

Modèles Décisionnels d'Interaction Homme-Robot

Abir-Beatrice Karami

▶ **To cite this version:**

Abir-Beatrice Karami. Modèles Décisionnels d'Interaction Homme-Robot. Interface homme-machine [cs.HC]. Université de Caen, 2011. Français. NNT : . tel-01076430

HAL Id: tel-01076430
https://hal.science/tel-01076430

Submitted on 22 Oct 2014

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

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

université de Caen
Basse-Normandie

U.F.R. : Sciences

ÉCOLE DOCTORALE : SIMEM

THÈSE

présentée par

Abir Béatrice Karami

et soutenue

le 12 Décembre 2011

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

spécialité : Informatique et applications
(Arrêté du 7 août 2006)

Modèles Décisionnels d'Interaction Homme-Robot

MEMBRES du JURY

Rachid Alami	Directeur de Recherche	LAAS-CNRS, Toulouse	(Rapporteur)
François Charpillet	Directeur de Recherche	INRIA-LORIA, Nancy	(Rapporteur)
Peter Ford Dominey	Directeur de Recherche	INSERM-CNRS, Lyon	
Laurent Jeanpierre	Maître de Conférence	U. Caen Basse-Normandie	(Encadrant)
Joelle Pineau	Associate Professor	U. McGill, Montreal	
Abdel-Illah Mouaddib	Professeur	U. Caen Basse-Normandie	(Directeur)

Mis en page avec la classe thloria.

To my parents ...

Acknowledgments

First of all, I would like to express my gratitude to my supervisor, Abdel-Illah Mouaddib, whose expertise, passion for research and optimism taught and motivated me considerably during these three years.

I would like to thank Laurent Jeanpierre for his guidance and enriching discussions. I would also like to thank Laetitia Matignon for her continuous support and inspirational advices. I thank you both for the time and effort you devoted in reviewing this thesis and for your assistance which I gratefully acknowledge.

For accepting to be part of the jury, I thank Mme. Joelle Pineau and Mr. Peter Ford Dominey. I also thank Mr. François Charpillet for his investment as a reporter. I wish also to express my appreciation and my gratitude to Mr. Rachid Alami for his investment as a reporter and for his welcome and enriching discussions during my collaboration with his team.

I thank all and every member of MAD team. I start with Simon Le Gloannec for his welcome and help on my arrival. I would like to express my warm thanks to Laetitia, Benoit, Lamia, Boris, Arnaud, Gregory, Nicolas and Mathieu for the friendly environment they create and for making those three years a lovely and interesting experience.

My precious Guillaume, thank you for supporting me especially in difficult times and for all your lovely surprises that lift my worries away. Thank you for being here and for being you.

Finally, I would like to thank my family; my sister Nada for always being around, my brother Hisham that I miss much and I would like to express my love and eternal gratitude to my parents, Fouad and Lorraine to whom I dedicate this thesis.

Abstract

This thesis is focused on decision models for human-robot interaction based on Markovian Decision Processes. First, we propose an augmented decision model that allows a companion robot to act considering estimated human intentions. This model addresses the problem of estimating the intention of the human by observing his actions. We proposed to simulate the behavior of a human to build a library of human action values toward his possible intentions. These values are integrated into the augmented Partially Observable Markov Decision Process (POMDP). Second, we present a coactive decision model that allows a robot in collaboration with a human to choose his behavior according to the progress of the shared task. This model is based on an augmented POMDP and allows the robot to act coactively to encourage the human actions and to perform the task in harmony with him. Third, we also propose a unified model for different types of human-robot interactions where the robot analyzes the needs of the human and acts accordingly. To overcome the complexity of POMDPs, the unified model divides the problem into several parts, the first estimates the human intention with a hidden Markov model (HMM) and another is responsible for choosing the corresponding type of interaction (collaboration, assistance, cooperation) using a Markov Decision Process (MDP). Finally, we propose a model that alternates between verbal interaction to infer the preference of the human using queries and non-verbal interaction in which preferences are estimated by observing the human actions. This model switches back to the verbal interaction when an ambiguity about the preferences is detected.

Outline: The manuscript contains: a detailed summery of the thesis and its contributions in French followed by the complete thesis in English including an introduction, the state of the art, the contributions and a conclusion.

Contents

Résumé étendu: Vers des robots compagnons intelligents partageant notre vie quotidienne **1**

1 Introduction 1

1.1 La problématique 1

1.2 Travaux existants 2

1.3 Apports de la thèse 3

2 État de l'Art: Les Modèles Markoviens 3

2.1 Les Chaînes de Markov et Chaînes de Markov Cachées 4

2.2 Les Processus Décisionnels Markoviens Observables 4

2.3 Les Processus Décisionnels Markoviens Partiellement Observables . . 6

3 Contributions 7

3.1 Un POMDP Augmenté pour inférer l'intention de l'humain 7

3.2 Le modèle de décision coactive 11

3.3 Un modèle décisionnel pour une sélection adaptative du type d'interaction 13

3.4 Modèles de coopération homme-robot verbal et non-verbal 18

4 Conclusion 22

I Introduction **23**

1 Introduction **25**

1.1 Motivation 25

1.2 Outline 26

II State of the Art **29**

2 Human-Robot Interaction and Companion Robots **31**

2.1 Human-Robot Interaction 31

2.1.1 HRI Paradigms 33

2.1.2 Robots Autonomy in HRI . . . 33
2.2 Companion Robots . . . 34
2.2.1 Inspiration from Human-Human Companionship . . . 35
2.2.2 Types of Interaction . . . 36
2.2.3 Types of Behavior . . . 37
2.3 Uncertainties in HRI and Companion Robots Systems . . . 39
2.3.1 HRI Environments . . . 39
2.3.2 Understanding Human Intentions . . . 40
2.4 Discussion . . . 41

3 Background on Theoretic Models 43
3.1 Belief-Desire-Intention Architecture . . . 43
3.2 Hierarchical Task Networks . . . 45
3.3 Bayesian Networks and Dynamic Bayesian Networks . . . 46
3.4 Markovian Models . . . 49
3.4.1 Markov Chains . . . 49
3.4.2 Hidden Markovian Models . . . 49
3.4.3 Markovian Decision Processes . . . 50
3.4.4 Partially Observable Markovian Decision Processes . . . 52
3.5 Discussion . . . 56

4 Related Work 59
4.1 Intention Recognition . . . 59
4.1.1 Approaches for Intention Recognition in HRI . . . 61
4.1.2 Ambiguity in Intention Recognition . . . 64
4.1.3 Learning . . . 65
4.2 Planning in HRI . . . 65
4.2.1 Dynamic Environments . . . 65
4.2.2 Planning Under Uncertainty . . . 66
4.2.3 Planning versus Interaction Type . . . 67
4.3 Discussion . . . 68

III Contributions 71

5 The Augmented Robot Decision Model for Human Intention Estimation 73
5.1 Motivation . . . 74

5.2 Evaluating Human Actions . . . 75
5.2.1 Modeling a Rational Human (The Human MDPs) . . . 75
5.2.2 The library of Human Action Values (Q-values) . . . 76
5.3 The Robot Decision Model . . . 78
5.3.1 States, Actions and Observations . . . 78
5.3.2 Rewards . . . 78
5.3.3 Transition and Observation Functions . . . 78
5.4 Illustration: Human-Robot Cooperation for Cleaning an Area . . . 80
5.4.1 Modeling the Human-MDPs for the Cooperative Mission . . . 81
5.4.2 Modeling the POMDP Decision Model for the Cooperative Mission . 82
5.5 Experimental Results . . . 84
5.5.1 State Space and Resolution Time . . . 85
5.5.2 Simulation Parameters . . . 85
5.5.3 Analyzing Simulations Results . . . 86
5.6 Discussion . . . 91

6 A Coactive Decision Model for Human-Robot Collaboration 93
6.1 Motivation . . . 94
6.2 Coactive Human-Robot Collaboration . . . 94
6.2.1 The Robot's CDM . . . 94
6.2.2 The Rational Human-MDPs for the Library of Q-values . . . 97
6.3 Illustrative Scenario: Handing Over an Object . . . 98
6.3.1 The CDM for Handing Over an Object Scenario . . . 98
6.3.2 Human MDPs for Handing Over an Object Scenario . . . 101
6.4 Experimental Results . . . 103
6.4.1 Simulations . . . 103
6.4.2 Integration on Real Robots . . . 105
6.5 Discussion . . . 108

7 A Decision Model for Adaptive Interaction-Type Selection 109
7.1 Motivation . . . 110
7.2 Definitions . . . 111
7.3 Model . . . 112
7.3.1 Level 1: The Human Intention Estimator (IE) . . . 114
7.3.2 Level 2: The Interaction-Class Selector (IS) . . . 115
7.3.3 Level 3: Choosing a Task and Applying a Type of Interaction . . . 117
7.4 A Companion Robot at Home Scenario . . . 119

7.5 Experimental Results 120
7.6 Discussion 122

8 Mixed Verbal and Non-Verbal Interaction for Inferring Human Preferences 123
8.1 Motivation 124
8.2 The Mixed Model for Human-Robot Cooperation 125
8.3 The Unified Framework 126
8.4 The Disjoined Framework 129
8.4.1 The Epistemic Dialog sub-Model 129
8.4.2 Intuitive HRI sub-Model 133
8.4.3 Switching between Epistemic and Intuitive Interactions 134
8.5 Experiments 135
8.5.1 Scenario of Cooperation using the Unified Framework 135
8.5.2 Scenario of Cooperation using the Disjoined Framework 135
8.5.3 Comparison Between Unified and Disjoined Frameworks 138
8.6 Discussion 139

IV Conclusion 141

9 Conclusion and Perspectives 143
9.1 Conclusion 143
9.2 Perspectives 144

Bibliography 145

List of Figures

1 Chaîne de Markov. 4
2 Le processus décisionnel du robot compagnon. 9
3 Une librairie de Q-valeurs créées à partir de Human-MDPs. 10
4 Le scénario coopératif de nettoyage des saletés. 10
5 Scénario de remise d'un objet à un humain avec Jido. 12
6 Scénario de remise d'un objet à un humain avec PR2. 13
7 Le modèle de décision HRMI. 16
8 Captures d'écran de la vidéo illustrant le scénario de HRMI. 17
9 L'architecture générale du modèle mixte pour une coopération homme-robot. . . 18
10 Flot de contrôle dans/entre les composants du modèle mixte. 19
11 L'environnement du scénario ≪ nettoyer une zone ≫. 20

2.1 A model of human-human cooperation. 35
2.2 A reactive agent behavior. 37
2.3 Reasoning about behavior. 38
2.4 Different sources of uncertainty in robotic environments. 40
2.5 Different sources of uncertainty in HRI environments. 40

3.1 A BDI architecture. 44
3.2 An HTN decomposition tree for a delivery problem. 47
3.3 A simple Bayesian Network represented as a Directed Acyclic Graph (DAG). . . 47
3.4 A generic Dynamic Bayesian Network structure consisting of 2 time slices, where t represents time. 48
3.5 A Markov Chain graph with its transition matrix T. 49
3.6 Museum visit example. 50

4.1 Inferring human intention by perceiving his actions. 60
4.2 Generating knowledge about the human by simulation. 61
4.3 A generic DBN model for intention recognition. 63
4.4 Different models of alternance between human and robot actions. 66

5.1 The high level process of the companion robot System. 75

5.2 The library of Q-values calculated from the Human-MDPs. 77
5.3 The environment of the cleaning an area domain. 80
5.4 Policy calculation-time and state space size for different problem sizes. 85
5.5 Cleaning area domain environment used for simulations. 86
5.6 The average of missions length and a histogram for those related to random human behavior simulations. 87
5.7 A histogram of the number of tasks achieved by the robot in 100 missions simulation. 87
5.8 Average of rewards per mission. 88
5.9 Frequency of robot actions with conflict with the real human intention over 100 missions simulations. 88
5.10 Comparing good robot estimations of human intentions using two methods for memory level of 0. 89
5.11 Comparing good robot estimations of human intentions using two methods for memory level of 50. 90
5.12 Comparing good robot estimations of human intentions using two methods for memory level of 99. 90

6.1 System Architecture: an Instance of LAAS Original Control Architecture. 105
6.2 Scenario handing object to human with PR2. 106
6.3 Scenario handing object to human with Jido. 107

7.1 The HRMI Decision Model. 112
7.2 Screen-shots of a video demonstrating three types of interactions. 120

8.1 The general architecture of the mixed model for the Human-Robot Cooperation. 125
8.2 The flow control in/between the components of the disjoined model. 130
8.3 Transition function values for one task (tk) preferences. An arrow connects a task state s and its next state s' and is labeled with the action a and the probability $T(s, a, s')$. 132
8.4 Cleaning area domain environment. 137

List of Tables

1 Relation entre les tâches et les différentes classes d'interaction. ≪ * ≫ signifie ≪ toutes valeurs admises ≫. 15
2 Une partie de l'interaction verbale/non-verbale pendant la réalisation de la mission coopérative. 21

3.1 A comparison between different Markovian models, BDI, DBN and HTN. 57

4.1 A Comparison between state of the art studies concerning approaches for intention recognition and companion robots planning. 70

6.1 Results of the handing object simulation with an engaged human. 104
6.2 Results of the handing object simulation with an occupied human. 105

7.1 Relevance between tasks and different interaction classes. " * " represents any possible value. 111
7.2 Experiments from HRMI model. 121

8.1 A dialog example between the human and the robot during the achievement of the mission (Unified Model). 136
8.2 Part of a cooperative verbal and non-verbal scenario between the human and the robot during the achievement of the mission (Disjoined Model). 137
8.3 Mixed verbal and non-verbal frameworks: model complexity and policy calculation-time. " - ". 139

Résumé étendu: Vers des robots compagnons intelligents partageant notre vie quotidienne

1 Introduction

1.1 La problématique

L'introduction des robots dans notre vie quotidienne a fait surgir de nouveaux défis pour les robots autonomes : s'adapter à l'existence des humains dans le même environnement et interagir avec eux. C'est une des principales raisons qui a mené à l'apparition d'un nouveau domaine de recherche appelé l'Interaction Homme-Robot (IHR).

Aujourd'hui, l'IHR est un thème de recherche très riche et en pleine expansion dans plusieurs directions. Cette thèse s'intéresse aux robots compagnons où l'IHR est considérée du point de vue de la conception de robots sociables qui interagissent avec les humains d'une manière naturelle. Un nombre croissant d'applications robotiques nécessite que les humains considèrent les robots comme des partenaires plutôt que des outils.

Les domaines d'applications concernant les robots compagnons varient en fonction du type d'interaction entre l'humain et le robot. Les robots compagnons peuvent agir seuls en respectant l'humain, son existence et ses besoins. Par example, ils peuvent accomplir des tâches non désirées par l'humain (pénibles, dangereuses, ...) [Cirillo *et al.*, 2009a]. Ils peuvent également assister l'humain pour qu'il réalise une tâche par lui-même, dans ce cas le robot doit détecter le besoin d'assistance et offrir les informations nécessaires à l'humain pour lui faciliter la réalisation de sa tâche. Un robot compagnon peut offrir, par example, des rappels concernant des activités quotidiennes pour les personnes âgées [Pineau *et al.*, 2003, Boger *et al.*, 2005, Duong *et al.*,]. Récemment, beaucoup d'intérêt s'est focalisé autour des applications de collaboration homme-robot pour réaliser une tâche commune où l'humain et le robot agissent ensemble en formant une équipe (chacun étant responsable de ses décisions) [Hoffman et Breazeal, 2008, Sisbot *et al.*, 2010].

Dans le domaine de l'IHR, le robot doit être impliqué dans l'interaction. Cela nécessite qu'il soit parfois non seulement réactif mais aussi qu'il ait des comportements différents pour inciter

une réaction de la part de l'humain. À ce sujet, [Johnson *et al.*, 2010] ont introduit le concept de « coactivité ». Un comportement coactif permet au robot non seulement d'exécuter sa part du travail mais aussi d'inciter l'humain à une interaction pour une activité jointe.

Beaucoup de difficultés font face à la prise de décision d'un robot lorsqu'il est en présence, interaction ou collaboration avec un humain [Klein *et al.*, 2004]. De nombreux efforts sont investis au sujet de la robotique humanoïde et ses applications dans l'IHR. Malheureusement, il n'y a pas encore une théorie puissante ou un système robuste pour la planification des tâches du robot compagnon.

Cette thèse aborde les capacités de raisonnement du robot dans ce contexte notamment pour répondre à des questions telles que :

- Comment inférer les intentions de l'humain afin de connaître ses besoins ?
- Comment se comporter et interagir pour assurer la satisfaction des besoins ?
- Comment aider l'humain à atteindre ses objectifs ?
- Comment planifier dans un environnement dynamique pour le bien-être de l'humain et comprendre ses intentions non observables ?

1.2 Travaux existants

Le problème de la reconnaissance de l'intention existait bien avant l'existence de l'IHR car on s'y intéresse depuis l'interaction homme-machine. En IHR, plusieurs domaines d'interaction nécessitent que le robot estime ou reconnaisse les intentions de l'humain. De nombreuses approches ont été utilisées dans la littérature pour la reconnaissance de l'intention. Ces approches sont basées sur des chaînes de Markov cachées (HMM) [Bui *et al.*, 2002, Bui, 2003, Nguyen *et al.*, 2005, Kelley *et al.*, 2008, Duong *et al.*,], des réseaux bayésiens dynamiques (DBN) [Pollack *et al.*, 2003, Schrempf et Hanebeck, 2005, Hui et Boutilier, 2006, Schmid *et al.*, 2007, Natarajan *et al.*, 2007, Schrempf *et al.*, 2007, Krauthausen et Hanebeck, 2009] ou les Q-valeurs des Processus de Décision Markovien [Fern *et al.*, 2007].

Pour leur capacité connue à planifier dans des environnements incertains et partiellement observables, les Processus Décisionnels Markoviens Partiellement Observables (POMDP) sont largement utilisés par la communauté d'IHR et dans divers contextes comme : la modélisation de l'incertitude dans les systèmes de reconnaissance de la parole [Williams, 2006, Schmidt-Rohr *et al.*, 2008a, Schmidt-Rohr *et al.*, 2008b, Young *et al.*, 2010], des robot assistants [Taha *et al.*, 2008, J. Pineau et Thrun, 2003, Fern *et al.*, 2007, Doshi et Roy, 2008, Hoey *et al.*, 2010], des robots sociables [Broz *et al.*, 2008, Broz *et al.*, 2011], le coût d'une action collaborative « cost-based Markov process » [Hoffman et Breazeal, 2007] et le coût d'interruption de l'humain [Armstrong-Crews et Veloso, 2007, Kamar *et al.*, 2009, Rosenthal et Veloso, 2011, Rosenthal et Veloso, 2011].

De nombreuses recherches en IHR se focalisent sur les robots collaboratifs et la planification des actions du robot afin de réaliser une tâche avec l'humain. Cependant, on trouve très peu

de travaux dans la littérature qui s'intéressent à des robots capables de planifier leur façon d'interagir avec les humains ou d'adapter leur comportement selon les besoins de l'humain.

1.3 Apports de la thèse

Nous allons introduire dans cette thèse plusieurs modèles décisionnels de l'Interaction Homme-Robot basés sur des Processus Décisionnels Markoviens. Ces modèles s'appliquent pour des problèmes d'interaction différents en permettant au robot compagnon de comprendre les intentions et les besoins de l'humain et d'agir en conséquence. Nos apports sont plus particulièrement :

- Un modèle appelé POMDP augmenté « augmented POMDP » qui considère l'intention de l'humain comme une variable cachée du POMDP. Pour avoir une estimation de cette variable, nous proposons de simuler le comportement de l'humain afin de construire une bibliothèque de valeurs d'action de l'humain par rapport à ses intentions possibles. Ces valeurs sont intégrées dans le modèle de décision augmenté du robot afin d'inférer l'intention de l'humain par l'observation de ses actions.

- Un modèle coactif de collaboration homme-robot pour réaliser une tâche partagée. Ce modèle est basé sur un POMDP augmenté et permet au robot d'avoir un comportement coactif, c'est à dire d'agir en harmonie avec l'humain et si nécessaire de guider ses actions afin de bien réaliser la tâche.

- Un modèle unifié d'interaction homme-robot multi-types permettant au robot de choisir le bon type d'interaction qui respecte l'intention de l'humain et son besoin d'interaction (assistance, collaboration, coopération).

- Un modèle mixte composé d'interactions verbales et non-verbales entre un robot et son partenaire humain. L'interaction verbale permet au robot d'envoyer des requêtes à l'humain afin de connaître ses préférences. L'interaction non-verbale permet au robot de réaliser des tâches et à la fois d'inférer les préférences de l'humain intuitivement (en observant ses actions). Enfin, le modèle mixte gère l'alternance entre les deux modèles selon l'ambiguïté sur les préférences.

Dans la version complète de la thèse (écrite en Anglais), nous présentons une version détaillée de ces modèles et des résultats expérimentaux pour montrer l'efficacité du modèle d'estimation de l'intention et montrer que le comportement du robot est bien adapté aux différentes situations. Nous discutons aussi quelques perspectives pour des recherches futures.

2 État de l'Art: Les Modèles Markoviens

Un modèle markovien est un modèle stochastique qui possède la propriété de Markov.

Definition 1 ***(Propriété de Markov)***
La propriété de Markov est satisfaite si l'état du système à l'instant $t+1$ ne dépend que de l'état à l'instant t.

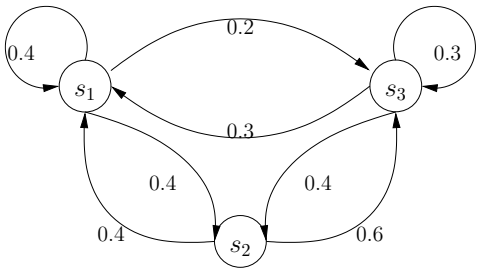
$$P(s_{t+1}|s_0, s_1, \ldots, s_t) = P(s_{t+1}|s_t)$$

Dans la suite, nous noterons s l'état à l'instant t et s' l'état à l'instant $t+1$.

2.1 Les Chaînes de Markov et Chaînes de Markov Cachées

Dans un système passif, une chaîne de Markov est constituée d'un ensemble d'états S et d'une fonction de transition T indiquant la probabilité $T(s'|s)$ de passer d'un état s à un état s'.

Dans une représentation sous forme de graphe (Figure 1), les nœuds du graphe correspondent aux états du système et les arêtes aux transitions entre les états. Chaque arête allant d'un état s à un état s' est étiquetée par une probabilité correspondant à la valeur de $T(s'|s)$.



$$T = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.4 & 0.0 & 0.6 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

Figure 1: Chaîne de Markov.

Dans certains cas, l'état du système peut être non-observable ou partiellement observable. De tels systèmes peuvent être modélisés par des chaînes de Markov cachées ou Hidden Markov Models (HMM) [Rabiner et Juang, 1986]. Le modèle est alors composé d'un ensemble d'états S, d'un ensemble d'observations Z, d'une fonction de transitions T et d'une fonction d'observation O. Cette dernière associe à chaque couple d'états (s, s') et à chaque observation $z \in Z$, une probabilité d'observation. $O(z|s, s')$ est la probabilité d'observer z sachant qu'on est passé de l'état s à l'état s'.

Plusieurs algorithmes basés sur les chaînes de markov sont décrits par [Rabiner et Juang, 1986]. Par example, pour une séquence d'observations, l'algorithme « Viterbi » calcule la séquence d'états la plus probable, l'algorithme « Forward » calcule la probabilité d'une séquence d'observations et l'algorithme « Baum–Welch » estime les probabilités initiales, la fonction de transition et la fonction d'observation du HMM.

2.2 Les Processus Décisionnels Markoviens Observables

Les Processus Décisionnels Markoviens Observables (MDP) sont les formalismes les plus courants pour des modèles de décision séquentiels.

Definition 2 *Un MDP est un tuple* $\langle S, A, T, R \rangle$ *où:*

- S *est un ensemble fini d'états* s*.*
- A *est un ensemble fini d'actions* a*.*
- $T : S \times A \to \prod(S)$ *est une fonction de transition markovienne donnant la probabilité de passer de l'état* s *à l'état* s' *quand l'action* a *est exécutée.*
- $R : S \times A \to \Re$ *est une fonction de récompense qui associe à chaque paire* (s, a) *la récompense obtenue par l'agent lorsqu'il exécute l'action* a *à partir de l'état* s*.*

Definition 3 *Une politique* π_{MDP} *d'un agent est une fonction* $\pi_{MDP} : S \to A$*, qui associe à chaque état du système* s*, une action* a *que l'agent doit exécuter.*

Une valeur V^π est définie afin d'évaluer les différentes politiques. Dans les problèmes à horizon fini $\mathcal{H}$, la valeur d'une politique pour un état s traduira l'espérance de la somme des récompenses espérées sur les $\mathcal{H}$ prochaines étapes en suivant la politique π à partir de l'état s, où $r^t = R(s, \pi(s))$.

$$V^\pi(s) = E\left[\sum_{t=0}^{\mathcal{H}} r_t\right] \qquad \forall s \in S$$

Dans le cas des problèmes à horizon infini ($\mathcal{H} = \infty$), l'espérance de gain est pondérée par un facteur d'atténuation γ.

$$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \qquad \forall s \in S$$

[Bellman, 1957] a montré que la fonction de valeur d'une politique peut être calculée par récurrence, grâce à *l'équation de Bellman* (Equation 1).

$$V_t^\pi(s) = R(s, \pi_t(s)) + \gamma \sum_{s' \in S} T(s, \pi_t(s), s') V_{t-1}^\pi(s') \tag{1}$$

La politique optimale π^* est la politique qui maximise la fonction de valeur.

$$\pi^*(s) = \arg\max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V^*(s') \right] \text{ où,}$$

$$V^*(s) = \max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V^*(s') \right]$$

L'équation de Bellman est à la base de plusieurs algorithmes de résolution des MDPs notamment Value Iteration [Bellman, 1957] et Policy Iteration [Howard, 1960]. L'algorithme de Value Iteration (Algorithme 1) consiste en une amélioration itérative de la valeur de chaque état du MDP en utilisant *l'équation de Bellman*. La valeur d'un état à l'itération t est calculée à partir de sa valeur à l'itération $t-1$. Le processus s'arrête lorsque la différence entre les valeurs

successives de tous les états est inférieure à un paramètre ϵ. La complexité d'une itération de cet algorithme est en $\mathcal{O}(|S^2||A|)$.

Algorithm 1: Value Iteration

Entrées: Un MDP $\langle S, A, T, R\rangle$, un facteur d'atténuation γ, une paramètre de précision ϵ.
Sorties : Politique optimale π.

Initialiser arbitrairement $V(s)$, $\forall s \in S$;
répéter
  $t = t + 1$;
  pour chaque $s \in S$ **faire**
    $V_t(s) = \max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{t-1}(s') \right]$
jusqu'à $\max_{s \in S} |V_t(s) - V_{t-1}(s)| \leq \epsilon$;
$\forall s,\ \pi(s) = \arg\max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_t(s') \right]$

2.3 Les Processus Décisionnels Markoviens Partiellement Observables

Un Processus Décisionnels Markoviens Partiellement Observable (POMDP) est défini par un tuple $\langle S, A, T, Z, O, R, b_0\rangle$ tel que : S, A, T, R ont les mêmes définitions que dans un MDP, et:

- Z est un ensemble fini d'observations.
- $O : S \times A \rightarrow \prod(Z)$ est une fonction d'observation donnant la probabilité $O(a, s', z)$ d'observer z depuis l'état s' quand l'action a est exécutée. $\sum_{z \in Z} O(a, s', z) = 1 \quad \forall (a, s')$.
- $b_0(s) = Pr(s_0 = s)$ est la probabilité que le système soit dans l'état s à l'instant $t = 0$.

Dans un POMDP, l'agent n'a pas une observabilité totale de son état, cependant, il maintient une croyance distribuée sur S. $b_t(s)$ est la probabilité que le système soit dans l'état s à l'instant t, sachant l'historique d'observations/actions que l'agent a reçues/effectuées et la croyance initiale $b_0(s)$:

$$b_t(s) = Pr(s_t = s | z_t, a_{t-1}, z_{t-1}, ..., a_0, b_0).$$

L'agent, à chaque instant, met à jour son état de croyance en appliquant la fonction de mise à jour [Cassandra *et al.*, 1994]

$$\begin{aligned} b'(s') &= \tau(b, a, z) \\ &= \frac{Pr(z|s, a, s', b) Pr(s'|a, b)}{Pr(z|a, b)} \\ &= \frac{\sum_{s \in S} O(s, a, s', z) T(s, a, s') b(s)}{Pr(z|a, b)} \end{aligned} \tag{2}$$

où a est la dernière action du robot, z est la dernière observation reçue et $Pr(z|a,b)$ est un facteur de normalisation:

$$Pr(z|a,b) = \sum_{s' \in S} \sum_{s \in S} O(s,a,s',z) T(s,a,s') b(s).$$

Definition 4 *Une politique π_{POMDP} d'un agent est une fonction $\pi_{POMDP} : b_t(s) \rightarrow A$, qui associe à chaque état de croyance du système $b(s)$, une action a que l'agent doit exécuter.*

L'approche classique optimale pour résoudre un POMDP est Value Iteration [Kaelbling *et al.*, 1998], où des itérations sont appliquées pour calculer des valeurs plus précises pour chaque état de croyance $V(b)$. L'équation 3 décrit la fonction de valeur (*l'équation de Bellman*) pour les POMDPs.

$$V_t(b) = \max_{a \in A} \left[\sum_{s \in S} b(s) R(s,a) + \gamma \sum_{z \in Z} Pr(z|a,b) V_{t-1}(\tau(b,a,z)) \right] \tag{3}$$

Une fois que les itérations conduisent à une convergence, une politique optimale associe l'action qui maximise $V(b)$ à tout état de croyance b.

$$\pi_t^*(b) = \arg\max_{a \in A} \left[\sum_{s \in S} b(s) R(s,a) + \gamma \sum_{z \in Z} Pr(z|a,b) V_{t-1} \tau(b,a,z) \right] \tag{4}$$

L'opération de mise à jour représentée dans l'équation 3 atteint une complexité de $O(|S|^2|V||Z||A|)$, où $|V|$ est le nombre de α vecteurs représentant la fonction de valeur [Sondik, 1978]. Dans certains cas, si le problème a des espaces d'action et d'observation très bornés, la complexité pourrait être $O(|S|^2|V|)$. Pour calculer une politique optimale, V doit être mis à jour sur la totalité de l'espace d'état de croyance, ce qui conduira à un calcul très coûteux pour les opérations de mise à jour entière.

Pour surmonter la complexité de résolution d'un POMDP d'une manière optimale, une grande variété d'algorithmes approximatifs ont été décrits pour diminuer la complexité et trouver des politiques acceptables pour un modèle POMDP [Smith et Simmons, 2004, Pineau *et al.*, 2006, Shani *et al.*, 2007, Dibangoye *et al.*, 2009].

Nous avons utilisé l'algorithme de « Value Iteration » classique pour résoudre les modèles de MDP de cette thèse, et le logiciel (public) ZMDP [Smith, 2005] pour les modèles de POMDP en appelant son algorithm « Focused Real-Time Dynamic Programming » .

3 Contributions

3.1 Un POMDP Augmenté pour inférer l'intention de l'humain

Nous nous intéressons aux scénarios d'IHR pour lesquels l'intention de l'humain n'est pas toujours connue du robot, mais où les actions de l'humain sont observables et peuvent être

détectées par le robot. Il faut alors que le système du robot soit à même de détecter/percevoir toutes les actions de l'humain qui concernent la mission humain-robot. De nombreux travaux étudient le problème de la perception des actions de l'humain via la vision [Poppe, 2010] ou via des capteurs embarqués [Zhu et Sheng, 2011], mais il ne s'agit pas du sujet de cette thèse. Notre objectif est de permettre au robot d'associer une action observée de l'humain à une intention possible de celui-ci. Ainsi, nous nous basons sur la théorie de la simulation par empathie pour proposer une façon d'évaluer les actions de l'humain, relativement à toutes les intentions qu'il peut avoir. Pour cela, nous intégrons dans le modèle décisionnel du robot les Q-valeurs associées aux actions de l'humain qui sont générées à partir de modèles de décision markoviens (Human-MDPs).

Les robots compagnons devraient considérer les humains comme des entités sociales dont le comportement est généré par des états mentaux sous-jacents. En se basant sur la théorie de la simulation par empathie (simulation theory of empathy, définition. 5), il est suggéré dans la littérature que l'on peut, en simulant les états mentaux d'une personne via une structure mentale similaire à la sienne, anticiper et comprendre le comportement des autres [Gray *et al.*, 2005]. Un robot compagnon disposant dans son système de suffisamment d'informations au sujet de l'humain peut réussir à faire de l'inférence sur les objectifs et croyances probables de celui-ci. Dans ce cas, le système du robot peut utiliser ses ressources non-seulement pour générer son propre comportement, mais également pour prédire et inférer celui de l'humain afin d'agir dans le respect de ses croyances et de ses objectifs.

Definition 5 *The Simulation theory of empathy [Rameson et Lieberman, 2009, Gallese et Goldman, 1998] proposes that we understand the thoughts and feelings of others by using our own mind as a model.*

Le modèle POMDP augmenté

Détecter les intentions de l'humain est une part essentielle de la tâche d'un robot compagnon. En conséquence, l'intention non-observable de l'humain doit être une des variables prise en compte pour la décision du robot. Il est rare que le robot connaisse de façon certaine l'intention de l'humain. Par contre, il peut maintenir une distribution de probabilités sur l'ensemble des intentions possibles. Pour pouvoir être utilisée, cette distribution devra être mise à jour à chaque fois que de nouvelles informations seront disponibles au sujet de l'humain ou de l'environnement. Cela permettra au robot de détecter les changements importants dans l'intention de l'humain, qu'ils soient dus à un changement d'avis de celui-ci ou à une mauvaise interprétation préalable du robot. On considère donc que le modèle décisionnel du robot compagnon est partiellement observable, ce qui nous a poussée à utiliser les POMDPs pour représenter ce modèle.

Le système du robot simule des politiques rationnelles pour l'humain et crée, à partir de ces politiques, une librairie de valeurs pour les actions de l'humain (Q-valeurs) qui seront ensuite intégrées au sein du modèle décisionnel du robot. Le système du robot construit pour

cela un ensemble de processus de décision markoviens associés à l'humain (Human-MDPs). Chaque Human-MDP (que l'on notera parfois MDP^h) est construit pour simuler, par empathie, un humain rationnel agissant conformément à l'une des intentions possibles. En résolvant le Human-MDP associé à une intention donnée, on génère des Q-valeurs qui représentent, pour cette intention et pour chaque état possible de l'humain, la valeur de chacune de ses actions. Ainsi, si on connaît l'état de l'humain et l'action exécutée, il suffit de comparer les Q-valeurs correspondantes issues de chaque Human-MDP afin de connaître l'importance de cette action relativement à chaque intention possible.

La figure 2 décrit, à haut niveau, le processus décisionnel du robot compagnon. Le robot commence par créer les Human-MDPs, puis résout ceux-ci afin d'obtenir la librairie de Q-valeurs qu'il intègre ensuite dans son propre modèle décisionnel.

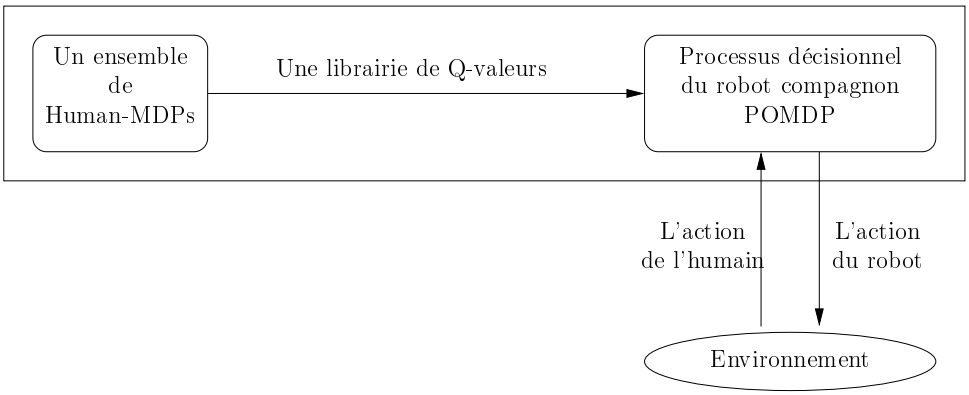


Figure 2: Le processus décisionnel du robot compagnon.

Il est impossible, pour le robot, de simuler de façon exacte le comportement de l'humain, et ce même si le robot dispose de toutes les informations au sujet de l'humain et de son état mental. Plusieurs raisons peuvent expliquer ce constat, telles que les émotions de l'humain ou encore des situations ne pouvant être prévues ni perçues par le robot. On peut par contre initialiser la simulation du comportement grâce à un modèle rationnel du comportement de l'humain, puis mettre à jour ce modèle en apprenant certaines variables au sujet de la personnalité de l'humain que le robot accompagne. Les modèles décrits dans cette thèse incluent uniquement le modèle rationnel de l'humain, sans améliorer celui-ci via l'apprentissage.

Afin de modéliser l'humain rationnel, le robot utilise des informations relatives à l'humain et à son environnement, aux actions possibles pour cet humain, à l'impact de ses actions, à l'objectif de l'humain et à ce qu'il faut et ne faut pas faire pour atteindre cet objectif. Ces informations correspondent alors aux états, actions, transitions et récompenses du Human-MDP.

On pose TK l'ensemble des tâches de la mission coopérative et $HI \subseteq TK$ l'ensemble des tâches que l'humain peut chercher à accomplir durant l'exécution de la mission humain-robot. En d'autres termes, HI contient toutes les tâches $hi \in HI$ que le robot doit considérer comme des intentions possibles de l'humain. Un Human-MDP (MDP^h_{hi}) est créé pour chaque intention $hi \in HI$ possible de l'humain. On obtient alors un ensemble de Human-MDPs (Chapter 5) :

$$\text{MDP}^h_{hi} = \left\langle S^h_{hi}, A^h_{hi}, T^h_{hi}, R^h_{hi}, \gamma^h \right\rangle \quad \forall hi \in HI$$

On calcule les Q-valeurs en résolvant ces Human-MDPs. On peut résoudre un MDP via l'algorithme classique de Value-Iteration [Puterman, 1994] ou via des algorithmes factorisés ou approximés [Boutilier *et al.*, 1999, Koller et Parr, 2000, Guestrin *et al.*, 2003, Guestrin *et al.*, 2011] afin de traiter des problèmes de grande taille.

