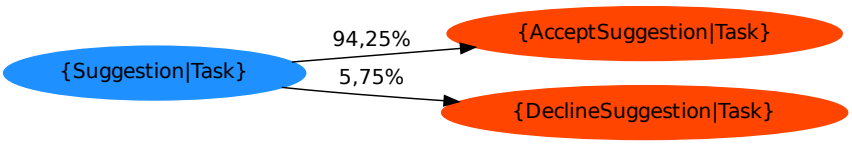

FIGURE A.8: Motif dialogique de suggestion d'action

	Task
S1 : bon on va quand même le lancer comme ça	*Suggestion*
A2 : oui	*AcceptSuggestion*

Dialogue A.21 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD05)

	Task
S1 : on va essayer de voir si on obtient des choses spécifiques pour les patients donc qu'ils soient spécifiques pour les médecins ou pour les étudiants en médecine	*Suggestion* *Inform*
A2 : d'accord	*AcceptSuggestion*

Dialogue A.22 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien AL07)

	Task	*AutoFB*
S1 : en mot clé bah écoutez je vous propose de mettre leucémie	*Suggestion*	
A2 : leucémie	*AcceptSuggestion*	*AutoPositive*

Dialogue A.23 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD02)

	Task
S1 : donc ce que je vous propose c'est qu'on remette effort du coup	*Suggestion*
A2 : oui / oui	*AcceptSuggestion*

Dialogue A.24 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD05)

	Task
S1 : et si on essayait de transformer le mal à la tête en migraine pour essayer de trouver un synonyme	*Suggestion*
A2 : oui / oui / bien sûr	*AcceptSuggestion*

Dialogue A.25 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD05)

	Task
S1 : bon alors donc on essaie quelque chose avec cancérologie et diagnostic	*Suggestion*
A2 : ouais	*AcceptSuggestion*

Dialogue A.26 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD06)

	Task	*AutoFB*
S1 : diagnostic de l'insuffisance rénale donc créatinine / analyse de sang / méthode / donc	*Inform*	
A2 : on peut regarder ça éventuellement	*Suggestion*	
S3 : on peut regarder éventuellement	*AcceptSuggestion*	*AutoPositive*

Dialogue A.27 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD07)

	Task
S1 : alors peut-être qu'on pourrait mettre dans quelque chose qui serait une espèce de mot clé taux de créatinine normal pour voir ce que ça donne	*Suggestion*
A2 : oui	*AcceptSuggestion*

Dialogue A.28 – Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD07)

Motif de requête

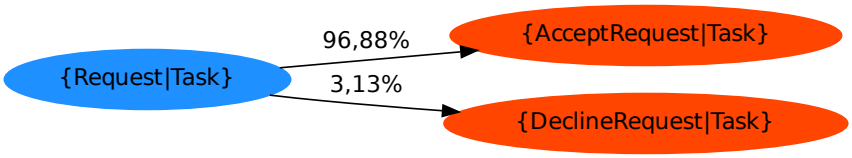

FIGURE A.9: Motif dialogique de demande de réalisation d'action (requête)

	Task
S1 : vous me dites si vous voyez des choses	*Request*
A2 : D'accord	*AcceptRequest*

Dialogue A.29 – Exemple de motif de type requête avec seconde partie de paire préférée (entretien VD05)

Motif d'offre

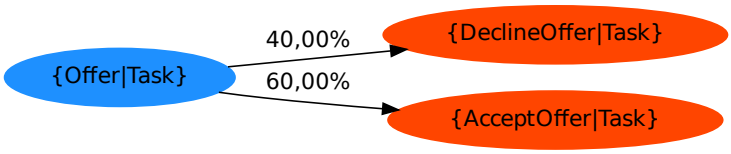

FIGURE A.10: Motif dialogique d'offre de réalisation d'action

	Task
S1 : est-ce que vous avez des questions sur un autre domaine ou ? ?	*Offer*
A2 : oui bah non je vois pas de :	*DeclineOffer*

Dialogue A.30 – Exemple de motif de type offre avec seconde partie de paire non-préférée (entretien AL10)

	Task
S1 : donc je vais sur le texte lui-même / je vais le visualiser [...]	*Offer*
A2 : ouais faut chercher là-dedans	*AcceptOffer*

Dialogue A.31 – Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD06)

	Task
S1 : par contre colon j'aurais presque envie de mettre diagnostic quand même parce que / parce que on va voir ce que ça donne	*Offer*
A2 : oui normalement c'est un diagnostic	*AcceptOffer*

Dialogue A.32 – Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD06)

	Task
S1 : je vais essayer / je vais retourner donc je retourne sur la page et je relance la recherche	*Offer*
A2 : ouais	*AcceptOffer*

Dialogue A.33 – Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD06)

Motif d'accord

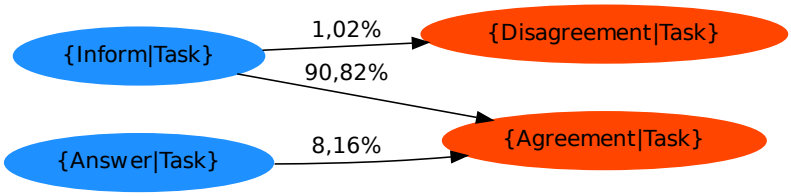

FIGURE A.11: Motif dialogique d'accord

	Task
S1 : mais ça doit pas être ça	*Inform*
A2 : non à mon avis non	*Agreement*

Dialogue A.34 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL04)

	Task
S1 : on a 115 articles	*Inform*
A2 : oui oui	*Agreement*

Dialogue A.35 – Exemple de motif de type accord avec seconde partie de paire préférée (entretien AL07)

	Task
S1 : donc que c'est un accès que n'importe qui peut avoir l'accès	*Inform*
A2 : tout à fait	*Agreement*

Dialogue A.36 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL10)

	Task
S1 : là on a 33 ressources alors effectivement / ça s'annonce bien	*Inform*
A2 : ah oui il y a des choses là	*Agreement*

Dialogue A.37 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL10)

	Task
S1 : donc là on a abouti à un truc qui n'a rien à voir	*Inform*
A2 : voilà	*Agreement*

Dialogue A.38 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL12)

	Task
S1 : bon on le trouve pas dans l'accès thématique	*Inform*
A2 : non	*Agreement*

Dialogue A.39 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien VD05)

	Task
S1 : on cherche mais on ne trouve pas toujours	*Inform*
A2 : exactement c'est tout à fait ça	*Agreement*

Dialogue A.40 – Exemple de motif d'accord avec seconde partie de paire préférée (entretien VD05)

	Task	*AutoFB*
S1 : ces deux là on les avait déjà vues	*Inform*	
A2 : déjà vues aussi	*Agreement*	*AutoPositive*
S3 : la troisième ?	*CheckQuestion*	
A4 : toujours les trois mêmes	*Answer*	
S5 : toujours les trois mêmes	*Agreement*	*AutoPositive*

Dialogue A.41 – Exemple de motif de type accord avec seconde partie de paire préférée (entretien VD06)

Motif de correction

FIGURE A.12: Motif dialogique de correction

	Task
S1 : vous êtes sociologue c'est ça	*Inform*
A2 : moi je suis linguiste	*Correction*

Dialogue A.42 – Exemple de motif de type correction en deux temps (entretien VD05)

Annexe B

Annexes du modèle

B.1 Événéments internes issus des jeux de dialogue

Le tableau B.1 présente les événements internes liés aux jeux de dialogue. Ces événements sont déclenchés dès lors que des conditions d'entrée, d'échec ou de succès sont atteintes par le tableau de conversation. Ces événements internes sont utilisés dans le jeu de contextualisation.

Événement interne	Description
atteintEntréeInitiateur(j)	Déclenché si les conditions d'entrée de l'initiateur sont atteintes
atteintEntréePartenaire(j)	Déclenché si les conditions d'entrée du partenaire sont atteintes
atteintSuccèsInitiateur(j)	Déclenché si les conditions de succès de l'initiateur sont atteintes
atteintSuccèsPartenaire(j)	Déclenché si les conditions de succès du partenaire sont atteintes
atteintÉchecInitiateur(j)	Déclenché si les conditions d'échec de l'initiateur sont atteintes
atteintÉchecPartenaire(j)	Déclenché si les conditions d'échec du partenaire sont atteintes
atteintEntrée(j)	Déclenché si la conjonction des conditions d'entrée de l'initiateur et du partenaire est atteinte.
atteintÉchec(j)	Déclenché si la disjonction des conditions d'échec de l'initiateur et du partenaire est atteinte.
atteintSuccès(j)	Déclenché si la disjonction des conditions de succès de l'initiateur et du partenaire est atteinte.
atteintSortie(j)	Déclenché si la disjonction des conditions de succès et d'échec de l'initiateur et du partenaire est atteinte.

Tableau B.1: Événements internes relatifs aux jeux de dialogue. j : JeuDeDialogue.

B.2 Jeu de communication de contextualisation

Le tableau B.2 présente le jeu de communication de contextualisation. Celui-ci permet d'établir un jeu de dialogue ou des combinaisons de jeux de dialogue.