La figure 3 montre l'utilité de ces Q-valeurs, via une librairie exemple. On peut voir, sur cette figure, que la valeur de l'action a^h_1 appliquée dans l'état s^h_1 est de 0,5 pour la tâche $hi_1 \in TK$ et de 0,2 pour la tâche $hi_2 \in TK$. Ces valeurs, si elles sont correctement utilisées dans le système décisionnel du robot, permettent de déduire que l'humain (lorsqu'il exécute l'action a^h_1 dans l'état s^h_1) cherche plus probablement à accomplir hi_1 que hi_2.

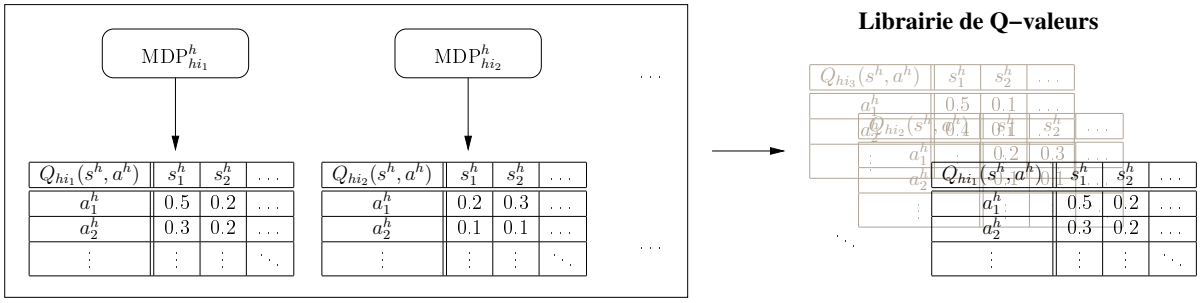


Figure 3: Une librairie de Q-valeurs créées à partir de Human-MDPs.

Expérimentations

Nous avons appliqué ce modèle décisionnel à un exemple dans lequel un robot compagnon coopère avec un humain afin de remplir une mission commune. La figure 4 décrit l'environnement partagé par le robot et l'humain dans lequel ils doivent nettoyer des saletés. Chaque saleté peut être nettoyée par l'humain seul, ou le robot seul. La mission est considérée comme terminée une fois que toutes les saletés ont été nettoyées.

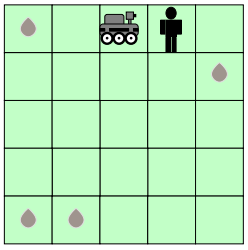


Figure 4: Le scénario coopératif de nettoyage des saletés.

Afin d'analyser les politiques produites, nous avons réalisé différentes simulations avec deux comportements différents pour l'humain. Le premier est le comportement rationnel, où l'humain choisit au hasard une des tâches possibles à faire, puis se comporte rationnellement (en suivant

les politiques de human-MDP) pour réaliser la tâche choisie. Le second est le comportement semi-rationnel, où l'homme suit un comportement rationnel, mais avec une probabilité de 30%, à chaque pas de temps, qu'il change son intention. Chaque simulation est initialisée dans l'état décrit à la figure 4 et se termine avec la fin de la mission (une fois les quatre saletés nettoyées).

Nous nous sommes concentrés sur les estimations de l'intention de l'humain faites par le robot à chaque pas de temps de chaque mission. Nous les avons comparées à l'intention réelle de l'humain rationnel et semi-rationnel. Nous avons remarqué qu'il y a un petit nombre de pas de temps où les estimations du robot ne correspondent pas à l'intention réelle de l'humain. On peut observer ces situations juste après un changement de l'intention réelle ou juste après que l'humain a fini de nettoyer une saleté. En effet, dans ces situations le robot n'a pas encore reçu assez d'observations lui permettant de bien estimer la nouvelle intention de l'humain.

Pour une version plus détaillée du modèle et des expérimentations, vous êtes invités à vous référer au chapitre 5.

3.2 Le modèle de décision coactive

Nous proposons un modèle décisionnel du robot compagnon en collaboration avec un humain pour une tâche commune (par exemple donner un objet à l'humain, déplacer une table avec lui ou encore remplir son verre). Pour de telles tâches, les robots doivent comprendre leurs partenaires humains et collaborer avec eux en tant que pairs pour bien réaliser ces tâches. À cette fin, nous proposons un modèle de décision coactive basé sur un POMDP augmenté. Cette coactivité permet une collaboration homme-robot harmonisée, elle permet aussi au robot de dévoiler l'intention de l'humain en cas d'ambiguïté. Le POMDP augmenté estime l'intention de l'humain en évaluant les actions de l'humain pour chaque intention possible (Q-valeurs).

Le grand défi pour les agents collaboratifs consiste à les faire agir ensemble pour arriver aux conditions optimales permettant la réalisation de la tâche partagée. L'agent collaboratif doit co-agir de manière à guider ou inciter son ou ses partenaires à collaborer vers les conditions de succès. Le robot compagnon doit aussi co-agir avec l'humain de la même manière, particulièrement quand l'humain est confus pendant la collaboration.

Ce domaine de recherche exige que le robot ait une capacité à réaliser une tâche en synergie avec un être humain, en particulier dans les situations de collaboration avec les personnes accompagnées. A ce sujet, [Johnson *et al.*, 2010] ont introduit le concept de « coactivité ». Un comportement coactif permet au robot non seulement d'exécuter sa part du travail mais aussi d'inciter l'humain à une interaction pour une activité jointe.

Cette contribution est motivée par la notion d'implémentation du concept de coactivité dans un système de IHR. Il est essentiel que le robot prenne en considération les réactions de l'humain. De plus, lorsque le robot est certain de l'intention de l'humain et que ce dernier ne collabore pas correctement, il doit le guider vers les actions qui lui permettent de poursuivre la collaboration vers la réalisation de la tâche. La réponse de l'humain à un tel comportement coactif peut être ensuite utilisée par le robot en tant qu'élément clef pour une meilleure compréhension de

l'intention de l'humain. Toutefois, tout ceci devrait être combiné avec une inspection attentive de l'intérêt de l'humain dans une telle collaboration. Le robot doit respecter son but principal en tant que compagnon qui consiste à respecter le confort de l'humain. Donc, dès que l'humain montre un désintéressement dans la collaboration, le robot ne devrait pas l'ennuyer en offrant son aide.

Le *Modèle de Décision Coactive* (CDM) proposé est basé sur un modèle de POMDP augmenté, CDM=$\langle S, A, Z, T, O, R, b_0 \rangle$. Toutefois le POMDP augmenté est modifié pour gérer le comportement coactif du robot. Il est important que le robot puisse différencier si l'humain est prêt à co-réaliser la tâche ou plutôt s'il a besoin d'y être incité. De plus, le robot doit être capable de détecter une situation où l'humain est occupé. Pour ces raisons, le CDM distingue deux catégories d'actions. La première catégorie d'actions est reliée au comportement coactif Ac. La seconde catégorie est reliée à la réalisation de la tâche At.

Donc, l'ensemble A des actions est défini comme :

$$A = A_c \cup A_t$$

Selon la dernière action effectuée par le robot, qu'elle soit coactive ou non, il est de la responsabilité du POMDP augmenté d'analyser l'action de l'humain observée ($z \in Z$) en tant qu'élément essentiel permettant une meilleure compréhension de la situation de collaboration. Cette analyse utilise une librairie de Q-valeurs créées à partir de deux « Human MDPs » définis par empathie. Le premier représente un humain collaborant, rationnel et intéressé à accomplir la tâche commune. Le deuxième représente un humain rationnel qui est occupé et n'est pas intéressé à accomplir la tâche commune.



Figure 5: Scénario de remise d'un objet à un humain avec Jido.

Ainsi, les Q-valeurs d'une action de l'humain sont un élément essentiel pour reconnaître s'il est intéressé ou non par la tâche. Ceci aidera le robot à décider, dans la prochaine étape, s'il doit commencer ou continuer à agir d'une manière coactive ou plutôt commencer ou continuer à co-réaliser la tâche avec l'humain, ou alors abandonner la tâche.

De plus, si la dernière action du robot était coactive, la Q-valeur de l'action de l'humain va clarifier le succès ou l'échec du comportement coactif.

La fonction de récompense du CDM équilibre la décision du robot entre l'impact négatif d'un comportement coactif persistant et le but principal qui est de réussir avec succès la tâche collaborative.

L'état de croyance initial b_0 représente l'incertitude concernant la volonté de l'humain de collaborer ou s'il est occupé.

Un modèle CDM pour un scénario démontrant la remise d'un objet à un humain a été appliqué en collaboration avec le LAAS-CNRS (Laboratoire d'analyse et d'architecture des Systèmes). Ce scénario a été appliqué sur deux robots du LAAS (Jido et PR2). Nous avons présenté différents simulations des comportements humain variés et montré l'aptitude du robot à s'adapter et co-agir pour une meilleure collaboration. Nous montrons des prises d'écrans de vidéos [1] réalisées lors de l'exécution des scénario avec Jido (figure 5) et avec PR2 (figure 6).

Une description complète du CDM ainsi que de l'architecture du robot est donnée dans le chapitre 6.



Figure 6: Scénario de remise d'un objet à un humain avec PR2.

3.3 Un modèle décisionnel pour une sélection adaptative du type d'interaction

Cette contribution cible deux problématiques. La première consiste à établir un système pour des robots compagnons. Ce système doit être capable de commuter entre différents types d'interactions afin de respecter les besoins de l'humain. La seconde problématique vise à modéliser ce système en utilisant une structure qui permette de surpasser les Processus de Décision Markovien Partiellement Observable (POMDP) pour des problèmes de taille importante. À cette fin, un modèle unifié d'Interaction Multi-type Homme-Robot (HRMI) est décrit. L'objectif est d'observer le comportement de l'humain, d'essayer de prédire et estimer son intention ou ses besoins et de réagir de façon appropriée (assister, coopérer, collaborer, ...). Il est impossible de résoudre un HRMI pour une application de taille importante en utilisant des POMDPs. Nous présentons une approche pour surmonter cette limitation. La problématique du HRMI est alors divisée en trois niveaux : le premier estime l'intention de l'humain ; le second sélectionne le type d'interaction approprié ; le troisième et dernier choisit parmi les politiques pré-calculées celle qui respecte l'intention de l'humain et le type d'interaction requis. Ainsi, le modèle d'HRMI inclut :

[1] `http://users.info.unicaen.fr/~akarami/demohri/JidoHRI_1.AVI`, `http://users.info.unicaen.fr/~akarami/demohri/JidoHRI_2.AVI`

- un modèle d'actualisation de la croyance de l'intention de l'humain. Ce modèle utilise plusieurs modèles MDPs de l'humain définis par rapport aux tâches réalisables par l'humain.

- un modèle MDP qui supporte la prise de décision du robot. Ces décisions sont orientées par les besoins de l'humain, elles correspondent à une interaction appropriée.

- un algorithme qui d'abord, définit la tâche à accomplir par le robot, puis choisit la politique pré-calculée pour la réaliser en respectant le type d'interaction.

Actuellement, la communauté IHR s'intéresse beaucoup aux robots compagnons qui assistent des personnes âgées. Le robot compagnon peut aider l'humain de différentes façons. Cela dépend des incapacités possibles de l'humain ou de ses préférences. Plus d'informations au sujet des contraintes et des désirs de l'humain devraient permettre au robot compagnon d'offrir la bonne aide au bon moment.

Pour que le robot soit capable de bien prendre sa décision, il doit déduire rapidement et avec précision l'intention de l'humain. De plus, il doit être capable de s'adapter rapidement à ses possibles changements d'intention.

Nous définissons I comme l'ensemble des classes d'interaction. Chaque classe d'interaction définit de quelle façon le robot aide l'humain. On peut distinguer dans la littérature d'IHR trois types d'interaction : Coopération CP, Assistance AS et Collaboration CL. En plus de celles-ci, une quatrième classe Confirmation CO est ajoutée pour le cas où le robot a besoin d'une confirmation de l'humain.

$$I = \{CP, AS, CL, CO\}$$

Nous présentons à présent nos définitions des trois principaux types d'interaction pour un robot compagnon qui partage le quotidien d'un humain.

Coopération Le robot peut coopérer avec l'humain en réalisant une tâche à sa place de façon à économiser son temps et ses efforts. Le robot doit être capable de réaliser la tâche seul (par example, nettoyer le tapis ou mettre la table) sans que cela ne dérange l'humain. Le choix de la tâche concomitante doit respecter l'intérêt et les préférences de l'humain.

Assistance Le robot compagnon peut assister l'humain en le guidant (typiquement le guider verbalement). Il doit être capable de détecter le besoin d'assistance de l'humain et de lui offrir la meilleure aide pour que l'humain soit capable de terminer sa tâche. Par exemple, en se référant au manuel du lave-vaisselle, le robot peut assister l'humain pour faire fonctionner la machine en le lisant étape par étape. Ce type d'interaction couvre aussi les robots assistant des personnes âgées atteintes de la maladie d'Alzheimer.

Collaboration Les tâches de collaboration regroupent les tâches qui nécessitent, pour être réalisées, la participation simultanée de l'humain et du robot (typiquement des actions

physiques). Le robot compagnon doit alors être présent lorsque l'humain montre un intérêt pour une tâche collaborative.

Le robot possède un ensemble des tâches possibles TK qui peuvent être réalisées par l'humain uniquement H, par le robot compagnon uniquement CR, par les deux ensembles $(H \wedge CR)$ ou par l'un d'entre eux $(H \vee CR)$. Une tâche $tk \in TK$ peut être définie comme $tk = \langle context, agent, policy, time, type \rangle$. *context* regroupe les informations sur la tâche par rapport au contexte du problème, en incluant les relations et les dépendances entre les tâches, ainsi que les conditions requises pour que la tâche soit réalisée. $agent \subset \{CR, H, H \vee CR, H \wedge CR\}$ caractérise qui peut réaliser la tâche. $policy \in \{possess, lack\}$ décrit si le robot possède un manuel ou une politique permettant d'assister l'humain dans la réalisation de la tâche. *time* est l'intervalle de temps normalement utilisé pour réaliser la tâche. Une tâche appartient à au moins un $type \subset I$ d'interaction selon les règles mentionnées dans le tableau 1.

Type d'interaction	Condition
Coopération	$tk = \langle *, \{CR, CR \vee H\}, *, *, \{CP\} \rangle$
Assistance	$tk = \langle *, \{H, CR \vee H\}, possess, *, \{AS\} \rangle$
Collaboration	$tk = \langle *, \{CR \wedge H\}, *, *, \{CL\} \rangle$
Confirmation	$tk = \langle *, \{H, CR \vee H\}, possess, *, \{CO\} \rangle$
Confirmation	$tk = \langle *, \{CR \wedge H\}, *, *, \{CO\} \rangle$

Table 1: Relation entre les tâches et les différentes classes d'interaction. « * » signifie « toutes valeurs admises ».

L'ensemble des tâches peut être représenté comme $TK = TK_h \cup TK_{cr}$, où le domaine de la variable *agent* pour TK_h est $\{H \vee CR, H, H \wedge CR\}$ et pour TK_{cr} est $\{H \vee CR, CR, H \wedge CR\}$. L'intention de l'humain peut être une des tâches réalisables par lui-même ou « ne rien faire », $intention \in TK_h \cup \{do_nothing\}$. La figure 7 présente les trois niveaux du système HRMI.

L'algorithme de haut niveau pour ce système fonctionne comme suit :

1. le système observe l'action de l'humain z^{ie}.
2. Niveau 1 : Le système d'actualisation de la croyance met à jour l'état de croyance sur l'intention de l'humain en utilisant la librairie de Q-valeurs: $update^{ie}(b'^{ie}|b^{ie}, z^{ie})$.
3. Niveau 2(a) : le sélecteur de classe d'interaction crée l'état s^{is} à partir de la croyance courante b^{ie}.
4. Niveau 2(b) : le sélecteur de classe d'interaction appelle la politique π^{is} pour choisir une classe d'interaction.
5. Niveau 3(a) : l'algorithme de sélection de tâche choisit la tâche à accomplir par le robot $tk \in TK_{cr}$.

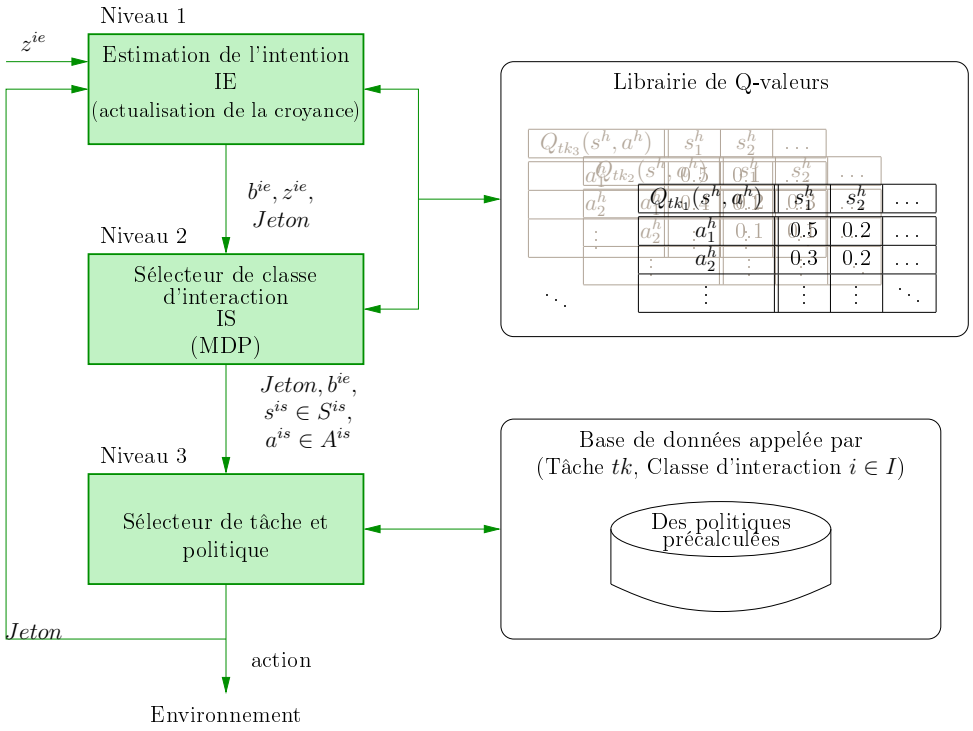


Figure 7: Le modèle de décision HRMI.

6. Niveau 3(b) : la politique appropriée est appelée depuis la base de données et est appliquée pour décider de l'action du robot.

7. Le robot applique son action et retourne à l'étape 1 de l'algorithme.

Le chapitre 7 décrit en détail les trois niveaux et présente plusieurs analyses sur la performance et l'extensibilité de la taille des problèmes considérés. Il est complété par une implémentation sur un robot réel pour un scénario inspiré de [RoboCup@home, 2011].

Ce scénario représentatif inclut trois tâches possibles et montre la capacité du modèle décisionnel à passer d'un type d'interaction à un autre tout en respectant les besoins de l'humain.

La première tâche est une tâche d'assistance (AS), elle consiste à trouver un livre sur l'étagère. Cette tâche peut être réalisée par l'humain seul ou avec l'assistance du robot ($agent = \{H\}$ et $policy = possess$). Si l'humain ne trouve pas le livre, le robot propose son assistance en cherchant dans sa base de données et lui indique sur quelle étagère le livre peut être trouvé.

La seconde tâche est de type collaboratif (CL), elle consiste à recharger l'imprimante en papier. Dans ce scénario, nous supposons que le robot connaît à chaque instant l'état de charge de l'imprimante en papier. Cependant, le robot n'est pas capable de la recharger seul, il attend une situation où l'humain est proche de l'imprimante pour collaborer. La tâche collaborative est réalisée à l'endroit où le robot apporte le papier à proximité de la machine, de façon à ce que l'humain n'ait plus qu'à la remplir. Ainsi la variable *agent* est définie comme suit : $agent = \{H \wedge CR\}$.

La troisième tâche est de type coopératif (CP), elle consiste à nettoyer les fenêtres. Cette tâche peut être réalisée seulement par le robot ($agent = \{CR\}$).



(a) Coopération



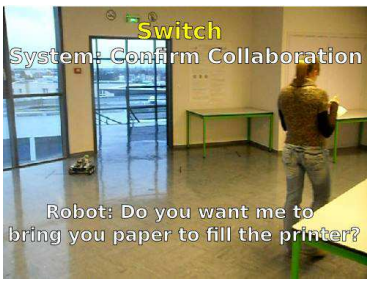
(b) Confirme Assistance



(c) Confirme Assistance



(d) Coopération



(e) Confirme Collaboration



(f) Confirme Collaboration



(g) Collaboration



(h) Collaboration



(i) Coopération



(j) Coopération

Figure 8: Captures d'écran de la vidéo illustrant le scénario de HRMI.

La figure 8 présente des captures d'écran de la vidéo[2] illustrant les trois types d'interaction entre l'humain et un robot koala. Au début, dans la figure 8(a), le robot commence à se déplacer

[2] `http://users.info.unicaen.fr/~akarami/demohri/demo_multi_interaction.avi`

vers les fenêtres pour les nettoyer. Pendant ce temps, l'humain hésite un moment en face de l'étagère, ce qui fait, dans la figure 8(b) et 8(c), que le robot propose son assistance pour trouver le livre et l'humain répond négativement. Dans la figure 8(e), la tâche de collaboration CL est confirmée lorsque l'humain s'approche de l'imprimante. Dans la figure 8(h), la tâche de collaboration est réalisée, le robot apporte le paquet de feuilles à l'humain. Enfin, la figure 8(j) montre que le robot retourne à la tâche coopérative où il nettoie la fenêtre.

3.4 Modèles de coopération homme-robot verbal et non-verbal

Coopérer avec un humain impose que le robot connaisse les préférences de son partenaire au sujet des tâches à réaliser de façon à satisfaire au mieux ses désirs pendant le déroulement de la mission (par exemple, effectuer les tâches indésirables). Le robot doit aussi ajuster rapidement son plan s'il observe un changement soudain de l'intention de l'humain au cours de la mission.

Les approches pour connaître l'intention de l'humain passent par des communication explicites (épistémiques) et/ou implicites (intuitives). Des communications explicites impliquent par exemple un système de dialogue. De telles communications semblent la solution évidente pour connaître les préférences du partenaire. L'inconvénient majeur est que le robot doit questionner l'humain en continu pour détecter ses changements d'intention à travers les incohérences de son discours. Pour éviter de poser des questions en continu, nous proposons de combiner le modèle épistémique explicite avec un modèle intuitif implicite. Ce dernier sera responsable de la coopération pendant l'observation des actions de l'humain pour deviner ses préférences via les Q-valeurs de ses actions. Il reste toujours possible de revenir sur des communications explicites pour lever toute ambiguïté détectée entre les préférences données par chacun des deux modèles.

Sur cette base, nous décrivons un modèle mixte qui permet au robot de passer d'interactions implicites intuitives à un dialogue épistémique explicite et vice-versa lors d'une mission coopérative avec un partenaire humain. L'architecture générale du modèle mixte est donnée dans la figure 9.

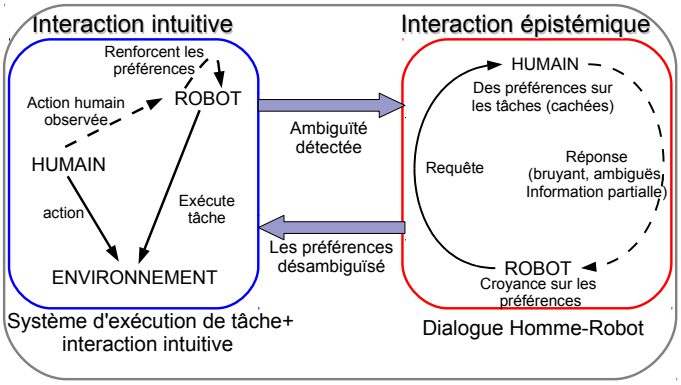


Figure 9: L'architecture générale du modèle mixte pour une coopération homme-robot.

Un de ces composants est un système de dialogue oral appelé modèle épistémique. Les actions du robot pendant cette phase d'interaction, sont de questionner l'humain. Les réponses peuvent

être alors bruitées ou ambiguës. Une fois l'ambiguïté levée le robot passe en mode d'exécution de tâches pour accomplir les tâches qui sont cohérentes avec les préférences de son partenaire. Le modèle de dialogue épistémique est représenté par le tuple :

$$\text{POMDP}_{ep} =< S_{ep}, A_{ep}, Z_{ep}, T_{ep}, O_{ep}, R_{ep}, b_{ep} > .$$

L'espace d'états rassemble l'ensemble des préférences de l'humain sur toutes les tâches. Ses préférences peuvent être :

- $s(tk) = to_do_by_robot$: l'humain préfère que ce soit le robot qui fasse la tâche tk,
- $s(tk) = to_do_by_human$: l'humain préfère faire la tâche tk lui même,
- $s(tk) = to_do_by_any$: l'humain n'a pas de préférences sur qui devrait faire la tâche tk (l'humain ou le robot),
- $s(tk) = undecided$: l'humain n'a pas encore choisi sa préférence pour la tâche tk,
- $s(tk) = unknown$: le robot n'a pas connaissance de la préférence de l'humain pour la tâche tk,
- $s(tk) = done$: la tâche tk est réalisée.

Le second composant est le modèle intuitif, appelé aussi le système d'exécution de tâche. La figure 10 présente un schéma plus détaillé de ce modèle.

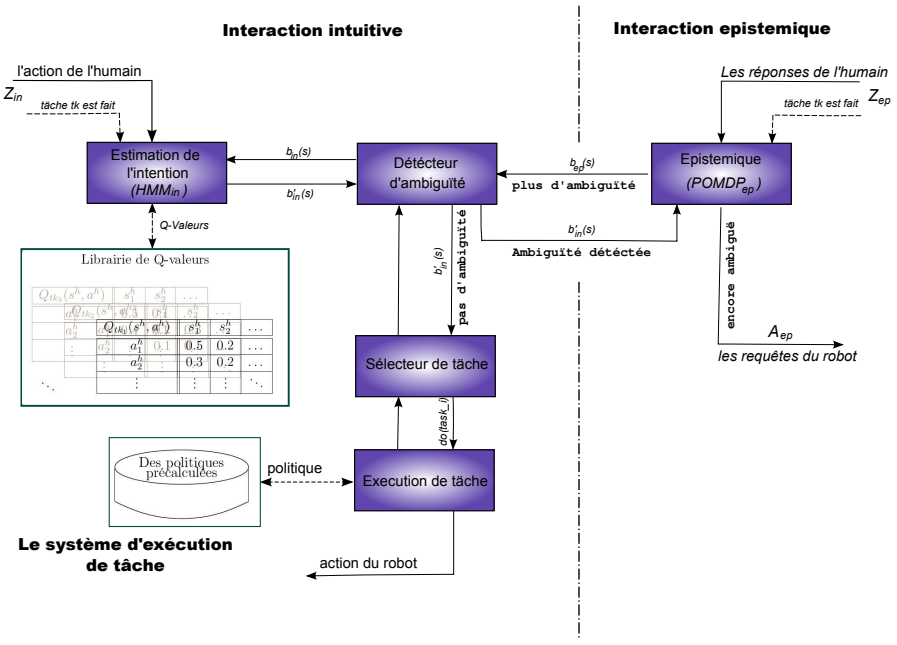


Figure 10: Flot de contrôle dans/entre les composants du modèle mixte.

Principalement, dans ce modèle, le robot choisit une tâche à réaliser et, tout au long de la réalisation de cette tâche, une reconnaissance intuitive des intentions du partenaire est active pour détecter tous changements dans ses préférences. Les croyances sur les préférences de l'humain (initialisées pendant le dialogue) sont renforcées par les préférences estimées via les Q-valeurs des actions observées de l'humain. Dans le cas d'une ambiguïté entre les deux croyances (épistémiques et intuitives) le robot revient sur un mode d'interaction épistémique pour lever l'ambiguïté sur la préférence ; autrement il continue d'exécuter les tâches en accord avec ses croyances sur les préférences de l'humain.

Nous avons effectué une expérience sous forme de scénario pour évaluer le comportement du modèle décrit, particulièrement le passage entre les interactions intuitives et épistémiques. Nous avons choisi le scénario de nettoyage des saletés où la mission consiste à nettoyer cinq saletés disposées à différentes positions dans l'environnement décrit dans la figure 11. Cette figure montre un numéro pour chacune des positions dans l'environnement, le numéro et la position de chacune des tâches (saleté à nettoyer) ainsi que la position initiale de l'humain.

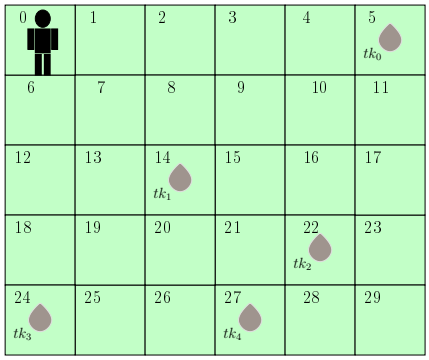


Figure 11: L'environnement du scénario ≪ nettoyer une zone ≫.

La table 2 montre une partie de l'interaction entre l'humain et le robot pendant la réalisation de la mission. Nous mentionnons que $s^*_{in} = \underset{s_{in} \in S_{in}}{\text{argmax}}(b_{in}(s_{in}))$ et $s^*_{ep} = \underset{s_{ep} \in S_{ep}}{\text{argmax}}(b_{ep}(s_{ep}))$. Le scénario décrit dans la table 2 ne montre pas les actions concernant l'exécution des tâches, mais plutôt comment le modèle intuitif est capable de détecter un changement dans les préférences de l'humain seulement en observant ses actions.

Les Q-valeurs pour ce scénario sont basées sur des MDP de déplacement, elles incluent la valeur de l'action depuis chaque position possible afin de se déplacer vers chacune des tâches.

La partie de l'interaction décrite souligne les différents comportements adoptés par le robot pendant sa coopération. Au début du scénario, l'état de croyance du modèle épistémique est initialisé avec une préférence *unknown* pour toutes les tâches. Le modèle épistémique vérifie les préférences de l'humain concernant toutes les tâches dans les étapes (1 à 5). Ensuite, le modèle passe au mode intuitif. On peut constater à l'étape 8 que la Q-valeur de l'action de l'humain n'était en faveur d'aucune de ses préférences, au contraire, il préférait une des tâches destinées au robot (tk_3 en position 24). L'état de croyance sur les préférences de l'humain indique à la fin

	Interaction épistémique:
1	$s^*_{ep} =< unknown, unkown, unkown, unkown, unkown >$
	Robot: « Should I do task tk_3? » Human: « yes! »
2	$s^*_{ep} =< unkown, unkown, unkown, to_do_by_robot, unkown >$
	Robot: « Should I do task tk_1? » Human: « no! »
3	$s^*_{ep} =< unkown, to_do_by_human, unkown, to_do_by_robot, unkown >$
	Robot: « Should I do task tk_2? » Human: « yes! »
4	$s^*_{ep} =< unkown, to_do_by_human, to_do_by_robot, to_do_by_robot, unkown >$
	Robot: « Should I do task tk_0? » Human: « any! »
5	$s^*_{ep} =< to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, unkown >$
	Robot: « Should I do task tk_4? » Human: « yes! »
	$s^*_{ep} =< to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$
	Interaction intuitive:
6	$s^*_{in} =< 0, to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$
	human: south
7	$s^*_{in} =< 6, to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$
	human: south
8	$s^*_{in} =< 12, to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$
	human: south
	$s^*_{in} =< 18, to_do_by_any, unkown, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	Interaction épistémique:
9	$s^*_{ep} =< to_do_by_any, unkown, to_do_by_robot, unkown, to_do_by_robot >$
	Robot: « Should I do task tk_3? » Human: « no! »
10	$s^*_{ep} =< to_do_by_any, unkown, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	Robot: « Should I do task tk_1? » Human: « yes! »
	$s^*_{ep} =< to_do_by_any, to_do_by_robot, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	Interaction intuitive:
11	$s^*_{in} =< 18, to_do_by_any, to_do_by_robot, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	human: south
12	$s^*_{in} =< 24, to_do_by_any, to_do_by_robot, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	human: $do(tk_3)$
13	$s^*_{in} =< 24, to_do_by_any, to_do_by_robot, to_do_by_robot, done, to_do_by_robot >$
	human: north
	$s^*_{in} =< 16, to_do_by_any, to_do_by_human, to_do_by_robot, done, to_do_by_robot >$
	Interaction épistémique:
14	$s^*_{ep} =< to_do_by_any, unkown, to_do_by_robot, done, to_do_by_robot >$

Table 2: Une partie de l'interaction verbale/non-verbale pendant la réalisation de la mission coopérative.

de l'étape 8 une préférence *unknown* pour la tâche tk_1 et *to_do_by_human* pour la tâche tk_3. Ces changements conduiront le détecteur d'ambiguïté à détecter les ambiguïtés dans les préférences et le modèle mixte retournera alors au mode d'interaction épistémique pour désambiguïser la croyance sur les préférences.

Le scénario démontre qu'en présence d'une librairie bien définie de Q-valeurs, le modèle intuitif est capable de détecter les ambiguïtés (s'il en existe) entre les préférences données par l'humain (en répondant aux requêtes) et ses préférences estimées lors de l'exécution. Une telle librairie assure le bon fonctionnement du modèle mixte verbal et non-verbal.

4 Conclusion

Nous présentons dans cette thèse plusieurs modèles décisionnels pour différents problèmes d'Interaction Homme-Robot, basés sur des Processus de Décision Markoviens Partiellement Observable (POMDP), des Processus de Décision Markoviens (MDP) et des Chaînes de Markov Cachées (HMM).

Dans la version complète de la thèse (écrite en Anglais), nous avons présenté dans une première partie : une introduction sur l'IHR et la forme d'intelligence qui est attendue de la part de robots compagnons interactifs. Ensuite, nous donnons une présentation rapide des modèles, approches et architectures utilisés dans la littérature pour modéliser ou planifier des systèmes robotisés pour des domaines d'interaction homme-robot. Enfin, nous étudions les travaux existants sur le sujet de robots compagnons et la prise de décision pour l'IHR.

Une deuxième partie contient nos quatre contributions avec leur motivations, modèles et expérimentations, simulations ou scénarios appliqués sur des robot physiques.

Enfin, le manuscrit se termine par une conclusion et donne des perspectives pour les travaux futurs.

Part I

Introduction

Chapter 1

Introduction

This thesis addresses the design of decision models for companion robots. Indeed, the introduction of robots in our daily lives have raised many challenges for autonomous robot systems. In addition, it lead to the birth of a new research domain called Human-Robot Interaction.

We are interested in providing companion robots with decision models that allow them to interact with humans in a natural and acceptable manner. Companion robots should be able to understand humans, adapt to their existence and respect their well-being and desires.

1.1 Motivation

A rising number of robotic applications require robots to be involved in the human environment. Humans have used machines as tools that they can manipulate. However, a better capacity is expected from an intelligent robot, it should be able to act as a partner to the human instead of being just a tool controlled by him.

Applications concerning companion robots vary according to the type of interaction with the human. A companion robot can act alone in the environment while being aware of the existence of the human and his needs, for example, achieving undesired tasks on his behalf [Cirillo *et al.*, 2009a]. It can also assist the human while achieving his own task, in this case, the robot should detect the need of assistance and offer the necessary information to the human that will help him achieving his task. Such interaction is mostly useful in companion robots for elderly people (to remind them of their daily activities) [Pineau *et al.*, 2003, Boger *et al.*, 2005, Duong *et al.*,]. Recently, a lot of research interest is focusing on applications where the human and the robot collaborate together to achieve a common task. In such interaction, the human and the robot form a team while each of them is responsible of his own decisions toward the success of the common task [Hoffman et Breazeal, 2008, Sisbot *et al.*, 2010].

When sharing a task with its partner, the companion robot should be able to behave according to the situation of the interaction. It is not sufficient for the robot to be reactive to what happens in the environment, it should also be able to behave differently as in inducing a reaction from the human. In this subject, [Johnson *et al.*, 2010] introduces the concept of "coactivity".

A coactive behavior allows the robot not only to execute its part of the work but also encourage the joint activity of the human.

There are a lot of difficulties facing the decision making for a robot in presence, interaction or collaboration with a human [Klein *et al.*, 2004]. A lot of efforts engaging in the subject of interactive robotics and its applications, however, we are still far from a robust system for companion robots. We will address in this thesis questions concerning how to understand the accompanied human and how to act and behave accordingly.

1.2 Outline

This thesis is divided into two main parts. We first give an overview of existed theoretic models in the literature and how they were employed to establish interactive robot models (Part II). Then we contribute by introducing several decision models for a companion robot in different conditions (Part III). We finally conclude and we present future research perspectives (Part IV). The remainder of this thesis is thus organized as follows:

State of the art

Chapter 2 In this chapter we explain the problem of interactive companion robots. It includes definitions related to the aimed subject and presents the challenges facing powerful and practical decision models for robots that share humans their daily lives.

Chapter 3 We present in this chapter models, approaches and architectures that were used in the literature to model or plan robotic systems for Human-Robot Interaction domains. The chapter presents classical approaches: Belief-Desire-Intention, Hierarchical Task Networks, Bayesian Networks and Markovian models. It ends with a comparison of the advantages and disadvantages of using such models for companion robots.

Chapter 4 We conclude the state of the art with an overview of related work in the literature concerning the subjects of human intention recognition and planning for interactive robots.

Contributions

Chapter 5 Our first contribution concerns a decision model that respects the human intention. This contribution proposes to build a library of human actions values that holds a value for each pair (human action, human intention). Those values are then integrated in a Partially Observable Markovian Decision Process which will give the model the ability to infer the human intention while observing his actions and make decisions accordingly.

Chapter 6 The contribution described in this chapter is motivated by a companion robot that chooses a type of behavior according to the progress in achieving the collaborative shared task and an estimated level of the human's engagement. We show that the proposed

decision model allows the companion robot to choose whether to behave normally or instead be coactive to incite the human's joint activity.

Chapter 7 This contribution addresses two problematics. First, establishing a system for a companion robot that is capable of choosing between different types of interaction according to the human's needs. Second, modeling the system using a framework that outperforms Partially Observable Markovian Decision Processes for large-scale problems. We contribute in this chapter with a unified model of Human-Robot Multi-type Interaction. The objective is to observe the human's behavior and try to predict/estimate the human's intention/need and therefore react appropriately (assist, cooperate, collaborate, ...).

Chapter 8 Our last contribution in this thesis combines different approaches to create a decision model for a cooperative robot. The proposed model is inspired by verbal interaction (queries) to reveal human preferences combined with non-verbal approaches (Chapters 5 and 7). A mixed model that allows the robot to infer the human preferences using verbal interaction and then intuitively reinforce the inferred preferences by observing the human actions while executing the tasks. We show that the model is able to switch between verbal and non-verbal interactions according to possible detection of ambiguity about the human preferences.

Conclusion In Chapter 9, we conclude this thesis by summarizing our contributions and draw some lines for future research.

Part II

State of the Art

Chapter 2

Human-Robot Interaction and Companion Robots

Contents

2.1 Human-Robot Interaction 31
2.1.1 HRI Paradigms 33
2.1.2 Robots Autonomy in HRI 33
2.2 Companion Robots 34
2.2.1 Inspiration from Human-Human Companionship 35
2.2.2 Types of Interaction 36
2.2.3 Types of Behavior 37
2.3 Uncertainties in HRI and Companion Robots Systems 39
2.3.1 HRI Environments 39
2.3.2 Understanding Human Intentions 40
2.4 Discussion 41

This chapter introduces definitions related to the subject of this thesis and the research fields of companion robots and Human-Robot Interaction. We discuss the expected level and form of intelligence in interactive companion robots while presenting several concepts that will be the base of the contributed work, e.g. robots behavior and their different possible types of interaction with the human. Finally, we discuss the challenges that faces the decision making of such robots as for human intention recognition and planning in uncertain environments.

2.1 Human-Robot Interaction

Human-Robot Interaction (HRI) is the study of interactions between humans and robots. It is a multidisciplinary research area with continuous contributions from different fields like engineering (electrical, mechanical, industrial and design), computer science (Human-Computer

Interaction, Artificial Intelligence, robotics, natural language understanding and computer vision), social sciences (psychology, cognitive science, communications, anthropology and human factors) and humanities (ethics and philosophy) [Feil-Seifer et Matarić, 2009].

Human-Computer Interaction (HCI) offers a rich resource for research and design in HRI. One of the main shared disciplines between HCI and HRI is the study of human factors. Much has been learned in the last three decades about how people perceive and think about computer-based technologies, about human constraints on interaction with machines, about the factors that improve usability and about the effects of technology on people and organizations. A great deal of these studies is applicable to robots [Kiesler et Hinds, 2004]. However, autonomous robots are a very different technology from desktop computers for several reasons, among them:

- Robots are mobile and they share physical space with humans.
- Robots have access to more information about the human and the environment.
- Being autonomous in real environments, robots have more control on their actions and decisions with important constraints (like time and resources).
- Autonomous robots have higher and more complicated levels of interaction with humans.

The HRI field studies how humans and robots interact and how best to design and implement robot systems that are capable of accomplishing interactive tasks in human environments. Some of its possible application domains are: entertainment, personal assistants, museum guidance, health-care, space exploration and rescue.

The original benchmarks for HRI were proposed by Isaac Asimov in his short story "Runaround" part of his "I, Robot" collection [Asimov, 1950], where he described the three laws of robotics as:"

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey orders given to it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law."

These three laws of robotics respect the rules of safe interaction, knowing that close interactions between the human and the robot can risk injuring the human. To avoid this issue, until recently, manufacturing robots were closed in a safe and human-free zones where no humans allowed in the robot workspace while it is working.

Artificial Intelligence (AI) studies the design of intelligent agents. An intelligent agent is a system that acts intelligently: it perceives its environment and uses this perception, in addition to prior knowledge and experiences, to take actions that maximize its chances of success in reaching

its goals [Poole *et al.*, 1998, Russell et Norvig, 2003]. Nowadays and with the advancements of Artificial Intelligence, autonomous robots could eventually have more intelligent behaviors, planning their actions in uncertain environments. These new capabilities will allow robots to work in close distance with humans keeping safety as a primer issue in addition to efficiency.

2.1.1 HRI Paradigms

In [Breazeal, 2004], they classified the field of HRI into four interaction paradigms. These are the following:"

- robot as tool;
- robot as cyborg extension;
- robot as avatar;
- robot as sociable partner (companion robots).

Each is distinguished from the others based on the mental model a human has of the robot when interacting with it. In the first paradigm, the human views the robot as a tool that is used to perform a task. The level of robot autonomy varies from complete tele-operation, to a highly self-sufficient system that need only be supervised at the task level. In the second paradigm, the robot is physically merged with the human to the extent that the person accepts it as an integral part of their body. In the third paradigm, the person projects himself through the robot in order to communicate with another from far away. The robot provides a sense of physical presence to the person communicating through it, and a sense of social presence to those interacting with it. The last paradigm speaks to the classic science-fiction fantasy of an artificial being. Interacting with it is like interacting with another socially responsive creature that cooperates with us as a partner."

2.1.2 Robots Autonomy in HRI

The relation between the human and the robot is partly defined with the level of the robot's autonomy. In robotics, there are several levels of autonomy that varies between being remotely tele-operated by the human and being fully autonomous:

- Adjustable/Sliding autonomy which refers to the ability of autonomous systems to operate with dynamically varying levels of independence, intelligence and control. Here, the system is able to incorporate human intervention when needed and to otherwise operate independently [Scerri *et al.*, 2004, Mouaddib *et al.*, 2010].
- Mixed-initiative refers to a flexible autonomy which support an efficient, natural interleaving of contributions by agents (human and robot) aimed at converging on solutions to problems [Allen *et al.*, 1999, Ferguson et Allen, 2007].

- Collaborative control is an approach that uses human-robot dialog (i.e., queries from the robot and responses, or lack of them, from the human), as the mechanism for adaptation. Collaborative control also allows robots to benefit from human assistance during perception and cognition and not just for planning and command generation [Fong, 2001].

Autonomy is a complex property in the HRI context. It is convenient for a robot to have a degree of autonomy when it is designed to stand in for a human in a given situation. Also, autonomy can speed up applications for HRI by not requiring human input. However, full autonomous systems are not the answer for sociable robots and they can lead to undesirable behavior [Feil-Seifer et Matarić, 2009, Johnson *et al.*, 2010]. Autonomy might lead to less efficient robots in social environments. First, with the increase of autonomy, the robot has less dependence on the human, but the human has more dependence on the robot, because the robot becomes the sole owner of certain information and decisions. Second, autonomy cannot help robots to overcome unexpected events (failures) when they occur, however, teamwork can. Third, and not least, in situations where robots have physical contact with the human, the latter must clearly retain authority. For example, rehabilitation should terminate if the human is in pain. For those reasons and more, adjustable autonomy allows an appropriate adjustment of both authority and autonomy. Moreover, [Johnson *et al.*, 2010] introduces coactive behavior (more details in Section 2.2.3) which is based on collaborative control allowing both parties to participate, in different ways, during interaction.

This thesis will be concentrated on companion robots (the fourth interaction paradigm), where HRI is viewed from the perspective of designing sociable robots that interact with people in a human-like way. There are a growing number of applications for robots that people can engage with as capable creatures or as partners rather than tools, yet little is understood about how to design robots to interact this way.

2.2 Companion Robots

HRI and intelligent robotic systems are important disciplines in the study of companion robots. Moreover, companion robots have a non-limited possibilities of interaction with humans and should respect social constraints whether in their behavior, appearance, or cognitive skills. [Dautenhahn, 2007] defined a companion robot as: "a robot that (i) makes itself 'useful', i.e. is able to carry out a variety of tasks in order to assist humans, e.g. in a domestic home environment, and (ii) behaves socially, i.e. possesses social skills in order to be able to interact with people in a socially acceptable manner."

The most important applications include companion robots for elderly and people with special needs in their homes or in assisted living facilities. Therefore, the design of a companion robot addresses deep issues into the nature of human social intelligence, as well as sensitive ethical issues to be able to interact with such vulnerable people. The roles that the companion robot can adopt are: to be effective machines performing tasks on human's behalf, assistants, companions

or even friends. All this depends on the human possible incapabilities or preferences. More information about the human desires and constraints would more probably lead the companion robot to offer the good type of companionship or help when needed. For example, if the human has the intention of doing a certain task like calling a friend, he might be able to achieve the task himself, however, he might need assistance in finding or remembering the phone number. In the first case, while the human is making his phone call, the robot can help the human by achieving some other tasks (like cleaning the room) on his behalf. In the second case, an assistant robot should offer assistance to the human to find/remember the phone number.

2.2.1 Inspiration from Human-Human Companionship

Many, in the field of HRI, study how humans interact between them in different situations and use those studies as an inspiration to how robots should interact with humans [Green *et al.*, 2008]. The study of companion robots behavior should be motivated by human-human companionship. As an example: to be able to build a robot system that cooperates with a human partner in an apartment, a similar model of human beings cooperating with each other is represented for the same type of cooperation. Let be, Bob and Ann are committed to the mission of cleaning their apartment. This mission can be divided into a number of tasks like cleaning bedroom, cleaning living room, doing laundry and washing dishes. If Ann observes Bob entering the bedroom and Ann believes in Bob's commitment to the mission, then Ann will believe also that he is planning to clean the bedroom and she will decide to do one of the other tasks like washing dishes. If Bob meanwhile finishes his task and observes Ann standing near the sink, Bob will believe that Ann is doing the dishes and he will decide to do one of the other tasks. This continues until all tasks are done. In conclusion, it is sufficient, in most cases, to observe the partner's actions in order to know what task he is trying to achieve.

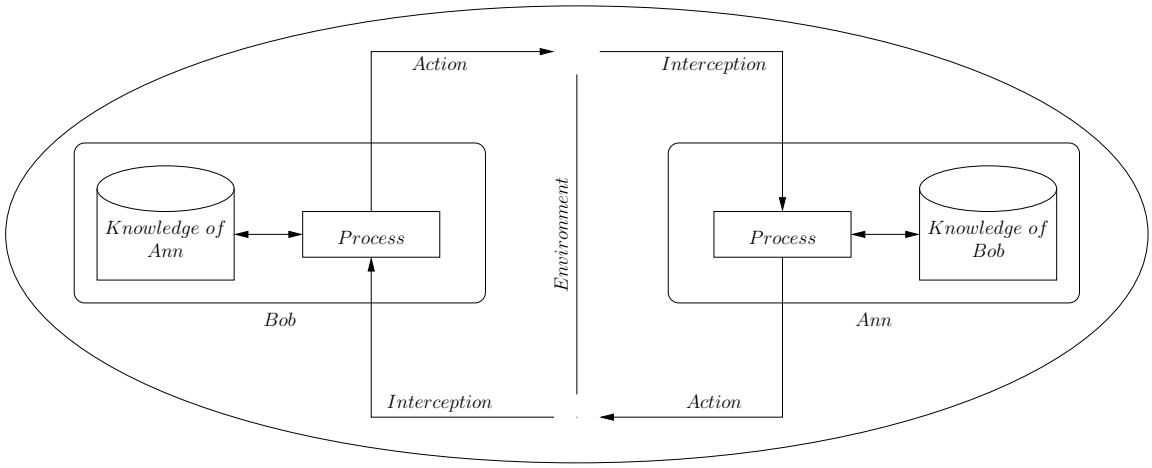


Figure 2.1: A model of human-human cooperation.

Figure 2.1 presents a simple model of human-human cooperation regarding the example of Bob and Ann. Each partner tries to build a knowledge of the other by observing him and, depending on this knowledge, attempt to proceed with the best interest of the cooperation.

2.2.2 Types of Interaction

Some domains in the literature discuss robot companions that help their human partners without necessarily interacting closely with them. Those robots can be seen as servant robots that are aware of the existence of a human partner in the same environment. Servant robots can be of help to a human by doing undesired tasks like cleaning. In [Cirillo *et al.*, 2009a] the robot attempts to recognize the human plan in order to cooperate by achieving its own tasks without disturbing the human plan.

Some other approaches were dedicated for robots as assistants to humans. Assistant robots normally have direct relations with people around them. They are supposed to detect a need of assistance from their human partner and offer the information that facilitate the task for him. For example, an autonomous mobile robotic assistant can provide elderly people in nursing homes with reminders about their daily activities [Pineau *et al.*, 2003]. Also, a cognitive assistive system is presented to help people with advanced dementia in their daily living activities [Boger *et al.*, 2005]. In the latter, they detailed a model for a hand-washing assistant which monitors the persons progress in his activity and suggests guidance in case of an unusual observed behavior. A similar work presents a system that learns a model of the house occupant's activities of daily living through observing what the occupant usually does during the days, then monitors the person's current activity to detect any abnormality and alert the caregiver [Duong *et al.*,].

Recently, some research interest is increasing about applications where a robot collaborates with a human partner to accomplish a common task. In Human-Robot Collaboration the concepts of master-slave or assistant relationship exist no more. The robot and the human act jointly each responsible for his decision to accomplish their common task as a human-robot team [Hoffman et Breazeal, 2008]. Acting jointly has been discussed more particularly for planning team members actions in multi-agent systems [Cohen et Levesque, 1991, Levesque *et al.*, 1990].

To formalize all the previously mentioned examples of interaction, the following presents our definition of the main three types of interaction for a companion robot sharing a human partner daily living activities:

Cooperation a robot can cooperate with a human by doing a task on his behalf to save him time or effort. The robot should be capable to do the task alone (cleaning carpet or the dinner table), and should not disturb the human by respecting the human's preferences and interests.

Assistance a companion robot can assist a human by guidance (typically spoken guidance). The robot should be able to detect the human's need of assistance and offer the best

guidance that will enable the human to complete his task. For example by accessing a manual of how to run the dishwasher the robot can assist the human by reading the steps. This type of interaction also covers cases like robot assistants for elderly people with dementia.

Collaboration a task that needs the participation of the human and the robot to achieve it (typically physical action) requires a collaboration between them. A robot companion should be of answer to the human when showing an interest in a collaboration task.

2.2.3 Types of Behavior

A robot in a human environment should choose its actions with respect to, both the human and its own, circumstances and goals. It should be flexible with dynamic environments and dynamic goals. Furthermore, it should learn from experience and make appropriate choices despite perceptual, computational and time limitations.

An intelligent and flexible agent can adopt different behaviors according to different situations. The following presents some chosen definitions and a simplified example to help describe possible behaviors of an agent while interacting with a human.

Definition 6 *"Reactive" as defined in the American Heritage Dictionary of the English Language:*

1. *Tending to be responsive or to react to a stimulus.*
2. *Characterized by reaction.*

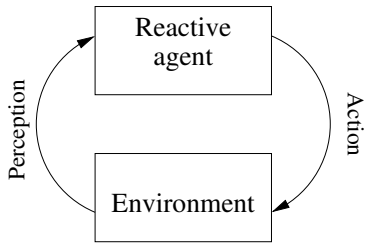


Figure 2.2: A reactive agent behavior.

Definition 7 *"Proactive" as defined in the American Heritage Dictionary of the English Language and Collins English Dictionary:*

1. *Acting in advance to deal with an expected difficulty; Anticipatory.*
2. *Tending to initiate change rather than reacting to events.*
3. *In psychology (learning theory): As an opposition to reactive.*

Definition 8 *"Deliberative" as defined in WordNet:*

1. *involved in or characterized by deliberation and discussion and examination;*

Definition 9 *"Coaction" as defined in the American Heritage Dictionary of the English Language:*

1. *An impelling or restraining force; a compulsion.*
2. *Joint action.*

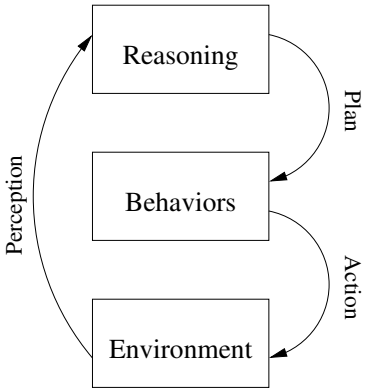


Figure 2.3: Reasoning about behavior.

Let's take for example the bar-robot that fills client's glasses with water. To be efficient in its interaction with clients, the bar-robot should be able to adapt its behavior with respect to its main job which is serving water and the situation of the human and his glass. A reactive robot mostly reacts to possible situations or possible human actions (Figure 2.2). If the human puts his empty glass in front of the robot, the robot will refill the glass with water. A robot that reasons about its behavior after perceiving the human's action is not considered as reactive (Figure 2.3). A proactive robot would not only react to situations, but maybe initiate the action by extending its hand with the water bottle towards a human who is standing near the bar with an empty glass. Being coactive completes proactivity, a proactive action is an anticipatory action which is acted in advance for certain anticipated future. However, coactive action is a guidance action which drives the other to act with us to actually achieve the anticipated future. Indeed, if the robot extended its arm and the human responded by extending its arm too, the human and the robot should act coactively to make it possible to fill the glass with water. A deliberative (communicative) robot is more characterized by communicating and deliberating plans with the human. It will be more verbally active, as in asking the human if he wants to fill his glass.

It is very important that the robot reasons carefully about its type of behavior and chooses, carefully as well, the actions to represent those behaviors. It is not desirable that robot actions cause any kind of confusion to the human.

2.3 Uncertainties in HRI and Companion Robots Systems

The environment of a companion robot, whether a hospital, an office, a home or an assisted living facility, is a cluttered and uncontrolled environment. To interact with its human partner and be operational in his environment, the robot must be provided with functionalities and capabilities to support it in surviving and succeeding in a complex, dynamic and uncertain environment.

During the day, the companion robot is supposed to perform operations involving navigation, communication, cooperation, assistance, etc. Some needed information to plan its decision might be accessible to the robot by perception, as for detecting objects, detecting human partner position or human vocal expressions. However, analyzed information from hardware devices and sensors are not totally reliable. Inevitable amount of doubt and uncertainty is resulted from such analysis.

Moreover, in order to assist or collaborate with a human (Section 2.2.2) the robot should choose its actions in a way that minimizes the expected cost of completing the human's task. However, the latter is not directly observable by the companion robot, which makes the problem of quickly inferring the human's goals/intentions from perceived information critically important.

Therefore, a robot is rarely able to describe the real environment "the true world", it acts only on its "beliefs of what is a true world".

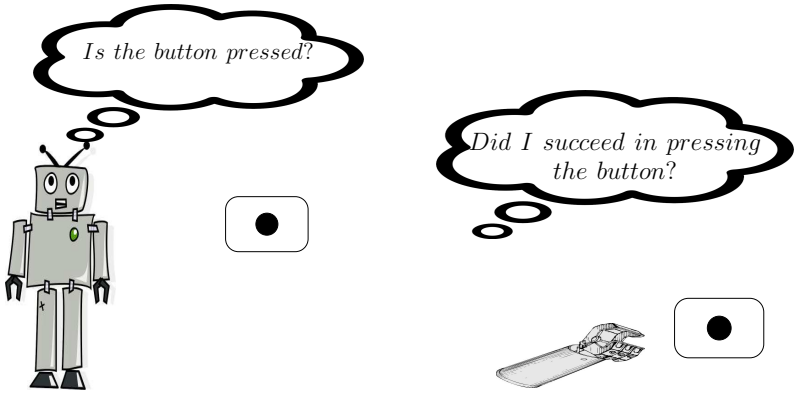
Human intention is one of the major topics of uncertainty in HRI domains but not the unique one. Human environments are meant first and foremost for human occupants. Therefore companion robots need to adapt and develop the ability to handle the uncertain and the dynamic nature of these environments and the uncertainties about the human's goals and intentions.

2.3.1 HRI Environments

Several reasons might prevent a companion robot from maintaining an exact image of the real environment, among them: intentional and unintentional changes in the environment state that might be caused by the human or the robot actions.

Figure 2.4 shows a representative example of a situation where the robot is uncertain about the state of a button in the environment. The robot in Figure 2.4(a) is not sure if the button is pressed or not. Furthermore, robots also are not sure if their actions were well applied and caused the good intended outcome. Figure 2.4(b), shows that the robot, after trying to press the button, is still not sure if the button is pressed or not.

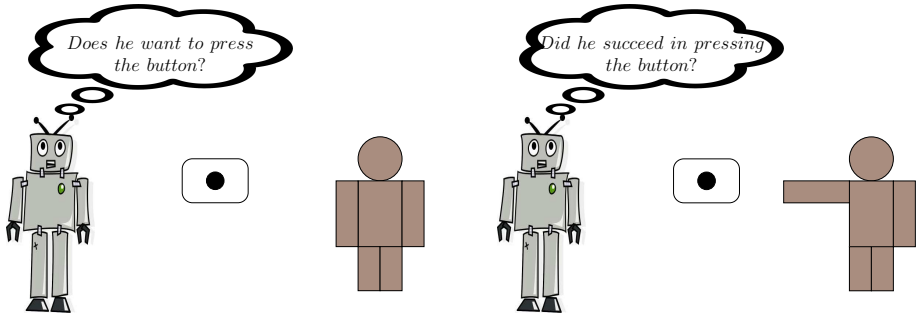
In HRI environments, the robot faces as well uncertainties related to the human. Figure 2.5(a) shows how the robot is uncertain about the human intention to press the button. In Figure 2.5(b), the robot observed the human trying to press the button but is not sure about the outcome of the human action.



(a) Uncertainty about environment states.

(b) Uncertainty about action's effects.

Figure 2.4: Different sources of uncertainty in robotic environments.



(a) Uncertainty about human intentions.

(b) Uncertainty about human actions effects.

Figure 2.5: Different sources of uncertainty in HRI environments.

2.3.2 Understanding Human Intentions

Definition 10 *The word "intention" as defined in the Fartex Trivia dictionary, Webster dictionary and Encarta respectively:*

1. *Intent implies a sustained unbroken commitment or purpose, while intention implies an intermittent resolution or an initial aim or plan.*
2. *An anticipated outcome that is intended or that guides your planned actions.*
3. *Something that somebody plans to do. The quality or state of having a purpose in mind.*

Those definitions emphasize two features of an intention; an intention has an *aim/goal* and a *plan*.

Intention/plan/activity recognition [Schmidt *et al.*, 1978] is a very expanded research subject as it is needed in many AI domains. The study of human intention concerned HCI community before [Babaian *et al.*, 2002, Hui et Boutilier, 2006, Tambe, 2008, Yorke-Smith *et al.*, 2009], and now it is one important subject concerning the HRI planning community. Without being able to

detect/share the intention of one another, the robot and the human will not be able to interact correctly.

The ability to appropriately understand the intention (or internal state) of the other is important for both the human and the robot in order to coordinate and synchronize their behavior. It allows them to work effectively, to correct misunderstandings before success is compromised, and to compensate for unexpected difficulties before a failure becomes manifest. Similarly to the Bob and Ann example, a companion robot during his cooperation with a human partner should be able to predict his partner's intentions. Using that information, the robot should be able to make better decisions for the mission and for the comfortability of its human partner. In another example, an assistant companion robot is incapable of offering assistance if it lacks the ability of detecting its human partner intention.

During social interaction with a robot partner, both parties need to appropriately convey their intended meaning to the other. There are different ways to communicate intention: verbal [Breazeal, 2004, Schmidt-Rohr *et al.*, 2008a], gestures [Jenkins *et al.*, 2007, Schmidt-Rohr *et al.*, 2008a] or implicit [Nguyen *et al.*, 2005, Fern *et al.*, 2007, Taha *et al.*, 2008]. An important role of Natural Language Processing and Image Processing is applied for verbal and gesture analysis to detect and match them to possible intentions. However, an implicit intention communication needs huge and precise knowledge of the partner in order to be able to implicitly match situations or any possible action to a possible intention.

Humans have various ways of doing things, some people are generally committed and others have more tendency to stall. It is one situation that risks a misunderstanding between the companion robot and the human. Moreover, human commitment to his goal/intention might change or be interrupted [Levesque *et al.*, 1990]. All those possibilities and uncertainties must be taken into account when the companion robot plans its actions.

2.4 Discussion

There are a lot of difficulties that faces the decision making of a robot in presence, interaction, collaboration with a human [Klein *et al.*, 2004]. Even though there are many efforts in the domain of humanoid robotics as companions, there is still no powerful theory or robust system for robot's task planning. A companion robot system, with respect to the constraints and limitations of operational resources, must be able to:

- cope with dynamic environments,
- cope with all kinds of uncertainty,
- understand its human partner and his possible needs,
- choose the way to interact with its partner,
- choose the best type of behavior,

- make decisions that does not harm the human in any way and respects the human and his desires.

In the following chapter (Chapter 3) we will be presenting theoretic approaches that have been used in the literature by the HRI community. In Chapter 4 we will show how those theoretic approaches were used in representing HRI dynamic environments, understanding perceived information about the human to infer his intentions, and finally planing robot decisions.

Chapter 3

Background on Theoretic Models

Contents

3.1 Belief-Desire-Intention Architecture 43
3.2 Hierarchical Task Networks 45
3.3 Bayesian Networks and Dynamic Bayesian Networks 46
3.4 Markovian Models 49
3.4.1 Markov Chains 49
3.4.2 Hidden Markovian Models 49
3.4.3 Markovian Decision Processes 50
3.4.4 Partially Observable Markovian Decision Processes 52
3.5 Discussion 56

This chapter includes a quick presentation of models, approaches and architectures that are used in the literature to model or plan robotic systems for Human-Robot Interaction domains. The chapter begins with presenting classical approaches: Belief-Desire-Intention architecture, Hierarchical Task Networks, Bayesian Networks and Dynamic Bayesian Networks followed by a detailed presentation of Markovian models: Markov chains, Hidden Markovian model, Markovian Decision Processes and Partially Observable Markovian Decision Processes. The chapter ends with a discussion including comparisons and relations between the presented models.

3.1 Belief-Desire-Intention Architecture

A Belief-Desire-Intention (BDI) architecture of practical reasoning provides a process of deciding which action to perform to reach goals. Practical reasoning involves two important processes: first, deciding what goals to achieve, then, how to achieve them.

The reasoning process in a BDI agent is shown in Figure 3.1. The BDI architecture includes seven main components [Weiss, 1999]:

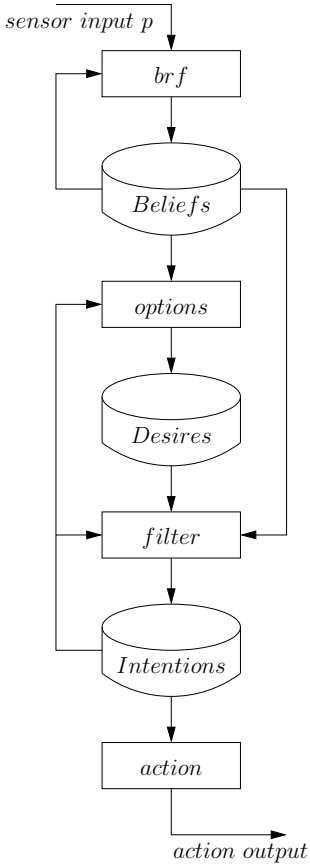


Figure 3.1: A BDI architecture.

- A set of beliefs ($Beliefs$), representing the agent's information about its current environment (including itself and other agents). Using the term belief rather than knowledge recognizes that what an agent believes may not necessarily be true.

- A belief revision function (brf), which takes a perceptual input P and the agent's current beliefs, and on the basis of these, determines a new set of beliefs.

$$brf : \wp(Beliefs) \times P \rightarrow \wp(Beliefs)$$

- A set of current desires ($Desires$), representing possible courses of actions available to the agent They represent objectives or situations that the agent would like to accomplish.

- An option generation function ($options$), which determines the options available to the agent (its desires) on the basis of its current beliefs about its environment and its current intentions.

$$options : \wp(Beliefs) \times \wp(Intentions) \rightarrow \wp(Desires)$$

- A set of current intentions ($Intentions$), representing the agent's current focus. An intention is a desire to which the agent has committed to trying to bring about. The set of active intentions must be consistent.

- A filter function ($filter$), which represents the agent's deliberation process and determines the agent's intentions on the basis of its current beliefs, desires and intentions.

$$filter : \wp(Beliefs) \times \wp(Desiers) \times \wp(Intentions) \rightarrow \wp(Intentions)$$

- An action selection function ($execute$), determines an action to execute on the basis of current intentions.

$$execute : \wp(Intentions) \rightarrow A$$

The state of a BDI agent at any given moment is a triple (B, D, I), where $B \subseteq Beliefs, D \subseteq Desiers$ and $I \subseteq Intentions$. The agent decision function $action : P \rightarrow A$ is defined in Algorithm 2.

Algorithm 2: function $action : P \rightarrow A$

Input: $p \in P$
Output: action to execute A

```
1 begin
2 |  B = brf(B, p);
3 |  D = options(D, I);
4 |  I = filter(B, D, I);
5 |  return execute(I);
6 end
```

The BDI model gives a clear functional decomposition. The main difficulty is knowing how to efficiently implement these functions. By design, the architecture does not have any lookahead deliberation or forward planning. The decided actions are planned for one step ahead.

3.2 Hierarchical Task Networks

Hierarchical Task Networks (HTN) is a planning paradigm which consists of finding a primitive decomposition of a given set of tasks (task network) [Ghallab *et al.*, 2004]. An HTN planner is provided with a set of tasks to be performed and a set of restrictions (often ordering constraints) that the tasks should satisfy. A plan is then formulated by repeatedly decomposing tasks into smaller subtasks until primitive, executable tasks are reached.

In the following, a formal definition of an HTN planning is presented as indicated in [Sohrabi *et al.*, 2009]:"

Definition 11 ***(HTN planning problem)***

An HTN planning problem is a 3-tuple $P = (s_0, w_0, D)$ where s_0 is the initial state, w_0 is a task network called the initial task network, and D is the HTN planning domain which consists of a set of operators and methods.

Definition 12 ***(Domain, Method)***

A domain is a pair $D = (O, M)$ where O is a set of operators and M is a set of methods. An operator is a primitive action, described by a triple $o = (name(o), pre(o), eff(o))$, corresponding to the operator's name, preconditions and effects.

A method m, is a 4-tuple $(name(m), task(m), subtasks(m), constr(m))$ corresponding to the method's name, a non-primitive task and the method's task network, comprising subtasks and constraints. Method m is relevant for a task t if there is a substitution σ such that $\sigma(t) = task(m)$. Several methods can be relevant to a particular non-primitive task t, leading to different decompositions of t. An operator o may also accomplish a ground primitive task t if their names match.

Definition 13 ***(Task, Task Netword)***

A task consists of a task symbol and a list of arguments. A task is primitive if its task symbol is an operator name and its parameters match, otherwise it is non-primitive.

A task network is a pair $w = (U, C)$ where U is a set of task nodes and C is a set of constraints. Each task node $u \in U$ contains a task t_u . If all of the tasks are primitive, then w is called primitive; otherwise it is called non-primitive.

Definition 14 ***(Plan)***

$\pi = o_1 o_2 ... o_k$ is a plan for HTN planning program $P = (s_0, w_0, D)$ if there is a primitive decomposition, w, of w_0 of which π is an instance."

Figure 3.2 shows a plan tree for a delivery problem with a root non-primitive task *deliver_objectAt*. The planner uses *methods* to decompose all the non-primitive tasks until reaching one of the three primitive tasks ($load_truck, unload_truck, drive_truck$).

3.3 Bayesian Networks and Dynamic Bayesian Networks

Bayesian networks are probabilistic graphical models representing joint probabilities of a set of random variables and their conditional independence relations.

A Bayesian Network (BN) consists of the following [Jensen, 2001]:"

- A set of variables and a set of directed edges between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables together with the directed edges form a Directed Acyclic Graph (DAG). A directed graph is acyclic if there is no directed path $A_1 \rightarrow \ldots \rightarrow A_n$ so that $A_1 = A_n$.

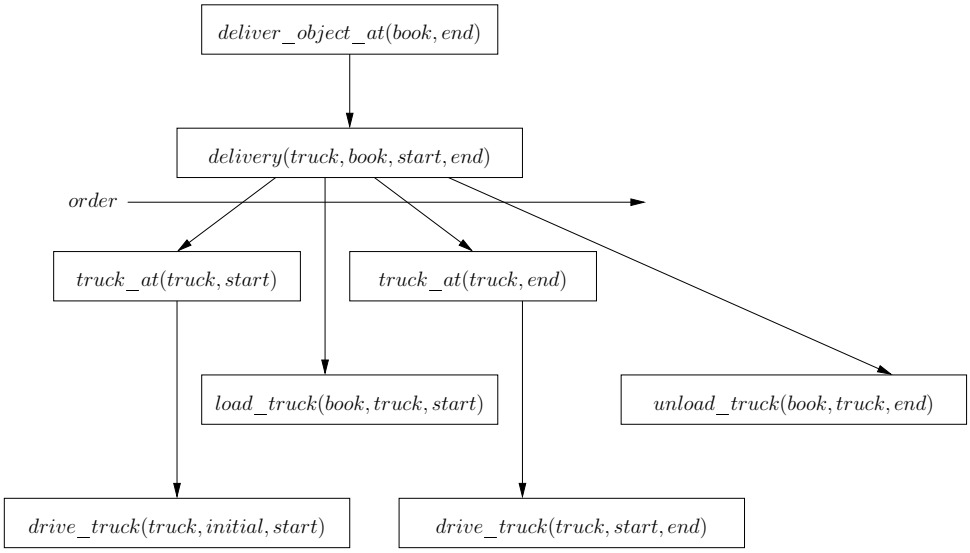


Figure 3.2: An HTN decomposition tree for a delivery problem.

- To each variable A with parents $B_1, \ldots, B_n$, a conditional probability table $P(A|B_1, \ldots, B_n)$ is attached. In case A has no parents, the table reduces to the unconditional probability table $P(A)$."

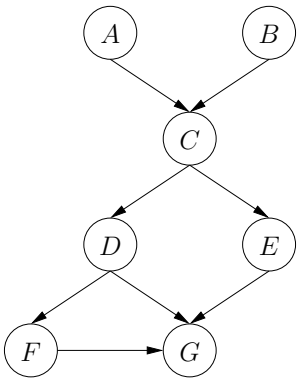


Figure 3.3: A simple Bayesian Network represented as a Directed Acyclic Graph (DAG).

Figure 3.3 shows a simple BN represented as a DAG. To define the whole structure of a BN, the probability distribution of all nodes that have parents must be specified given the prior probability of the root nodes. In Figure 3.3, knowing $P(A), P(B)$, the probabilities to specify in this BN are $P(C|A, B), P(E|C), P(D|C), P(F|D)$ and $P(G|D, E, F)$. Those probabilities are calculated using probability theory and the Bayes' rule.

The *Bayes' rule* for computing posterior probability $P(X|Y)$ (Equation 3.1) is calculated given the prior one $P(X)$, and the likelihood $P(Y|X)$ that Y will materialize if X is true:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.1}$$

Bayesian Networks are a way to represent the belief about the variables of a system and the relationships that exist between these various variables. The purpose of such representations is to infer some belief about some processes of events in the system. Static Bayesian Networks (SBNs) work with beliefs from a single time instant which make it impossible to model systems that evolve over time. As a result, Dynamic Bayesian Networks (DBNs) have been developed to overcome this limitation [Mihajlovic et Petkovic, 2001]. A DBN is made up of interconnected time slices of SBNs, and the relationships between two neighboring time slices follow the Markovian Property (see Section 3.4), which means that random variables at time $t+1$ may depend on variables at time t and possibly on other variables from the same time slice $(t+1)$. Figure 3.4 illustrates this approach where two time slices are interconnected by temporal relations, which are represented by the arcs joining particular variables from two consecutive time slices.

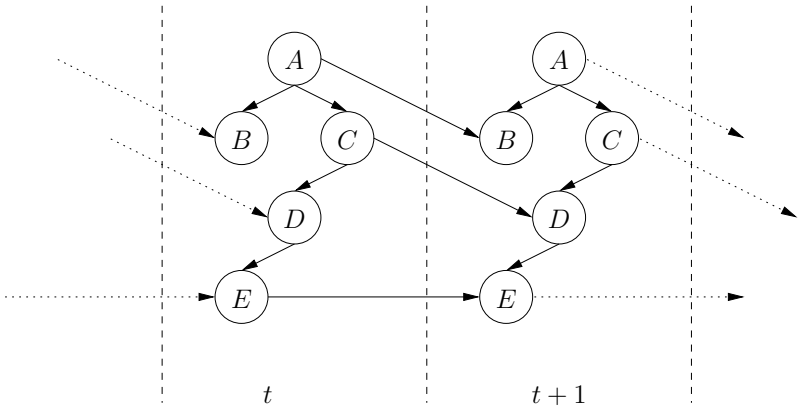


Figure 3.4: A generic Dynamic Bayesian Network structure consisting of 2 time slices, where t represents time.

DBN variables do not need to be directly observable. A DBN consists of probability distribution function on the sequence of T hidden-state variables $X = \{x_0, \ldots, x_{T-1}\}$ and the sequence of T observable variables $Y = \{y_0, \ldots, y_{T-1}\}$, where T is the time boundary for the given investigated event. This can be expressed by the following:

$$P(X,Y) = \prod_{t=1}^{T-1} pr(x_t|x_{t-1}) \prod_{t=1}^{T-1} pr(y_t|x_t) pr(x_0), \qquad \text{where,}$$

$pr(x_t|x_{t-1})$ is the transition probability distribution function that specifies time dependencies between the variables, $pr(y_t|x_t)$ is the observation probability distribution function that specifies dependencies of observation nodes regarding to other notes at time slice t and, finally, $pr(x_0)$ is the initial probability distribution in the beginning of the process.

DBNs provide a unified hierarchical probabilistic framework for sensory information representation, integration and inference over time [Mihajlovic et Petkovic, 2001, Li, 2005]. In addition, DBNs provide the ability to predict the influence of possible future actions through its temporal causality.

3.4 Markovian Models

In probability theory, a Markovian model is a stochastic model that assumes the Markov property.

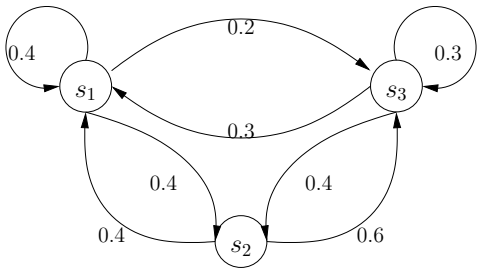
Definition 15 ***(Markov Property)***
A first order Markov property is satisfied when the system state at time $t+1$ depends only on its immediate past, which means the system state at time t.

$$P(s_{t+1}|s_0, s_1, \ldots, s_t) = P(s_{t+1}|s_t)$$

In the following, the state at time t will be denoted with s and the state at time $t+1$ with s'.

3.4.1 Markov Chains

A Markov Chain (the simplest of Markovian models) is composed of a set of states S and a transition function $T(s'|s)$ which gives the probability of passing from state $s \in S$ to state $s' \in S$. In a finite state space, a Markov Chain can be represented with a graph as shown in Figure 3.5. In this graph, the nodes represent the states and the arrows represent the transitions and their probabilities $T(s'|s)$. The transitions can similarly be presented in a matrix with probabilities of transitioning between any pair (s, s').



$$T = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.4 & 0.0 & 0.6 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

Figure 3.5: A Markov Chain graph with its transition matrix T.