B.3 Jeu de dialogue de question à choix multiples

Le tableau B.3 présente le jeu de dialogue de question à choix multiples. Le but est une question q telle que q : ALTQ. Les *conditions d'entrée* précisent que le partenaire ne doit pas déjà être engagé sur une proposition qui résout la question, ou sur le fait qu'il ne connaît pas

α	β
prop.entrée(x, j)	C_{ctx}(y, acc.entrée(y, j)|ref.entrée(y, j)|prop.entrée(y, j' ; j)|prop.entrée(y, j'$\rightsquigarrow j$) ,**Crt**) C($\{x,y\}$,j,**Sugg**)
prop.entrée(x, j' ; j)	C_{ctx}(y, acc.entrée(y, j' ; j)|ref.entrée(y, j' ; j) ,**Crt**) C($\{x,y\}$,j',**Sugg**)
prop.entrée(x, j'$\rightsquigarrow j$)	C_{ctx}(y, acc.entrée(y, j'$\rightsquigarrow j$)|ref.entrée(y, j'$\rightsquigarrow j$) ,**Crt**) C($\{x,y\}$,j',**Sugg**)
prop.entrée(x, j'$<j$)	C_{ctx}(y, acc.entrée(y, j'$<j$)|ref.entrée(y, j'$<j$) ,**Crt**) C($\{x,y\}$,j',**Sugg**) C_{ctx}(x, prop.entrée(x, j"$<j$),**Crt**)
ref.entrée(x, j)	C($\{x,y\}$,j,**Ina**)
ref.entrée(x, j' ; j)	C($\{x,y\}$,j,**Ina**) C($\{x,y\}$,j',**Ina**)
ref.entrée(x, j'$\rightsquigarrow j$)	C($\{x,y\}$,j,**Ina**) C($\{x,y\}$,j',**Ina**)
ref.entrée(x, j'$<j$)	C($\{x,y\}$,j',**Ina**)
acc.entrée(x, j)	C($\{x,y\}$,j,**Crt**) C_{ctx}(y,atteintSuccèsInitiateur(j)|atteintÉchecInitiateur(j) $\Rightarrow$ C_{ctx}(y,prop.sortie(y, j),**Crt**),**Crt**) C_{ctx}(x,atteintSuccèsPartenaire(j)|atteintÉchecPartenaire(j) $\Rightarrow$ C_{ctx}(x,prop.sortie(x, j),**Crt**),**Crt**) C_{ctx}(y,poursuit(y, j),**Crt**) C_{ctx}(x,prop.entrée(x, j'$<j$) ,**Crt**) C_{ctx}(y,prop.entrée(y, j'$<j$),**Crt**)
acc.entrée(x, j' ; j)	Idem que acc.entrée(x, j') C_{ctx}(y,atteintSuccès(j') $\Rightarrow$ C_{ctx}(y,acc.entrée(y, j),**Crt**),**Crt**)
acc.entrée(x, j'$\rightsquigarrow j$)	Idem que acc.entrée(x, j') C_{ctx}(y,atteintEntrée(j) $\Rightarrow$ C_{ctx}(y,acc.entrée(y, j),**Crt**),**Crt**)
acc.entrée(x, j'$<j$)	Idem que acc.entrée(x, j') prio(j',j,T_i)
poursuit(x, j)	C_{ctx}(y,poursuit(y, j)|prop.sortie(y, j),**Crt**)
prop.sortie(x, j)	C_{ctx}(y,acc.sortie(y, j)|ref.sortie(y, j),**Crt**)
acc.sortie(x, j)	fermerJeu(x,j,T_i) + suppression des engagements contextualisés par le jeu de contextualisation en rapport avec j
ref.sortie(x, j)	/

TABLEAU B.2: Jeu de communication de contextualisation. j : JEUDEDIALOGUE, j' : JEUDEDIALOGUE.

de réponse à la question. Les *conditions de succès* sont atteintes quand le partenaire est engagé sur une proposition qui résout la question. Les conditions d'échec indiquent que le jeu est un échec lorsque le partenaire s'engage sur le fait qu'il ne peut pas trouver la réponse. Les règles précisent que l'initiateur s'engage à jouer un acte *ChoiceQuestion*. Une fois cet acte joué, le partenaire est engagé à jouer soit un acte *Answer*, soit un acte *ExecNegativeAutoFB*. Les règles déclarent que le partenaire peut donner autant de réponse *à propos* qu'il peut, et seulement une réponse résolvante. Pour finir, les effets stipulent que, *dans le contexte de ce jeu*, jouer un acte *ExecNegativeAutoFB* engage le partenaire sur la proposition **fail** (q).

j=**interrogationAChoixMultiple**(q)		
	Initiateur (x)	Partenaire (y)
Entrée		C(y,p,**Ina**) avec **resolves** (p, q) et C(y,**fail** (q),**Ina**)
Succès	C(y,p,**Crt**) avec **resolves** (p, q)	C(y,p,**Crt**) avec **resolves** (p, q)
Échec	C(y,**fail** (q),**Crt**)	C(y,**fail** (q),**Crt**)
Règles	choiceQuestion(x, q)	choiceQuestion(x, q) $\Rightarrow$ $\mathrm{C_j}$(y,answer(y, p)| execNegativeAutoFB(y, q),**Crt**) avec **rel** (p, q) answer(y, s) $\stackrel{*}{\Rightarrow}$ $\mathrm{C_j}$(y,answer(y, p)| execNegativeAutoFB(y, q),**Crt**) avec **strictRel** (s, q), **rel** (p, q)
Effets		execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)

Tableau B.3: Jeu de dialogue d'interrogation à choix multiples. q : ALTQ

Annexe C

Annexes de la mise en œuvre

C.1 Exemples de code

Cette section présente deux jeux définis à l'aide des primitives fournies dans Dogma : le jeu de dialogue de requête et un extrait du jeu de communication d'évaluation. Le langage Scala permet la définition de nouveaux *opérateurs* qui se révèlent être des méthodes « classiques » d'une classe. Cette caractéristique du langage permet de rendre l'implémentation des règles du jeu proche des spécifications.

Le listing C.1 présente la classe `RequestGame` qui définit le jeu de dialogue de requête précédemment présenté. Les cinq premières lignes du listing définissent la classe comme étant composée de trois attributs : l'initiateur (`initiator`), le partenaire (`partner`) et le but du jeu (`goal`). Cette classe étend la classe `DialogueGame` de la librairie Dogma qui requiert la définition de cinq méthodes correspondant aux cinqs champs d'un jeu de dialogue (conditions d'entrée, de succès, d'échec, règles et effets).

La méthode `getEntryConditionsFor` permet de déterminer les *conditions d'entrée* pour un interlocuteur donné en paramètre (`speaker`). Les conditions sont de type `GameCondition` : il s'agit d'un type qui réfère à une fonction prenant un *commitment store* en entrée et retournant un booléen (vrai si les conditions sont réunies, faux sinon). Cette méthode déclare que les conditions d'entrée sont toujours réunies pour l'initiateur (ligne 8), et que les conditions sont réunies pour le partenaire s'il n'est pas déjà engagé sur l'action proposée (lignes 10 et 11). Ces dernières conditions sont vérifiées via la condition `InactiveActionCommitement` qui permet de contrôler la présence d'un engagement en action extra-dialogique en état **Ina** dont le débiteur est `partner`, le créditeur est `initiator` et dont le contenu correspond au troisième paramètre.

Les méthodes `getSuccessExitConditionsFor` (ligne 15) et `getFailureExitConditionsFor` (ligne 19) permettent de spécifier, respectivement, les *conditions de succès* et les *conditions de sortie* pour un interlocuteur donné (`speaker`). Elles font usage de la condition `ExistActionCommitment` qui permet de tester la présence d'un engagement dans le *commitment store*. Cette vérification est réalisée sans prendre en compte le paramètre temporel de l'engagement. Lorsque l'état de l'engagement n'est pas explicité (e.g., ligne 16), il est supposé qu'il s'agit de l'état **Crt**. La méthode `getSuccessExitConditionsFor` permet de vérifier la présence d'un engagement en action extra-dialogique sur le but du jeu. La méthode `getFailureExitConditionsFor` permet de vérifier la présence d'un engagement en action extra-dialogique sur le but du jeu en état **Fal**.

La méthode `getRulesFor` (ligne 24) permet de déterminer les règles du jeu de dialogue pour un interlocuteur donné (`speaker`) et pour un contexte donné (`instance`) correspondant au jeu de dialogue. Cette méthode retourne un ensemble d'engagements dialogiques (`Set[AnyGameCommitment]`). Elle spécifie que l'initiateur du jeu contracte un engagement sur l'événement dialogique *Request* dont le contenu est le but du jeu (lignes 28-29), et que le partenaire contracte un engagement sous la forme d'une *règle de production* (lignes 33-36). Cette

règle de production est sous une forme conditionnelle (via l'opérateur `==>`, ligne 34) et d'une alternative (via l'opérateur[1] `*|`, ligne 35). Elle stipule que le partenaire (`partner`) est engagé à jouer un *AcceptRequest* ou un *DeclineRequest* dès lors que l'initiateur joue un acte *Request*. Cette règle de production implique l'action de création d'un engagement définie dans DOGMA (`Create`, ligne 34).