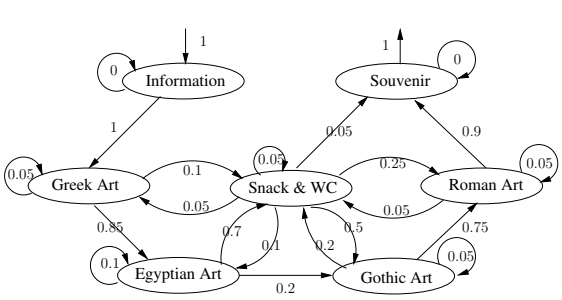
3.4.2 Hidden Markovian Models

A Hidden Markovian Model (HMM) [Rabiner et Juang, 1986] is a Markov chain for which the state is only partially observable. An HMM is composed of a set of states S, a set of observations Z, a transition function $T(s'|s)$ and an observation function $O(z|s, s')$. The latter gives the probability of observing z knowing that the system started in state s and ended in state s'.

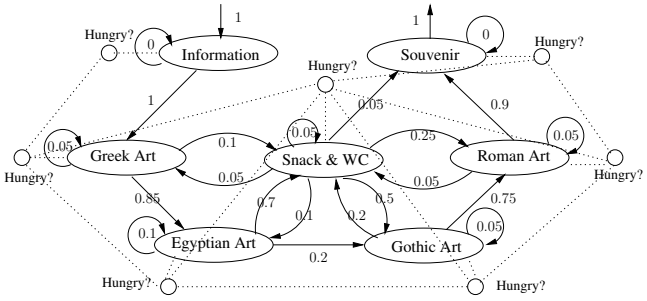
Figure 3.6 presents an example of an HMM problem. Considering a museum with four main exhibitions (Greek Art, Egyptian Art, Gothic Art and Roman Art) in addition to an information room, a souvenir room, a snack bar and rest rooms. Figure 3.6(a) presents statistical probabilities on the sequence that visitors usually take during their visit to the museum. It is presented as

a Markov chain and it shows that an important number of the visitors follows the sequence of exhibitions as the following: Greek, Egyptian, Gothic then Roman exhibitions with a break in the snack bar in the middle of the visit (after the Egyptian and before the Gothic exhibition). Other visitors, with lower percentages, follow the same visiting sequence while taking their break in the snack bar in different steps of the visit (after the Greek, Gothic or Roman exhibitions).

Assuming an intelligent system in the museum that needs to represent a visitor's sequence of the visit for reason of assistance or guidance and that a time step of the system represent one hour. At each time step t each visitor can be located in one of the seven possible locations shown as circles in Figure 3.6(a). In the following time step $t + 1$ the visitor moves to a next location with a probability indicated on the arrow between the corresponding circles. However, it is important for the intelligent system to know the level of hunger of the visitor, which will give an idea of the location from where the visitor will go to the snack bar to eat. Figure 3.6(b) describes this problem with a Hidden Markovian Model where the system state is represented by the location of the visitor which is an observable part of the state and the visitor's level of hunger (not, little or very hungry) which is the hidden non-observable part of the state.



(a) Museum visit Markov chain: probabilities on museum visit sequences.



(b) Museum visit hidden Markovian model: visitor "hungry?" hidden state.

Figure 3.6: Museum visit example.

Several well-known algorithms for HMMs exist [Rabiner et Juang, 1986]. For example, given a sequence of observations, the Viterbi algorithm will compute the most-likely corresponding sequence of states, the Forward algorithm will compute the probability of the sequence of observations and the Baum–Welch algorithm will estimate the starting probabilities, the transition function and the observation function of a hidden Markovian model.

3.4.3 Markovian Decision Processes

The most common representations for sequential decision models in decision-theoretic planning are Markovian Decision Processes.

Definition 16 *A Markovian Decision Process (MDP) is represented by a tuple* $\langle S, A, T, R$ *where:*

- *S is a finite set of states that represent the environment for an agent.*
- *A is a finite set of the agent's actions.*
- *$T : S \times A \to \prod(S)$ is a transition function.*
- *$R : S \times A \times S \to \Re$ is the reward function*

The set of states: A finite set of all states denoted $S = \{s_0, s_1, \dots\}$ includes all possible states representing the world. A state of the world at time t, represents all information of the world that are related to the problem at this time.

The set of actions: A finite set of possible agent actions, denoted $A = \{a_0, a_1, \dots\}$. At each time t, the agent effects an action a_t which has stochastic effects on the world state.

The transition function: A state transition probability distribution, $T(s, a, s') = Pr(s'|s, a)$ is the probability of transitioning from state s to state s' after doing action a, where $\sum_{s' \in S} T(s, a, s') = 1 \quad \forall (s, a)$.

The reward function: A mapping $S \times A \times S$ to the agent's immediate reward for making action a while being in state s and ending in state s'.

Definition 17 *An MDP policy π_{MDP} is a function $\pi : S \to A$, which associates an action to each MDP state.*

Definition 18 *A Horizon H is the number of actions (time steps) the system will take during its life time. The term infinite-horizon is used when H is infinite.*

A value function is defined to evaluate a policy by calculating its long-term expected reward. For an infinite-horizon problem the value function is defined as:

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \qquad \forall s \in S,$$

where γ is a discount factor and r_t is the reward at time t.

[Bellman, 1957] showed that the value function can be calculated using the *Bellman equation* (Equation 3.2).

$$V_t^{\pi}(s) = R(s, \pi_t(s)) + \gamma \sum_{s' \in S} T(s, \pi_t(s), s') V_{t-1}^{\pi}(s') \tag{3.2}$$

An optimal policy π^* is the policy that maximizes the long-term expected reward:

$$\pi^*(s) = \arg\max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V^*(s')\right] \text{ where,}$$

$$V^*(s) = \max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V^*(s') \right]$$

There are several algorithms to solve an MDP, classically Value Iteration [Bellman, 1957] and Policy Iteration [Howard, 1960] which are based on *Bellman equation*. The Value Iteration algorithm (Algorithm 3) consists in improving the value of each MDP state by using *Bellman equation*, where the state value at time t is calculated from its value at time $t-1$. The iteration stops when the difference between successive values of all states is less that a precision creterion ϵ. The complexity of this algorithm is $\mathcal{O}(|S^2||A|)$.

Algorithm 3: Value Iteration

Input : An MDP $\langle S, A, T, R$, a discount factor γ, a precision criterion ϵ.
Output: Optimal policy π.

Randomly initialize $V(s)$, $\forall s \in S$;
repeat
  $t = t + 1$;
  forall $s \in S$ **do**
    $V_t(s) = \max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{t-1}(s') \right]$
until $\max_{s \in S} |V_t(s) - V_{t-1}(s)| \leq \epsilon$;
$\forall s,\ \pi(s) = \arg\max_{a \in A} \left[R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_t(s') \right]$

Another value function called Q-value function represents a value for each pair (s, a) which corresponds to the value of taking the action a from the state s and then continuing according to the current policy π. This function is represented in Equation 3.3.

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^\pi(s') \tag{3.3}$$

Reinforcement Learning

If the agent has no knowledge about its environment, the transition and reward functions will be hard to calculate. In this case, the agent can directly learn its policy using reinforcement learning algorithms[Kaelbling *et al.*, 1996, Diuk et Littman, 2009, Szepesvari, 2010] as for Q-learning [Sutton et Barto, 1998] that does not require prior knowledge about the transition function.

3.4.4 Partially Observable Markovian Decision Processes

A **POMDP** relies on a probabilistic model that is represented by a tuple $\langle S, A, T, Z, O, R, b_0$ where: S, A, T and R are the same as in an MDP and:

- Z is a finite set of observations.

- $O : S \times A \times S \rightarrow \prod(Z)$ is a discrete probability distribution over Z. $O(s, a, s', z) = Pr(z|s, a, s')$ is the probability that transitioning from state s to state s' by doing the action a will produce the observation z, where $\sum_{z \in Z} O(s, a, s', z) = 1 \quad \forall (s, a, s')$.

- $b_0(s) = Pr(s_0 = s)$ is the probability of being in state s at time $t = 0$.

Given that the state is not directly observable, the agent instead maintains a belief distribution over S. b_0 is the initial state probability distribution and $b_t(s)$ is the probability that the system is in state s at time t, given the history of all observations/actions the agent received/affected and the initial belief state b_0:

$$b_t(s) = Pr(s_t = s|z_t, a_{t-1}, z_{t-1}, ..., a_0, b_0).$$

Knowing the last applied action a and the recent observation z, the agent calculates a new belief state $b'(s')$ by applying the belief update function [Cassandra *et al.*, 1994]:

$$\begin{aligned} b'(s') &= \tau(b, a, z) \\ &= \frac{Pr(z|s, a, s', b)Pr(s'|a, b)}{Pr(z|a, b)} \\ &= \frac{\sum_{s \in S} O(s, a, s', z)T(s, a, s')b(s)}{Pr(z|a, b)} \end{aligned} \tag{3.4}$$

where $Pr(z|a, b)$ is a normalizing factor defined as:

$$Pr(z|a, b) = \sum_{s' \in S} \sum_{s \in S} O(s, a, s', z)T(s, a, s')b(s).$$

Definition 19 *A POMDP policy π_{POMDP} is a function $\pi_{POMDP} : b_t \rightarrow a$, which associates an action to each POMDP belief state.*

The classical optimal approach to solve a POMDP is the value iteration approach [Sondik, 1971, Kaelbling *et al.*, 1998], where iterations are applied in order to compute more accurate values for each belief state $V(b)$. Equation (3.5) describes the value function (Bellman's Equation) for POMDPs which assigns the best value for a belief state depending on a chosen action added up with the best rewarded values the agent could receive up to time t.

$$V_t(b) = \max_{a \in A} \left[\sum_{s \in S} b(s)R(s, a) + \gamma \sum_{z \in Z} Pr(z|a, b)V_{t-1}(\tau(b, a, z)) \right] \tag{3.5}$$

Once iterations lead to a convergence, an optimal policy is defined by mapping the action that gives the maximum value given by $V(b)$.

$$\pi_t^*(b) = \arg\max_{a \in A} \left[\sum_{s \in S} b(s) R(s,a) + \gamma \sum_{z \in Z} Pr(z|a,b) V_{t-1}(\tau(b,a,z)) \right] \tag{3.6}$$

It is well known that the value function $V(b)$ can be represented as a finite collection of $|S|$-dimensional vectors known as α vectors. Thus, V is both piecewise linear and convex [Sondik, 1978]. The backup operation represented in (Equation 3.5) reaches a complexity of $O(|S|^2|V||Z||A|)$, where $|V|$ is the number of α vectors representing the value function [Sondik, 1978]. In some cases, if the problem has very bounded action and observation spaces the complexity might get to $O(|S|^2|V|)$. Knowing that for computing an optimal policy, V must be updated over the entire belief space, this will lead to a very expensive computation for the whole backup operations.

To overcome the complexity of optimally solving a POMDP, a large variety of approximate approaches were described to decrease the complexity of finding acceptable policies. Some approaches consists in finding an exact solution for an approximate model as in Q_{MDP} based on the underlying MDP [Littman *et al.*, 1995] or grid-based approximations [Hauskrecht, 2000, Zhou et Hansen, 2001]. Other approaches consists in finding an approximate solution of the exact POMDP model as in point-based approaches [Smith et Simmons, 2004, Pineau *et al.*, 2006, Shani *et al.*, 2007, Dibangoye *et al.*, 2009]. Point-Based approaches compute the value function V only over a finite sub-set of belief states B instead of computing an optimal value function over the entire belief space. As a result, the complexity of the value function is bounded by $|B|$. Point-based approaches differ in their method to collect the reachable belief states and the way they order the point-based backups on the collected beliefs. The publicly released ZMDP software package [Smith, 2005] was used to solve some of the POMDP models described in Part III of this thesis, by calling one of its solvers: Focused Real-Time Dynamic Programming (FRTDP). The following presents a quick description for some approaches of solving POMDPs.

The Q_{MDP} approach

The Q_{MDP} [Littman *et al.*, 1995] is a sub-optimal solution based on the Q-value function of the underlying MDP of the solved POMDP. In a POMDP, the Q function for action a notated $Q_a(b)$ is the expected reward for a policy that starts in belief state b, takes action a and then behaves optimally. By choosing the action that has the largest Q-value for a given belief state, an agent can behave optimally. To find Q-functions for POMDPs, the Q_{MDP} approach proposes to make use of the Q-values of the underlying MDP. That is, by temporarily ignoring the observation model and find the $Q_{MDP}(s,a)$ values for the MDP consisting of the transitions and rewards only. Using the Q_{MDP} values, it is possible to estimate the Q value for a belief state b (using

the probabilities of being in a state $b(s)$) for each possible action as:

$$Q_a(b) = \sum_{s \in S} b(s) Q_{MDP}(s, a)$$

This approach assumes that after the next action, the agent will have no more uncertainty about its state. Thus, the action with largest long-term reward from all states will be the one chosen at each step. The Q_{MDP} algorithm is a suboptimal policy that, nevertheless, can yield acceptable results when applied to medium or large sized environments. The main drawback is that Q_{MDP} policies will not take actions to gain information. For instance, a "look around without moving" action and a "stay in place and ignore everything" action would be indistinguishable with regard to the assumption of one-step uncertainty. This can lead to situations in which the agent loops forever without changing belief state.

Grid-based approaches

The key of finding truly optimal policis in POMDPs is to cast the problem as a completely observable continuous-space MDP where the state set will be the set of belief states [Cassandra *et al.*, 1994]. A family of approximate solvers were established for finding a solution to solve MDPs with continuous state space [Hauskrecht et Kveton, 2003, Li et Littman, 2005], one of the first of this family is the grid-based approach [Zhou et Hansen, 2001, Bonet, 2002, Hauskrecht, 2000].

A finite grid is placed over the belief simplex, values are computed for points in the grid, and interpolation is used to evaluate all other points in the simplex. Therefore the value function over the continuous belief space can be approximated by the finite set of the grid points that estimates the value of an arbitrary point of the belief space by relying only on the points of the grid and their associated values. Different approaches were proposed related to the emplacement of the points and the regularity of the grid to approximate the calculation of the value function as efficiently as possible [Hauskrecht, 2000, Zhou et Hansen, 2001].

Topological Order Planning

Topological Order Planning (TOP) [Dibangoye *et al.*, 2009], is an approximate POMDP solver that uses the topological order of the underlying MDP to find good belief space trajectories. TOP groups together states into layers, creating an acyclic layer graph. Layers are solved in reversed topological order, starting with layers that contain goal states. Belief space trajectories are directed towards the solvable layers of the model. Once a layer has been solved, trajectories that reach that layer can be terminated, resulting in shorter trajectories and thus less backups. The policies yield from TOP algorithm are interesting when the represented problem contains a significant topological structure.

Heuristic Search Value Iteration

Heuristic Search Value Iteration (HSVI) [Smith et Simmons, 2004] is an approximate POMDP solution algorithm that provides provable bounds on the reward obtained by the polices it produces with respect to the optimal policy. The algorithm stores a compact representation of the upper and lower bounds of the value function over the belief state. After initializing the lower bound according to the worst case reward for each action and the upper bound by assuming full observability, the bounds are refined through heuristic search by making local updates to specific beliefs. The beliefs to update are chosen by heuristic depth-first search of a tree that represents how beliefs evolve according to the actions and observations. The policy is obtained directly from the piecewise linear convex representation of the value function given by the lower bound. HSVI can be modified to yield an anytime algorithm by iteratively adjusting the convergence threshold for the bounds.

Focused Real-Time Dynamic Programming

Focused Real-Time Dynamic Programming (FRTDP) is another heuristic search-based approximation algorithm that is similar to HSVI. FRTDP maintains, as HSVI, upper and lower bounds on the value function. However, it chooses states in a way that avoids revisiting states that do not improve. FRTDP stores an explicit graph, a finite data structure holding information about states which have already been visited during search. As in HSVI, the goal of the heuristic is to achieve the greatest reduction in the width between reward bounds. This algorithm also uses an adaptive termination criteria to prevent unnecessary exploration of long, low reward trials. FRTDP is typically faster to converge than HSVI, at the expense of having far greater memory costs because of the need to store the explicit graph. A description of FRTDP applied to MDP models is given in [Smith et Simmons, 2006].

3.5 Discussion

The chapter presented different models that are used in the literature to describe and solve HRI problems. This discussion aims to compare the different advantages and properties of the previously mentioned models. Table 3.1, presents comparisons between the different Markovian models, BDI, DBN and HTN. The subjects of comparison are: handling uncertainty, complete/partial observability, passive/active model, planning horizon and optimality.

The table shows the ability of HMMs, POMDPs, BDI and DBN to handle uncertainty and partial observability. HTNs, in their formal model, are not able to handle uncertainty and partial observability, however, [Bouguerra et Karlsson, 2005] extended HTN model to overcome both limitations. DBNs represent a generalization of systems that model dynamic events such as HMMs.

An active model is a model that represent actions (planning and execution). Markov chains, HMMs, DBNs and HTNs are considered as passive models. HTNs are used to hierarchically

	Uncertainty	Observability	Active/Passive	Planning Horizon	Optimality
Markov Chains	true	complete	passive	-	-
HMMs	true	partial	passive	-	-
MDPs	true	complete	active	long-term	optimal
POMDPs	true	partial	active	long-term	optimal
BDI	true	partial	active	one-step	sub-optimal
DBN	true	partial	passive	-	-
HTNs	false	complete	passive	-	-

Table 3.1: A comparison between different Markovian models, BDI, DBN and HTN.

formulate a plan and decompose it to primitive actions but they are not actually involve in the action execution. BDI architecture is used for deciding goal directed agent actions in dynamic environments and to guide execution in real time.

As a heuristic approach the BDI model is likely to be sub-optimal. Moreover, BDI is a one step planner and include no look-ahead planning.

Many efforts concentrated on comparing and finding relations between those different models (POMDP/BDI [Schut *et al.*, 2002], BDI/HTN [de Silva et Padgham, 2004] and BDI/MDP [Simari et Parsons, 2006]). Some were interested in hybrid models as for the hybrid POMDP/BDI model [Nair et Tambe, 2005] used for multi-agent teams.

The multi-agent systems community are mostly interested in BDI, Multi-agent MDP [Boutilier,] and Decentralized POMDPs [Bernstein et Zilberstein, 2000]. These approaches are used when planning for more than one agent in the team.

Chapter 4

Related Work

Contents

4.1 Intention Recognition 59
4.1.1 Approaches for Intention Recognition in HRI 61
4.1.2 Ambiguity in Intention Recognition 64
4.1.3 Learning 65
4.2 Planning in HRI 65
4.2.1 Dynamic Environments 65
4.2.2 Planning Under Uncertainty 66
4.2.3 Planning versus Interaction Type 67
4.3 Discussion 68

This chapter will discuss related work to the HRI problematics described in Chapter 2 that employ approaches from Chapter 3. The described related-work aims to solve collaborative and assistive HRI domains in regard to: intention recognition, human-robot communication and planning. This chapter ends with a comparison between the suggested systems in the literature and what they offer as solutions to the difficulties facing companion robotic systems.

4.1 Intention Recognition

Following to the description of human intention in Section 2.3.2, the ability to understand a partner's actions in terms of goals and other mental states is an important element of cooperative, collaborative and assistive behaviors. Therefore, interactive robots need to recognize their human-partner's goals and intentions. For instance, in a human-robot collaborative task scenario, the robot should be capable of tracking human actions and use belief and goal inferences to anticipate the human's needs and to plan actions to provide the human with timely and relevant collaboration. Such a capability will enable robots to behave based on implicit rather than explicit commands from humans.

Definition 20 *Intention Recognition is the procedure of recognizing the intention of an agent using, as clues, the actions of this agent or the changes in the environment caused by the agent's actions.*

Definition 21 *Plan Recognition is the procedure of recognizing the intention of the agent as well as his plan to achieve this intention.*

Definition 22 *A Plan is a sequence of actions that leads an agent to achieve an intended goal.*

There are many aspects of the utility of intention recognition for robot companion applications. Cleaning robot machines, that are aware of the human plan of the day, are more likely to avoid noisy cleaning tasks in the same room where the human is resting [Cirillo *et al.*, 2009b]. Recognizing the human intention, in human robot collaboration, is practical for the robot in order to plan the best action to collaborate with the human [Hoffman et Breazeal, 2007]. In elderly assistance, suspicious activities recognition helps in detecting a need of assistance such as a reminder to take medication [Pollack *et al.*, 2003, Myers *et al.*, 2007], or assist cognitively impaired people in completing certain tasks [Adlam *et al.*, 2009, Hoey *et al.*, 2010].

To be able to infer the human intention, the robot should be able to evaluate the human actions towards his possible intentions. To do this, as described in Figure 4.1, the robot needs to perceive human actions and use them next to existing knowledge in order to translate those actions into possible intentions. The robot, therefore, can plan most appropriate decisions with respect to the human inferred intention. The existing knowledge can be of the form of relational task hierarchy [Natarajan *et al.*, 2007], policies and Q-functions [Fern *et al.*, 2007], trajectories [Nguyen *et al.*, 2005], etc. This knowledge can be self-constructed by the robot through simulation [Gray *et al.*, 2005] or learned by observation [Taha *et al.*, 2008] or simulated then reinforced by learning [Fern *et al.*, 2007] or manually given by the user.

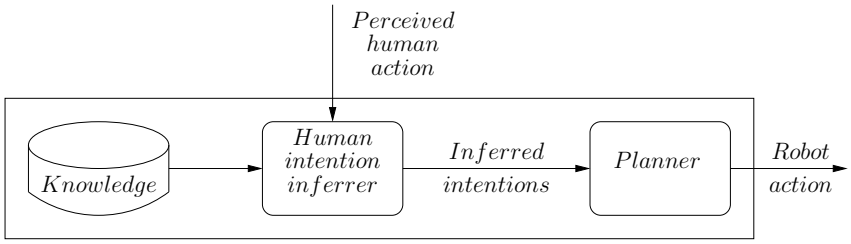
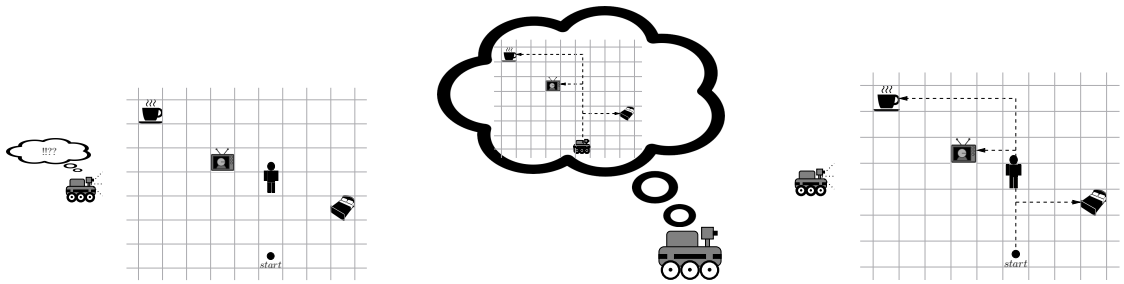


Figure 4.1: Inferring human intention by perceiving his actions.

Definition 23 *The Simulation theory of empathy [Rameson et Lieberman, 2009, Gallese et Goldman, 1998] proposes that we understand the thoughts and feelings of others by using our own mind as a model.*

The simulation theory is a theory of how we understand others. The theory holds that humans anticipate and make sense of the behavior of others by activating mental processes that, if carried into action, would produce similar behavior.

To be able to accompany humans, robots need to understand people as social entities whose behavior is generated by underlying mental states such as intentions, beliefs, desires, feelings, attentions, etc. Using similar brain structure to simulate the underlying mental states of the other person can help in anticipating and understanding his behavior. This similarity have dual use: to be able to generate similar behavior as humans and also predict and infer the same in them [Gray *et al.*, 2005].



(a) The robot perceiving the human actions.

(b) The robot simulating possible human intentions (trajectories) by empathy.

(c) The robot matches simulation with the human world.

Figure 4.2: Generating knowledge about the human by simulation.

Figure 4.2 shows an example of a robot trying to infer its human partner's intention. Possible human intentions in this example include: going to bed to rest, watch tv or make coffee. Figure 4.2(b) shows how the robot can generate knowledge about the human by simulating trajectories by empathy. In Figure 4.2(c), the robot matches its generated knowledge about the human with the real human world, allowing it to make inferences about the human's likely goals that would arise given these circumstances within the associated robot system. Different approaches are proposed to actualize this matching, some of those are to be detailed in the following of this section. Incorrect inferences present an opportunity for learning [Fern *et al.*, 2007]. Ideally, even incorrect inferences should at some level seem plausible to the human. This will assist with efficient error recovery and reduce the chances of making the same incorrect inference in the future.

4.1.1 Approaches for Intention Recognition in HRI

Different domains require different kinds of information to infer or recognize: informative robots might need to infer their human partner's belief states to anticipate his informational needs, collaborative robots are more driven by the human's goal states to physically help achieve those goals, and servant-communicative robots need to infer the human spoken dialogs and gestures to understand how they can serve him. Main approaches used in the literature for intention

recognition are now detailed, they use formalism based on HMMs, DBNs and Q-value functions from MDPs.

Intention recognition using HMMs:

Trained HMMs have been used for activity understanding, showing a significant potential for their use in activity modeling and inferring intent [Kelley *et al.*, 2008, Yokoyama et Omori, 2010]. Using the Baum-Welch algorithm [Rabiner et Juang, 1986], it is possible to apply repeated execution of a given activity to provide the data used to estimate the model transition probabilities [Kelley *et al.*, 2008]. As a result of training, the robot has a set of HMMs, one for each activity. The recognition problem, then, consists of inferring the intent of the actions from the trained HMMs. Toward this end, the robot monitors the human behavior and computes the likelihood that the sequence of observations has been produced by each applicable HMM using the Forward Algorithm [Rabiner et Juang, 1986]. To recognize the intent of an agent, Kelley et al. considered the intentional state emitted by the model with highest probability. The Viterbi Algorithm [Rabiner et Juang, 1986] detects the most probable sequence of hidden states.

HMMs are a powerful tool for modeling sequential phenomena, and have been knowingly used in applications involving speech and sound [Rabiner, 1989].

More formalisms based on HMMs have been proposed for domains with special properties, like: Hierarchical Hidden Markovian Model (HHMM) [Nguyen *et al.*, 2005], Abstract Hidden Markovian Model (AHMM) [Bui *et al.*, 2002], Abstract Hidden Markovian Memory Model (AHMEM) [Bui, 2003], Switching Hidden Semi-Markovian Model (SHSMM) [Duong *et al.*,].

Intention recognition using DBNs:

Bayesian Networks and DBNs are known for their capability for exploiting the causal dependency structure of the given domain. They were used in many human intention recognition domains [Hui et Boutilier, 2006, Schmid *et al.*, 2007, Natarajan *et al.*, 2007, Krauthausen et Hanebeck, 2009] and suspicious activity recognition [Pollack *et al.*, 2003].

Figure 4.3 shows a generic DBN model for intention recognition inspired from [Schrempf et Hanebeck, 2005]. They feature one node for the hidden intention state in every time step. As human intentions are often influenced by external circumstances, they represent these environmental influences by a node containing the "domain knowledge". Possible actions are given as nodes depending on the intention. Measurement nodes layer is proposed to be added to the network to enable robots to reconstruct observations from sensor measurements.

To represent temporal behavior of a human, an edge is introduced from the intention node in time step t to the intention node in time step $t+1$. This enables the network to cope with the human "changing his/her mind". The human actions depend on his intentions. These actions do not depend on other actions in the same time step. However, actions may depend on the action performed in the preceding time step. Hence, an edge from every action to its corresponding node in the next step is drawn. These edges contain information on how likely it is, that the

same action is performed twice, given a certain intention. Edges from one action in time step t to a different action in time step $t+1$ are possible as well, introducing information on the likelihood of successive actions. Since sensor measurements depend only on the action at the current time step and not on previous measurements, no edges are drawn between measurement in two different time steps. The edges connecting two time steps represent the transition model as in HMMs.

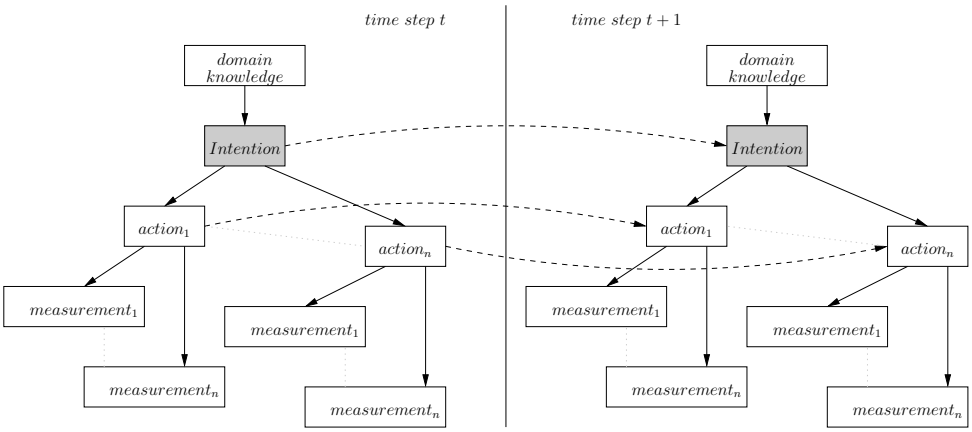


Figure 4.3: A generic DBN model for intention recognition.

To estimate the human intention, the estimator computes a probability density over the intention given the measurements and the domain knowledge using Bayesian forward and backward inference.

Different particular DBN models were proposed as in Hybrid Dynamic Bayesian Networks (HDBNs) [Schrempf *et al.*, 2007] for continuous and discrete valued states and Quantitative Temporal Bayesian Networks (QTBNs) [Pollack *et al.*, 2003] for domains where the Markov property is violated.

Intention recognition using Q-value functions:

As mentioned before, some approaches simulates human plans then uses observed human actions to estimate the human intention. For such estimations, [Fern *et al.*, 2007] suggest to build an MDP for each human intention g including the environment states and the human actions and then solve the Q-value function. The Q-value function gives the expected cost of executing agent action a in world state w and then acting optimally to achieve intention g using only human actions $Q(a, w, g)$. Computationally, the main obstacle to this approach is computing the Q-value function, which need only be done once for a given application domain. Then an intention distribution G is maintained where the probability of the human having intention g conditioned on observation sequence is $Pr(g|O_t)$. Given the human Q-value functions calculated from the MDPs, it is straightforward to incrementally update the intention distribution upon each of the human's actions $Pr(g|O_t) = (1/Z)P(g|O_{t-1})Q(a, w, g)$, where Z is a normalizing constant. That is, the distribution is adjusted to place more weight on intentions that are more

likely to cause the agent to execute action a in w. The accuracy of intention estimation relies on how well the Q-value functions truly reflects the human.

4.1.2 Ambiguity in Intention Recognition

It is possible to obtain ambiguous results from intention recognition models, where the likelihood of many possible human intentions are equal. As in the example of the bar-robot described in Section 2.2.3, the situations where the human is standing near the bar with an empty glass provide insufficient information to the robot to know if he wants to fill his glass or not. In this case, it is impossible for the robot, to infer the intention of the human, therefore, to decide what to do.

Some have thought of possible ways that can trigger a change in this ambiguity. In [Schrempf *et al.*, 2005, Schmid *et al.*, 2006, Yorke-Smith *et al.*, 2009], they used models with proactive (Definition 7) behavior to trigger a clarifying reaction from the human. The robot proactively supposes one of the intentions is true and acts in consequence then monitors the reaction of the human to better recognize his intentions. In the example of the bar-robot, the robot can proactively choose that the intention of the human is to fill his glass, so it proactively extends its hand with the bottle towards the human and waits for the reaction. The robot should have the means to choose which intention to suppose true and which proactive action is the safest and most probable to reveal a human reaction.

It is possible to use communication to solve ambiguity in intention recognition. Verbal communication is the most studied form of communication for ambiguity detection. Using verbal communication, the robot can ask the human about his preferences over intentions. However, audio signal on a mobile robot have poor reliability. When the robot detects a change in the human intention, the robot must decide whether conflicting evidence has been introduced by a speech recognition error or by an actual new user intention. Therefore, the mission is limited by the noisy and stochastic speech input of the human which leaves to the robot's system to overcome the problem of uncertainty in speech recognition [Williams et Young, 2007, Schmidt-Rohr *et al.*, 2008a].

Cost of Communication The ultimate goal of a dialog system is, mostly, to receive a human's request, to acquire information from him or to give information to him. A human can be considered for the robot as a source of information. When uncertain about the state of the world or the state of the human, some systems allow the robot to ask the human to get information in order to be able to continue with its tasks. Also, when collaborating with the human, the robot might need to communicate information to the human that are important to the task achievement. In all previous situations, however, the robot must interrupt the human to initiate a dialog, which is not always appropriate. The robot should be able to decide when to query the human [Kaupp *et al.*, 2010] and it must make trade-offs between the *gain* of gathering additional information and the *cost* of interrupting him [Kamar *et al.*, 2009, Rosenthal et Veloso, 2011].

Spoken dialog managers have benefited from using stochastic planners. Recent research has shown that MDPs and POMDPs can be useful for generating effective dialog strategies [Doshi et Roy, 2008, Roy *et al.*, 2000].

4.1.3 Learning

Learning approaches can be used in activity recognition [Kelley *et al.*, 2008] where a human model of an activity is learned by observation. Once having a learned human model for each of the human activities, a system can detect an abnormality in the human actions and propose assistant [Hoey *et al.*, 2010] or alert a caregiver [Duong *et al.*,]. Q-learning 3.4.3 can be used to learn a human model and then evaluate observed human actions for all possible activities in order to recognize his intention [Fern *et al.*, 2007].

In this thesis, we do not use learning mechanisms, however, it is our priority as a future work to integrate learned human models into some of our proposed contributions.

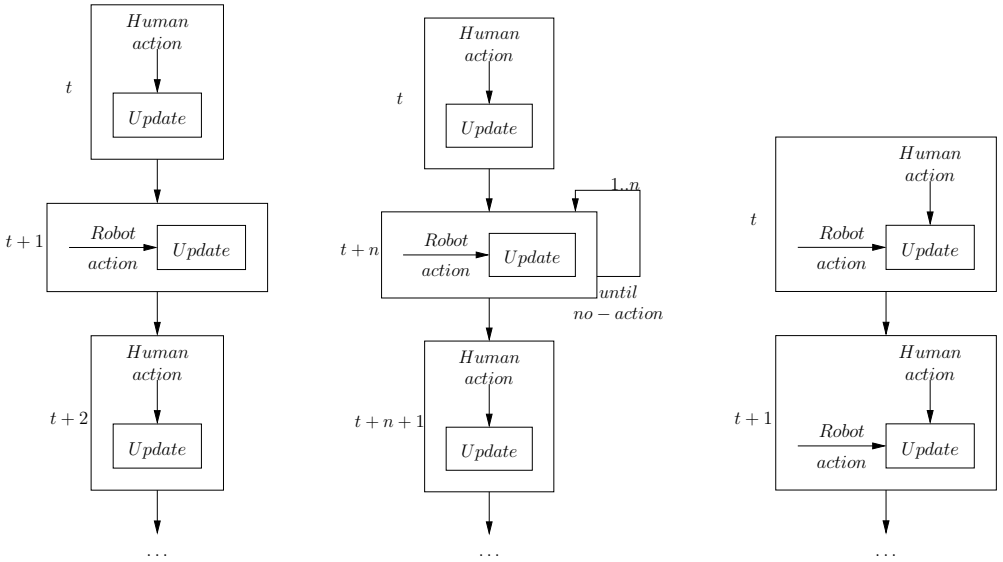
4.2 Planning in HRI

Most HRI environments are full of uncertainty and partially observable, they are mostly represented with non observable, hidden states like human intentions. This is a challenge for robotic systems to be able to represent such environments in a way that allows optimal or near optimal planning. Environment representation must balance between two problems. First, it is important to choose how to model companion robots environments in order to be able to plan actions considering all possible situations. Second, robot systems act online in real human environments, this makes time a very important aspect in decision making. Larger environment representations mean more needed time to treat those representations and to make decisions.

4.2.1 Dynamic Environments

Environment representations must include information that represent all needed facts about the environment regarding the problem domain, in addition to information about the human who is a part of this environment. Changes in the environment can be caused by the human, by the robot or other ways. The changes by the robot can be known with uncertainty by knowing the executed robot action. Changes made by the human can be known with uncertainty if the human action is observed or can be tracked by the robot. Spontaneous changes in the environment (not necessarily caused by the human nor the robot) can be represented by a probability distribution so that the system is aware that they may occur.

Supposing a robotic system that follows a time-step protocol to update its knowledge about the state of the environment, the literature includes three possible ways for the environment to evolve at each time step. Figure 4.4(a) shows a *simple turn taking* type of evolving where the human and the robot switch turns and only one action from one of them is applied at each time step, as in [Hoffman et Breazeal, 2007]. Figure 4.4(b) shows a special case from the previous



(a) Simple turn taking human and robot actions.

(b) One/Several human/robot actions.

(c) Parallel human and robot actions.

Figure 4.4: Different models of alternance between human and robot actions.

model, where the human and the robot switch turns, however, the robot takes turn over several time steps, applying an action at each of them, until its planner has no more useful actions to do, and after that, the human takes turn for only one time step to make one action, this is presented in [Fern *et al.*, 2007] and will be referred to as *one/several turn taking*. Finally, Figure 4.4(c), shows the case where both human and robot apply in *parallel* an action at each time step, as in [Taha *et al.*, 2008, Rosenthal et Veloso, 2011].

The update blocks in Figure 4.4 updates the robot knowledge about the environment using the robot action, the representation model and the observations (human action and changes in the environment).

4.2.2 Planning Under Uncertainty

The sense of expectation is very important for planning. To plan optimally and under uncertainty, the agent has to consider not only what will "most likely" happen but everything that "may" happen. POMDPs have proved very useful for modeling, planning and learning under uncertainty and for long term horizons [Cassandra *et al.*, 1994, Kaelbling *et al.*, 1998], which make them appropriate to use for companion robots systems.

Uncertainty about the changes, caused by possible human or robot actions, can be represented in POMDPs by the transition function. Human actions can be dealt with as observations, and the human intention is represented as a non observable part of the state (hidden state). Some approaches that use POMDP planners have used to infer human intentions: trained HMMs [Taha *et al.*, 2008], DBNs [Natarajan *et al.*, 2007] and Q-functions [Fern *et al.*, 2007].

Companion robots are mostly intruders to human environments. Humans are free in choosing their actions, and robots must adjust and plan for best interaction with the human (collaborate with, cooperate with or assist the human). As part of this adjustment, robots must be able to infer human intentions and choose their actions almost instantly. This will prevent situations where human loose patience, especially in human-robot collaboration including joint tasks and related manipulation of objects [Hoffman et Breazeal, 2007], or in assisting and detecting abnormality for elderly people in their daily living activities [Boger *et al.*, 2005].

As shown in Figure 4.4 a robot decision must be taken:

- each time step while observing the human action at each of them.
- each two time steps while observing the human action from the previous time step.
- each time step except when the planner fails to choose an action, one human action is observed once before the sequence of actions starts.

Each of these models is chosen depending on the domain to solve, however, the second model (Figure 4.4(b)) prevents the human from acting while the robot still believes that it can make important actions for its mission.

When the system has not enough information to decide an action, the best decision for a reactive robot system can be to wait for a time step or more expecting future observations to give better information about the situation, consequently, being able to make a decision. When the lack of information is related to the human, as described in Sections 2.2.3 and 4.1.2, it is possible to behave proactively/coactively instead of waiting. The challenge for those planners is to select a robot action that urges the human to react in a way that unravels his intention or guide him to collaborate. The corresponding robot actions need to be executed with care, since the recognized intention is uncertain and the human is meant to close the loop of intention recognition.

4.2.3 Planning versus Interaction Type

Planning depends on the relation between the human and the robot during the interaction:

Planning for human-robot team (collaboration): For a human-robot team sharing a task, the planning can be assigned as followed: the human plans for both team members and the robot follow the human orders, the robot plans for both team members and the human follows the robot orders or both team members are responsible for their own actions with respect to the team goal and the actions of the other member. The first two categories have a resemblance to multi-agent planning for a task [Wooldridge, 2002, Nair et Tambe, 2005]. Companion robots systems should not be considered in those two categories because in the first the robot is considered as a tool and not an intelligent system, and the second category does not totally respect the ethics of companion robots. However, it is possible that the robot proposes different plans and then negotiates with the human, who

chooses the optimal plan for the task and his preferences [Montreuil *et al.*, 2007, Alili *et al.*, 2009]. In the first category, which is the most appropriate to companion robots, the robot plans its actions while observing the human actions. For example, in collaborative assembly tasks, the robot must be able to recognize human's basic movements like grasping or lifting and be able to place these in the context of the task. This would allow the robot to plan its actions in the best way to collaborate with the human. For instance, if the robot understands that the plan of the human is to assemble two pieces together, it can collaborate by passing the right type of screwdriver to the human.

Planning for a human aware robot (cooperation): Here the robot plans his actions with respect of the human existence and the constraints of this existence on choosing the task to do. Human intention in this category is an important information to the robot planner. However the robot will not be interested in the exact human intention but it will help in planning to choose the robot's tasks with respect to the human plan (intention).

Planning for assistance: Assistant robots are mostly interested in the current task the human is doing. When a need for assistance is detected (abnormality detection) the robot planner must first verify this need. If confirmed, it plans for the best action to assist the human knowing the current situation and the level of progress of the intended task.

4.3 Discussion

This chapter included related work in the fields of HRI and companion robots, especially in the subject of human intention recognition and planning under uncertainty.

The chapter discussed the proposed state of the art models for representing human information and approaches for recognizing human intentions and deal with ambiguity. It also discussed the uncertainty in companion robots environments and the non-observability of some human variables and the high utility of POMDPs as planners under those circumstances. Section 3.4.4 discussed that POMDPs are not tractable for large state space and mentioned some proposed approximate algorithms in the literature.

Several research projects have concentrated on applicable companion robots and put their robots in real experimental situations. In [Dautenhahn *et al.*, 2005], they explored people's perceptions and attitudes towards the idea of a future robot companion for the home. The approach was adopted using questionnaires and human-robot interaction trials to derive data from 28 adults:"Results indicated that a large proportion of participants were in favor of a robot companion and saw the potential role as being an assistant, machine or servant. Few wanted a robot companion to be a friend. House-hold tasks were preferred to child/animal care tasks. Human-like communication was desirable for a robot companion, whereas human-like behavior and appearance were less essential". This proves the humans acceptance of robots in their daily life. However, it also reminds us of the many important features that are still needed to fulfill the expectations of the accompanied humans.

To conclude the state of the art, Table 4.1 shows a comparison between different studies that were referenced in this chapter in regard to their focus problem, capabilities and the approaches they used. The symbol "-" is used when information are not available or not related. The table includes for each referenced study the intention recognition approach (HMM, DBN or Q-value functions), the robot action planner (POMDP or other), the type of interaction between the human and the robot (collaboration, cooperation or assistance), the robot behavior type (reactive or proactive) and the human robot action turn taking in time steps (*simple turn taking, one/several turn taking and parallel*). The "*" in the planner column indicates a planner that is not presented in the section. For example [Cirillo *et al.*, 2009b] uses PTLPlan (probabilistic conditional planner), [Hoffman et Breazeal, 2007] uses an anticipatory action selector after calculating the cost of possible action sequences, [Pollack *et al.*, 2003] uses local search approach called planning by rewriting and [Schmid *et al.*, 2007] uses a human expert entry of possible set of actions for each possible human intention where the planner calculates the best action if the intention is known and unique or else it calculates the best proactive action if the belief over the intention is very ambiguous.

Important advancements have been achieved in the domain of intention recognition for HRI. However, most accomplishments concerning decision making for companion robots have been specialized in one type of interaction or even a certain interactive task.

We are interested, in this thesis, in decision models that give a companion robot the ability to recognize the human intention and act accordingly by exploring new and original directions. Indeed, **we are interested in high level models that enables companion robots to switch between different types of interactions and different types of behaviors depending on the recognized human intention or his needs. To our knowledge, this issue has not been addressed before.**

	Intention Recognition	Planner	Interaction Type	Behavior	action/time step
[Nguyen *et al.*, 2005]	HHMM	-	-	-	-
[Kelley *et al.*, 2008]	HMM	-	-	-	-
[Schrempf et Hanebeck, 2005]	DBN	-	-	-	-
[Cirillo *et al.*, 2009b]	HMM	*	cooperation	reactive	parallel
[Hoffman et Breazeal, 2007]	-	*	collaboration	reactive	simple turn taking
[Pollack *et al.*, 2003]	QTBN	*	assistance	reactive	parallel
[Hoey *et al.*, 2010]	action recognition/tracking	POMDP	assistance	reactive	parallel
[Natarajan *et al.*, 2007]	DBN	POMDP	collaboration	reactive	one/several turn taking
[Fern *et al.*, 2007]	Q-functions	POMDP	collaboration	reactive	one/several turn taking
[Taha *et al.*, 2008]	-	POMDP	assistance	reactive	parallel
[Boger *et al.*, 2005]	action recognition/tracking	POMDP	collaboration	reactive	parallel
[Schmid *et al.*, 2007]	BN	*	-	proactive	-
[Schmidt-Rohr *et al.*, 2008a]	HMM	POMDP	assistance	proactive	parallel

Table 4.1: A Comparison between state of the art studies concerning approaches for intention recognition and companion robots planning.

Part III

Contributions

Chapter 5

The Augmented Robot Decision Model for Human Intention Estimation

Contents

5.1 Motivation 74
5.2 Evaluating Human Actions 75
5.2.1 Modeling a Rational Human (The Human MDPs) 75
5.2.2 The library of Human Action Values (Q-values) 76
5.3 The Robot Decision Model 78
5.3.1 States, Actions and Observations 78
5.3.2 Rewards 78
5.3.3 Transition and Observation Functions 78
5.4 Illustration: Human-Robot Cooperation for Cleaning an Area . 80
5.4.1 Modeling the Human-MDPs for the Cooperative Mission 81
5.4.2 Modeling the POMDP Decision Model for the Cooperative Mission 82
5.5 Experimental Results 84
5.5.1 State Space and Resolution Time 85
5.5.2 Simulation Parameters 85
5.5.3 Analyzing Simulations Results 86
5.6 Discussion 91

This chapter proposes a companion robot decision model that is augmented with a capability of estimating the human intention. The described decision model can be solved with POMDP

The work presented in this chapter was published in the proceedings of 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2009 [Karami *et al.*, 2009] and in the proceedings of the 5th ACM/IEEE International Conference on Human Robot Interaction, HRI 2010 [Karami *et al.*, 2010]

techniques. The chapter explains how the decision model is enriched to help in estimating human intentions while making decisions in regard to those estimations.

The chapter starts with a motivation for simulating rational human policies to generate a library of human action values (Q-values). Furthermore, a description of how to integrate the Q-values into the robot decision model, while clarifying the role of this integration in updating the estimations over the human intention. This chapter also describes a domain example of a human-robot cooperation. The example consists of a shared mission divided into a number of tasks. It illustrates, throw a real scenario, the use of the described decision model and the integration of the human Q-values. Different experiments are presented to show scalability and performance analysis and to compare results of different policies using simulations with a human acting randomly, rationally or semi-rationally. The experiments also statistically compare the estimated human intention with the real chosen task by the rational and semi-rational human to demonstrate the efficiency of the presented model in estimating the human intention.

5.1 Motivation

As pointed in Chapter 4, companion robots should understand people as social entities whose behavior is generated by underlying mental states. Inspired from the simulation theory of empathy (Definition 23), it was suggested in the literature that with similar brain structure to simulate the underlying mental states of the other person, it is possible to anticipate and understand the behavior of others. A companion robot who has enough information about the human in its own system can infer the human's likely goals and beliefs. In this case, the robot's system uses its resources not only to generate its own behavior, but also to predict and infer the same in the human in order to behave with respect to the human's beliefs and goals.

The decision model is proposed for scenarios where the human's intention is not always known to the companion robot, however, human actions are observable and can be detected by the robot. Therefore, it is necessary that the robot system is able to detect/perceive all human actions that are related to the human-robot mission. A lot of research is concerned with the subject of perception of the human actions based on vision [Poppe, 2010] or wearable sensors [Zhu et Sheng, 2011], though it is not a subject of this thesis. Our concern is for the robot to be able to link an observed human action to a possible human intention. Indeed, with inspiration from the simulation theory of empathy, we suggest a way to evaluate human actions towards all human intentions by integrating human action Q-values generated from Markovian decision models (Human-MDPs) into the robot decision model.

Tracking human intentions is very important to the robot's duty as a companion to the human. Consequently, the non observable human intention must be part of the robot's decision variables. It is rare that the robot knows for sure the human intention, however, it can keep a probability distribution over all possible intentions. To be used effectively, this probability distribution must be updated each time new information about the human and the environment is available. This will allow the robot to detect important changes in the human intention,

whether they are due to change in human's mind or previous bad interpretation. For those reasons, the companion robot decision environment is considered partially observable which motivated us to use POMDPs to model the companion robot decision problem.

5.2 Evaluating Human Actions

This section will explain how the robot system simulates rational human policies and creates from them a library of human action values (Q-values) that are later integrated in the robot's own decision model (as explained in Section 5.3). By simulating rational human behavior in its own system, the robot can evaluate human actions toward all intentions.

The robot system creates a set of Human Markovian Decision Processes (Human-MDPs). Each Human-MDP (denoted MDP^h) is modeled to simulate by empathy a rational human acting towards one of the possible intentions. Solving a Human-MDP, associated to a certain intention, will yield the associated Q-values which hold the value of each human action in each human state given this intention. Therefore, for a certain human state and action, the Q-values from different Human-MDPs give an idea of the importance of this action towards the different intentions.

Fig. 5.1 shows the high level process of the companion robot decision making. The robot starts by creating the Human-MDPs and solving them to get a library of Q-Values which are then integrated in the robot's decision model.

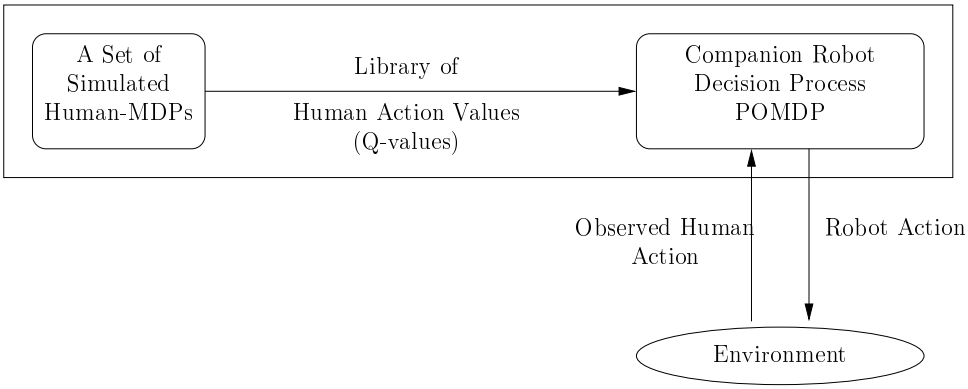


Figure 5.1: The high level process of the companion robot System.

5.2.1 Modeling a Rational Human (The Human MDPs)

It is impossible for the robot to simulate the exact human behavior, even if the robot has access to all information about the human and his mental state. This may be due to different reasons like human emotions or situations that are not predictable nor perceivable by the robot. However, the human behavior simulation can be initialized using a model of a rational human acting in a similar environment toward a certain goal, where it is possible later on to update the model by learning certain variables about the human's personality whom the robot is accom-

panying (Section 4.1.3). The described model include only the rational human model without enhancement by learning.

To be able to model the rational human, the robot uses information concerning the human, the environment, the human possible actions, the changes produced by the human actions, the human goal and the main must/must-not do to achieve this goal. Those information are represented in the Human-MDP set of states, set of actions, transition and reward functions.

Let TK be a set of all tasks concerning the human-robot mission and let $HI \subseteq TK$ a set of tasks that the human might intend to do at any time during the mission. In other words, HI includes all possible tasks that the robot should consider as a possible human intention.

A Human-MDP model (MDP^h_{hi}) is created for each possible human intention $hi \in HI$. This results in a set of Human-MDP models:

$$\text{MDP}^h_{hi} = \left\langle S^h_{hi}, A^h_{hi}, T^h_{hi}, R^h_{hi} \right\rangle \quad \forall hi \in HI \quad \text{, where:}$$

S^h_{hi} represents the set of all possible states, each state includes human and environment variables that are usually common for all tasks in addition to possible variables that concern the human intended task hi. A^h_{hi} represents the set of human possible actions related to achieving his intended task hi. T^h_{hi} is the transition function that gives the probability of transiting between two states by doing an action: $T^h_{hi}(s^h, a^h, s'^h)$, $\forall s^h, s'^h \in S^h$ and $a^h \in A^h$. R^h_{hi} is the reward function that gives the assigned cost or gain for doing a certain action from a certain state, $R^h_{hi}(s^h, a^h)$, $\forall s^h \in S^h$ and $a^h \in A^h$.

However, the idea of creating the Human-MDPs is to evaluate all possible human actions in all possible situations and for all possible tasks. For this reason, a union of all states and actions is created and used in all Human-MDPs, as the following:

$$S^h = \bigcup_{hi \in HI} S^h_{hi}, \qquad A^h = \bigcup_{hi \in HI} A^h_{hi}$$

When creating the Human-MDP ($\text{MDP}^h_{hi_1}$), variables that are related to S_{hi_2} but not to S_{hi_1} are set to a default value and the $\text{MDP}^h_{hi_1}$ model causes no change in their values. Moreover, actions in A_{hi_2} but not in A_{hi_1} cause no change in the $\text{MDP}^h_{hi_1}$ model. The reward function should be defined in accordance between all Human-MDPs; actions with the same level of importance to achieve a task should have close reward values.

As a result of creating the union of states and actions, the Human-MDPs are finally represented as the following:

$$\text{MDP}^h_{hi} = \left\langle S^h, A^h, T^h_{hi}, R^h_{hi} \right\rangle \quad \forall hi \in HI.$$

5.2.2 The library of Human Action Values (Q-values)

The Q-values are calculated by solving the described Human-MDP models. MDPs can be solved using classic Value Iteration [Puterman, 1994](Section 3.4.3) or using factored or other

approximate MDP algorithms [Boutilier *et al.*, 1999, Koller et Parr, 2000, Guestrin *et al.*, 2003, Guestrin *et al.*, 2011] for large scale models.

Algorithm 4 illustrates the calculation of the Q-values by solving the Human-MDPs using classic Value Iteration. During convergence (lines 4:13), the library of Q-values $Q_{hi}(s^h, a^h)$ is filled with the updated value function of the corresponding Human-MDP (V_{hi}).

Algorithm 4: Calculation of Human Action Values (Q-values)

Input : $\text{MDP}^h_{hi} = \langle S^h, A^h, T^h_{hi}, R^h_{hi} \rangle \quad \forall hi \in HI$, , a discount factor γ, a precision criterion ϵ.

Output: A Library of Human Action Values (Q-values): $Q_{hi}(s^h, a^h)$.

`// Running value iteration for all Human-MDPs`

1 **forall** $hi \in HI$ **do**
2 **forall** $s^h \in S^h$ **do**
3 Initialize $V_{hi}(s^h) = V'_{hi}(s^h) = 0$;
4 **repeat**
5 **forall** $s^h \in S^h$ **do**
6 Initialize $max_action_value = 0$;
7 $V_{hi}(s^h) = V'_{hi}(s^h)$;
8 **forall** $a^h \in A^h$ **do**
9 $Q_{hi}(s^h, a^h) = R^h_{hi}(s^h, a^h) + \gamma \sum_{s'^h \in S^h} T(s^h, a^h, s'^h) V_{hi}(s'^h)$;
10 **if** $Q_{hi}(s^h, a^h) >$ max_action_value **then**
11 Update $max_action_value = Q_{hi}(s^h, a^h)$;
12 $V'_{hi}(s^h) = max_action_value$;
13 **until** $\max_{\forall s^h \in S^h} |V'_{hi}(s^h) - V_{hi}(s^h)| \leq \epsilon$;

To illustrate the utility of the Q-values, Figure 5.2 shows an example of a library of Q-values. We can read from the figure (left side) that applying action a^h_1 from state s^h_1 has a value of 0.5 and 0.2 for tasks hi_1 and tk_2 respectively. Those values, if well used by the robot system decision model, can lead to an estimation (when the human is observed doing action a^h_1 from state s^h_1) that he is more probably intending to do task hi_1 than task hi_2.

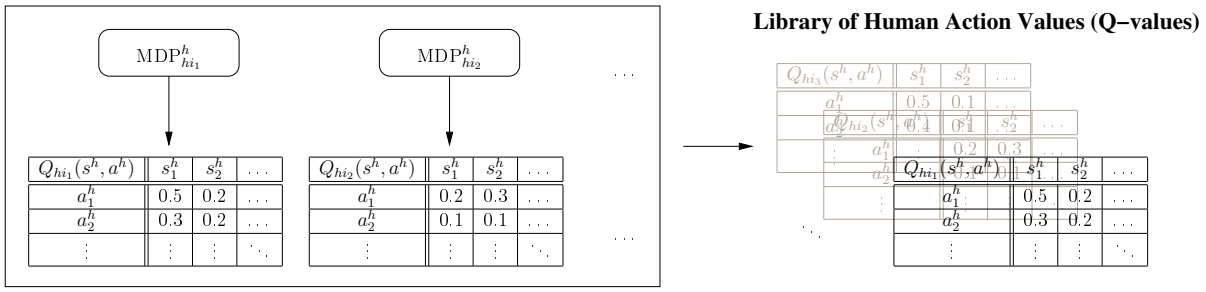


Figure 5.2: The library of Q-values calculated from the Human-MDPs.

5.3 The Robot Decision Model

The use of POMDPs, to model the companion robot decisional problem, is convenient regarding the uncertainty and partial observability concerning the environment, the human and his unknown intentions. In addition, systems with off-line planners (like a POMDP planned policy) are more dynamical than those with on-line planners, which is a sensitive point in companion robots.

In the following, the robot decision model is presented as an augmented POMDP, defined by a tuple $\langle S, A, Z, T, O, R, b_0 \rangle$:

5.3.1 States, Actions and Observations

It is expected that the robot system represents all needed variables related to its decision process. An augmented POMDP state $s \in S$ should include essential information about the robot S_{cr}, the human S_h, the human intention ($S_{hi} = HI$) and the environment (including information about the tasks) S_e. Therefore, each state $s \in S$ includes four groups of variables.

$$S = S_{cr} \times S_h \times S_{hi} \times S_e.$$

The companion robot system does not know at all times the exact state of the world. Therefore, it keeps a probability distribution over the set of states. $b(s)$ represents this belief and b_0 is given to be the initial belief state of the system.

The set of actions A represents all robot actions that can be applied during the mission. The set of observations Z represents all possible observed human actions that are related to the mission Z_h where $Z_h \subset A^h$, in addition to possible observations related to the environment Z_e. Therefore, $Z = Z_h \cup Z_e$.

5.3.2 Rewards

The reward function is the key for controlling the behavior of the companion robot. It must be defined in a way that motivates the companion robot to do actions that respect the human's intention and the assigned mission and avoid actions that cause a conflict with the human or the mission's success.

5.3.3 Transition and Observation Functions

The changes in the state of the environment are caused by the human and the robot who is sharing the environment with the human. The transition function T gives a probability distribution over the next state of the environment $s' \in S$ knowing the current state $s \in S$ and the robot latest action $a \in A$. It represents the uncertainty of the output of the robot's action, in addition to the uncertainty of the output of all possible human actions knowing that the transition function has no knowledge of the latest human action.

However, when the robot observes information about the human Z_h or the changes in the environment Z_e, the observation function O will help in recovering part of the uncertainty regarding the transition function. This will help refining the belief over the current state of the environment in addition to better analyzing the behavior of the human toward his possible intentions.

Integration of the Q-values into the companion robot decision model

This section presents how the library of Q-values is used to infer the human intention from the observed human action. Indeed, when the human is rational, he performs the actions that maximize the value of the achievement of his intended task. Given a human intention $s'_{hi} \in S_{hi}$, the performed human action $(z = a^h) \in Z^h$ from a state $s^h \in S^h$ has the value of $Q_{s'_{hi}}(s^h, z)$. Therefore, It is possible to consider the library of Q-values as a source for a probability distribution over possible observed human actions z towards possible intentions. This idea has motivated us to integrate the Q-values into the robot decision model through the observation function.

The part of the observation function concerned about this integration is the perceived human actions Z_h. The observation function gives the probability of observing z knowing the couple (s, s'). In other words, it gives the probability of observing the human doing action z knowing his current state and his intended task that justifies his action. As explained previously, if the human is rational, this probability can be given by the human action value from the Q-values. However, the corresponding Human-MDP state s^h should be derived from the POMDP state s in order to extract the exact Q-value. This derivation can be done easily by choosing the subset of variables from s corresponding to the variables of s^h. The observation probability is defined in Equation 5.1 where λ is a normalizing factor and $s'_{hi} \in HI$ is the human intention. We notice that the observation function in this decision model does not depend on the action of the robot $a \in A$.

$$pr(z|s, s') = \lambda Q_{s'_{hi}}(s^h, z) \tag{5.1}$$

$$\lambda = \frac{1}{\sum_{z \in Z} Q_{s'_{hi}}(s^h, z)}$$

The transition function (Equation 5.2) is calculated for all state variables following the resolved cooperation problem. However, the estimated human intention s_{hi} can transition with equal probabilities $\frac{1}{|\text{possible } s'_{hi}|}$ to any possible intention including the same.

$$pr(s'|s, a) = pr(s'_{cr}, s'_h, s'_e|s, a) * \frac{1}{|\text{possible } s'_{hi}|} \tag{5.2}$$

Memorizing the estimated human intention for longer terms

It is possible to enrich the companion robot decision model with a property that will help it memorizing for more time steps the estimated human intention. This property has negative and

positive aspects. The negative aspect consists in the fact that the robot needs several time-steps to "forget" a badly estimated intention which might lead to a behavior far from optimal during those time steps. The positive aspect consists in the ability of avoiding estimations caused by a brief human hesitation in case of irrational human behavior.

To add the memory property to the robot model, the variable $maintain \in [0, 1]$ is introduced in the transition function. When the value of *maintain* is high, the transition function represents a higher probability that the human will keep his intention ($s'_{hi} = s_{hi}$) and smaller probabilities $\frac{1-maintain}{|\text{possible } s'_{hi}|}$ that the human will change his intention to $s'_{hi} \neq s_{hi}$. Equation 5.3 represents the transition function enriched with memory.

$$pr(s'|s,a) = \begin{cases} pr(s'_{cr}, s'_h, s'_e|s,a) * \text{maintain} & \text{if } s'_{hi} = s_{hi} \\ pr(s'_{cr}, s'_h, s'_e|s,a) * \frac{1-\text{maintain}}{|\text{possible } s'_{hi} \neq s_{hi}|} & \text{if } s'_{hi} \neq s_{hi} \end{cases} \tag{5.3}$$

5.4 Illustration: Human-Robot Cooperation for Cleaning an Area

This section presents a domain example for a companion robot that cooperates with a human to achieve a shared mission. Figure 5.3 shows the shared environment which consists of a robot, a human and some soil-spots that need to be cleaned. Each soil-spot can be cleaned by the human or by the robot. The mission is considered done when all soil-spots are cleaned.

We remind that the robot decision process has no control over the human actions. The human, however, is considered as an important subject of the environment who makes changes in it and adds constraints over the decision making.

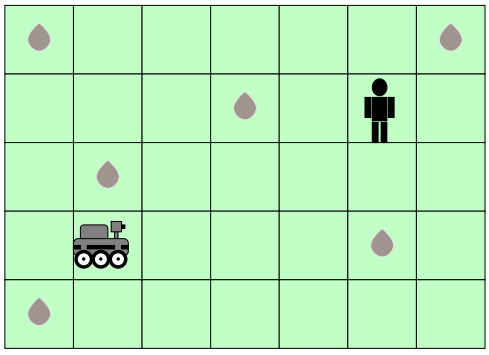


Figure 5.3: The environment of the cleaning an area domain.

This example is presented to illustrate the use of augmented POMDPs for cooperative human-robot missions. In addition, the modeled example is used in the next section to present experiment results regarding the human intention estimations and the robot behavior resulted from the calculated POMDP policy. Therefore, the presented example is simplified to focus on the previously mentioned points and avoid details of the particular problem to solve.

It is important that the system is able to well estimate the human intention and that the robot decisions from the POMDP policy respect those estimations. For instance, if the robot estimates that the human has interest in cleaning the up-right soil-spot, it should respect the human's will and choose another soil-spot to complete the mission.

The robot's belief over the human's intended soil-spot is subject to the observations that it receives. This belief might not be exactly true because of a bad observation analysis or possible sudden change of the human's real intention. One of the important goals of this approach is to compute a policy that would adapt fast enough to any possible change in the environment's variables including human's intention with respect to the mission's success.

The reward function can play a great role in the decision making process. For example, giving less importance to the human's intention and more importance to completing the mission can cause a totally different robot behavior.

The following is the formalization of the cooperative domain example including the Human-MDPs and augmented POMDP model.

The environment space is discretized and represented by a number of possible positions.

$$\text{POS} = \{p_0, \ldots, p_{max}\}.$$

The tasks are defined by the soil-spots positions, where N is the total number of soil-spots.

$$TK = \{tk_i^p | i \in \{0, \ldots, N-1\}, p \in POS\}.$$

The human and the robot can both move at each time step one position east, south, west, north or stay in the same position. The spot is cleaned when the robot or the human has the same position as the soil-spot for 2 successive time-steps.

There is no preference upon the soil-spots, therefore, they may be handled in any order. However, generalizing to any priority or manipulation restrictions (for example, it is preferable that the human cleans the greasy soil-spots) is just a matter of adjusting the reward function.

5.4.1 Modeling the Human-MDPs for the Cooperative Mission

Each task consists of cleaning a soil-spot. Therefore, the number of tasks $|TK|$ is equal to the number of soil-spots. We consider that a human intention can be any of the tasks $HI = \{hi_i^p | hi_i^p = tk_i^p, i \in \{0, \ldots, N-1\}, p \in POS\}$. Therefore, for each task of cleaning one soil-spot a Human-MDP is created. $\text{MDP}_{hi_i^p}^h = \left\langle S^h, A^h, T_{hi_i^p}^h, R_{hi_i^p}^h \right\rangle \quad \forall hi_i^p \in HI$, where:

- S^h represents all possible human states at any time-step which includes the human position in the environment ($s^h = pos^h$). All Human-MDPs share exactly the same set of states.

$$S^h = \{pos^h | pos^h \in POS\}.$$

- A^h represents the set of human possible movement actions in the environment. In this example, the human actions are exactly the same for all tasks.

$$A_h = \{east^h, south^h, west^h, north^h, wait^h\}$$

- $T^h_{hi^p_i}$ represents, for simplicity, deterministic transitions of the human position. It is also similar for all tasks.

$$T^h_{hi^p_i}(pos^h, a^h, pos'^h) = \begin{cases} 1 & \text{if } pos'^h \text{ is the correspondant next position.} \\ 0 & \text{elsewise.} \end{cases}$$

- The reward function $R^h_{hi^p_i}$, however, is different for each Human-MDP model. For instance, in the Human-MDP of the task tk^p_i, the reward is assigned when the human position pos^h is equal to the associated soil-spot position p and he chooses action $wait^h$.

$$R^h_{hi^p_i}(pos^h, a^h) = \begin{cases} r > 0 & \text{if } pos^h = p \text{ and } a = wait^h \\ 0 & \text{elsewise} \end{cases}$$

5.4.2 Modeling the POMDP Decision Model for the Cooperative Mission

All variables in the POMDP decision model are considered as completely observable except for the human intention. At each time-step, the robot's system observes the human's latest action and calculates a new probability over his intention.

States, Actions and Observations

The state space represents the robot, the human and the environment variables,

$$S = S_{cr} \times S_h \times S_{hi} \times S_e.$$

- The robot variables contain the robot position, $S_{cr} = \{pos^r | pos^r \in POS\}$.
- The human variables contain the human position, $S_h = \{pos^h | pos^h \in POS\}$.
- The human intention can be one of the defined tasks or *Other*. The intention *Other* is added to represent the fact that the human is not interested in participating in the mission.

$$S_{hi} = HI \cup \{Other\}.$$

- The environment variables contain the status of the tasks where each task is represented with the fact that it is *not-done* or *done*,

$$S_e = \{stat^{tk^p_i} | tk^p_i \in TK, stat^{tk} \in \{\text{not-done}, done\}\}.$$

The set of robot actions includes its movement actions: moving south, moving west, moving north, moving east or wait.

$$A = \{south, west, north, east, wait\}\,.$$

At each time step, the robot perceives one of the 5 possible human actions.

$$Z = \{oSouth, oWest, oNorth, oEast, oWait\}\,.$$

Transition Function

As described in Equation 5.2, the transition function is computed as the following:

$$pr(s'|s,a) = pr(s'_{cr}, s'_h, s'_e|s,a) * \frac{1}{|\text{possible } s'_{hi}|}$$

For simplicity, the transitions of the robot variables are considered deterministic in the POMDP model. The new robot position depends on its old position and its action.

$$pr(s'_{cr}|s_{cr},a) = pr(pos^r, a, pos'^r)$$

Similarly, the transitions of the status of the tasks are deterministic from *not-done* to *done* when the human or the robot cleans the corresponding soil-spot. The new status of each task depends on its old status and the human's and robot's old and new positions.

$$pr(stat'^{tk_i^p} = done|s,a,pos'^h) = \begin{cases} 1 & \text{if } (stat^{tk_i^p} = \text{done}) \text{ or } (pos^r = pos'^r = p) \text{ or} \\ & \quad (pos^h = pos'^h = p) \\ 0 & \text{elsewise} \end{cases}$$

However, the transitions of the human variables are not deterministic as the human action a^h is not known to the transition function. Therefore, the new human position might change with equal probabilities to any possible position knowing the 5 human possible actions.

$$pr(s'_h|s_h,a) = \begin{cases} \frac{1}{|\text{possible } pos'^h|} & \text{if } \exists a^h \in A^h | pr(pos^h, a^h, pos'^h) > 0 \\ 0 & \text{elsewise} \end{cases}$$

Similarly, the transition of the human intention variable is not known. Therefore, the human intention stays the same or changes to any other possible intention with equal probabilities (no memory):

$$pr(s'_{hi}|s,a) = \frac{1}{|\text{possible } s'_{hi}|}$$

or with a maintain probability that enriches the robot with a memorizing capability as shown in Equation 5.3:

$$pr(s'_{hi} = s_{hi}|s,a) = \text{maintain}, \qquad pr(s'_{hi} \neq s_{hi}|s,a) = \frac{1 - \text{maintain}}{|\text{possible } s'_{hi} \neq s_{hi}|}$$

Observation Function

The observation function as described in Equation 5.1 gives the probability of observing z knowing the current and the next state of the system $pr(z|s,s')$. Unfortunately, the tool we use to solve POMDPs in this thesis [Smith, 2005] does not accept such observation functions, though it accepts the classical observation function $pr(z|a,s')$ which lacks the current state of the system s. The current state s is only needed to extract the human variables s^h that is used to get the appropriate Q-value. We can bypass the need of the current state s under the assumption that the transition function in the Human-MDP is deterministic. This assumption makes it possible to conclude s^h from s'^h and z, where s'^h is derived from s' and z is known. If the assumption is not satisfied, other POMDP solvers must be used.

Therefore, we re-formalized the observation function (Equation 5.4) to avoid the need of the current state (s). The appropriate Q-value is extracted by concluding the current human position pos^h from his next position pos'^h and the action that made him reach this next position z.

$$pr(z|s') = \lambda Q_{s'_{hi}}(pos^h, z) \quad \text{where, } T^h(pos'^h|s^h,z) > 0 \tag{5.4}$$

Reward function

The robot is awarded when it cleans a soil-spot and when the mission is totally completed. On the other hand, a penalty is assigned to the robot when it cleans a soil-spot that is estimated to be intended by the human. However, this penalty is assigned based on the estimation of the robot over human intentions, not on the real human intention. The reward function is defined as follows:

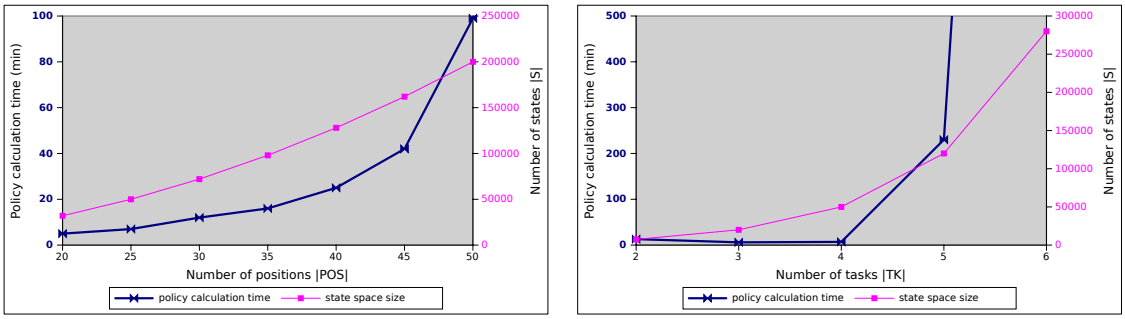
$$R(s,a,s') = \begin{cases} r_1 > 0 & \text{if } \exists tk_i^p \in TK | stat^{tk_i^p} = \text{not-done and } pos^r = p \text{ and } a = wait \\ r_2 > r_1 & \text{if } stat'^{tk_i^p} = \text{done} \quad \forall tk_i^p \in TK \\ r_3 < 0 & \text{if } \exists tk_i^p \in TK | stat^{tk_i^p} = \text{done and } pos^r = p \text{ and } s_{hi} = tk_i^p \end{cases} \tag{5.5}$$

5.5 Experimental Results

The POMDP model, detailed in the previous section, was implemented and solved with the publicly released ZMDP solver [Smith, 2005] using the default FRTDP algorithm. The library of Q-values was created using Algorithm 4 with a discount factor $\gamma = 0.95$ and a precision criterion $\epsilon = 0$.

5.5.1 State Space and Resolution Time

Figure 5.4 presents the scalability analysis for different models with different position POS and tasks TK dimensions and the corresponding needed time to solve the model. For all models, the solver was stopped when the regret $(upperBound(b0) - lowerBound(b0))$ stays unchanged for more than 80 seconds. The solver used a 6 cores Intel(R) Xeon(TM) 2.13 GHz machine with 16 GB of memory.



(a) $|TK| = 4$, $|POS| = \{20, 25, 30, 35, 40, 45, 50\}$ (b) $|POS| = 25$, $|TK| = \{2, 3, 4, 5, 6\}$

Figure 5.4: Policy calculation-time and state space size for different problem sizes.

Figure 5.4(a) shows policy calculation times and state space sizes for a problem with different numbers of possible positions $|POS| = \{20, 25, 30, 35, 40, 45, 50\}$ and a fixed number of tasks $|TK| = 4$. Figure 5.4(b) shows policy calculation times and state space sizes for a problem with different task space sizes $|TK| = \{2, 3, 4, 5, 6\}$ and a fixed number of possible positions $|POS| = 25$. We notice that policies calculation times grows exponentially with the number of tasks than with the number of possible positions. However, the states space sizes grows linearly with the number of positions and exponentially with the number of tasks. The number of possible states in the POMDP model is calculated as following:

$$\begin{aligned} |S| &= |S_{cr}| * |S_h| * |S_{hi}| * |S_e| \\ &= |POS| * |POS| * |TK + 1| * 2^{|TK|} \end{aligned}$$

We notice that a problem with 25 possible positions and 6 tasks needs hours to reach an accepted policy. Problems with 7 tasks might take days and with more than that are impossible to solve.

5.5.2 Simulation Parameters

To analyze the produced policies, we ran various simulations with different human behaviors. All simulations started with the initial state of the environment (as shown in Figure 5.5) and ended with the end of the mission (all 4 soil-spots are cleaned).

There were 3 calculated POMDP policies for three different models. The first model has a transition function with no maintained memory, the second has a transition function with

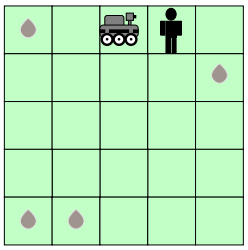


Figure 5.5: Cleaning area domain environment used for simulations.

maintain= 0.5 and the last has transition function with *maintain*= 0.99. The higher the memory level is, the more the robot model is confident that the human will keep the same intention in the next time step. This means that in the model with no maintained memory, the robot's model of the human will be updated only with the knowledge of the latest human action. Therefore, the robot decision will depend only on the latest human action.

During simulations we used three simulated human behaviors. The first is random, where the human's actions were chosen randomly. The second is rational, where the human randomly choses one of the possible tasks to do, then behaves rationally (following the Human-MDPs policies) to achieve the chosen task. The third is semi-rational, where the human behaves as in rational behavior but with a 30% chance that he would change his intention of at each time step.

For each of the three robot policies (with different memory models) and for each type of simulated human behavior, we ran 100 complete missions simulations.