La méthode `getPropositionalEffectsFor` (ligne 40) fonctionne de la même manière que la méthode `getRulesFor`. Elle permet de définir les effets des actes *AcceptRequest* et *DeclineRequest* produits par le partenaire (lignes 46-49). Les règles de production font intervenir les opérations de création (`Create`, ligne 47) et d'échec (`Failure`, ligne 49) fournies par DOGMA.

```
class RequestGame private (
  initiator: Interlocutor,
  partner: Interlocutor,
  goal: ActionProposition) extends DialogueGame(...) {
  // Entry and exit conditions
  def getEntryConditionsFor(speaker: Interlocutor): GameCondition = {
    if (isInitiator(speaker)) {
      GameCondition.success
    } else {
      InactiveActionCommitment(partner, initiator, goal.action) &&
        InactiveActionCommitment(partner, initiator, goal.action.neg)
    }
  }

  def getSuccessExitConditionsFor(speaker: Interlocutor): GameCondition = {
    ExistActionCommitment(C(partner, initiator, goal.action))
  }

  def getFailureExitConditionsFor(speaker: Interlocutor): GameCondition = {
    ExistActionCommitment(C(partner, initiator, goal.action, Fal))
  }

  // Rules
  def getRulesFor(speaker: Interlocutor)(instance: DialogueGameInstance): Set[
      AnyGameCommitment] = {
    if (isInitiator(speaker)) {
      Set(
   // Request
       C(instance, initiator, partner,
         Request(initiator, goal)))
    } else {
      Set(
   // Production rule
       C(instance, partner, initiator,
         Request(initiator, goal) ==> Create(partner,
           C(instance, partner, initiator, AcceptRequest(partner, goal) *|
               DeclineRequest(partner, goal)))))
    }
  }
```

1. L'opérateur d'alternative commence par un « * » pour des raisons de priorité inhérentes au langage.

```
  // Effects
  def getPropositionalEffectsFor(speaker: Interlocutor)(instance: DialogueGameInstance)
      : Set[AnyGameCommitment] = {
    if (isInitiator(speaker)) {
      Set()
    } else {
      Set(
        // AcceptRequest and DeclineRequest
        C(instance, partner, initiator,
          AcceptRequest(partner, goal) ==> Create(partner, C(partner, initiator, goal.
              action))),
        C(instance, partner, initiator,
          DeclineRequest(partner, goal) ==> Failure(partner, C(partner, initiator,
              goal.action))))
    }
  }

}
```

Listing C.1: Extrait du code du jeu de dialogue de requête dans DOGMA

Le listing C.2 présente un extrait du jeu de communication d'évaluation. Les quatre premières lignes définissent le jeu de communication comme impliquant plusieurs interlocuteurs (`speakers`). Ce jeu hérite de la classe `CommunicationGame` de la librairie DOGMA qui nécessite la définition de la méthode `getRulesFor`. Cette méthode permet d'obtenir les règles du jeu pour l'interlocuteur `speaker`. Elle retourne un ensemble d'engagements dialogiques (`Set[AnyGameCommitment]`).

Les règles du jeu sont spécifiées en tant que *règles de production conditionnelles persistantes* (via l'opérateur `=*=>`, lignes 9 et 14). La première règle spécifie les *effets directs* des actes informatifs *Inform*, *Answer* et *Disagreement* (lignes 7-10). Elle fait intervenir une alternative (via l'opérateur `*|`) qui définit que l'intervention d'un événement dialogique *Inform*, *Answer* ou *Agreement* conduit à engager le locuteur sur son contenu sémantique (via l'opération de création `Create`, ligne 10). Le contenu sémantique est spécifié par une variable `P` qui s'unifie avec celui de l'événement intervenu (pourvu que sa fonction et son interlocuteur soient identiques à celles de la description). La seconde règle spécifie les *effets directs* de l'acte *Disagreement* (lignes 12-15). Elle conduit à engager le locuteur du *Disagreement* sur la négation de son contenu sémantique (`P.neg`, ligne 15). Les autres règles du jeu d'évaluation sont spécifiées sur le même modèle.

```
class EvaluationGame private (speakers: Interlocutor*)
  extends CommunicationGame
  with GameVariables {

  def getRulesFor(speaker: Interlocutor): Set[AnyGameCommitment] = Set(
    // InformativeAct(_, p) -> C(_,p)
    C(this, speaker, otherSpeaker(speaker),
      Inform(speaker, P) *| Answer(speaker, P) *| Agreement(speaker, P)
        =*=>
        Create(speaker, C(speaker, otherSpeaker(speaker), P))),
    // Disagreement(_, p) -> C(_, ~p)
    C(this, speaker, otherSpeaker(speaker),
      Disagreement(speaker, P)
```

```
        =*=>
15      Create(speaker, C(speaker, otherSpeaker(speaker), P.neg))))
    ...
}
```

Listing C.2: Extrait du code du jeu de communication d'évaluation

C.2 Traces de fonctionnement du système

Ces annexes présentent trois exemples supplémentaires de traces générées par DOGMA. L'annexe C.2.1 présente le cas du refus d'entrée dans un jeu de dialogue suite à un acte de proposition d'entrée. L'annexe C.2.2 présente un jeu de question ouverte dont l'exécution conduit à l'échec du jeu. Enfin, l'annexe C.2.3 présente un exemple de correction.

C.2.1 Refus d'entrée dans le jeu

Cet exemple présente un jeu de dialogue de requête sur l'action ajouterMotCle(**grippe**) qui forme le but du jeu. Il débute par un acte de contextualisation de proposition d'entrée dans ce jeu (*« Je peux vous demander de réaliser une action ? »*). Cet acte établit l'engagement conjoint sur le jeu de dialogue en état **Sugg**.

Tours: (prop.entrée(x, g4))

	C_x	C_y
C_p	∅	∅
C_a(**Crt**)	∅	∅
C_a(**Fal**)	∅	∅
C_{dA}	∅	∅

C_{xy}^{Jeu} :

Sugg.	:	Request(ajouterMotCle(**grippe**))=g4
Ouv.	:	
Ferm.	:	
Saillants	:	∅

Actes attendus :
ctx, ev:
acc.entrée(y, g4) | ref.entrée(y, g4) | prop.entrée(y, Ys⇝g4) | prop.entrée(y, Ys ; g4)

L'exploitation du tableau de conversation spécifie que les seuls actes attendus sont de la part de l'interlocuteur y. Il s'agit pour lui d'accepter ou de refuser l'entrée dans le jeu, ou bien de proposer une combinaison de jeu. Le terme `Ys` représente une variable devant être remplacée par

une proposition de jeu valide. L'interlocuteur y refuse l'entrée dans le jeu.

Tours: (ref.entrée(y, g4), prop.entrée(x, g4))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\emptyset$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	
Ferm.	:	
Saillants	:	$\emptyset$

Le jeu suggéré est alors retiré. Le tableau de conversation retourne dans sa situation initiale, à l'exception de l'historique du dialogue qui garde la trace de la tentative d'ouverture du jeu de dialogue.

C.2.2 Échec d'un jeu de question/réponse

Cette trace présente l'exécution d'un jeu de dialogue de question ouverte sur la question *« Quel est un synonyme d'eczéma ? »* ($?X$.synonyme(**eczema**, X)) initié par l'interlocuteur x. La proposition de jeu est acceptée par l'interlocuteur y. Le jeu de question ouverte est alors établi entre les interlocuteurs, conduisant au tableau de conversation ci-après.

Tours: (acc.entrée(y, g5), prop.entrée(x, g5))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\lfloor$setQuestion(x, q)$\rfloor_{:g5}$	$\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g5}$, $\lfloor$setQuestion(x, q) $\Rightarrow$ C_{g5}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g5}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g5}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g5}$

C_{xy}^{Jeu}:

Sugg.	:	
Ouv.	:	OpenInterrogation(?X.synonyme(**eczema**, X))=g5
Ferm.	:	
Saillants	:	g5

q=?X.synonyme(**eczema**, X)

Actes attendus :
g5:
setQuestion(x, q)
ctx, ev:
prop.entrée(y, Ys<g5), prop.entrée(x, Ys<g5)

Les interlocuteurs sont chacun engagés sur les règles du jeu d'interrogation ouverte. L'initiateur est engagé à poser la question tandis que le partenaire est conditionnellement engagé à y répondre. L'acte attendu dans le cadre de ce jeu est la question de l'initiateur. Celui-ci

produit cet acte.

Tours: (setQuestion(x, q), acc.entrée(y, g5), prop.entrée(x, g5))

	C_x	C_y
C_p	$\emptyset$	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)$\rfloor_{:g5}$, $\lfloor$execNegativeAutoFB(y, q) $\Rightarrow$ C(y,**fail** (q),**Crt**)$\rfloor_{:g5}$, $\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g5}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g5}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	OpenInterrogation(?X.synonyme(**eczema**, X))=g5
Ferm.	:	
Saillants	:	g5

q=?X.synonyme(**eczema**, X)

Actes attendus :

g5:
answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q)

ctx, ev:
prop.entrée(y, Ys<g5), prop.entrée(x, Ys<g5)

L'interrogation produite par x conduit à engager le partenaire à répondre à la question. Ce dernier peut répondre par : (i) une réponse *résolvante* (permettant l'atteinte du succès du jeu), (ii) une réponse *pertinente mais non résolvante* (engageant le locuteur à produire une

autre réponse), ou par (iii) un *« Je ne sais pas »* (*ExecNegativeAutoFB*). L'interlocuteur y choisit cette dernière option.