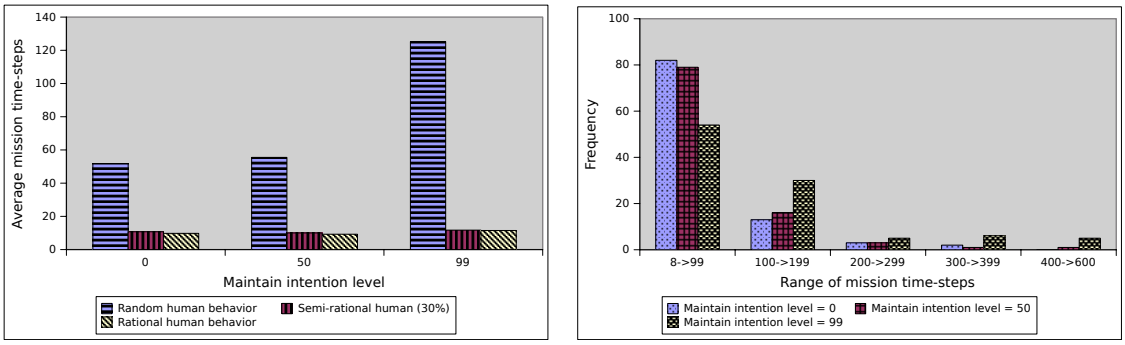
5.5.3 Analyzing Simulations Results

Needed time-steps to complete the missions

Figure 5.6(a) shows the average number of needed time-steps to complete the missions for all 9 simulation combinations. The number of needed time-steps average between 9 and 12 except for simulations with random human behavior. Figure 5.6(b) shows a histogram of the mission lengths for simulations with random human behavior. We notice that contrary to simulations with rational and semi-rational human behaviors, some missions reaches 600 time-steps before they end. This is a natural result when the human moves randomly and the robot's belief over the human intention is changing accordingly. The robot in this case waits a longer time-steps before being able to choose a task that is undesired by the human.

Did the robot do his share of work?

Figure 5.7 shows the number of missions where the robot achieved $(0, 1, 2, 3,$ or $4)$ tasks. The rest of the tasks were achieved by the human, obviously. We notice that for simulations with a random human, the robot achieved two or more tasks in more than 80% of the missions. However, with low memory levels for semi-rational and random human behavior, higher number

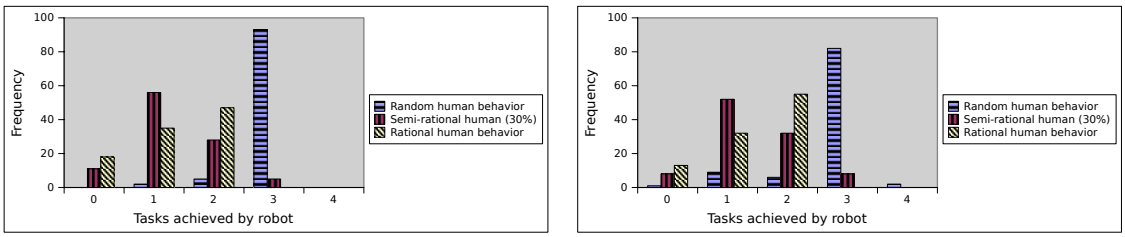


(a) Average of needed time-steps/mission according to different maintain intention levels.

(b) A histogram showing the frequency of chosen ranges of needed time-steps for simulated missions with random human behavior.

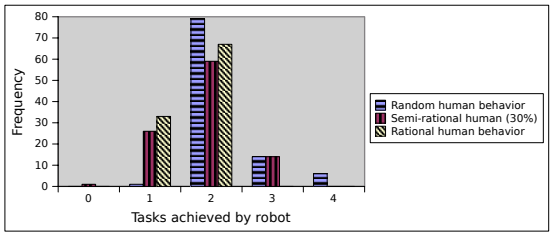
Figure 5.6: The average of missions length and a histogram for those related to random human behavior simulations.

of missions end with 0 tasks done by the robot. This is due to the higher rapidity and reactivity in the change of the belief state over the human intention at each observation of the human action. The robot in this case prefers to observe the human actions to better understand his intention than acting with a high probability of getting a penalty.



(a) No maintained memory.

(b) *maintain*= 0.5.



(c) *maintain*= 0.99.

Figure 5.7: A histogram of the number of tasks achieved by the robot in 100 missions simulation.

Did the robot act as modeled?

Figure 5.8 shows the average reward gained by the robot. The reward is calculated at each time step using the reward function (Equation 5.5) for the dominant state of the current belief state and the robot action (chosen by the POMDP policy). The rewards here are calculated following the robot's estimations over the human intention. This means that the robot is not penalized

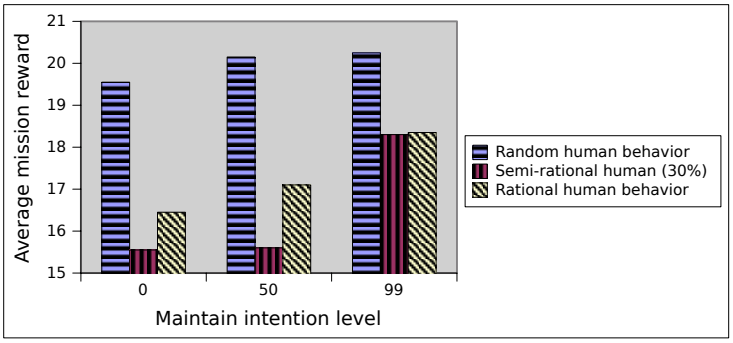
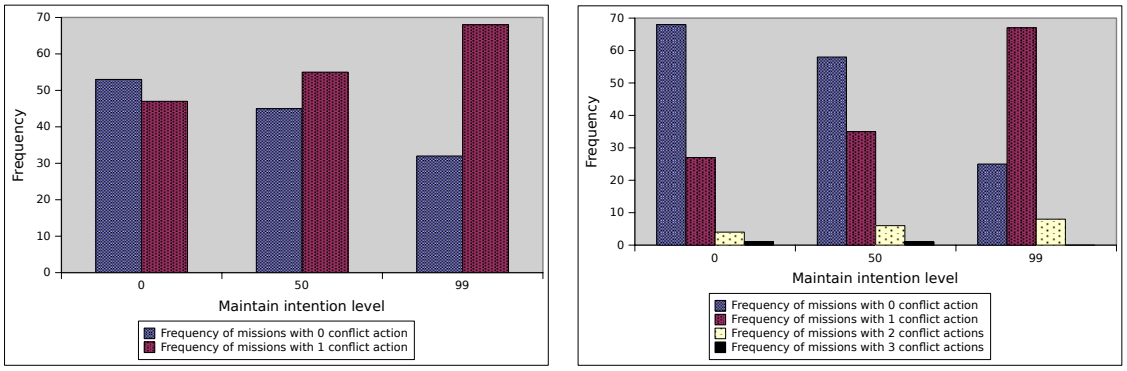


Figure 5.8: Average of rewards per mission.

if it cleans a soil-spot that it believes falsely estimated as not intended by the human. The resulted policies did not **at any time** choose an action that was in conflict with the estimated human intention. A conflict is declared when the robot cleans a soil-spot that is intended by the human. So the question that we should answer is: Did the robot make any conflict action with the real human intention during the simulations?



(a) Frequency of robot actions with conflict with the rational human real intention.

(b) Frequency of robot actions with conflict with the semi-rational human real intention.

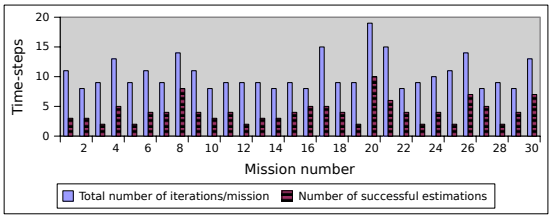
Figure 5.9: Frequency of robot actions with conflict with the real human intention over 100 missions simulations.

To answer this question, we calculated the number of times the robot did a conflict action with the real human (semi-rational and rational) intention. At first, Figure 5.9(a) shows for the rational human behavior simulations, the number of missions (out of 100) that the robot did not even one conflict action, and the number of missions it did only one conflict action with the human's real intention. We notice in Figure 5.9(b) the existence of few missions with 2 or 3 conflict actions with the semi-rational human's real intention. The total number of missions with more than 1 conflict action is less than 10% knowing that the semi-rational human have a 30% probability of changing intention at each time-step. We also notice that the existence of missions with one conflict action gets remarkably higher with the increase of memory level which prevents the robot from quickly update the new human intention. We should also note that the

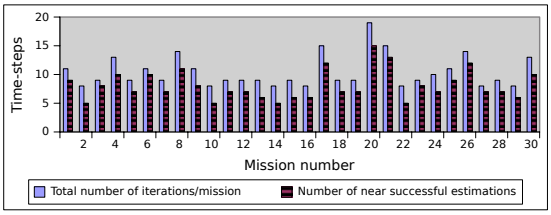
simulated rational and semi-rational human behaviors are not implemented to be particularly cooperative with the robot.

Why were there conflict action? Was the model able to integrate the real human intention?

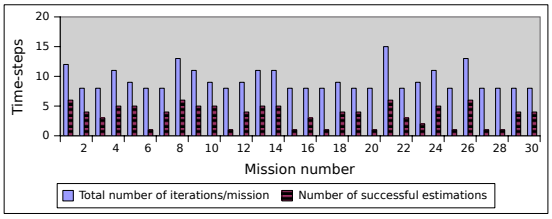
To answer to those questions, we focused on the robot's estimations over the human's intention at each time step of each mission and compared them to the rational and semi-rational human's real intention. Figures 5.10 5.11 5.12 (left side) presents for a representative sample of 30 missions, the total number of time-steps for each mission and the number of time-steps where the dominant estimated human intention was equal to the real human intention at this time-step. We notice that the number of good estimations are very low and should produce a number of conflict actions greater than found in Figure 5.9.



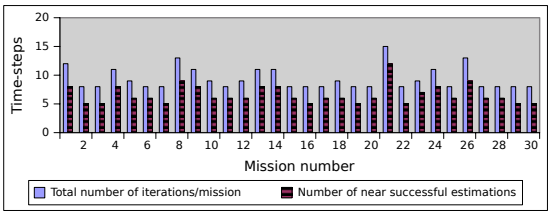
(a) Semi-rational human, maintain level = 0, the real human intention is the highest estimated intention.



(b) Semi-rational human, maintain level = 0, the real human intention is highly estimated intentions.



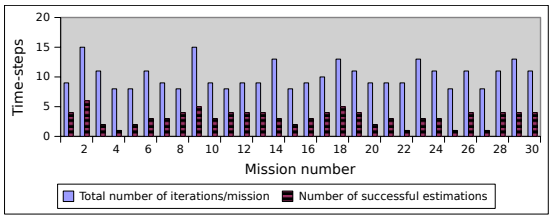
(c) Rational human, maintain level = 0, the real human intention is the highest estimated intention.



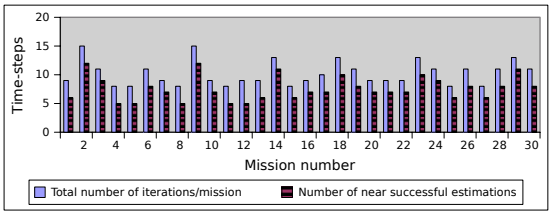
(d) Rational human, maintain level = 0, the real human intention is highly estimated intentions.

Figure 5.10: Comparing good robot estimations of human intentions using two methods for memory level of 0.

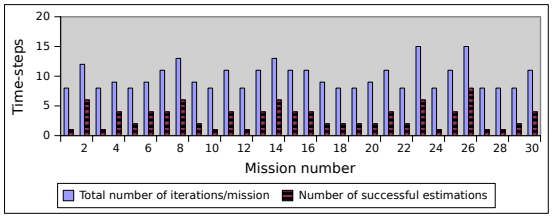
However, Figures 5.10 5.11 5.12 (right side) presents for the same samples, the total number of time-steps for each mission and the number of time-steps where the real human intention at this time-step was highly estimated to be the human's intention (its probability is higher than the average of all possible intentions). This probability is high enough to prevent the robot from doing a conflict action. We notice that here we have better estimations of the human's real intention. Still, there are small number of time-steps where the robot's estimations do not match with the human's intention. These might be a result of many cases, like the time-steps just after the human changes his intention or the time-steps just after the human finishes cleaning a soil-spot.



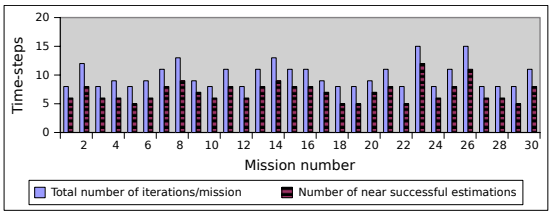
(a) Semi-rational human, maintain level = 50, the real human intention is the highest estimated intention.



(b) Semi-rational human, maintain level = 50, the real human intention is highly estimated intentions.

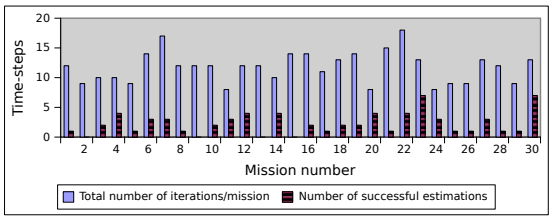


(c) Rational human, maintain level = 50, the real human intention is the highest estimated intention.

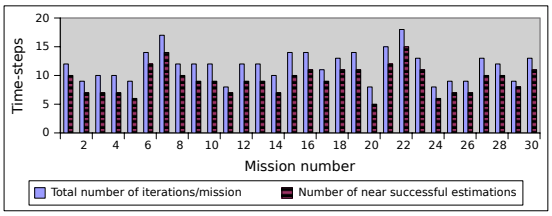


(d) Rational human, maintain level = 50, the real human intention is highly estimated intentions.

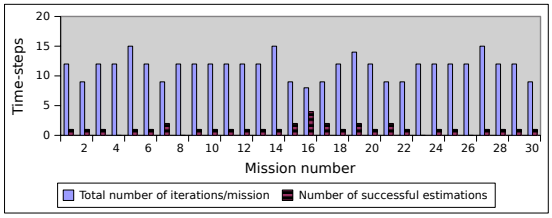
Figure 5.11: Comparing good robot estimations of human intentions using two methods for memory level of 50.



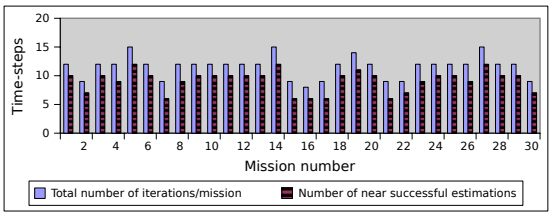
(a) Semi-rational human, maintain level = 99, the real human intention is the highest estimated intention.



(b) Semi-rational human, maintain level = 99, the real human intention is highly estimated intentions.



(c) Rational human, maintain level = 99, the real human intention is the highest estimated intention.



(d) Rational human, maintain level = 99, the real human intention is highly estimated intentions.

Figure 5.12: Comparing good robot estimations of human intentions using two methods for memory level of 99.

5.6 Discussion

We proposed in the chapter a decisional POMDP model for a companion cooperative robot that is enriched with the capability of estimating the human intention. The suggested model integrated human action values (Q-values), calculated by simulating a rational human, into the observation function. This model is able to estimate the human intention at each time step (after observing the human action) and make appropriate decisions regarding the estimated intentions. Moreover, the model is adaptive to any change in the human intention once the latter starts to be reflected in the human actions. In the experiments, we showed that the robot was able to highly estimate the real human intention except during few time-steps when estimations are adjusting to a new intention.

The Q-values in this contribution are calculated using simulations of a rational human. Human intention estimations can be more accurate if the Q-values represent accurately the behavior of the accompanied human while achieving the tasks. Therefore, those values can be reinforced (learned) online [Fern *et al.*, 2007] while observing the real human during the mission. This will allow the model to be more adopted to the human's nature of doing things. It is possible also to learn the human's lack of interest in the mission, for example, recognizing randomness in actions might be a sign of disinterest for some humans.

We proposed also in this model to add a property that allows the robot to memorize the latest estimated intention. We discussed the negative and positive effects of such a memory. It is also preferable to adapt the memory level to the size of the environment. As we noticed in Figure 5.9(b), the number of missions with one conflict action increases proportionally with the increase of the memory level. This is because the environment space is too small and the memory level is too high according to the environment space. This does not give the robot enough time-steps to adjust (forget) the previously estimated intention before the human actually achieves the new intention.

We notice that the suggested model can not deal with the following problems:

- Multi-type of interactions between the human and the robot.

- Large size problems.

- Adapted behavior in collaborative tasks (coactivity).

In the following chapters (6 and 7) we will address those problems.

Chapter 6

A Coactive Decision Model for Human-Robot Collaboration

Contents

6.1 Motivation 94
6.2 Coactive Human-Robot Collaboration 94
6.2.1 The Robot's CDM 94
6.2.2 The Rational Human-MDPs for the Library of Q-values 97
6.3 Illustrative Scenario: Handing Over an Object 98
6.3.1 The CDM for Handing Over an Object Scenario 98
6.3.2 Human MDPs for Handing Over an Object Scenario 101
6.4 Experimental Results 103
6.4.1 Simulations 103
6.4.2 Integration on Real Robots 105
6.5 Discussion 108

This chapter proposes a robot decision model for a collaborative task with a human (handing object to the human, moving a table with the human, filling human's glass with water ...). For such tasks, robots are expected to understand their human partners and collaborate as peers to better achieve the shared task. We propose a Coactive Decision Model to formalize such a problem using an augmented POMDP. The coactivity allows the robot to collaborate in harmony with the human to achieve the shared task, it also gives the robot the possibility of revealing the human intention in case of ambiguity. The augmented POMDP estimates the human intention by evaluating the observed human actions towards possible intentions (Chapter 5). The experiments in this chapter use an example of a robot task to hand over an object to the human. We consider different scenarios of human behavior and show the ability of the robot to adapt and coact for better collaboration.

6.1 Motivation

With the advancement of robotic research, robots are becoming increasingly familiar in situations where they will collaborate with a human as partners or teammates to achieve a shared task (as for a robonaut that helps humans work and explore in space [NASA, 2011] or a nursebot that helps elderly people live their daily life in nursing homes [Pineau *et al.*, 2003]). One important challenge in agents collaboration is acting together to reach the best conditions to achieve the shared task. A collaborative agent should coact in a way that incites the others to collaborate and reach the conditions of success. Similarly, a companion robot should coact with the human notably when the latter shows confusion about the how-to collaborate.

Such research domains requires a shift of focus from robot autonomy to a higher capacity to achieve a task in synergy with a human especially in collaborative situations with accompanied persons. On this subject, [Johnson *et al.*, 2010] introduced the concept of "coactivity". As mentioned in Section 2.2.3, a coactive behavior permits the robot not only to perform its part of the work but also encourage the joint activity of its human partner.

This chapter is motivated by the idea of implementing the concept of coactivity in an HRI framework. It is important that the robot consider the human's reactions and retortions. Moreover, when the robot is certain about the human intention but the human is not collaborating as he should, the robot should coactivly guide him to act so as to achieve the collaborative task. The human's response to such coactive behavior can be used by the robot as a key to better understand the human intention.

As example, the robot can offer simple guidance to allow a successful and flexible collaboration with the human. However, all of this should be combined with a careful inspection about the human's interest in such a collaboration. The robot should respect its main purpose as a companion which is the comfortability and well-being of the human. Therefore, once the human show his disinterest in collaborating, the robot should not annoy the human by offering guidance.

Managing the inherent uncertainty induced by working with humans in dynamic environments is also important. We suggest in this chapter to use the augmented POMDP model that we proved in Chapter 5 a solution for handling uncertainty about the human and adapt it to create a coactive decision framework for Human-Robot Collaboration for a shared task.

6.2 Coactive Human-Robot Collaboration

The proposed Coactive Decision Model (CDM) is based on an augmented POMDP model (Chapter 5). However, the augmented POMDP is adjusted to incorporate the coactive behavior of the robot.

6.2.1 The Robot's CDM

The CDM is represented by a POMDP tuple $\langle S, A, Z, T, O, R, b_0 \rangle$.

The Set of States S

A state $s \in S$ represents variables about the companion robot, the human and the task itself,

$$S = S_{cr} \times S_h \times S_t.$$

The companion robot states S_{cr} include physical robot's states regarding the collaborative task. The human's states S_h include physical human's states regarding the collaborative task, the human's existence in the interaction area (e.g. in the same room as the robot) in addition to all information about the human that can help the robot differentiate the case where the human is ready to co-achieve the shared task from the case where he needs to be impelled into co-acting towards the success of the shared task. The human's non observable variable *engagement* represents the fact that the human is engaged in the collaboration or occupied with something different than the shared task. Finally the state of the task S_t include variables related to the advancement of the task achievement.

The initial belief state b_0 holds uncertainty about the engagement of the human. The belief state $b(s)$ is then updated at each time-step knowing the robot's latest action and the corresponding observed human action.

The Set of Actions A

The CDM distinguishes two categories of actions in its set of actions A. The first category is related to the coactive behavior when needed A_c and the second category is related to the actual collaborative achievement of the shared task A_t. Therefore, the set of actions A in a CDM is defined as following:

$$A = A_c \cup A_t$$

The first group of actions are verbal coactive actions to attract the attention of the human and to guide/incite him to perform an action that favors the task's success. Robot's coactive actions help in revealing the human intention towards the task. A negative human reaction to a coactive robot action is a sign of disinterest in the shared task unless a failure in interpreting the human's actions or a very brief human distraction. For this reason we use two levels of coactive actions: soft level and strong level. A negative human reaction to a strong level coactive action is a strong sign of his unwillingness (occupied or not interested) to collaborate.

The Set of Observations Z

Observations include the human actions that are relevant to the shared task, in addition to information about the advancement of the task.

The Transition Function T

The transition function $T(s, a, s') = pr(s'|s, a)$ (Equation 6.1) memorizes the latest step value of the hidden variable of the state (*engagement*) with a maintain probability of 70%. Where $s, s' \in S$ and $a \in A$.

$$pr(s'|s,a) = \begin{cases} pr(s'_{cr}, s'_h, s'_t|s,a) * \text{maintain} & \text{if } engagement = engagement' \\ pr(s'_{cr}, s'_h, s'_e|s,a) * (1 - \text{maintain}) & \text{if } engagement \neq engagement' \end{cases} \tag{6.1}$$

The Observation Function O

The observation function gives the probability of observing the human action $z \in Z$ knowing the robot did action $a \in A$ and the system ended in state $s' \in S$.

The effect of the observed human action on the non observable hidden state *engagement* is given in (Equation 6.2) using an evaluation of human actions Q-Values (Section 6.2.2) toward possible *engagement* values. λ is a normalizing factor, s^h and s'^h are Human MDP states (Section 6.2.2) derived from s and s' respectively.

$$pr(z|a, s') = \lambda \mathrm{Q}_{engagement'}(s^h, z), \qquad \text{where } pr(s'^h|s^h, z) > 0 \tag{6.2}$$

The Reward Function R

The CDM reward function balances between the negativity of persistent coactive behavior and the main goal of successfully achieve the collaborative shared task. To avoid disturbing the human, the model penalizes all soft and strong coactive actions when the human is found willing and engaged to collaborate. This indicates the fact that it is not necessary to guide a human who does not need guidance and is acting correctly (engaged) by himself toward the collaboration's success. However, if the human is showing willingness but not acting correctly, a guidance from the robot is necessary. Therefore, a small reward is given for soft and strong coactive actions when the human is not showing engagement to the task.

$$R(s,a) = \begin{cases} penalty_1 < 0 & \text{if } (engagement = engaged \text{ and } a \text{ is a soft level incite}) \\ penalty_2 < penalty_1 & \text{if } (engagement = engaged \text{ and } a \in \text{ is a strong level incite}) \\ reward_1 > 0 & \text{if } (engagement <> engaged \text{ and } a \in \text{ is a soft level incite}) \\ reward_2 > reward_1 & \text{if } (engagement <> engaged \text{ and } a \in \text{ is a strong level incite}) \end{cases} \tag{6.3}$$

To assure the balance between the penalties and rewards of coactive actions (soft and strong) in different situations of engaged or occupied human, trials showed that the values of $penalty_1, penalty_2, reward_1$ and $reward_2$ can be found by solving the following equations:

$$v_1 * reward_1 + (1 - v_1) * penalty_1 \leq v_1 * reward_2 + (1 - v_1) * penalty_2 \qquad \forall v_1 \in [0.6, 1]$$

$$v_1 * reward_1 + (1 - v_1) * penalty_1 > v_1 * reward_2 + (1 - v_1) * penalty_2 \qquad \forall v_1 \in [0, 0.6[$$

where the value of v_1 represents the probability that the human is occupied and $(1-v_1)$ represents the probability that the human is engaged.

6.2.2 The Rational Human-MDPs for the Library of Q-values

Based on the latest robot action, whether it is coactive or not, the augmented POMDP is responsible for analyzing the human's action ($z \in Z$) as a key to better understand the collaborative situation. As shown in the augmented POMDP model (Chapter 5), the Q-value of a human action is a key to know if the human is interested or not in the shared task. This will help the robot to decide, in the next time-step, whether to start/keep acting coactively in order to incite the human or start/keep co-achieving the task with the human. Moreover, if the robot's latest action was coactive, the Q-value of the human action will clarify the success or failure of the coactive/guiding behavior.

Sociology studies can be applied in this work to create a library of human action values. This can be done by observing the human while achieving the task of handing object and later evaluate his chosen actions toward his real interest. The created library will be similar to the Q-Values library and will help in calculating the observation function of the CDM.

Since we were not able to do the sociology studies, this chapter describes how we used instead two Human-MDPs to generate the library of human action values (Q-values). Each of the Human-MDPs calculates an optimal policy of a simulated human with a different intention, one for a human who is interested in the collaboration and who is engaged in the shared task and another for a human who is occupied. The resulted library can be accessed to know the value of the human action toward one possible intention ($Q_{engaged}(s^h, a^h), Q_{occupied}(s^h, a^h)$). The library of Q-values is used in Equation 6.2 to calculate the observation function of the Coactive Augmented POMDP model.

The two Human-MDP models are exactly similar except for the reward function.

$$MDP^h_{engaged} = \left\langle S^h, A^h, T^h, R^h_{engaged} \right\rangle, \qquad MDP^h_{occupied} = \left\langle S^h, A^h, T^h, R^h_{occupied} \right\rangle.$$

To simulate a rational human behavior to achieve a task, the Human MDP state should represent all information related to the human, the task (receive an object) and the environment. Therefore, the state $s^h \in S^h$ holds human variables, task variables and a variable that represents the fact that the human is being incited/guided by the robot to successfully achieve the task. Therefore,

$$S^h = S_h \times S_t \times incited,$$

where $incited \subset A_c$ represents the fact that the robot tried to incite the human in its last action. A Human MDP state $s^h \in S^h$ is created from the POMDP state $s \in S$ and the latest robot action if it was a coactive action.

The set of actions A^h includes all possible human actions that are related to the task achievement and can be received by the POMDP model as observations.

The transition function T^h transitions the human and task variables depending the problem to solve.

For the Human MDP of a simulated human that is engaged and collaborating to take the object, the reward function $R_{engaged}$ rewards actions in favor of the collaboration and penalizes all actions that show disinterest of the human in collaborating. For the Human MDP of a simulated busy human, the reward function $R_{occupied}$ rewards actions in favor of not collaborating and penalizes actions that shows the human interest in collaborating especially responding properly to the robot guidance.

In the next section, we present an example of human robot collaboration and use it to illustrate how a CDM can be instantiated. We note that the chosen variables and calculated functions are example related, however, the general CDM model presented in this section which is based on an augmented POMDP can be used for similar human robot collaboration tasks where coactive robot actions are needed.

6.3 Illustrative Scenario: Handing Over an Object

A collaborative task was chosen as example where the robot is supposed to hand over an object to a person with the assumption that this person has previously asked for the object and will be handed the object while sitting on the chair. In this example, the robot should lead the human to sit on the chair and insure that the human is engaged in taking the object by making sure that the human is looking at the robot and is aware that the robot is willing to hand him the object. Meanwhile, the robot advances his arm towards the human to reach a position that facilitate the task but respect the secure and personal space of the human. Finally, when the human holds the object and is ready to receive it, the robot releases the object to him.

This section describes the corresponding CDM model as an augmented POMDP in addition to the Human MDPs that are defined by empathy and used to calculate the observation function of the augmented POMDP.

6.3.1 The CDM for Handing Over an Object Scenario

The Sets of States, Actions and Observations

For the handing object task, the robot's physical state regarding the interaction includes two variables. The first is the robot's *hand_position* which can be in initial position (close to the robot), wait position (middle) or extended position (close to the human). The waiting position permits the robot to wait for new information about the human to better decide whether to extend its arm to hand the object or to go back to its rest position. The second variable is the robot's *hand_situation* which represents the fact that the robot's hand is ready to release the

object or not. Therefore,

$$S_{cr} = hand_position \times hand_situation \qquad \text{where,}$$

$$hand_position \in \{initial, wait, extended\} \quad \text{and} \quad hand_situation \in \{not_ready, ready\}.$$

The state of the human includes three variables. The first is about the human presence which can be sitting on the chair, standing close to the robot or far away from the interaction area. The second variable concerns the human's sight direction which can be looking at the robot or looking elsewhere. Finally, the hidden (non-observable) variable of the state called *engagement* which concerns the fact that the human is engaged to achieve the collaborative task or occupied with something else. Therefore,

$$S_h = presence \times looking \times engagement \qquad \text{where,}$$

$$presence \in \{sitting, nearby, far_away\}, \qquad looking \in \{at_robot, otherway\} \quad \text{and,}$$

$$engagement \in \{engaged, occupied\}.$$

The state of the task consists of one variable that represents the agent that is currently holding the object,

$$S_t = object_holder \qquad \text{where,}$$

$$object_holder \in \{robot, human\}.$$

Following the assumption that the human has asked for the object prior to the beginning of the task, the initial belief state is chosen with a high probability that the human is interested and engaged in achieving the task. During the mission, however, the POMDP policy investigates the engagement variable value and, from there, directs the robot to the final state. In case ($engagement = engaged$), the final state includes ($hand_position = initial, object_holder = human$), however, if the ($engagement = occupied$), the final state includes ($hand_position = initial, object_holder = robot$).

Physical actions concerning the robot's hand movements which include: move hand to wait position, move hand back to initial position, move hand to extend position, move hand back to wait position, release object from hand or do nothing. Therefore, $A_t = \{go_wait, back_to_initial, extend, back_to_wait, stop, release, nothing\}$. The soft level coactive actions includes ping sit (ps), ping look (pl) and ping take (pt). The strong level includes strong ping sit (sps), strong ping look (spl) and strong ping take (spt). The ping sit and strong ping sit actions are to incite the human to sit on the chair. Ping look and strong ping look are to incite the human to look at the robot which is a sign of awareness and acceptance of the task advancement. Ping take and strong ping take are to incite the human to reach the hand of the robot to catch the object. The robot would try to incite the human to take the object (pt, spt) once the human is sitting on the chair and showing willingness to take the object and the robot extends its hand to

an accepted and comfortable position which allows the human to reach the object. Therefore, $A_c = \{ps, sps, pl, spl, pt, spt\}$.

Accordingly, the set of robot's actions is defined as:

$$A = \{go_wait, back_to_initial, extend, back_to_wait, stop, release, nothing, ps, sps, pl, spl, pt, spt\}.$$

The first group of observations is related to the human existence: human entered the visibility space and is nearby the robot (ob_nearby), human left the visibility space and is far from the robot (ob_far_Away) or human entered the task space which means he is sitting on the chair ($ob_sitting$). The second group of observations is related to the human awareness about the task advancement: human shares task context by looking at the robot ($ob_looking_at_robot$), human ignores task context by looking away from the robot ($ob_looking_away$) or human is ready to catch the object (ob_hand_ready). Finally, an observation related to the task progress which consists of observing the object in the human's hand ($ob_object_with_human$).

All observations are detected by different modules (localization, object detection, ...) and are sent as signals to the robot's system in a way that at each time-step the CDM receives one of the following possible observations:

$$Z = \{ob_nearby, ob_far_Away, ob_sitting, ob_looking_at_robot, ob_looking_away,$$

$$ob_hand_ready, ob_object_with_human, ob_nothing\}.$$

It is possible that more than one observation is generated in the same time-step. A solution for such situations is presented in Section 6.4.2.

The Transition Function T

The transition function is represented in Equation 6.1 where actions ($go_to_wait, back_to_initial$, $extend, back_to_wait$) deterministically and accordingly transition the value of $hand_position$. For example:

$$pr(hand_position' = extended | hand_position = wait, a = extend) = 1,$$

$$pr(hand_position' = wait | hand_position = extended, a = back_to_wait) = 1.$$

One of the following variables can change value at each transition: $looking$, $presence$ and $handSituation$. This represents the change in the current state caused by the possible received observation.

The robot's action $release$ can transition the value of the variable $object_holder$ from $robot$ to $human$ or transition the system to a $fail$ state. This depends on whether or not the system observes $ob_object_with_human$ after applying the action $release$.

The Observation Function O

For simplicity and without loss of generality, the function is defined deterministically over the non hidden human related variables of the state, those are (*presence*, *looking*, *hand_situation* and *object_with*). For example:

$$pr(z = ob_sitting|a, presence' = sitting) = 1,$$

$$pr(z = hand_ready|a, hand_situation' = ready) = 1,$$

$$pr(z = ob_object_with_human|a, object_holder' = human) = 1.$$

The effect of the observed the human action on the non observable hidden state *engagement* is represented in Equation 6.2.

The Reward Function R

In addition to the rewards described in Equation 6.3, the handing object model assigns a penalty when the robot's hand is extended and the human is occupied. This encourages the robot to go back to wait position when the human is occupied. For final states, a small reward is assigned for *back_to_initial* action when the human leaves the visibility space and a higher reward for the same action when the object is released to the human.

$$R(s,a) = \begin{cases} penalty_1 < 0 & \text{if } (engagement = engaged \text{ and } a \in \{ps, pl, pt\}) \\ penalty_2 < penalty_1 & \text{if } (engagement = engaged \text{ and } a \in \{sps, spl, spt\}) \\ reward_1 > 0 & \text{if } (engagement <> engaged \text{ and } a \in \{ps, pl, pt\}) \\ reward_2 > reward_1 & \text{if } (engagement <> engaged \text{ and } a \in \{sps, spl, spt\}) \\ penalty_3 < penalty_2 & \text{if } (hand_position = extended \text{ and } engagement <> engaged) \\ reward_1 > 0 & \text{if } (presence = far_away \text{ and } a = back_to_initial) \\ reward_3 \gg reward_2 & \text{if } (object_holder = human \text{ and } a = back_to_initial) \end{cases}$$

6.3.2 Human MDPs for Handing Over an Object Scenario

The Set of States S^h

As described in Section 6.2.2 the Human MDP state include human's states, task's state and a state that represent the fact that the human was incited by the latest robot action. Therefore,

$$S^h = presence \times looking \times pinged \times object_holder,$$

where $(looking, presence, objectOwner)$ have the same values as previous and *pinged* represents the type of ping the human received if the last robot action was a ping,

$$pinged \in \{ps, pl, pt, sps, spl, spt, no_ping\}.$$

The Set of Actions A^h

The possible human actions, that are related to the task achievement and can be received by the POMDP model as observations, are:

$$A^h = \{sit, stand, go_away, look_at_robot, look_otherway, take_object, do_nothing\}.$$

The Transition Function T^h

The Human MDP state variables *presence*, *looking* and *object_holder* transition deterministically according to the associated human actions $\{sit, stand, go_away\}$, $\{look_at_robot, look_away\}$ and $\{take_object\}$. However, the variable *pinged* has the possibility to change to any possible *pinged* value uniformly. Non possible next ping values include situations like ping sit when the human is already sitting.

The Reward Function R^h

Equation 6.4 describes the reward function for an engaged human in the collaboration. It rewards actions in favor of collaborating and well receiving the object from the robot. Such actions present the intention of the human of taking the object as in sitting on the chair and looking at the robot, whether by himself or incited by the robot's guidance. However, the $R_{engaged}$ function penalizes all actions that show disinterest of the human in taking the object as looking away from the robot or standing or leaving the interaction area. The following summarizes all situations:

$$R_{engaged}(s^h, a^h) = \begin{cases} r_1 > 0 & \text{if } (pinged = nothing \text{ and } a^h \in \{sit, look_at_robot, take_object\}) \\ r_2 > r_1 & \text{if } (pinged = pt \text{ and } a^h = take_object) \\ r_3 > r_2 & \text{if } (pinged = spt \text{ and } a^h = take_object) \\ r_4 >= r_3 & \text{if } (pinged = pl \text{ and } a^h = look_at_robot) \\ r_5 > r_4 & \text{if } (pinged = spl \text{ and } a^h = look_at_robot) \\ r_6 >= r_5 & \text{if } (pinged = ps \text{ and } a^h = sit) \\ r_7 > r_6 & \text{if } (pinged = sps \text{ and } a^h = sit) \\ p_1 < 0 & \text{if } (a^h = look_otherway) \\ p_2 < p_1 & \text{if } (a^h \in \{stand, do_nothing\}) \\ p_3 < p_2 & \text{if } ((a^h = go_away) \text{ or } \\ & (pinged <> nothing \text{ and } a^h = do_nothing)) \end{cases} \tag{6.4}$$

Equation 6.5 describes the reward function for a Human MDP simulating an occupied human. The function rewards actions in favor of not collaborating. Such actions presents the occupation of the human and his unwillingness to take the object as in not looking at the robot or leaving

the interaction area. The function also penalizes actions that shows the human interest in collaborating especially responding properly to the robot guidance/pings.

$$R_{occupied}(s^h, a^h) = \begin{cases} r_1 > 0 & \text{if } (a^h = look_otherway) \\ r_2 > r_1 & \text{if } (a^h \in \{stand, do_nothing\}) \\ r_3 > r_2 & \text{if } ((a^h = go_away) \text{ or} \\ & (pinged <> nothing \text{ and } a^h = do_nothing)) \\ p_1 < 0 & \text{if } (pinged = nothing \text{ and } a^h \in \{sit, look_at_robot, take_object\}) \\ p_2 < p_1 & \text{if } (pinged = pt \text{ and } a^h = take_object)) \\ p_3 < p_2 & \text{if } (pinged = spt \text{ and } a^h = take_object) \\ p_4 <= p_3 & \text{if } (pinged = pl \text{ and } a^h = look_at_robot) \\ p_5 < p_4 & \text{if } (pinged = spl \text{ and } a^h = look_at_robot) \\ p_6 <= p_5 & \text{if } (pinged = ps \text{ and } a^h = sit) \\ p_7 < p_6 & \text{if } (pinged = sps \text{ and } a^h = sit) \end{cases} \tag{6.5}$$

6.4 Experimental Results

The experiments of this chapter show results of the CDM model for handing object to the human. The Human MDPs were solved with classic Value Iteration algorithm for MDPs and the Coactive Augmented POMDP policy was calculated with the ZMDP solver in less than 30 minutes.

The initial belief state is described as:

$$b_0 = \{(initial \times not_ready \times nearby \times otherway \times engaged \times robot) = 95\%,$$

$$(initial \times not_ready \times nearby \times otherway \times occupied, human) = 0.05\%\}$$

6.4.1 Simulations

Tables 6.1 and 6.2 present results of simulations for two scenarios where a human user chooses a possible action at each time-step. The first shows results of the shared task with a human that was interested in taking the object, while the second shows results with a human that was occupied and not interested in taking the object at this time. The tables show at each step of the scenarios the current state of the robot, the human and the task, the belief over the human intention of taking the object, the chosen action from the POMDP policy and finally the observation received after the action was applied.

We detail, for example, steps 4,5 and 6 of Table 6.1. Before those steps the human showed clear interest in taking the object by sitting down and looking at the robot. The belief state in step 4 represents this fact with $pr(engaged) = 0.997$. The chosen action of the policy was

Step	presence	looking	hand_situation	hand_position	object_holder	engaged	occupied	action (a)	observation (z)
1	nearby	otherway	not_ready	initial	robot	0.95	0.05	-	sitting
2	sitting	otherway	not_ready	initial	robot	0.93	0.07	go_wait	at_robot
3	sitting	at_robot	not_ready	wait	robot	0.996	0.004	extend	nothing
4	sitting	at_robot	not_ready	extended	robot	0.997	0.003	pt	otherway
5	sitting	otherway	not_ready	extended	robot	0.5	0.5	spl	nothing
6	sitting	otherway	not_ready	extended	robot	0.05	0.95	back_wait	nothing
7	sitting	otherway	not_ready	wait	robot	0.52	0.48	spl	at_robot
8	sitting	at_robot	not_ready	wait	robot	0.98	0.02	extend	nothing
9	sitting	at_robot	not_ready	extended	robot	0.99	0.01	pt	ready
10	sitting	at_robot	ready	extended	human	0.99	0.01	release	with_human
11	sitting	at_robot	ready	extended	human	0.99	0.01	back_wait	-
12	sitting	at_robot	ready	wait	human	0.99	0.01	back_initial	-

Table 6.1: Results of the handing object simulation with an engaged human.

to incite the human to take the object with a soft ping take ($a_4 = pt$), which presents that the robot hand is extended and it is safe for the human to reach for the object. Meanwhile the system observes that the human interest was interrupted by something else ($z_4 = otherway$). At time-step 5, the updated belief state shows ambiguity after the observed interruption. The chosen action was to strong ping the human to look at the robot ($a_5 = spl$) which is an attempt to incite the human to continue the task. However, the human did not respond positively to this attempt where he kept looking other way ($z_5 = nothing$). At time-step 6, the updated belief state shows a lack of engagement from the human's side with $pr(engaged) = 0.05$ and the following action was for the robot to back up the hand position to the wait position. This decision does not mean that the robot ended the task, but it is a way to give the human the space to leave the chair without disturbing him. Moreover, the following steps in this scenario shows a second attempt of the robot to attract the human attention, which will end with success in steps 7 and 8 and successfully end the task with the object delivered to the human in step 10 and the final robot decision is to return the hand to the initial position.

Table 6.2 presents another case, where the human shows a strong lack of interest in the shared task, and after few robot attempts the human leaves the interaction area entirely ($z_7 = far_away$) which leads the robot to move its hand back to the initial position ($a_8 = back_to_initial$) and the system reaches an end state where the task is ended without being able to deliver the object.

We noticed during simulations situations in which the human stays close to the robot but is occupied doing something. The robot in such case keeps trying to incite the human to collaborate until the situation ends in one of the final states (object handed or human leaving the interaction area). To handle the negative insistence of the robot, the decision *back_to_initial* is taken and the mission is forced to end when the belief state reaches more than 96% occupied.

Step	presence	looking	hand_situation	hand_position	object_holder	engaged	occupied	action (a)	observation (z)
1	nearby	otherway	not_ready	initial	robot	0.95	0.05	-	nothing
2	nearby	otherway	not_ready	initial	robot	0.58	0.42	ps	nothing
3	nearby	otherway	not_ready	initial	robot	0.12	0.88	sps	sitting
4	sitting	otherway	not_ready	initial	robot	0.57	0.43	go_wait	at_robot
5	sitting	at_robot	not_ready	wait	robot	0.98	0.02	extend	otherway
6	sitting	otherway	not_ready	extended	robot	0.9	0.1	pl	stand
7	nearby	otherway	not_ready	extended	robot	0.29	0.7	sps	far_away
8	far_away	otherway	not_ready	extended	robot	0.0	1.0	back_wait	nothing
9	far_away	otherway	not_ready	wait	robot	0.0	1.0	back_initial	nothing

Table 6.2: Results of the handing object simulation with an occupied human.

6.4.2 Integration on Real Robots

The work presented in this chapter was realized in collaboration with LAAS CNRS laboratory (Laboratoire d'Analyse et d'Architecture des Systèmes). The described decision model was an outcome of this collaboration and it was applied and tested on LAAS robots (Jido, PR2) for the scenario of handing over an object to the human. The used architecture on the robots is an instance of the LAAS general architecture [Alami *et al.*, 1998].

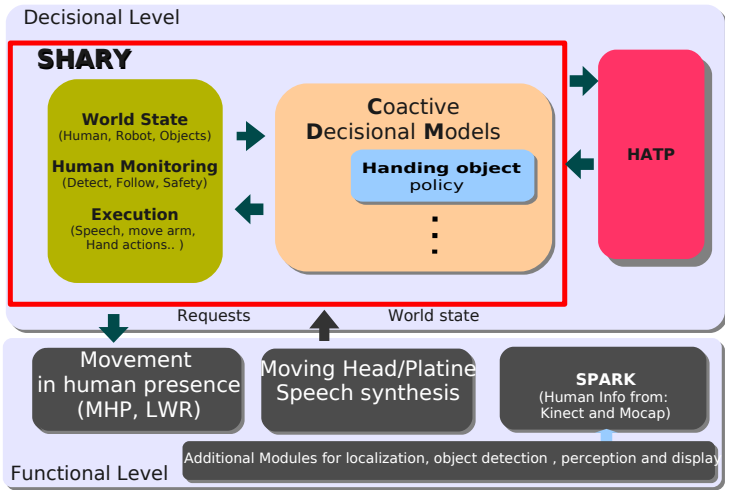


Figure 6.1: System Architecture: an Instance of LAAS Original Control Architecture.

Figure 6.1 shows the adapted architecture that include a set of Coactive Decision Models (CDM) for different collaboration tasks. The decisional layer consists of two main components: SHARY (Supervision for Human Aware Robot Ynteraction), that is in charge of cooperative task achievement and HATP (Human Aware Task Planner), a high level task planner.

SHARY [Clodic *et al.*, 2008] is a task supervision and execution controller. It has been adapted here to invoke its associated decisional scheme from the set of "Coactive Decisional Models" whenever it has to achieve a collaborative human-robot shared task.

HATP [Montreuil *et al.*, 2007] [Alili *et al.*, 2009] has the capacity to synthesize plans for human-robot teamwork that respect social conventions and that favor acceptable collaborative behaviors based on a representation of the human abilities, preferences and desires.

The original LAAS control architecture has addressed several aspects of Human-Robot Collaboration, a high level task planner [Clodic *et al.*, 2008] which generates partially ordered plans involving coordinated and collaborative human and robot tasks, the HRI-enabled task supervision and execution framework (SHARY) [Clodic, 2007], inspired from joint intention theory [Cohen et Levesque, 1991] and more particularly the work of Clark [Clark, 1996] on teamwork and communication acts that support collaborative task achievement [Clodic *et al.*, 2007]. The supervision system is not only responsible for the refinement and the correct execution of the robot task, but also for the appropriate set of communication and monitoring activities within and around task realization. It is also in charge of monitoring human commitment and activities in order to provide appropriate response based on the current context. The robot control architecture includes also a motion synthesis layer that takes into account the safety and the preferences of humans [Koay *et al.*, 2007] when they share space with a robot [Sisbot *et al.*, 2010].

The work presented in this chapter builds on these capabilities for collaborative task achievement and consists of formalizing collaborative tasks achievement as a "Coactive Decision Models" using augmented POMDPs.

The handing over an object task was applied with PR2 (Figure 6.2) and Jido (Figure 6.3). The high level robot decisions related to the collaboration with the human were fed by the CDM policy. However, the functional level was responsible for physically applying those actions with respect with the human's existence. The arm movements were controlled by a motion planner that produces human-friendly motion [Sisbot *et al.*, 2010] validated through user studies [Dehais *et al.*, 2011].



Figure 6.2: Scenario handing object to human with PR2.

During each time-step of the task (3 seconds), all received information about the human and the task are stacked to be treated at the end of the time-step in the order they were received. Each information that is different than the current state is sent as observations to the CDM



Figure 6.3: Scenario handing object to human with Jido.

model. Then the belief update function updates the belief state using the received observation and the last decided robot action. In case of several observations at the same time-step, the belief update function is called as many times as needed with a robot action *do_nothing* unless for the last observation where it uses the latest chosen robot action by the policy.

Screen shots were taken while performing similar scenarios to ones described in Tables 6.1 and 6.2 with PR2 Robot (Figure 6.2) and Jido Robot (Figure 6.3).

6.5 Discussion

This contribution is an initial step in using Coactive Decision Models to create a library of coactive companion robot policies for daily collaborative tasks. Such library of N POMDP policies will have the complexity of solving one CDM collaborative model.

This chapter also focused on the robot's decisions and behavior when collaborating with a human. The approach uses an Augmented POMDP to better understand the human intentions. The proposed Coactive Decision Model permits the robot not only to adapt to human's inferred intentions but also to try to coactively incite him to collaborate toward the success of the shared task and facilitate his joint activity.

In the described model, as long as the human is in the interaction area (did not leave the room) the robot will keep trying to incite him to get involve in the collaborative task. This behavior is unacceptable if the human is busy doing something else. Therefore, to better control the coactive actions, we will be working in the future on adding another hidden state to the CDM which represent the level on which the human is annoyed. This level increases when the robot tries to incite the human to collaborate and decreases when the human reacts positively to the incitement.

The model is far from perfect as it is not evident to parameter the reward functions for the Human MDPs and the CDM model. The illustrative scenario in this chapter was modeled manually, it will be more interesting to learn them automatically. The experimental results lack detailed evaluations as they were tested and simulated by the developers. Further work includes user studies to show the validity of the interaction with strangers.

Chapter 7

A Decision Model for Adaptive Interaction-Type Selection

Contents

7.1 Motivation 110
7.2 Definitions 111
7.3 Model 112
7.3.1 Level 1: The Human Intention Estimator (IE) 114
7.3.2 Level 2: The Interaction-Class Selector (IS) 115
7.3.3 Level 3: Choosing a Task and Applying a Type of Interaction . . . 117
7.4 A Companion Robot at Home Scenario 119
7.5 Experimental Results 120
7.6 Discussion 122

This chapter addresses two problematics. First, establishing a system for a companion robot that is capable of switching between different types of interaction according to the human's needs. Second, modeling the system using a framework that outperforms Partially Observable Markov Decision Processes (POMDPs) for large-scale problems. For this end, the chapter describes a unified model of Human-Robot Multi-type Interaction (HRMI). The objective is to observe the human's behavior and try to predict/estimate the human's intention/need and therefore react appropriately (assist, cooperate, collaborate, ...). As discussed in Section 4.2, POMDPs are largely used by the community of collaborative and assistive Human-Robot Interaction (HRI). However, results presented in Chapter 6 shows that employing augmented-POMDPs to solve an HRMI problem of large-scale applications is intractable. We present an approach to overcome this limitation by dividing the problem of HRMI into three levels: first, estimate the human intention; second, select the appropriate type of interaction; last, choose one of the

Part of the work presented in this chapter will appear in the proceedings of the 5th European Conference on Mobile Robots [Karami et Mouaddib, 2011]

pre-calculated policies with respect to the human intention and the needed type of interaction. Therefore, the HRMI model includes: a belief update model concerning the intention of the human using multiple human MDP models defined by empathy for each specific human task; an MDP model that supports the robot's decision making towards the human needs by selecting the appropriate interaction type; finally, an algorithm that first defines the task to accomplish by the robot, then chooses the pre-calculated policy to achieve it with respect to the type of interaction. The chapter, afterward, presents some interesting performance and scalability analysis and an implementation on a real robot showing the feasibility of the approach using a representative scenario inspired from [RoboCup@home, 2011].

7.1 Motivation

Nowadays, a vast research interest is progressing towards companion robots that share their environment with people like elders in their home or in assisted living facilities. During daily and long-term interactions, a companion robot can be of help to a human in different ways. This depends on the human possible incapabilities or preferences. More information about the human's desires and constraints would more probably lead the companion robot to offer the good help when needed.

We consider a Human-Robot Multi-type Interaction (HRMI) model where the companion robot observes a human and acts appropriately for the best help. Indeed, to best address this problem, the robot must be able to: estimate the human's intention (what task is he trying to achieve?), then infer the kind of help (interaction) that he might need, then reason the best action to help the human. In addition to being able to accurately and quickly infer the human's intentions, the companion robot should be able to adapt to the human's possible change in desire over time.

Though POMDPs offer an appealing theoretical framework, finding a tractable approximate solution for real size applications is highly complicated. We noticed in Chapter 6 that the proposed augmented POMDP model was able to solve Human-Robot Cooperation problems of very small sizes that are not sufficient to describe real life applications. An HRMI environment can similar to the Human-Robot Cooperation environment, however, tasks related to collaboration and assistance are added to the problem which cause an augmentation in the representation of the environment, human intentions, possible actions to do. This shows the inability of augmented-POMDPs to solve HRMI problems of real life applications.

Therefore, instead of using POMDPs, we propose an outperforming model that solves the problem on three levels without loosing the adaptability and the performance of handling uncertainty of POMDPs. First, the system observes the human action and estimates his intention using multiple Human MDPs defined by empathy and a Hidden Markov Model (HMM) where the hidden state corresponds to the human intention. Second, an MDP model helps in inferring the kind of help the human needs using the belief state updated by the HMM. Third, knowing the kind of interaction needed to help the human and his intention, the robot applies an action

from previously computed policies (a database containing a policy for each task achieved under each type of interaction if possible).

7.2 Definitions

This section includes various definitions related to the proposed HRMI model. First, the chosen types of interaction are presented, followed by the definition of a **task** and how it can be related to one or more types of interaction. Finally, the description of the human intention and what it represents.

We define I as a set of **interaction classes** $I = \{IC_1, IC_2, IC_3, \dots\}$. Each class of interaction differs in the way the companion robot helps the human. As described in Section 2.2.2, the HRI literature distinguishes three types of interaction (which will be subjects of this chapter): Cooperation CP, Assistance As and Collaboration CL. For reasons explained in the next section, we add Confirmation CO to the types of interaction.

$$I = \{CP, AS, CL, CO\}$$

The robot holds a list of all possible **tasks** that can be done by the human and/or the robot. A task $tk \in TK$ can be defined, for example, as $tk = \langle context, agent, policy, time, type \dots \rangle$. *context* holds information about the task regarding the context of the problem; it might include relations and dependencies between tasks, conditions to achieve, related databases to access or update information. $agent \subset \{CR, H, H \vee CR, H \wedge CR\}$, is the possible agent that can achieve the task; a task can be achieved uniquely by the companion robot, uniquely by the human, by any of them or by both of them. $policy \in \{possess, lack\}$ is the fact that the robot possesses or lacks a policy or a manual of how to assist the human if he needs assistance to do the defined task. *time* is the expected needed time-steps to achieve the task. Finally, a task is considered of one or more types of interaction following the rules mentioned in Table 7.1, $type \subset I$ include all possible types of interaction that can be used to achieve this task.

Type of interaction	Condition
Cooperation	$tk = \langle *, \{CR, CR \vee H\}, *, *, \{CP\} \rangle$
Assistance	$tk = \langle *, \{H, CR \vee H\}, possess, *, \{AS\} \rangle$
Collaboration	$tk = \langle *, \{CR \wedge H\}, *, *, \{CL\} \rangle$
Confirmation	$tk = \langle *, \{H, CR \vee H\}, possess, *, \{CO\} \rangle$
Confirmation	$tk = \langle *, \{CR \wedge H\}, *, *, \{CO\} \rangle$

Table 7.1: Relevance between tasks and different interaction classes. " * " represents any possible value.

The set of tasks can be represented as $TK = TK_h \cup TK_{cr}$, where the domain of the variable "*agent*" for TK_h is $\{H \vee CR, H, H \wedge CR\}$, and for TK_{cr} is $\{H \vee CR, CR, H \wedge CR\}$.

The human's **intention** can be one of the human's tasks or nothing at all, $intention \in TK_h \cup \{do_nothing\}$.

7.3 Model

This section presents the HRMI decision control model that allows a robot to observe a human and detects his possible intention (intended task) and infer any possible need of assistance or collaboration that might require the robot's involvement. The companion robot system considers the human needs as priority (assistance, collaboration), otherwise helping a human by doing a cooperative task comes in second.

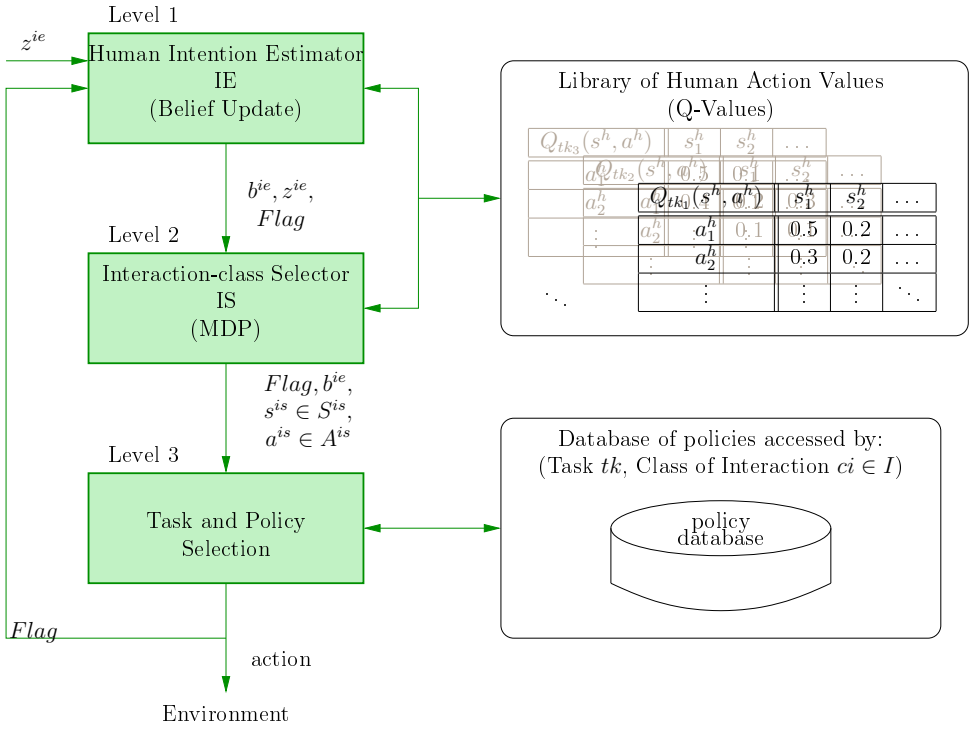


Figure 7.1: The HRMI Decision Model.

Figure 7.1 presents the three levels of the system. In the first level, at each time-step, the human Intention Estimator (IE) observes the human action (z^{ie}) and updates the estimation over the possible human intentions ($update^{ie}(b'^{ie}|b^{ie}, z^{ie})$). The IE uses a library of human action value functions (Q-Values) calculated off-line. In the second level, the Interaction-class Selector (IS) receives the updated estimations and uses them to efficiently switch the type of interaction in a way that matches the human desires. The third level uses all information from the first and second levels to choose a task for the robot based on the estimated intention and the chosen class of interaction. By selecting the policy of the robot's task from the policies database, the robot applies the appropriate action from this policy. Afterwards, the system hands the control back to the IE and along with the new observed human action, a new time-step begins. We note

that levels 2 and 3 are responsible for adapting the robot's behavior according to any change in the human intention estimations.

The high level algorithm of the system is as the following:

1. The system observes the human action z^{ie}.
2. Level 1: The belief update function updates the belief over the human intention,

$$update^{ie}(b'^{ie}|b^{ie}, z^{ie}).$$

3. Level 2(a): The Interaction-class Selector creates the state s^{is} from the current belief b^{ie}.
4. Level 2(b): The Interaction-class Selector calls the policy π^{is} to choose an interaction class.
5. Level 3(a): The Task Selector algorithm chooses a task to achieve by the robot $tk \in TK_{cr}$.
6. Level 3(b): The appropriate policy from the database is used to decide the robot's action.
7. The robot applies its action and a new time-step begins.

The following contains few details related to the model and the decision control (confirmation and the control flag) and the rest of the section will present a detailed description of the three levels of the decision model.

Confirming with the human

The fact that the human intention is of type AS does not necessarily prove that the human needs assistance to achieve his task (he is possibly capable of achieving it by himself). In order to avoid unneeded gestures of assistance or collaboration which might confuse the interaction with the human, a confirmation policy is added to the possible robot tasks ($TK_{cr} \cup$ confirm). This allows the robot to ask the human for a confirmation once the system detects a possible need of assistance (will be explained later in this section) or once the estimated human intention is of type CL. The confirmation policy is simply a query question as: *(Do you need my help in doing task 'x'?)* and the human answer can simply be a *yes* or *no*. There is no need of any kind of confirmation if the human intention is of type CP, in such a case, the robot will choose to cooperate by doing another task of type CP.

The Flag

The $Flag$ can be considered as a messenger between the system levels. It represents the current class of interaction or other information that can affects the decided type of interaction between the robot and the human. The $Flag$ can be assigned with a new value in Levels 2 and 3. If the observed human action, which is received from Level 1, contains a human answer to a query *(yes,no)*, Level 2 updates the $Flag$ value with the human answer. Also, when a possible need of assistance is detected in Level 2, the $Flag$ value is updated to point this fact. Level 3 updates

the $Flag$ value depending on the class of interaction of the chosen robot task. Possible $Flag$ values can be:

- $NeedAssistance$: a possible need of assistance is detected.
- $DoingX$: the robot follows a policy to achieve a task of type X where: $X \in I$.
- $Confirm$: the robot is confirming with the human his need of help.
- Yes : the human confirmed with (yes).
- No : the human confirmed with (no).

7.3.1 Level 1: The Human Intention Estimator (IE)

The IE is the part of the system that observes the human actions and tries to translate those actions into intentions using the library of human action values (Q-Values). In a similar way to the augmented POMDP described in Chapter 5, the IE uses a number of Human MDPs (equal to the number of tasks involving the human TK_h) which are defined by empathy and calculated off-line to produce the library of Q-Values.

The IE belief update function is used to calculate the belief over the human intentions at each time-step after observing his latest action (z^{ie}). This function is defined using a Hidden Markov Model (HMM) where the human intention is represented as the hidden part of the model.

The IE is represented by a tuple $\langle S^{ie}, Z^{ie}, T^{ie}, O^{ie}, b_0^{ie} \rangle$. The set of states $S^{ie} = \langle S_{hc}^{ie} \times S_{hi}^{ie} \rangle$ represents the human context variables including information about the human regarding the context of the problem (S_{hc}^{ie}) and the human intention ($S_{hi}^{ie} \subset TK_h$). The possible observations are the possible related human actions: $Z^{ie} \subset A^h$. The transition function $T^{ie} = Pr(s'^{ie}|s^{ie})$ gives the probability of ending in state s'^{ie} knowing the current state s^{ie} where $s^{ie}, s'^{ie} \in S^{ie}$, $s_{hi}^{ie}, s_{hi}'^{ie} \in S_{hi}^{ie}$ and $s_{hc}^{ie}, s_{hc}'^{ie} \in S_{hc}^{ie}$.

$$pr(s'^{ie}|s^{ie}) = \begin{cases} pr(s_{hc}'^{ie}|s^{ie}) \times p & \text{if } s_{hi}^{ie} = s_{hi}'^{ie} \\ pr(s_{hc}'^{ie}|s^{ie}) \times \frac{1-p}{|s_{hi}'^{ie}|} & \text{if } s_{hi}^{ie} \neq s_{hi}'^{ie} \end{cases}$$

The property of memorizing the human's intention through the transition function (T^{ie}) is introduced by adding a probability of $p \in [0, 1]$ that the human will keep his intention in the next time-step and a probability of $1 - p$ that he will change his intention. However, the transition probabilities concerning the human's context variables s_{hc}^{ie} depends on the context of the problem. $O^{ie} = pr(z^{ie}|s^{ie}, s'^{ie})$ is the observation function which is computed using the library of Q-Values. Equation 7.1 describes the probability of each observation knowing the current state (s^{ie}) and the next state (s'^{ie}). λ in Equation 7.1 is a normalizing factor and s^h represent the human MDP state derived from the current IE state (s^{ie}) which can transition via the observed human action to the human MDP state (s'^h) derived from the next IE state (s'^{ie}).

$T^h_{s'^{ie}_{hi}}$ is the transition function of the human MDP corresponding to the intended human task s'^{ie}_{hi} (see Section 5.3.3 for details).

$$pr(z^{ie}|s^{ie}, s'^{ie}) = \lambda Q_{s'^{ie}_{hi}}(s^h, z^{ie}), \qquad \text{where: } T^h_{s'^{ie}_{hi}}(s^h, z^{ie}, s'^h) > 0, \tag{7.1}$$

The Human Intention Estimator receives at each time-step the human action z^{ie} and uses the belief update function (Equation. 7.2), where δ is a normalizing factor, to calculate the new belief state (b'^{ie}) and gives it to the Interaction-class Selector. Any possible change in the human intention is usually reflected on his actions and therefore reflected on the new belief over the human intentions (b'^{ie}).

$$b'^{ie}(s'^{ie}) = \delta \sum_{s^{ie} \in S^{ie}} pr\left(z^{ie}|s^{ie}, s'^{ie}\right) pr\left(s'^{ie}|s^{ie}\right) b^{ie}(s^{ie}) \tag{7.2}$$

7.3.2 Level 2: The Interaction-Class Selector (IS)

The IS is responsible for choosing the appropriate type of interaction with respect to the estimated human intention. It starts by analyzing the received belief state (b^{ie}) for better estimation of the human needs. Then it uses a pre-calculated policy MDP^{is} to choose one of the different interaction classes: Cooperation CP; Assistance AS; Collaboration CL.

The IS model

The IS model is represented with a Markov Decision Process such as $MDP^{is} = \left\langle S^{is}, A^{is}, T^{is}, R^{is} \right\rangle$. The set of states $S^{is} = \left\langle S^{is}_{hc} \times S^{is}_{hi} \times S^{is}_{Flag} \right\rangle$ includes the variables: S^{is}_{hc} that represents human context related information, $S^{is}_{hi} \subset TK_h$ that is the human intended task and $S^{is}_{Flag} \subset Flag$ that holds the $Flag$ value. The set of actions $A^{is} = \{doCP, doAS, doCL, doConfirm\}$ includes the decided interaction class or a confirmation. The IS transition function $T^{is} = pr(s'^{is}|s^{is}, a^{is})$ is calculated in two parts as in Equation 7.3.

$$T^{is} = pr(s'^{is}_{Flag}|s^{is}, a^{is}) * pr(\{s'^{is}_{hc}, s'^{is}_{hi}\}|s^{is}) \tag{7.3}$$

The first part is responsible for the probability of changing the $Flag$ variable knowing the action $a^{is} \in A^{is}$. The second part is responsible for the transition of the human context variables (s^{is}_{hc}) and human intention (s^{is}_{hi}) (Equation 7.4), where $s^{is}, s'^{is} \in S^{is}$, $s^{is}_{hi}, s'^{is}_{hi} \in S^{is}_{hi}$ and $s^{is}_{hc}, s'^{is}_{hc} \in S^{is}_{hc}$. The property of memorizing the human's intention is introduced by adding the probabilities $p/1-p$ that the human will keep/change his intention in the next time-step.

$$pr(\{s'^{is}_{hc}, s'^{is}_{hi}\}|s^{is}) = \begin{cases} pr(s'^{is}_{hc}|s^{is}) \times p & \text{if } s^{is}_{hi} = s'^{is}_{hi} \\ pr(s'^{is}_{hc}|s^{is}) \times \frac{1-p}{|s'^{is}_{hi}|} & \text{if } s^{is}_{hi} \neq s'^{is}_{hi} \end{cases} \tag{7.4}$$

The reward function $R^{is}(s^{is}, a^{is})$ is defined in a way where a confirmation of collaboration is rewarded if the human intention is of type CL. However, a confirmation for assistance is only rewarded if $Flag = PossibeNeed$. $DoCL, DoAS$ are rewarded when human intention is of type CL, AS respectively and $Flag = Yes$. $DoCP$ is rewarded else-wise.

Employing the IS policy for selecting the type of interaction

The IS MDP policy π^{is} is calculated off-line. Once the system is on-line the policy is called to choose the best interaction class. As the state of the human is not totally observable, the IS state is created on-line as described in Algorithm 5.

Algorithm 5: The creation of the ICS state

Input : $Flag$, $b^{ie}(s), z^{ie}$, previous_state_intention
Output: s^{is}

$st^* = \underset{s \in S^{ie}}{\text{argmax}}(b^{ie}(s))$;
$st_2^* = \underset{s \in S^{ie} - \{st^*\}}{\text{argmax}}\ (b^{ie}(s))$;

`// Check for human answers`
if $Flag = Confirm$ *and* $z^{ie} = yes$ **then**
  $Flag = yes$;
else if $Flag = Confirm$ *and* $z^{ie} = no$ **then**
  $Flag = no$;
else if $Flag = Confirm$ *and* $z^{ie} \neq yes$ *and* $z^{ie} \neq no$ **then**
  $Flag = no$;
  set-timer;

`// initialize` s^{is} `with argmax` S^{ie}
Initialize $s^{is} =< st^*_{hc}, st^*_{hi}, Flag) >$
if $Flag = yes$ **then**
  $s^{is} =< st^*_{hc}, previous_state_intention, Flag) >$;
else if $Flag = no$ **then**
  set previous_state_intention to done
else if $(\underset{\beta \in S^{ie}}{\text{variance}}(b^{ie}(\beta)) > \underset{\alpha \in \{st^*, st_2^*\}}{\text{variance}}(b^{ie}(\alpha)))$ **then** `// ambiguity in` b^{ie}
  forall $(tk \in TK_h)$**do**`// check for need of assistance`
    if $(\{AS\} \subset tk_{type}$ *and tk is-not-done)* **then**
      if $(st^*_{hc}$ *relates with* $tk_{context})$ **then**
        $s^{is} =< st^*_{hc}, tk, NeedAssistance >$;

Using the belief state that the IS receives from level 1 ($b^{ie}(s)$) at each time-step, the algorithm creates the corresponding IS state s^{is} that will be used by the IS MDP policy π^{is} to decide the interaction type ($\pi^{is} : s^{is} \rightarrow a^{is}$).

Lines 1 and 2 of the algorithm define the first and second dominant states of the belief state. Line 15 tests if the dominant state st^* has a distinct belief probability from the other states. This is done by comparing the variance between the first two dominant states and the variance between all states. If the variance between the probabilities of the two dominant states is higher than the variance between all states probabilities then the dominant state is considered highly distinct and the algorithm can recognize a distinct intention. When observing a human's answer to a $doConfirm$ action (from the previous time-step) the algorithm will change the $Flag$ value to match the human's answer *yes* or *no* (Lines 3:6). In case of a confirmation action by the robot without an answer from the human, the algorithm in lines (7:9) sets the task to done and a timer to reset the task to not done after a certain time, this choice was made to avoid annoying the human with an endless series of confirmations.

If the latest robot action was a confirmation for assistance or collaboration, the algorithms verifies the human answer ($Flag$) in lines (10:14). If the human answer was a *yes* then the intended task is set to be the human intention in the previous time-step (the task that the robot was confirming about). However, if the human answer was a *no*, then the system supposes that the human is not needing help in doing the task, the task is set to done. This choice was made to prevent an endless series of confirm actions when the human's answer is no. It is possible to set a timer in order to switch such tasks after a while as not done to be able to detect future need of help (collaboration, assistance) for any of them.

Considering a possible situation where the human intention is of type AS and the human actions are indistinct and not actually of advantage to achieving the intended task. A consequence of such situation is a certain ambiguity in the belief state over the human intentions. This ambiguity will be detected in line 15 of Algorithm 5. The ambiguity in the belief state, however, might be a sign of one of different circumstances like a lack of interest in all intentions, or a need of assistance. Therefore, lines (16:19) test the possibility that the human needs help in doing one of the AS tasks. The test checks for a relation between the human context variables st^*_{hc} and the context variables of any possible intended task of type AS. This relation might be a similar location or any fact that reveals a human interest in this task. In case the test passes, a state is created with the related assistance task and the $Flag$ is assigned the value $NeedAssistance$. This will normally lead the IS policy π^{is} to choose $doConfirm$.

After the Algorithm creates the IS state (s^{is}) and the IS policy π^{is} is called to choose an interaction type, both the decision and the IS state are passed to Level 3 in order to choose a targeted task for the robot and the corresponding robot action to achieve this task.

7.3.3 Level 3: Choosing a Task and Applying a Type of Interaction

Algorithm 6 describes the selection of the task that the robot should achieve knowing the IS state (s^{is}) and the decided type of interaction (a^{is}) received from Level 2. The algorithm also describes how the value of the $Flag$ is changed to inform the other levels of the system, in the following time-step, of the applied type of interaction.

Algorithm 6: Task Selection

Input : s^{is}, a^{is}.
Output: $tk \in TK$, $Flag$.

1 **if** $(a^{is} = doConfirm)$ **then**
2 $\quad tk = confirm\ s^{is}_{hi}$;
3 $\quad Flag = confirm$;
4 **else if** $(a^{is} = doAS\ and\ \{AS\} \subset (s^{is}_{hi})_{type}\ and\ s^{is}_{Flag} = yes)$ **then**
5 $\quad tk = s^{is}_{hi}$;
6 $\quad Flag = DoingAS$;
7 **else if** $(a^{is} = doCl\ and\ \{CL\} \subset (s^{is}_{hi})_{type}\ and\ s^{is}_{Flag} = yes)$ **then**
8 $\quad tk = s^{is}_{hi}$;
9 $\quad Flag = DoingCL$;
10 **else**
11 $\quad$ **forall** $(t \in TK_r)$ **do**
12 $\quad\quad$ **if** $(t_{agent} = CR)$ **then** `// priority robot only`
13 $\quad\quad\quad tk = t$;
14 $\quad\quad\quad Flag = DoingCP$;
15 $S^* = \phi$;
16 **while** *tk is not defined* **do**
17 $\quad st^0 = \underset{s \in S^{ie} - S^*}{\operatorname{argmin}} (b^{ie}(s))$; `// Get next argmin`
18 $\quad$ **if** $st^0_{hi} \in TK_r\ and\ (\{CL\} \subset (st^0_{hi})_{type}\ or\ \{AS\} \subset (st^0_{hi})_{type})$ **then**
19 $\quad\quad tk = st^0_{hi}$;
20 $\quad\quad Flag = DoingCP$;
21 $\quad\quad S^* = S^* \cup \{st^0\}$;

Lines 1:3 of the algorithm relates to the situation where the IS decided action is to confirm a task ($a^{is} = doConfirm$). In this case, the chosen task to achieve is to confirm the need of help for the intended human task (s^{is}_{hi}). Otherwise, lines 4:9 shows the chosen task to achieve if the human intention s^{is}_{hi} is of type AS or CL and he confirmed positively his need of help ($s^{is}_{Flag} = yes$). In Line 11, the algorithm searches for tasks of type CP that only the robot can do ($tk_{agent} = CR$) which has the priority at this point. Finally, in Lines 15:21, if non of the prior cases are true, the algorithm looks for a task of type CP that is doable by the robot and is least intended by the human. According to the chosen task, the algorithm sets the $Flag$ value to hold the exact type of interaction that the robot is/will-be applying.

After running Algorithm 6, Level 3 accesses the database of pre-calculated policies (Figure 7.1) and calls the optimal action from the appropriate policy knowing the decided robot task ($tk \in TK_{CR}$) and decided type of interaction $Flag \in I$. A task might have several pre-calculated policies in the database, one for each type of interaction that can achieve this task.

Level 3 ends the system's time-step by sending the $Flag$ value to level 1.

7.4 A Companion Robot at Home Scenario

This section describes a small representative HRMI scenario inspired from [RoboCup@home, 2011]. The scenario was applied on a koala robot to show the ability of the decision model to switch between the different types of interaction according to the inferred human needs.

The representative scenario includes three tasks. The first is a task of type assistance (AS), it consists of finding a book in the bookcase that is a task that can be done by the human alone (with or without assistance). Therefore, $agent = \{H\}$ and $policy = possess$. If the human is not able to find the book, the robot offers his assistance by asking the human about the name of the book then accessing a database to get the book's emplacement in the bookcases and inform the human of the exact shelve number to find the book.

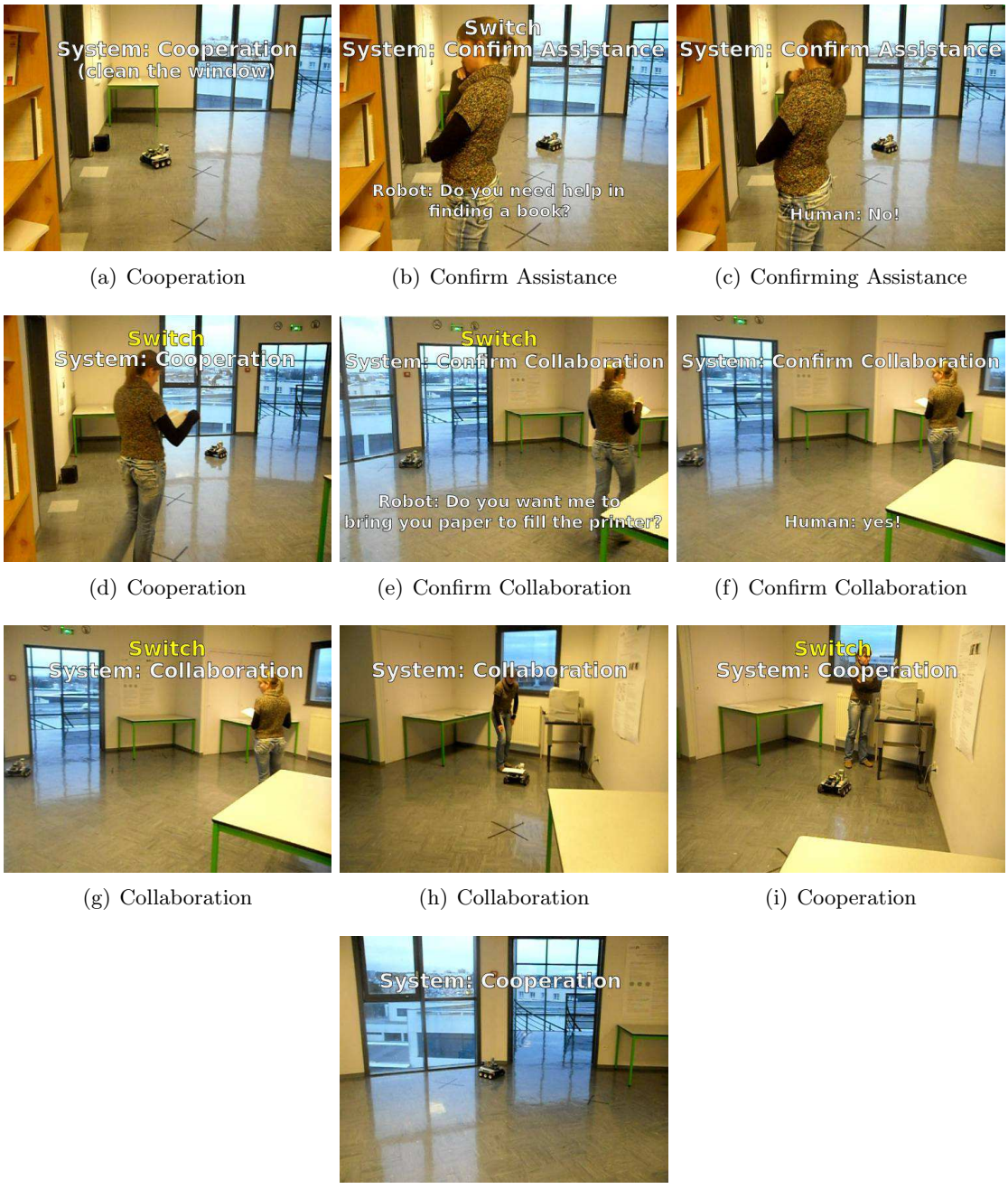
The second task is of type collaboration (CL), it consists of filling the printer with paper and $agent = \{H \wedge CR\}$. In this scenario we suppose that the robot can observe the fact that the printer has no more paper inside. However, lacking the ability to fill the printer by itself, the robot waits for a situation where the human is near the printer (looking for his printed papers) to collaborate with him to fill the printer. The collaboration for this task is done in such a way that the robot brings the paper to a position near the printer and the human takes them and fills the printer.