Tours: (execNegativeAutoFB(y, q), setQuestion(x, q), acc.entrée(y, g5), prop.entrée(x, g5))

	C_x	C_y
C_p	$\emptyset$	**fail** (q)
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\lfloor$answer(y, P) avec **strictRel** (P, q) $\stackrel{*}{\Rightarrow}$ C_{g5}(y,answer(y, P2) avec **rel** (P2, q) | execNegativeAutoFB(y, q),**Crt**)$\rfloor_{:g5}$

C^{Jeu}_{xy}:

Sugg.	:	
Ouv.	:	OpenInterrogation(?X.synonyme(**eczema**, X))=g5
Ferm.	:	
Saillants	:	g5

q=?X.synonyme(**eczema**, X)

Actes attendus :
ctx, ev:
prop.sortie(y, g5), prop.sortie(x, g5)
prop.entrée(y, Ys<g5), prop.entrée(x, Ys<g5)

L'acte *ExecNegativeAutoFB* produit par l'interlocuteur y a pour conséquence de l'engager sur la proposition **fail** (q) (par satisfaction des *effets* du jeu de question ouverte). L'état d'échec du jeu est atteint pour les deux interlocuteurs. Ils peuvent alors proposer la sortie du jeu de dialogue.

C.2.3 Jeu d'évaluation : exemple de correction

Ce dernier exemple présente un cas de correction d'une proposition. Il se base sur la supposition qu'un acte *Inform* peut être produit à n'importe quel moment par un interlocuteur. Autrement dit, un interlocuteur peut a priori s'engager sur n'importe quelle proposition à n'importe quel moment. Cette prérogative générale permet de souligner que le but premier du dialogue est l'échange d'information. Elle peut sembler trop forte (e.g., elle permet des répétitions à l'infini). Néanmoins, nous ferons l'hypothèse qu'elle tient pour garder notre exemple simple.

Le locuteur x affirme que son interlocuteur y est un humain (*« Vous êtes un humain. »*).

Tours: (inform(x, humain))

	C_x	C_y
C_p	humain	$\emptyset$
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\emptyset$

C_{xy}^{Jeu} : Sugg. : / Ouv. : / Ferm. : / Saillants : $\emptyset$

Actes attendus :
ctx, ev:
agreement(y, humain) | disagreement(y, humain) | correction(y, humain, P2)
avec **correct** (humain, P2)

Cette affirmation conduit à engager x sur la proposition « humain ». L'interlocuteur y peut alors affirmer son accord (via l'acte *Agreement*), ou son désaccord (via les actes *Disagreement* et *Correction*). Ce dernier corrige l'interlocuteur y en lui signifiant qu'il n'est pas un interlocuteur humain mais bel et bien une machine.

Tours: (correction(y, humain, machine), inform(x, humain))

	C_x	C_y
C_p	humain	¬humain, machine
C_a(**Crt**)	$\emptyset$	$\emptyset$
C_a(**Fal**)	$\emptyset$	$\emptyset$
C_{dA}	$\emptyset$	$\emptyset$

C_{xy}^{Jeu} : Sugg. : / Ouv. : / Ferm. : / Saillants : $\emptyset$

Actes attendus :
ctx, ev:
acceptCorrection(x, humain, machine) | declineCorrection(x, humain, machine)

Cet acte de correction engage l'interlocuteur y sur le fait qu'il est une machine et qu'il n'est pas humain (une correction est également un désaccord). L'interlocuteur x peut accepter cette correction (via l'acte *AcceptCorrection*) ou la refuser (via l'acte *DeclineCorrection*).

L'interlocuteur x accepte cette correction.

Tours: (acceptCorrection(x, humain, machine), correction(y, humain, machine), inform(x, humain))

	C_x	C_y
C_p	machine	¬humain, machine
C_a(**Crt**)	∅	∅
C_a(**Fal**)	∅	∅
C_{dA}	∅	∅

C_{xy}^{Jeu}:

Sugg.	:	
Ouv.	:	
Ferm.	:	
Saillants	:	∅

L'acceptation de la correction engage l'interlocuteur x sur le fait que son partenaire est une machine. En outre, son engagement sur le fait que son partenaire est un humain est supprimé.

Liste des dialogues

1.1 Un exemple de projet conjoint d'échange d'information (entretien VD06) 12
1.2 Exemple de pré-séquence de clôture 13
1.3 Exemple de séquences emboîtées 13
1.4 Exemple de séquences enchaînées 14
1.5 Exemple de dialogue entremêlant avancement de la tâche et gestion de l'interaction (entretien VD06) 16
1.6 Exemple de dialogue annoté avec des actes multi-niveaux 23
2.1 Exemple de réponses à propos dans l'approche de Ginzburg 47
2.2 Exemples de la relativité de la notion de résolution dans l'approche de Ginzburg 47
2.3 Exemple de questions dépendantes dans l'approche de Ginzburg 47
3.1 Jeux de dialogue de demande d'information puis d'aide, *bid* explicite 59
3.2 Jeu de dialogue de demande d'information, phase d'entrée et de sortie implicite . 63
3.3 Exemple de dialogue de séquencement de jeux de recherche d'information. 69
3.4 Emboîtement intentionnellement fonctionnel 75
3.5 Emboîtement interactionnellement fonctionnel 76
4.1 Exemple de dialogue de réparation de requête (VD08) 94
4.2 Exemple de dialogue de réparation de requête (VD06) 95
4.3 Exemple de dialogue de construction de requête initiale avec la tactique de téléportation (VD02) 97
4.4 Exemple de dialogue de construction de requête initiale avec la tactique d'orienteering (AL05) 98
5.1 Exemple de segment fonctionnel sur plusieurs tours de parole (entretien VD08) . 108
5.2 Exemple d'un motif de type question oui/non avec seconde partie de paire préférée (entretien AL03) 132
5.3 Exemple d'un motif de type question oui/non avec seconde partie de paire non-préférée (entretien VD04) 132
5.4 Exemple d'un motif de type vérification avec seconde partie de paire préférée (entretien AL12) 132
5.5 Exemple d'un motif de type vérification avec seconde partie de paire non-préférée (entretien VD05) 133
5.6 Exemple de motif de type vérification avec altération du projet conjoint (entretien AL11) 133
5.7 Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien VD04) 134
5.8 Exemple d'un motif de type question ouverte avec seconde partie de paire non-préférée (entretien VD06) 134
5.9 Exemple de motif de type question à choix multiples avec seconde partie de paire préférée (entretien AL06) 135

5.10 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien AL07) . . . 136
5.11 Exemple de motif de type suggestion avec seconde partie de paire non-préférée (entretien VD06) . . . 136
5.12 Exemple de motif de type requête avec seconde partie de paire préférée (entretien VD07) . . . 136
5.13 Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD07) . . . 137
5.14 Exemple de motif de type offre avec seconde partie de paire non-préférée (entretien AL07) . . . 137
5.15 Exemple de motif d'accord avec seconde partie de paire préférée (entretien VD02) 138
5.16 Exemple de motif de type correction en trois temps (entretien VD04) . . . 139
5.17 Exemple de longue prise de parole de l'expert (entretien AL10) . . . 140
A.1 Exemple de dialogue annoté (entretien VD06) : une succession de deux suggestions 223
A.2 Exemple de dialogue annoté (entretien VD02) : sélection de qualificatif dans la construction de la première requête. . . . 225
A.3 Exemple de dialogue annoté (entretien VD05) : demande de verbalisation du besoin d'information . . . 226
A.4 Exemple de dialogue annoté (entretien AL07) : question/réponse suivie de clarifications . . . 227
A.5 Exemple de dialogue annoté (VD08) : segment fonctionnel sur plusieurs tours de parole. . . . 228
A.6 Exemple d'un motif de type question oui/non avec seconde partie de paire préférée (entretien AL03) . . . 250
A.7 Exemple d'un motif de type vérification avec seconde partie de paire préférée (entretien AL3) . . . 250
A.8 Exemple d'un motif de type vérification avec seconde partie de paire préférée (entretien VD2) . . . 251
A.9 Exemple d'un motif de type vérification avec seconde partie de paire non-préférée (entretien VD04) . . . 251
A.10 Exemple d'un motif de type vérification avec seconde partie de paire non-préférée (entretien VD05) . . . 251
A.11 Exemple d'un motif de type vérification positive (entretien AL06) . . . 251
A.12 Exemple d'un motif de type vérification positive (entretien VD05) . . . 251
A.13 Exemple d'un motif de type vérification positive (entretien VD07) . . . 252
A.14 Exemple d'un motif de type vérification negative (entretien VD02) . . . 252
A.15 Exemple d'un motif de type vérification negative (entretien VD04) . . . 252
A.16 Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien AL06) . . . 252
A.17 Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien AL06) . . . 253
A.18 Exemple de motif de type question ouverte avec seconde partie de paire préférée (entretien AL10) . . . 253
A.19 Exemple de motif de type question à choix multiples avec seconde partie de paire préférée (entretien VD07) . . . 253

A.20 Exemple de motif de type question à choix multiples avec seconde partie de paire préférée (entretien VD07) . . . 254
A.21 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD05) . . . 254
A.22 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien AL07) . . . 254
A.23 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD02) . . . 254
A.24 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD05) . . . 255
A.25 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD05) . . . 255
A.26 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD06) . . . 255
A.27 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD07) . . . 255
A.28 Exemple de motif de type suggestion avec seconde partie de paire préférée (entretien VD07) . . . 255
A.29 Exemple de motif de type requête avec seconde partie de paire préférée (entretien VD05) . . . 256
A.30 Exemple de motif de type offre avec seconde partie de paire non-préférée (entretien AL10) . . . 256
A.31 Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD06) . . . 256
A.32 Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD06) . . . 257
A.33 Exemple de motif de type offre avec seconde partie de paire préférée (entretien VD06) . . . 257
A.34 Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL04) 257
A.35 Exemple de motif de type accord avec seconde partie de paire préférée (entretien AL07) . . . 257
A.36 Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL10) 258
A.37 Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL10) 258
A.38 Exemple de motif d'accord avec seconde partie de paire préférée (entretien AL12) 258
A.39 Exemple de motif d'accord avec seconde partie de paire préférée (entretien VD05) 258
A.40 Exemple de motif d'accord avec seconde partie de paire préférée (entretien VD05) 258
A.41 Exemple de motif de type accord avec seconde partie de paire préférée (entretien VD06) . . . 258
A.42 Exemple de motif de type correction en deux temps (entretien VD05) . . . 259

Listings

A.1 Schéma d'annotation pour un acte de dialogue dans le processus d'annotation. . . . 217
C.1 Extrait du code du jeu de dialogue de requête dans DOGMA 266
C.2 Extrait du code du jeu de communication d'évaluation 267

Liste des algorithmes

6.1 Algorithme de mise à jour d'un engagement dialogique en action suite à l'occurrence d'un événement. 155
6.2 Algorithme de mise à jour du tableau de conversation suite à l'occurrence d'un événement externe. 156
7.1 Structure de l'algorithme de gestion de dialogue basé sur les jeux de dialogue. . . 186

Bibliographie

[Ales 2012] Z. Ales, G. Dubuisson Duplessis, O. Serban et A. Pauchet. *A Methodology to Design Human-Like Embodied Conversational Agents.* In AAMAS 2012 Workshop, Human-Agent Interaction Design and Models, pages 34–49, 2012.

[Allen 1980] J. Allen et C. R. Perrault. *Analyzing intention in utterances.* Artificial Intelligence, vol. 15, pages 143–178, 1980.

[Allen 1995] J. Allen, K. Schubert, G. Ferguson, P. Heeman, C.H. Hwang, T. Kato, M. Light, N. Martin, B. Miller et M. Poesio. *The TRAINS project : A case study in building a conversational planning agent.* Journal of Experimental & Theoretical Artificial Intelligence, vol. 7, no. 1, page 7–48, 1995.

[Allen 1997] J. Allen et M. Core. *DAMSL : Dialog act markup in several layers.* Draft of manual, 1997.

[Allen 1999] J. Allen. *Mixed-initiative interaction.* Intelligent Systems and their Applications, IEEE, vol. 14, no. 5, pages 14–16, 1999.

[Allen 2000] J. Allen, G. Ferguson, B.W. Miller, E.K. Ringger et T. Sikorski-Zollo. *Dialogue systems : From theory to practice in TRAINS-96.* In Handbook of Natural Language Processing, pages 347–376, Edimbourg, Ecosse, 2000.

[Allwood 1992] J. Allwood. *On Dialogue Cohesion.* Gothenburg Papers in Theoretical Linguistics 65, 1992.

[Allwood 1994] J. Allwood. *Obligations and options in dialogue.* Think Quarterly, vol. 3, page 9–18, 1994.

[Allwood 1995] J. Allwood. *An activity based approach to pragmatics.* Rapport technique, Gothenburg Papers In Theoretical Linguistics, University of Göteborg, 1995.

[Allwood 2001] J. Allwood. *Capturing differences between social activities in spoken language.* Pragmatics and Beyond New Series, page 301–320, 2001.

[Asher 2003] N. Asher et A. Lascarides. Logics of conversation. Cambridge University Press, 2003.

[Aust 1995] H. Aust, M. Oerder, F. Seide et V. Steinbiss. *The Philips automatic train timetable information system.* Speech Communication, vol. 17, no. 3-4, page 249–262, 1995.

[Austin 1962] J.L. Austin. How to do things with words. Oxford University Press, Oxford, 1962.

[Barwise 1981] J. Barwise et J. Perry. *Situations and attitudes.* The Journal of Philosophy, vol. 78, no. 11, page 668–691, 1981.

[Bates 1979] M. J. Bates. *Information search tactics.* Journal of the American Society for information Science, vol. 30, no. 4, page 205–214, 1979.

[Bates 1989] M. J. Bates. *The design of browsing and berrypicking techniques for the online search interface.* Online Information Review, vol. 13, no. 5, page 407–424, 1989.

[Bates 1990] M. J. Bates. *Where should the person stop and the information search interface start ?* Information Processing and Management, vol. 26, no. 5, page 575–591, 1990.

[Bickmore 2006] T. Bickmore et T. Giorgino. *Health dialog systems for patients and consumers.* Journal of Biomedical Informatics, vol. 39, no. 5, page 556–571, 2006.

[Bickmore 2009] T. Bickmore, D. Schulman et G. Shaw. *DTask and LiteBody : Open Source, Standards-Based Tools for Building Web-Deployed Embodied Conversational Agents.* In Intelligent Virtual Agents, page 425–431, 2009.

[Bilange 1991a] E. Bilange. *Modélisation du dialogue oral finalisé personne-machine par une approche structurelle.* PhD thesis, Université de Rennes I, 1991.

[Bilange 1991b] E. Bilange. *A task independent oral dialogue model.* In Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics, page 83–88, 1991.

[Bohus 2009a] D. Bohus et E. Horvitz. *Open-world dialog : Challenges, directions, and prototype.* In Proceedings of IJCAI'2009 Workshop on Knowledge and Reasoning in Practical Dialogue Systems, 2009.

[Bohus 2009b] D. Bohus et A. I. Rudnicky. *The RavenClaw dialog management framework : Architecture and systems.* Computer Speech & Language, vol. 23, no. 3, page 332–361, 2009.

[Bos 2003] J. Bos, E. Klein, O. Lemon et T. Oka. *DIPPER : Description and Formalisation of an Information-State Update Dialogue System Architecture.* In 4th SIGdial Workshop on Discourse and Dialogue, page 115–124, Sapporo, Japan, 2003.

[Bratman 1987] M.E. Bratman. Intention, plans, and practical reason. Harvard University Press, Cambridge, 1987.

[Broder 2002] A. Broder. *A taxonomy of web search.* In ACM Sigir forum, volume 36, page 3–10, 2002.

[Bunt 1996] H. Bunt. *Dynamic interpretation and dialogue theory.* Dialogue and instruction, 1996.

[Bunt 2000] H. Bunt. *Dialogue pragmatics and context specification.* Abduction, Belief and Context in Dialogue, vol. 1, page 81–150, 2000.

[Bunt 2007] H. Bunt, R. Morante et S. Keizer. *An empirically based computational model of grounding in dialogue.* In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, page 283–290, 2007.

[Bunt 2009] H. Bunt. *The DIT++ taxonomy for functional dialogue markup.* In AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts, pages 13–24, 2009.

[Bunt 2010] H. Bunt. *DIT++ Taxonomy of Dialogue Acts (Release 5, May 2010)*, 2010. http ://-dit.uvt.nl/.

[Bunt 2011a] H. Bunt. *Interpretation and Generation of Dialogue with Multidimensional Context Models.* In A. Esposito, A. Esposito, R. Martone, V. Müller et G. Scarpetta, editeurs, Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces. Theoretical and Practical Issues, volume 6456 of *Lecture Notes in Computer Science*, pages 214–242. Springer Berlin / Heidelberg, 2011.

[Bunt 2011b] H. Bunt. *Multifunctionality in dialogue.* Computer Speech & Language, vol. 25, no. 2, page 222–245, 2011.

[Bunt 2011c] H. Bunt. *The semantics of dialogue acts.* In Proceedings of the 9th International Conference on Computational Semantics (IWCS 2011), page 1–14, Oxford, 2011.

[Bylander 1991] T. Bylander. *Complexity results for planning.* In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, volume 1, page 274–279, 1991.

[Carberry 1990] S. Carberry. Plan recognition in natural language dialogue. MIT Press, 1990.

[Carletta 1996] J. Carletta. *Assessing agreement on classification tasks : the kappa statistic.* Computational linguistics, vol. 22, no. 2, page 249–254, 1996.

[Carletta 1997] J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C. Kowtko et A. H. Anderson. *The reliability of a dialogue structure coding scheme.* Computational linguistics, vol. 23, no. 1, page 13–31, 1997.

[Cassell 2000] J. Cassell, J. Sullivan, S. Prevost et E. F. Churchill. Embodied conversational agents. MIT press, 2000.

[Castelfranchi 1995] Cristiano Castelfranchi. *Commitments : From Individual Intentions to Groups and Organizations.* In ICMAS, volume 95, page 41–48, 1995.

[Castelfranchi 2000] C. Castelfranchi, F. Dignum, C. M. Jonker et J. Treur. *Deliberative normative agents : Principles and architecture.* In Intelligent Agents VI. Agent Theories, Architectures, and Languages, page 364–378. Springer, 2000.

[Chaib-Draa 2006] B. Chaib-Draa, M. A. Labrie, M. Bergeron et P. Pasquier. *Diagal : An agent communication language based on dialogue games and sustained by social commitments.* Autonomous Agents and Multi-Agent Systems, vol. 13, no. 1, page 61–95, 2006.

[Chaignaud 1996] N. Chaignaud. *Étude cognitive et informatique de la résolution d'un problème : analyse, modélisation et implantation.* PhD thesis, Université Paris Nord, 1996.

[Chaignaud 2000] N. Chaignaud, I. Cheikhrouhou et A. Nguyen-Xuan. *Generalisation of a cognitive model for problem solving with incomplete information.* ICCM'00, page 58–69, 2000.

[Clark 1989] H.H. Clark et E.F. Schaefer. *Contributing to discourse.* Cognitive science, vol. 13, no. 2, page 259–294, 1989.

[Clark 1996] H.H. Clark. Using language, volume 4. Cambridge University Press, 1996.

[Cohen 1960] J. Cohen. *A coefficient of agreement for nominal scales.* Educational and psychological measurement, vol. 20, no. 1, page 37–46, 1960.

[Cohen 1979] P. Cohen et C. R. Perrault. *Elements of a plan-based theory of speech acts.* Cognitive Science : A Multidisciplinary Journal, vol. 3, no. 3, page 177–212, 1979.

[Cohen 1990a] P. Cohen et H. Levesque. *Rational interaction as the basis for communication.* In P. Cohen, J. Morgan et M. Pollack, editeurs, Intentions in communication, pages 221–255. MIT Press, Cambridge, 1990.

[Cohen 1990b] P. R. Cohen et H. J. Levesque. *Intention is choice with commitment.* Artificial intelligence, vol. 42, no. 2, page 213–261, 1990.

[Cohen 1997] P. Cohen. *Dialogue modeling.* In Survey of the state of the art in Human Language Technology, page 204–210, 1997.

[Cooper 2010] R. Cooper et S. Larsson. *Accommodation and reaccommodation in dialogue*. Presuppositions and Discourse : Essays Offered to Hans Kamp, pages 101–124, 2010.

[Cunningham 2011] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li et W. Peters. Text processing with GATE (version 6). 2011.

[Darmoni 2000] S. Darmoni, JP. Leroy, F. Baudic, M. Douyere, J. Piot et B. Thirion. *CISMeF : a structured health resource guide*. Methods of Information in Medicine, vol. 39, pages 30–35, 2000.

[Dastani 2001] M. Dastani, J. Hulstijn et L. Van der Torre. *Negotiation protocols and dialogue games*. In Proceedings of the 5th International Conference on Autonomous Agents, page 180–181, 2001.

[Dubuisson Duplessis 2013] Guillaume Dubuisson Duplessis. *Analyse haut niveau du corpus Cogni-CISMeF*. Rapport technique, Juillet 2013.

[Ekman 1999] P. Ekman. *Basic emotions*. Handbook of cognition and emotion, vol. 4, page 5–60, 1999.

[Farges 2013] C. Farges. Étude de la dynamique conversationnelle homme-homme dans un cadre de négociation. Master's thesis, INSA de Rouen, 2013.

[Ferguson 1996] G. Ferguson, J. Allen et B. Miller. *TRAINS-95 : Towards a Mixed-Initiative Planning Assistant*. In Proceedings of the Third Conference on Artificial Intelligence Planning Systems (AIPS-96), pages 70–77, Édimbourg, Écosse, 1996.

[Ferguson 1998] G. Ferguson et J. Allen. *TRIPS : An Intelligent Integrated Problem-Solving Assistant*. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), pages 567–573, 1998.

[Ferguson 2007] G. Ferguson et J. Allen. *Mixed-Initiative Systems for Collaborative Problem Solving*. AI magazine, vol. 28, no. 2, page 23, 2007.

[Fikes 1972] R. E. Fikes et N. J. Nilsson. *STRIPS : A new approach to the application of theorem proving to problem solving*. Artificial intelligence, vol. 2, no. 3, page 189–208, 1972.

[Finin 1994] T. Finin, R. Fritzson, D. McKay et R. McEntire. *KQML as an agent communication language*. In Proceedings of the third international conference on Information and knowledge management, page 456–463, 1994.

[FIPA 2000] FIPA. *Foundation for Intelligent Physical Agents, Communicative Act Library Specification*. http://www.fipa.org, 2000.

[Geach 1962] P. T. Geach. Reference and generality : An examination of some medieval and modern theories, volume 88. Cornell University Press Ithaca, 1962.

[Geertzen 2007] J. Geertzen, V. Petukhova et H. Bunt. *A multidimensional approach to utterance segmentation and dialogue act classification*. In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 140—149, 2007.

[Geertzen 2009a] J. Geertzen. *Dialogue act recognition and prediction*. Ph.D. dissertation, Tilburg University, 2009.

[Geertzen 2009b] J. Geertzen et H. Bunt. *Measuring annotator agreement in a complex hierarchical dialogue act annotation scheme*. In Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, page 126–133, 2009.

[Ginzburg 1994] J. Ginzburg. *An update semantics for dialogue.* In Proceedings of the first International Workshop on Computational Semantics, Tilburg, The Netherlands, 1994. H. Bunt et al.

[Ginzburg 1995a] J. Ginzburg. *Resolving questions, I.* Linguistics and Philosophy, vol. 18, no. 5, page 459–527, 1995.

[Ginzburg 1995b] J. Ginzburg. *Resolving questions, II.* Linguistics and Philosophy, vol. 18, no. 6, page 567–609, 1995.

[Ginzburg 1996] J. Ginzburg. *Interrogatives : Questions, Facts, and Dialogue.* In The Handbook of Contemporary Semantic Theory, volume 5(18), page 359–423. Blackwell Publishers, 1996.

[Ginzburg 2010] Jonathan Ginzburg. *Relevance for dialogue.* In Aspects of semantics and pragmatics of dialogue., page 121–129, 2010.

[Ginzburg 2012] J. Ginzburg. The interactive stance. OUP Oxford, 2012.

[Golovchinsky 2008] G. Golovchinsky, P. Qvarfordt et J. Pickens. *Collaborative information seeking.* Information Seeking Support Systems, 2008.

[Greaves 2000] M. Greaves, H. Holmback et J. Bradshaw. *What is a conversation policy ?* In Issues in agent communication, page 118–131. Springer, 2000.

[Grice 1957] H. P. Grice. *Meaning.* The philosophical review, vol. 66, no. 3, page 377–388, 1957.

[Grice 1975] H. P. Grice. *Logic and conversation.* 1975, page 41–58, 1975.

[Grosz 1986] B. Grosz et C. Sidner. *Attention, intentions, and the structure of discourse.* Computational Linguistics, vol. 12, no. 3, pages 175–204, 1986.

[Grosz 1996] B. J. Grosz et S. Kraus. *Collaborative Plans for Complex Group Action.* Artificial Intelligence, vol. 86, no. 2, pages 269–357, 1996.

[Hamblin 1970] C. L. Hamblin. Fallacies. Methuen, London, 1970.

[Hearst 2009] M. Hearst. Search user interfaces. Cambridge University Press, 2009.

[Hjelm 2005] D. Hjelm, A-C. Forslund, S. Larsson et A. Wallentin. *DJ GoDiS : Multimodal menu-based dialogue in an asychronous ISU system.* In Proceedings of the 9th workshop on the semantics and pragmatics of dialogue., 2005.

[Horvitz 1999a] E. Horvitz. *Principles of mixed-initiative user interfaces.* In Proceedings of the 1999 ACM SIGCHI Conference on Human Factors in Computing Systems, page 159–166, New York, 1999. ACM Press.

[Horvitz 1999b] E. Horvitz. *Uncertainty, action, and interaction : in pursuit of mixed-initiative computing.* Intelligent Systems and their Applications, IEEE, vol. 14, no. 5, pages 17–20, 1999.

[Horvitz 2007] E. Horvitz. *Reflections on Challenges and Promises of Mixed-Initiative Interaction.* AI Magazine, vol. 28, no. 2, page 3, 2007.

[Hulstijn 2000a] J. Hulstijn. *Dialogue games are recipes for joint action.* In Proceedings of the Forth Workshop on the Semantics and Pragmatics of Dialogue (Gotalog'00), 2000.

[Hulstijn 2000b] J. Hulstijn. *Dialogue models for inquiry and transaction.* PhD thesis, University of Twente, 2000.

[Jokinen 2010] K. Jokinen et M. McTear. Spoken dialogue systems, volume 5. Morgan & Claypool, 2010.

[Kamp 1981] H. Kamp. *A Theory of Truth and Semantic Representation*. In J. A. G. Groenendijk, T. M. V. Janssen et M. B. J. Stokhof, editeurs, Formal Methods in the Study of Language, volume 1, page 277–322. Mathematisch Centrum, Amsterdam, 1981.

[Keizer 2006] S. Keizer et H. Bunt. *Multidimensional Dialogue Management*. In Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, page 37–45, Sydney, Australia, Juillet 2006. Association for Computational Linguistics.

[Kowtko 1991] J. C. Kowtko, S. D. Isard et G. M. Doherty. *Conversational games within dialogue*. In Proceedings of the ESPRIT Workshop on Discourse Coherence, 1991.

[Kronlid 2008] F. Kronlid. *Steps towards Multi-Party Dialogue Management*. PhD thesis, Göteborg University, 2008.

[Lambert 1991] L. Lambert et S. Carberry. *A tripartite plan-based model of dialogue*. In Proceedings of the 29th annual meeting on Association for Computational Linguistics, page 47–54, 1991.

[Landragin 2013] F. Landragin. Man-machine dialogue. design and challenges. Wiley-ISTE, London, UK, 2013.

[Larsson 1998] S. Larsson. *Coding schemas for dialogue moves*. Unpublished paper, available at https://sites.google.com/site/larssonstaffan/, 1998.

[Larsson 2000a] S. Larsson, P. Ljunglof, R. Cooper, E. Engdahl et S. Ericsson. *GoDiS-an accommodating dialogue system*. In Proceedings of ANLP/NAACL-2000 Workshop on Conversational systems, page 7–10, 2000.

[Larsson 2000b] S. Larsson et D.R. Traum. *Information state and dialogue management in the TRINDI dialogue move engine toolkit*. Natural language engineering, vol. 6, no. 3&4, page 323–340, 2000.

[Larsson 2002a] S. Larsson. *Issue-Based Dialogue Management*. Ph.D. dissertation, Department of Linguistics, Göteborg University, Göteborg, 2002.

[Larsson 2002b] S. Larsson, A. Berman, L. Grönqvist et F. Kronlid. *TRINDIKIT 3.0 Manual*. Trindi Deliverable D, vol. 6, 2002.

[Larsson 2003] S. Larsson. *Interactive communication management in an issue-based dialogue system*. In Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue (DiaBruck), page 75–82, 2003.

[Larsson 2011] S. Larsson, A. Berman et J. Villing. *Adding a Speech Cursor to a Multimodal Dialogue System.* In INTERSPEECH, page 3319–3320, 2011.

[Levin 1977] J. A. Levin et J. A. Moore. *Dialogue games : Metacommunication structures for natural language interaction*. Cognitive Science, vol. 1, no. 4, pages 395–420, 1977.

[Levinson 1983] S.C. Levinson. Pragmatics. Cambridge University Press, 1983.

[Lewin 2000] I. Lewin. *A formal model of conversational game theory*. In Fourth Workshop on the Semantics & Pragmatics of Dialogue, 2000.

[Lewis 1969] D. K. Lewis. Convention : A philosophical study. Cambridge MA : Harvard University Press, 1969.

[Lewis 1979] D. K. Lewis. *Scorekeeping in a language game*. Journal of philosophical logic, vol. 8, pages 339–359, 1979.

[Lieberman 1997] H. Lieberman. *Autonomous interface agents.* In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, page 67–74, 1997.

[Lieberman 2009] H. Lieberman. *User interface goals, AI opportunities.* AI Magazine, vol. 30, no. 4, page 16, 2009.

[Litman 1990] D.J. Litman et J.F. Allen. *Discourse processing and commonsense plans.* In P. Cohen, J. Morgan et M. Pollack, editeurs, Intentions in communication, page 365–388. MIT Press, Cambridge, 1990.

[Ljunglöf 2009] P. Ljunglöf. *trindikit.py : An open-source Python library for developing ISU-based dialogue systems.* In Proc. of IWSDS'09, 1st International Workshop on Spoken Dialogue Systems Technology Workshop, Kloster Irsee, Germany, 2009.

[Lochbaum 1994] K.E. Lochbaum. *Using collaborative plans to model the intentional structure of discourse.* In Computational Linguistics, 1994.

[Lochbaum 1998] K.E. Lochbaum. *A collaborative planning model of intentionnal structure.* Computationnal linguistic, vol. 24, no. 4, pages 525–572, 1998.

[Loisel 2008] A. Loisel. *Modélisation du dialogue Homme-Machine pour la recherche d'informations : approche questions-réponse.* PhD thesis, INSA Rouen, 2008.

[Loisel 2011] A. Loisel, G. Dubuisson Duplessis, N. Chaignaud et J-Ph Kotowicz. *Un système de dialogue fondé sur l'étude d'interactions humaines.* In Sixième Journées Francophones sur les Modèles Formels d'Interaction (MFI-2011), Juin 2011.

[Loisel 2012] A. Loisel, G. Dubuisson Duplessis, N. Chaignaud et J-Ph Kotowicz. *A Conversational Agent for Information Retrieval Based on a Study of Human Dialogues.* In Joaquim Filipe et Ana Fred, editeurs, Proceedings of the 4th International Conference on Agents and Artificial Intelligence, volume 1, pages 312–317. SciTePress, 2012.

[Luzzati 1989] D. Luzzati. *Recherches sur le dialogue homme-machine, modèles linguistiques et traitement automatique.* Thèse d'état, Université de la Sorbonne, 1989.

[Mann 1988] W. C. Mann. *Dialogue games : Conventions of human interaction.* Argumentation, vol. 2, no. 4, page 511–532, 1988.

[Mann 2002] W. C. Mann. *Dialogue macrogame theory.* In Proceedings of the 3rd SIGdial workshop on Discourse and dialogue, volume 2, page 129–141, 2002.

[Manning 1999] C. D. Manning et H. Schütze. Foundations of statistical natural language processing. MIT press, 1999.

[Marchionini 1989] G. Marchionini. *Information-seeking strategies of novices using a full-text electronic encyclopedia.* Journal of the American Society for Information Science, vol. 40, no. 1, page 54–66, 1989.

[Marchionini 2008] G. Marchionini et R. White. *Find what you need, understand what you find.* International Journal of Human-Computer Interaction, vol. 23, no. 3, page 205–237, 2008.

[Martin 1999] D. L. Martin, A. J. Cheyer et D. B. Moran. *The Open Agent Architecture : A framework for building distributed software systems.* Applied Artificial Intelligence, vol. 13, no. 1-2, page 91–128, 1999.

[Maudet 2001] N. Maudet. *Modéliser les conventions des interactions langagières : la contribution des jeux de dialogue.* Thèse de doctorat, Université Paul Sabatier, Toulouse, France, Mai 2001.

[Maudet 2002a] N. Maudet. *A la recherche de la structure intentionnelle dans le dialogue.* Traitement automatique des langues, vol. 43, no. 2, page 71–98, 2002.

[Maudet 2002b] N. Maudet et B. Chaib-draa. *Commitment-based and dialogue-game based protocols : new trends in agent communication languages.* The Knowledge Engineering Review, vol. 17, no. 2, pages 157–179, 2002.

[Maudet 2003a] N. Maudet. *Negotiating dialogue games.* Autonomous Agents and Multi-Agent Systems, vol. 7, no. 3, page 229–233, 2003.

[Maudet 2003b] N. Maudet et P. Muller. *Pour un dialogue entre jeux et rhétorique.* Herzig, A., Chaib-draa, B., et Mathieu, P., éditeurs, Modeles Formels de l'Interaction, page 189–198, 2003.

[McBurney 2002] P. McBurney, S. Parsons et M. Wooldridge. *Desiderata for agent argumentation protocols.* In Proceedings of the first International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), page 402–409, 2002.

[McTear 2004] M. McTear. Spoken dialogue technology : toward the conversational user interface. Springer-Verlag New York Inc, 2004.

[Moeschler 1989] J. Moeschler. Modélisation du dialogue : représentation de l'inférence argumentative. Hermes, 1989.

[Morge 2013] M. Morge, F. Delecroix et J-C. Routier. *A Virtual Selling Agent which is Persuasive and Adaptive.* In Sascha Ossowski, editeur, Agreement Technologies, volume 8 of *Law, Governance and Technology Series*, pages 625–645. Springer Netherlands, 2013.

[O'Day 1993] V. L. O'Day et R. Jeffries. *Orienteering in an information landscape : how information seekers get from here to there.* In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems, page 438–445, 1993.

[Odersky 2008] M. Odersky, L. Spoon et B. Venners. Programming in scala : a comprehensive step-by-step guide. Artima Inc, 2008.

[Orkin 2009] J. Orkin et D. Roy. *Automatic learning and generation of social behavior from collective human gameplay.* In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, volume 1 of *AAMAS '09*, page 385–392, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[Orkin 2010] J. Orkin, T. Smith, H. Reckman et D. Roy. *Semi-automatic task recognition for interactive narratives with EAT & RUN.* In Proceedings of the Intelligent Narrative Technologies III Workshop, page 3, 2010.

[Orkin 2013] J. Orkin. *Collective Artificial Intelligence : Simulated Role-Playing from Crowdsourced Data.* PhD thesis, Massachusetts Institute of Technology, 2013.

[Oshry 2007] M. Oshry, R. Auburn, P. Baggia, M. Bodell, D. Burke, D. C. Burnett, E. Candell, J. Carter, S. McGlashan et A. Lee. *Voice eXtensible Markup Language (VoiceXML) 2.1.* W3C Recommendation, 2007.

[Pasquier 2003] P. Pasquier et B. Chaib-draa. *Engagements, intentions et jeux de dialogue.* Herzig, A., Chaib-draa, B., et Mathieu, P., rédacteurs, Modèles formels de l'interaction, Actes des Secondes Journées Francophones (MFI'03), page 289–294, 2003.

[Pasquier 2005] P. Pasquier. *Aspects cognitifs des dialogues entre agents artificiels : l'approche par la cohérence cognitive.* PhD thesis, Université Laval, Canada, 2005.

[Pauchet 2006] A. Pauchet. *Modélisation cognitive d'interactions humaines dans une cadre de planification multi-agents*. PhD thesis, Université Paris Nord, 2006.

[Petukhova 2007] V. Petukhova et H. Bunt. *A multidimensional approach to multimodal dialogue act annotation*. In Proceedings of the seventh international workshop on computational semantics IWCS-7, page 142–153, 2007.

[Petukhova 2009a] V. Petukhova et H. Bunt. *Dimensions in communication*. Rapport technique, Tilburg University, 2009.

[Petukhova 2009b] V. Petukhova et H. Bunt. *The independence of dimensions in multidimensional dialogue act annotation*. In Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, page 197–200, 2009.

[Poesio 1997] M. Poesio et D. R. Traum. *Conversational actions and discourse situations*. Computational intelligence, vol. 13, no. 3, page 309–347, 1997.

[Poesio 1998a] M. Poesio et A. Mikheev. *The predictive power of game structure in dialogue act recognition : Experimental results using maximum entropy estimation*. In Fifth International Conference on Spoken Language Processing, 1998.

[Poesio 1998b] M. Poesio et D. Traum. *Towards an Axiomatization of Dialogue Acts*. In Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology), page 207–222, 1998.

[Polanyi 1984] L. Polanyi et R. Scha. *A syntactic approach to discourse semantics*. In Proceedings of the 10th international conference on Computational linguistics, page 413–419, 1984.

[Pollack 1990] M. E. Pollack. *Plans as complex mental attitudes*. In P.R. Cohen, J. Morgan et M. E. Pollack, editeurs, Intentions in Communication, pages 77–104. MIT Press, 1990.

[Prévot 2004] L. Prévot. *Structures sémantiques et pragmatiques pour la modélisation de la cohérence dans des dialogues finalisés*. PhD thesis, Université Paul Sabatier, 2004.

[Pulman 1998] S. Pulman. *The TRINDI project : Some preliminary themes*. In Proceedings of the Twente Workshop on Language Technology, 1998.

[Purver 2004] M. Purver. *The Theory and Use of Clarification Requests in Dialogue*. PhD thesis, King's College, University of London, 2004.

[Rao 1995] A. S. Rao et M. P. Georgeff. *BDI agents : From theory to practice*. In Proceedings of the first international conference on multi-agent systems (ICMAS-95), page 312–319, 1995.

[Reed 1998] C. Reed. *Dialogue frames in agent communication*. In Proceedings of the third International Conference on Multiagent Systems, page 246–253, 1998.

[Rich 2001] C. Rich, C.L. Sidner et N. Lesh. *Collagen : Applying Collaborative Discourse Theory to Human-Computer Interaction*. AI Magazine, vol. 22, no. 4, page 15–25, 2001.

[Rich 2007] C. Rich et C.L. Sidner. *DiamondHelp : A Generic Collaborative Task Guidance System*. AI Magazine, vol. 28, no. 2, 2007.

[Rich 2012] C. Rich et C. L. Sidner. *Using Collaborative Discourse Theory to Partially Automate Dialogue Tree Authoring*. In Intelligent Virtual Agents, page 327–340, 2012.

[Rijsbergen 1979] C. J. V. Rijsbergen. Information retrieval. Butterworth-Heinemann, Newton, MA, USA, 2nd édition, 1979.

[Roulet 1987] E. Roulet, A. Auchlin, J. Moeschler, C. Rubattel et M. Schelling. L'articulation du discours en français contemporain, volume 11. P. Lang, 1987.

[Sacks 1974] H. Sacks, E. A. Schegloff et G. Jefferson. *A simplest systematics for the organization of turn-taking for conversation.* Language, page 696–735, 1974.

[Sadek 1991] D. Sadek. *Attitudes mentales et interaction rationnelle : vers une théorie formelle de la communication.* PhD thesis, Université de Rennes I, 1991.

[Schank 1977] R. Schank et R. Abelson. *Scripts, plans, goals, and understanding : An inquiry into human knowledge structures.* Hillsdale, NJ, 1977.

[Schegloff 1973] E. A. Schegloff et H. Sacks. *Opening up closings.* Semiotica, vol. 8, no. 4, page 289–327, 1973.

[Schroder 2010] M. Schroder. *The SEMAINE API : towards a standards-based framework for building emotion-oriented systems.* Advances in HCI, vol. 2010, page 2–2, 2010.

[Searle 1969] J.R. Searle. Speech acts – an essay in the philosophy of language. Cambridge Univ Pr, 1969.

[Searle 1985] J. R. Searle et D. Vanderveken. Foundations of illocutionary logic. Cambridge University Press, 1985.

[Searle 1990] J. Searle. *Collective Intentions and Actions.* In P. R. Cohen, J. Morgan et M. E. Pollack, editeurs, Intentions in Communication, pages 401–415. Cambridge : MIT Press, 1990.

[Serban 2013] O. Serban et A. Pauchet. *AgentSlang : A fast and reliable platform for Distributed Interactive Systems.* In Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on, page 35–42, 2013.

[Shah 2008] C. Shah. *Toward Collaborative Information Seeking (CIS).* In Proceedings of Collaborative Exploratory Search workshop at JCDL 2008, 2008.

[Singh 1991] M. P. Singh. *Social and Psychological Commitments in Multiagent Systems.* In AAAI Fall Symposium on Knowledge and Action at Social and Organizational Levels, page 104–106, 1991.

[Singh 1998] M. P. Singh. *Agent communication languages : Rethinking the principles.* IEEE Computer, vol. 31, no. 12, page 40–47, 1998.

[Singh 1999] M. P. Singh. *An ontology for commitments in multiagent systems.* Artificial Intelligence and Law, vol. 7, no. 1, page 97–113, 1999.

[Singh 2008] M. P. Singh. *Semantical Considerations on Dialectical and Practical Commitments.* In AAAI, volume 8, page 176–181, 2008.

[Sperber 1989] D. Sperber et D. Wilson. La pertinence : Communication et cognition. Les éditions de Minuit, 1989.

[Stalnaker 1979] R. C. Stalnaker. *Assertion.* In Syntax and Semantics, volume 9. Cole, P., 1979.

[Stolcke 2000] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema et M. Meteer. *Dialogue act modeling for automatic tagging and recognition of conversational speech.* Computational linguistics, vol. 26, no. 3, page 339–373, 2000.

[Sutcliffe 1998] A. Sutcliffe et M. Ennis. *Towards a cognitive theory of information retrieval.* Interacting with computers, vol. 10, no. 3, page 321–351, 1998.

[Swartout 2006] W. R. Swartout, J. Gratch, R. W. Hill Jr, E. H. Hovy, S. Marsella, J. Rickel et D. R. Traum. *Toward Virtual Humans.* AI Magazine, vol. 27, no. 2, pages 96–108, 2006.

[Tecuci 2007] G. Tecuci, M. Boicu et M. T. Cox. *Seven aspects of mixed-initiative reasoning : An introduction to this special issue on mixed-initiative assistants.* AI Magazine, vol. 28, no. 2, page 11, 2007.

[Teevan 2004] J. Teevan, C. Alvarado, M. S. Ackerman et D. R. Karger. *The perfect search engine is not enough : a study of orienteering behavior in directed search.* In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR'04), page 415–422, 2004.

[Telang 2012] P. R. Telang, M. P. Singh et N. Yorke-Smith. *Relating goal and commitment semantics.* In Programming Multi-Agent Systems, page 22–37. Springer, 2012.

[Traum 1992] D. R. Traum et E. A. Hinkelman. *Conversation Acts in Task-oriented Spoken Dialogue.* Computational intelligence, vol. 8, no. 3, page 575–599, 1992.

[Traum 1994a] D. R. Traum. *A computational theory of grounding in natural language conversation.* PhD thesis, University of Rochester, 1994.

[Traum 1994b] D. R. Traum et J. Allen. *Discourse obligations in dialogue processing.* In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, page 1–8, 1994.

[Traum 1997] D. R. Traum. *A reactive-deliberative model of dialogue agency.* In Intelligent Agents III Agent Theories, Architectures, and Languages, page 157–171. Springer, 1997.

[Traum 1999] D. R. Traum. *Computational models of grounding in collaborative systems.* In Psychological Models of Communication in Collaborative Systems-Papers from the AAAI Fall Symposium, page 124–131, 1999.

[Traum 2003] D. R. Traum et S. Larsson. *The information state approach to dialogue management.* In Current and new directions in discourse and dialogue, page 325–353. Springer, 2003.

[Traum 2004] D. R. Traum. *Issues in multiparty dialogues.* In Advances in agent communication, page 201–211. Springer, 2004.

[Vanderveken 1990] D. Vanderveken. Meaning and speech acts. Cambridge University Press, 1990.

[Wallace 2003] R. Wallace. *The elements of AIML style.* Alice AI Foundation, 2003.

[Walton 1995] D. N. Walton et E.C.W. Krabbe. Commitment in dialogue : Basic concepts of interpersonal reasoning. NY : State University of New York Press, 1995.

[Weisser 2003] M. Weisser. *SPAACy A semiautomated tool for annotating dialogue acts.* International journal of corpus linguistics, vol. 8, no. 1, page 63–74, 2003.

[Weizenbaum 1966] J. Weizenbaum. *ELIZA - a computer program for the study of natural language communication between man and machine.* Communications of the ACM, vol. 9, no. 1, pages 36–45, Janvier 1966.

[Winograd 1986] T. Winograd et F. Flores. Understanding computers and cognition : A new foundation for design. Ablex, 1986.

[Wittgenstein 1958] L. Wittgenstein. Philosophical investigations, volume 255. Blackwell Oxford, 1958.

[Yolum 2004] P. Yolum et M. P. Singh. *Reasoning about commitments in the event calculus : An approach for specifying and executing protocols*. Annals of Mathematics and Artificial Intelligence, vol. 42, no. 1-3, page 227–253, 2004.

[Yuan 2008] T. Yuan, D. Moore et A. Grierson. *A Human-Computer Dialogue System for Educational Debate : A Computational Dialectics Approach*. International Journal of Artificial Intelligence in Education, vol. 18, no. 1, page 3–26, 2008.

[Yuan 2011] T. Yuan, D. Moore, C. Reed, A. Ravenscroft et N. Maudet. *Informal logic dialogue games in human-computer dialogue*. Knowledge Engineering Review, vol. 26, no. 2, page 159–174, 2011.