The third task is of type cooperation (CP) and it consists of cleaning the windows of the building. This task can be done only by the robot, $agent = \{CR\}$.

Figure 7.2 includes screen-shots of a video demonstrating the described scenario with three types of interactions. With the absence of the human, the robot starts by achieving a cooperative task (cleaning the window) as shown in Figure 7.2(a). Later, the human enters the room and after he hesitates in front of the bookcase, the robot detects a possible need of assistance and switches types of interaction to confirming. Figure 7.2(b) shows the robot offering assistance in finding a book. Figure 7.2(c) shows the human answering the confirmation with a "no" which leads the robot to switch back to cooperation as shown in Figure 7.2(d). Meanwhile, the human grabs a book from the bookcase and moves towards the printer. The robot switches to confirm collaboration with the human to fill the printer with paper as shown in 7.2(e). Figure 7.2(f) shows the human confirming positively the collaboration which leads the robot to switch to collaboration as shown in Figure 7.2(g). The robot in Figure 7.2(h), bringing paper near the printer, the human takes them and fill the paper with them. After ending the collaborative task, the robot switches back to cooperation and moves to continue cleaning the windows as shown in Figures 7.2(i) 7.2(j).

The implementation of the human intention estimation for this scenario was based on the human position as human context variables. The printer position was the context variable for the filling printer task and the bookcase position was the context variable for looking for a book task.

`http://users.info.unicaen.fr/~akarami/demohri/demo_multi_interaction.avi`



(a) Cooperation (b) Confirm Assistance (c) Confirming Assistance

(d) Cooperation (e) Confirm Collaboration (f) Confirm Collaboration

(g) Collaboration (h) Collaboration (i) Cooperation

(j) Cooperation

Figure 7.2: Screen-shots of a video demonstrating three types of interactions.

7.5 Experimental Results

The results we show in this section concern problems with three classes of interaction. We present some performance results including off-line calculation times measured for different sizes of the problem. For scalability analysis we show in Table 7.2 results for 6 experiments. Experiments were done on Linux Intel 6 cores, with 20Go Memory. We considered the tasks context variables

are their positions in the human-robot shared area and the human context variable is his position as well. The HMDPs are calculated to plan the human path to each possible task. Tasks types are divided equally and randomly to CP, CL and AS. In those experiments we presented the areas with 10 to 100 rooms each with 10 to 20 accessible positions. This is considered large enough to cover scenarios of a house, an apartment or a hotel. Table 7.2 presents the number of positions $|POS|$, the number of tasks $|TK|$, number of ICS states and the offline calculation time. As we notice, the HRMI calculation times allow us to solve real-life robot companion scenarios contrary to a POMDP model (Figure 5.4). The approach we present in this chapter solves problems more than 100 times the size of problems solved using the POMDP model (Chapter 5) and with acceptable calculation time.

Exp.	$\|POS\|$	$\|TK\|$	$\|S^{is}\|$	total time (minutes)
1	200	25	4.5×10^4	< 1
2	200	50	9×10^4	< 1
3	400	50	1.8×10^5	2
4	400	100	3.6×10^5	5
5	1200	100	1.08×10^6	21
6	1200	400	4.32×10^6	480 (8 hours)

Table 7.2: Experiments from HRMI model.

We recall that the parts that are calculated offline are: the HMDPs policies, the HIE transition and observation functions, the MDP^{is} policy. The more the scenario is complicated, the more time is needed in creating the MDP models and in solving them. In our results, the majority of calculation time served for the creation of the transition function of the MDP^{is} model described in Equation 7.3. We also recall that once the system is online, the robot decisions are taken instantly.

It is hard to evaluate the HRMI model and its algorithms. We have created a problem with 10 tasks of different types (mostly collaboration and assistance) and we simulated a rational human behavior to randomly chose tasks and randomly need help (random *yes* and *no* answers). Algorithm 5 of the HRMI model guarantees the switch to the appropriate type of interaction under the condition that the context variables $(tk_{context}, s^{ie}_{hc}, s^{is}_{hc})$, which are responsible for relating the human actions to his intended task, are well defined and lead to no confusion. In the following we describe some analysis over simulations using the decision model:

- To detect the need of assistance, Algorithm 5 checks for ambiguity in the belief state. During simulation, the human action that triggers this ambiguity is doing nothing (e.g. no displacement). However, it is possible to adjust the needed number of time-steps in which the human does nothing in order to reach ambiguity. This is related to the Q-Value of the human action (*do_nothing*) when his context variables are related to his intended task context variables (e.g. his position is equal to the position of his intended task).

- When confirming a task of type assistance and the human answer is negative: the model assumes that such a reaction from the human means that he is capable of doing the task by himself. This is why the Algorithm 5 sets the task as done in such situations.
- When confirming a task of type collaboration and the human answer is negative: knowing that a collaborative task **cannot** be done by the human alone, the model assumes a confusion in the human intention estimation (during simulations, this confusion happened when two tasks are very close to each other). To avoid multiple confirmation while the confusion is not solved, the model propose to set the task as done for certain time and then add it back after a small period of time. The period of time must be relative to the type of activity in the solved problem.

7.6 Discussion

We presented a framework for a companion robot that is able to interact with a human using different types of interaction (assisting, collaborating, cooperating) depending on the recognized intention and needs of the human. To our knowledge, there is no prior work on adaptive multi-interaction systems in HRI. However, assistive and collaborative individual models have a large interest in the literature.

The model can solve cooperative missions as in Chapter 5 if only the set of tasks include only cooperative tasks. It can also integrate coactive policies for collaborative tasks in the policy database.

Our results show that by dividing the decision model into different components, we are able to solve real life problem sizes with acceptable calculation time, and compute policies that act reasonably as expected. Those policies handle the problem of uncertainty over the human intention and also overcomes the limitation of POMDPs.

An important amount of time is dedicated for manually creating the sub models (Human MDPs, IS, IE and a database of policies for each couple (task, interaction)). Nevertheless, this is a normal difficulty when solving HRI problems with a large task space whether models are created manually or learned from observation.

Chapter 8

Mixed Verbal and Non-Verbal Interaction for Inferring Human Preferences

Contents

8.1 Motivation 124
8.2 The Mixed Model for Human-Robot Cooperation 125
8.3 The Unified Framework 126
8.4 The Disjoined Framework 129
8.4.1 The Epistemic Dialog sub-Model 129
8.4.2 Intuitive HRI sub-Model 133
8.4.3 Switching between Epistemic and Intuitive Interactions 134
8.5 Experiments 135
8.5.1 Scenario of Cooperation using the Unified Framework 135
8.5.2 Scenario of Cooperation using the Disjoined Framework 135
8.5.3 Comparison Between Unified and Disjoined Frameworks 138
8.6 Discussion 139

We discussed in Chapter 5 a decision model that concerns a robot cooperating with a human partner to achieve a common mission. The mission consists of several tasks and the objective of the robot is to infer the human's intention and act accordingly with respect to the mission's success. The human intentions were inferred by observing the human's actions. In this chapter, we present a mixed model that combines a verbal human-robot interaction with the previous model (non-verbal). Using verbal interaction, the robot can ask (query) the human about his preferences over the possible tasks. We propose two different frameworks to model the mixed

The work presented in this chapter was published in AAAI 2010 Fall Symposium (Dialog with Robots) [Matignon *et al.*, 2010]

verbal and non-verbal interaction problem. The first consists of a **unified POMDP model** that handles both interactions by switching between two types of actions. The second separates the verbal interaction model from the non-verbal interaction model and will be referred to as **disjoined model**. Both frameworks allow the robot to switch accurately between verbal and non-verbal interactions. The verbal interaction (will be referred to as epistemic) aims at disambiguating the human's preferences. The non-verbal interaction (will be referred to as intuitive) consists in achieving the cooperative mission while inferring human's intention based on the observed human actions. The beliefs over human's preferences computed during the epistemic interaction are then reinforced in the course of the mission execution by the intuitive interaction. However, a detected ambiguity in the belief over preferences might lead to a potential switch to the epistemic interaction. Using such a decision model allows the robot to detect the changes in the human's preferences, consequently adjust its plan and switch between both kinds of interactions. Experimental results on a scenario inspired from robocup@home outline various specific behaviors of the robot during the cooperative mission.

8.1 Motivation

Cooperating with a human requires the robot to be aware of its partner's preferences upon the tasks so as to effectively satisfy the human's desires during the mission. Given that the human's preferences are a state of mind, a robot should be able to infer its partner's preferences, or at least the probable ones. Using those information, the robot should make better decisions for the cooperative mission, e.g. performing the human's non-favorite tasks. The robot should also adjust its plan rapidly in case of a sudden change in the human's intentions during the mission.

Several successful approaches are interested in inferring the human's intentions to enhance the cooperation between the robot and the human (see Section 4.1). They vary between applications with implicit (intuitive) or explicit (epistemic) communication for the cooperation.

Explicit communication as for spoken dialog seems to be the obvious solution to access the human's state of mind and infer his preferences. However, spoken dialog systems are complex for several reasons. First, the system observes the human answers during the dialog via automated speech recognition and language parsing which are imperfect technologies. Second, each human answer (even if it could be observed accurately) provides incomplete information about the human's intention, so the system must assemble evidence over time. Third, because the human might change his intention at any point during the dialog, inconsistent evidence could either be due to speech recognition error or due to a modified intention. Thus the challenge for the dialog agent would be to interpret conflicting evidence in the human answers to estimate his intention. Finally, the agent must make trade-offs between the cost of gathering additional information (increasing its certainty upon the human's intention, but prolonging the conversation) and the cost of making a final decision that might not match the human's intention. For all of these reasons, the spoken dialog problem can be regarded as planning under uncertainty. Many researchers have found POMDP frameworks suitable for designing a robust dialog agent in spoken

dialog systems. These researches range from robot system that interacts with the elderly in a nursing home [Pineau *et al.*, 2003], automated system that assists people with dementia [Hoey *et al.*, 2010], flight agent assisting the caller to book a flight ticket [Williams, 2006, Young *et al.*, 2010] or a wheelchair directed by her patient [Doshi et Roy, 2008].

A drawback concerning explicit communication for human-robot cooperative missions is related to the fact of detecting a change in the human's preferences. This can be solved by querying the human continuously and assembling conflicting evidence in the human answers. However, the robot should avoid to constantly ask queries since too much questions could annoy the human. For this reason, we propose to combine the epistemic explicit model with an implicit intuitive model. The latter will be responsible for achieving the cooperative mission while observing the human's actions and inferring his preferences using the Q-values of those actions. When an ambiguity between the inferred preferences from both models is detected, a switch to the explicit communication is probably useful.

This chapter describes a unified model that allows the robot to switch between epistemic explicit interaction and intuitive implicit interaction for a cooperative mission with a human partner.

8.2 The Mixed Model for Human-Robot Cooperation

The general architecture of the mixed model is shown in Figure 8.1. One of its components is a human-robot spoken dialog interaction called epistemic interaction. Robot's actions during this interaction are queries asked to the human with potentially noisy or ambiguous answers. The robot must choose queries that disambiguate the human's preferences and build a belief over them despite uncertainty in the observed responses. Once sufficiently certain and based on this belief, the robot switches to task execution to perform the tasks that satisfy the human's preferences.

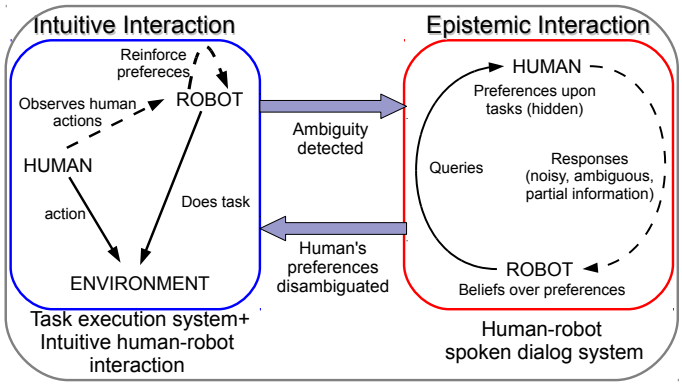


Figure 8.1: The general architecture of the mixed model for the Human-Robot Cooperation.

The second component is the task execution system also called intuitive interaction. During this interaction, the robot chooses tasks to achieve while respecting the human's preferences and the mission's success. However, while achieving the tasks, the human's preferences may change and one challenging issue is to detect this change. For this reason, an intuitive human-robot cooperation is introduced during the task execution to detect the change in preferences. The beliefs over human's preferences (inferred during the dialog) are reinforced by comparing them with the inferred intentions from observing non-verbal human actions (see Chapter 5). In case of ambiguity, the robot returns to the epistemic interaction to disambiguate the human's preferences; otherwise, it continues to execute tasks according to its belief over the human's preferences. Therefore the mixed model provides an accurate switch between both kinds of verbal (epistemic) and non-verbal (intuitive) interactions.

8.3 The Unified Framework

The unified framework consists of a unified POMDP model [Matignon *et al.*, 2010]. The POMDP model is responsible for both verbal interaction and tasks execution. However, detecting a change in the human intention is not detailed in this framework, only an observation in the list of the POMDP observations expressed the fact that another part of the system detected a change in the human intention.

The human and the robot share a mission of N tasks $TK = \{tk_1, tk_2, ..., tk_N\}$. The human has preferences upon the tasks modeled as his internal state s_h which may change over the course of the mission. Preferences are, for each task t_i of the mission:

- $s_h(tk) = to_do_by_human$ if the human would rather do the task tk;
- $s_h(tk) = to_do_by_robot$ if the human would rather the robot did the task tk;
- $s_h(tk) = undecided$ if the human has not yet decided his preference upon the task tk.

The Unified POMDP Model:

States

The state space brings together the set of human's preferences upon the tasks of the mission (non-observable) and the status of each tasks (*done* or *not_done*) that is observable. The human's preferences can be $to_do_by_human, to_do_by_robot, undecided$, for any tasks of the mission that has the status *not_done*. The state s is then characterized by a function that associates, at each task $tk \in TK$ of the mission, either the human's preference or the status of the task if *done*.

Initially, the belief state is uniform among all states where all the tasks are *not_done*. While the robot is executing tasks, it can detect a change in the human's preferences thanks to the observation of the human actions. Indeed, an intuitive method that builds a belief over all possible human's intentions given the observed human actions is used. If the beliefs over human's

intentions computed with the intuitive method do not match the beliefs calculated during the dialog, then the robot has detected a change in the human's intentions and should restart a spoken dialog.

Actions

Possible actions for the robot include queries to the human asked during the dialog. The robot can choose from three kinds of queries: it can choose to ask a general question such as "Which task should I do ?", to confirm a preference upon a specific task tk such as "Should I do the task tk ?" and to greet the human and ask "How can I help you?". The robot can also choose to achieve a task of the mission. We assume the robot has a list of predefined policies to accomplish correctly each task of the mission. The action $do(tk)$ then leads the robot to follow the corresponding policy. The robot may also choose to wait, for instance because remaining tasks are preferred by the human. The robot action set is: $A = \{wait, do(tk_1),\ ...\ do(tk_N), confirm(tk_1),\ ...\ confirm(tk_N), ask, greet\}$.

Observations

The observation set includes different ways of partially or fully communicating the human's preferences. In reply to a general question or a greeting, observations consist of N observations $\{prefdo(tf_1), ..., prefdo(tf_N)\}$ associated with each of the N tasks plus the $prefdo(\emptyset)$ observation. Observations *yes* and *no* stand for positive and negative confirmations in response to confirm queries. Observations *not yet* stands for a not yet decided response. The robot may also observe *nothing*. Finally, the robot may observe $hdid(tk)$ when the human has just achieved the task tk. The robot observation set is: $Z = \{hdid(t_1),\ ...\ hdid(t_N), nothing, prefdo(t_1),\ ...\ prefdo(t_N), prefdo(\emptyset), yes, no, not\ yet\}$.

Transition Function

The effects of the robot action a on a state s are relatively clear: when the robot does a task, the task status changes to *done*. Other actions like queries or wait action do not modify the state. However, the transition from state s to s' is not only defined by the robot action, but also by the human actions and intentions. Indeed, we assume the human has a small probability to change his preferences during the dialog. This change in preference might be dependent on the robot action. However, we assume in the presented model, without loss of generality, that changes in human preferences are independent from the robot action. As well, the task status that can change to *done* when the human did a task. We suppose that: the human does only tasks he would rather do, i.e. tasks whose preferences are *to_do_by_human* or *undecided*; the human will keep same preferences with a probability pKI; the human might do a task he is intended with a probability $pHDo$; otherwise the human changes his preferences upon tasks that are *not_done* to another preferences chosen uniformly. Once the human has decided his preference upon tk

($s(th) \in \{to_do_by_human, to_do_by_robot\}$), he cannot return to a not yet decided preference, yet $s(tk)$ can switch between *to_do_by_human* and *to_do_by_robot*. Thus:

- $T(s, a, s' = s) = pKI$
- $T(s, a, s' \in \texttt{hDo}(s)) = \frac{pHDo}{size(\texttt{hDo}(s))}$
- $T(s, a, s' \in \texttt{hI}(s)) = \frac{1-pKI-pHDo}{size(\texttt{hI}(s))}$ if $size(\texttt{hDo}(s)) \neq 0$
- $T(s, a, s' \in \texttt{hI}(s)) = \frac{1-pKI}{size(\texttt{hI}(s))}$ if $size(\texttt{hDo}(s)) = 0$

where $\texttt{hDo}(s)$ is the set of all reachable states from s when the human does one task that he prefers; and $\texttt{hI}(s)$ is the set of all possible permutations of preferences upon *not yet done* tasks in s. We obtain the same probabilities when $a = do(tk)$ except that the status of the task tk in s' is *done*.

Observation Function

Based on the most recent action a and the future state (s') of the system, the robot has a model of the observation z it may receive. First the observation function $O(a, s', z)$ gives in a deterministic way the observation $z = hdid(tk)$ when the human just did a task tk. The robot also observes *nothing* when it waits or does a task: $O(a \in \{wait, do(tk)\}, s', z = nothing) = 1$.

The observation function also encodes both the words the human is likely to use to reply to the queries and the speech recognition errors that are likely to occur. We suppose speech recognition errors are different according to the kind of queries. If the robot made a general query or a greeting, then it observes the right answer with a probability $pAsk$. Thus, when $a \in \{ask, greet\}$:

- $O(a, s', z = prefdo(\emptyset)) = pAsk$ if $\texttt{nbRDo}(s') = 0$ and $\texttt{nbNYet}(s') = 0$
- $O(a, s', z = not\ yet) = pAsk$ if $\texttt{nbRDo}(s') = 0$ and $\texttt{nbNYet}(s') \neq 0$
- $O(a, s'(t_i) = 1, z = prefdo(t_i)) = \frac{pAsk}{\texttt{nbRDo}(s')}$

where $\texttt{nbRDo}(s')$ is the number of tasks in s' that the robot can do according to human's preferences and $\texttt{nbNYet}(s')$ the number of tasks in s' upon which the human is not decided yet. If the robot made a confirm query, then it observes the right answer with a probability $pConf$. Thus, when $a = confirm(tk)$:

- $O(a, s'(tk) = to_do_by_robot, z = yes) = pConf$
- $O(a, s'(tk) = to_do_by_human, z = no) = pConf$
- $O(a, s'(tk) = undecided, z = not\ yet) = pConf$

In both queries ask and $confirm$, the robot observes in addition to the right answer an arbitrary response uniformly from the remaining possible replies $prefdo(tk), prefdo(\emptyset), yes, no, not\ yet, nothing$.

Reward Function

The robot is rewarded for greeting the human in the beginning of the mission. The reward function also specifies how much the human is willing to tolerate ask versus $confirm$ queries thanks to the choice of the reward for a general query ask. The robot is penalized for doing a task preferred by the human. It is also penalized if it does a task although there remains undecided tasks or if it waits although it would better do a task or query the human. Finally, the robot gets a high reward when it does a human's non-favorite task and all the human preferences upon the tasks have been given. It also gets a high reward when it waits while all remaining tasks are human's favorite tasks.

8.4 The Disjoined Framework

This section describes a disjoined verbal and non-verbal model for human-robot cooperative missions. Contrary to the unified model, this framework describes the verbal interaction and the non verbal intuitive interaction while achieving tasks using tow separate components (sub-models). In addition, the disjoined model switches between the two components using an ambiguity detector which is described in detail.

A cooperative mission is composed of a set of N tasks $TK = \{tk_1, tk_2, ..., tk_N\}$, each of these tasks can be done by the human or by the robot. However, the robot should choose his tasks with respect to the human's preferences. Therefore, the robot queries the human about his preference toward each of the N tasks. Figure 8.2 shows a detailed schema of the intuitive and epistemic components and the flow control in the disjoined model. The following will describe the epistemic interaction, the intuitive interaction and how/when the system switches between the two components.

8.4.1 The Epistemic Dialog sub-Model

The system starts in the epistemic interaction which is a POMDP dialog policy. This policy chooses queries as robot's actions (A_{ep}) and receives the human's answers as observations (Z_{ep}). The POMDP also receives an observation when a task is done (if the human achieved a task while answering the queries of the robot). Once the human's preferences are disambiguated the control goes to the intuitive interaction.

The epistemic dialog sub-model is represented by a tuple:

$$\text{POMDP}_{ep} =< S_{ep}, A_{ep}, Z_{ep}, T_{ep}, O_{ep}, R_{ep}, b_{ep} > .$$

States S_{ep}

The state space brings together the set of human's preferences upon the tasks and the status of each task. If a task tk is done, the task state is $s(tk) = done$, otherwise the task state holds the

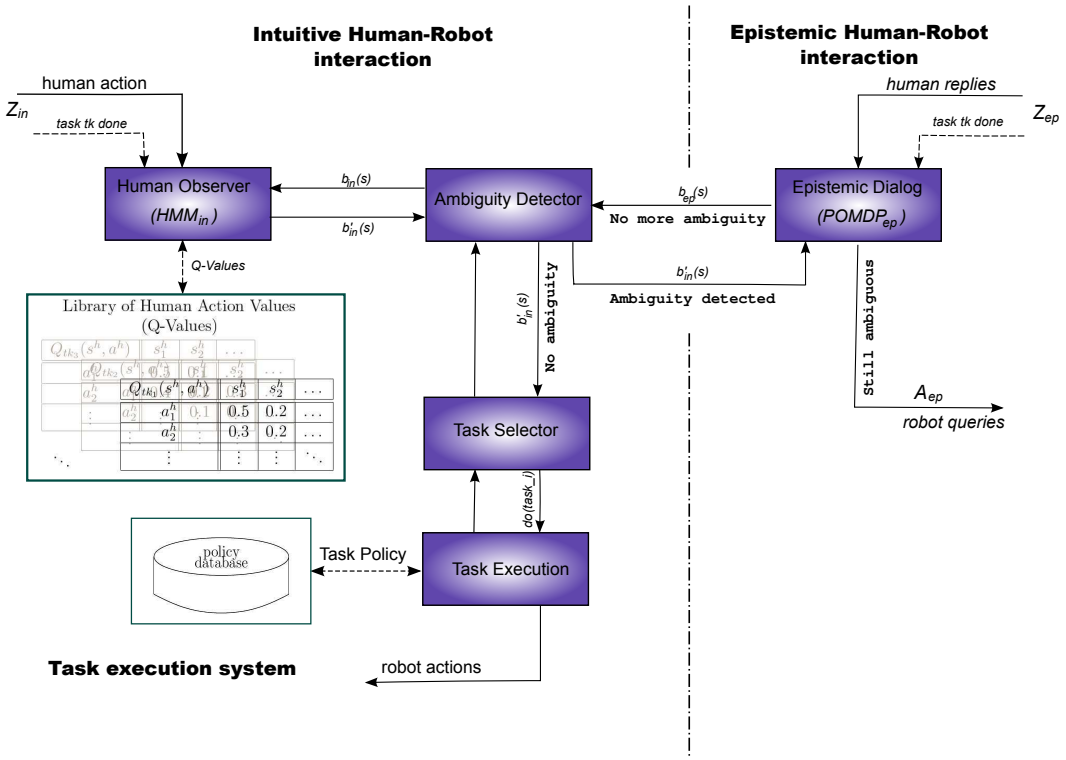


Figure 8.2: The flow control in/between the components of the disjoined model.

human's preference over it. For each task $tk \in TK$, an example of possible tasks' states can be $s(tk) \in \{to_do_by_robot, to_do_by_human, to_do_by_any, undecided, unknown, done\}$ such as:

- $s(tk) = to_do_by_robot$: the human prefers that the robot does the task tk,
- $s(tk) = to_do_by_human$: the human prefers to do the task tk himself,
- $s(tk) = to_do_by_any$: the human has no preferences upon who should do the task tk (the human or the robot),
- $s(tk) = undecided$: the human has not yet decided his preference about the task tk,
- $s(tk) = unknown$: the robot has no knowledge of the human's preference about the task tk,
- $s(tk) = done$: the task tk is done.

The state space is then the set of all tasks' states: $S_{ep} =< s(tk_1), s(tk_2), ..., s(tk_N) >$. The initial belief state $b_{ep}(s)$ in the beginning of the mission holds the task state $unknown$ for all the mission's tasks, such as $b_0 = (pr(s =< unknown, unknown, ..., unknown >) = 1)$. A possible

intermediate state can be as in:

$$\begin{aligned} b_t = (&pr(s_{ep} =< unknown, to_do_by_any, ..., unknown >) = 0.8, \\ &(pr(s_{ep} =< unknown, undecided, ..., unknown >) = 0.1, \\ &(pr(s_{ep} =< unknown, to_do_by_human, ..., unknown >) = 0.1) \end{aligned}$$

A final state is reached when none of the tasks has the state *unknown* which means that all preferences are disambiguated, as in:

$$\begin{aligned} b_{t+k} = (&pr(s_{ep} =< to_do_by_robot, to_do_by_any, ..., to_do_by_robot >) = 0.7, \\ &pr(s_{ep} =< to_do_by_robot, undecided, ..., to_do_by_human >) = 0.05, \\ &pr(s_{ep} =< to_do_by_human, to_do_by_robot, ..., to_do_by_robot >) = 0.05, \\ &pr(s_{ep} =< to_do_by_robot, to_do_by_human, ..., to_do_by_any >) = 0.05, \ldots) \end{aligned}$$

Actions A_{ep}

The epistemic actions include queries where the robot asks the human to confirm his preference upon a specific task tk such as "Should I do the task 'clean the table' ?".

$$A_{ep} = \{conf(tk_1), conf(tk_2), ..., conf(tk_N)\}.$$

Observations Z_{ep}

The set of observations holds the possible human answers to the queries and the possible observation that one of the tasks is done.

The verbal observations related to human's answers are processed using a speech recognition system. When querying about task tk, possible answers can be:

- *ob_yes*: the human's preference over the task tk is *to_do_by_robot*.
- *ob_no*: the human's preference over the task tk is *to_do_by_human*.
- *ob_any*: the human's preference over the task tk is *to_do_by_any*.
- *ob_maybe*: the human's preference over the task tk is *undecided*.
- *ob_nothing*: observed when no audio observations have been made (the human did not answer the query or the speech recognition system failed).

The robot can also receive the observation $ob_done(tk)$ when the task tk is done.

Therefore, the set of observations is:

$$Z_{ep} = \{ob_yes, ob_no, ob_any, ob_maybe, ob_nothing, ob_done(tk_1), ..., ob_done(tk_N)\}.$$

Transition function T_{ep}

Figure 8.3 shows an example of transition function values for one task state.

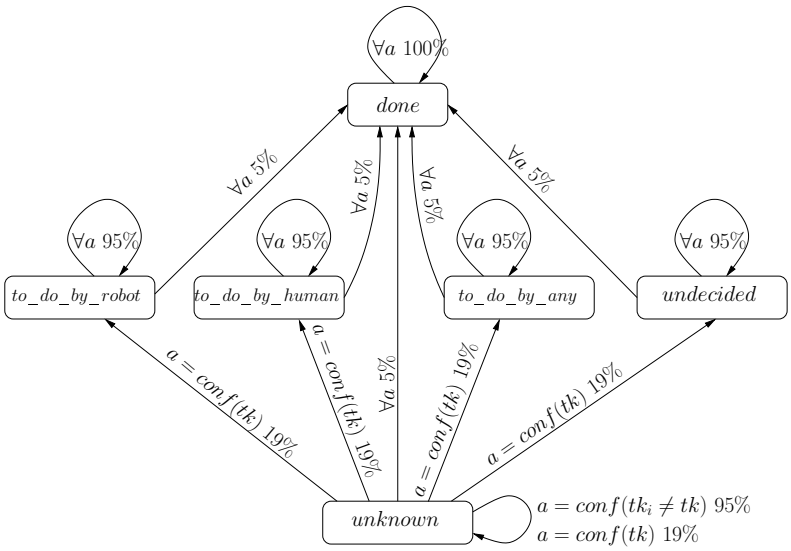


Figure 8.3: Transition function values for one task (tk) preferences. An arrow connects a task state s and its next state s' and is labeled with the action a and the probability $T(s, a, s')$.

The tasks' values transitions are as the following:

- a task that is done stays done,
- a task that is not done can transfer to done during the dialog,
- asking a query $conf(tk)$ about the task tk on which the robot has no knowledge ($s(tk) = unknown$) brings knowledge about this task except if the speech recognition system fails,
- no change in the human's preferences is taken into account during the epistemic interaction.

Observation function O_{ep}

The observation function gives the probability of observing $z_{ep} \in Z_{ep}$ from state $s'_{ep} \in S_{ep}$ after doing action $a_{ep} \in A_{ep}$. Observations that represent the fact that a task is done are not dependent on the action a_{ep}, this can be described as:

$$pr(done(tk)|a, s'(tk) = done) = 1, \qquad pr(done(tk)|a, s'(tk) \neq done) = 0. \tag{8.1}$$

The observation function reflects the uncertainty of the speech recognition system. This means that the probability of observing ob_yes assigns a small probability that the human's answer was something else and the speech recognition system analyzed it as yes. For example,

$$pr(z_{ep}|conf(tk), s'(tk) = to_do_by_robot) = \begin{cases} \beta & \text{if } z_{ep} = ob_yes \\ 1 - \beta & \text{otherwise} \end{cases} \tag{8.2}$$

where $\beta \in [0, 1]$ is the probability of the success of the recognition system. Equations 8.1, 8.2 are then normalized to respect the condition: $\sum_{z_{ep} \in Z_{ep}} pr(z_{ep}|a_{ep}, s'_{ep}) = 1 \quad \forall (a_{ep}, s'_{ep})$.

Reward function R_{ep}

The reward function $R(s, a, s')$ assigns a positive reward when the system reaches a state with no task preference as unknown.

$$R(s, a, s' | \forall tk \in TK, s'(tk) \neq unknown) = reward > 0 \tag{8.3}$$

8.4.2 Intuitive HRI sub-Model

Once the human's preferences are disambiguated the control switches to the intuitive interaction. As shown in Figure 8.2, the task selector in the intuitive sub-model chooses a task with respect to the human's preferences and a robot action is executed from the corresponding task policy (policy database). While executing tasks, the intuitive sub-model monitors the human's non-verbal actions to determine if his actions match the preferences that were inferred during the epistemic interaction. This is done by matching the human's actions to possible preferences using a Hidden Markovian Model (HMM_{in}) and then comparing the belief state $b_{in}(s)$ from the HMM_{in} with the belief state $b_{ep}(s)$ from the epistemic interaction (Section 8.4.3).

The HMM_{in} state space includes the tasks' states in addition to the human variables that are related to the tasks achievement. The observations set of the HMM_{in} includes the possible human actions related to the tasks and observations to inform that a task is done. The HMM_{in} belief state $b_{in}(s)$ is initialized at the beginning of the intuitive interaction with the initial human context variables and the beliefs over preferences from the epistemic interaction. The observation function of the HMM_{in} uses the Q-values of the observed human actions to reinforce/update the human's preferences (see Section 5.2, 5.3 and Section 7.3.1). The HMM_{in} transition function covers all possible transitions related to the human context variables. Moreover, the transitions of the tasks states values are described as the following:

- a task that is *done* stays *done*,
- any task state can transition to *done*,
- a task state *to_do_by_human* can stay the same or transition to *unknown* (if the observed human action is not in favor of the task),
- any task state value (including *unknown*) can stay the same or transition to *to_do_by_human* (if the observed human action is in favor of the task).

After each robot executive action, an observed human action is received by the Hidden Markovian Model HMM_{in}. A new belief state $b'_{in}(s)$ is updated and the Ambiguity Detector

tests for ambiguity between $b'_{in}(s)$ and $b_{ep}(s)$ (see Section 8.4.3). In case the test fails and no ambiguity was found, the intuitive sub-model continues by executing another robot action and receiving another human non-verbal action. However, if an ambiguity is detected the disjoined model switches back to the epistemic interaction to disambiguate the belief over the human's preferences.

8.4.3 Switching between Epistemic and Intuitive Interactions

Switching from epistemic interaction to intuitive interaction occurs when all human's preferences are sufficiently disambiguated, i.e. probability of preference unknown for all tasks is less than a small value ϵ.

$$\forall s_{ep} \in S_{ep}, \forall tk \in TK, b_{ep}(s_{ep}(tk) = unknown) < \epsilon.$$

Switching from intuitive to epistemic interactions occurs when ambiguity is detected. To describe how ambiguity is detected, different situations must be studied:

1. If the human expressed his preferences to do tasks tk_i, tk_j himself during the epistemic interaction: while achieving task tk_i, the human actions might **not be** of advantage toward task tk_j. This might lead the belief state $b_{in}(s)$ to be updated for task tk_j from the value *to_do_by_human* to the value *unknown* which is not necessarily the true fact about the human's preferences.

2. If the human expressed his preferences to do task tk_k himself and for the robot to do task tk_l. While achieving task tk_k, it is probable that some of the human actions **be** of advantage toward task tk_l as well. This might lead the belief state $b_{in}(s)$ to be updated for task tk_l from the value *to_do_by_robot* to the value *to_do_by_human* which is not necessarily the true fact about the human's preferences.

Taking the previous two points into account, ambiguity is detected if one of the following conditions is true where $s^*_{in} = \underset{s_{in} \in S_{in}}{\text{argmax}}(b_{in}(s_{in}))$ and $s^*_{ep} = \underset{s_{ep} \in S_{ep}}{\text{argmax}}(b_{ep}(s_{ep}))$:

condition 1: If $\forall tk \in TK, s^*_{in}(tk) = unknown$.

condition 2: If $\exists tk_i \in TK : (s^*_{ep}(tk_i) = to_do_by_robot, s^*_{in}(tk_i) = to_do_by_human)$ and $\nexists tk_j \neq tk_i \in TK : s^*_{in}(tk_j) = to_do_by_human$.

The first condition consists in all tasks preferences are *unknown*. The second condition consists in having a task that the human expressed his preference to *to_do_by_robot* (during epistemic interaction), however, the intuitive interaction updated the preferences in a way that only this task has the preference *to_do_by_human*. This means that there is no confusion with another task and it is very possible that the human actually changed his preferences.

When ambiguity is detected the disjoined framework switches to the epistemic sub-model. In the first condition the epistemic belief state is initiated with a belief state b_{ep} that has all

tasks preferences as *unknown*. However, in the second condition, the epistemic belief state is initiated with the intuitive belief state values after changing all tasks values with ambiguous preference to *unknown*.

8.5 Experiments

8.5.1 Scenario of Cooperation using the Unified Framework

The purpose of this scenario is to test the behavior resulting from the computed unified POMDP policy and especially the accurate choice of actions. Table 8.2 shows a scenario with a mission composed of 5 tasks between the human and the robot. This scenario outlines various specific behaviors of the robot during the cooperation. We note in step 4 the reinforcement of the preference upon the task 0 that is due to the increased probability of the change of the human's intention over time since the last confirmation (step 1). In steps 10 and 13, we notice that the robot has switched to the dialog actions for two different reasons. At step 10, after finishing all its assigned tasks, the robot checks the possible change in the human's preferences that might have occurred during the execution period. At step 13, the robot receives an observation from the intuitive system that declares an observed human's intent change. For this, it reinitializes all the remaining tasks to an equal probability and starts to re-inferring the new human's preferences using queries.

We also performed experiments with a human and a mobile koala robot. In order to be able to realize the verbal communication between the robot and the human, we integrated a speech recognition module for the robot to interpret the human speech answers; and a speech synthesizer for the robot to convert its queries into speech. Audio observations are processed using the Sphinx-4 open-source speech recognition system [Walker *et al.*, 2004] and the FreeTTS open-source speech synthesizer was used for the text-to-speech conversion [Walker *et al.*, 2002]. We chose a mission composed of 4 tasks. The video shows the execution of a complete mission composed of 4 tasks that are labeled *task pink, task green, task blue* and *task yellow*.

8.5.2 Scenario of Cooperation using the Disjoined Framework

The purpose of this scenario is to test the behavior resulting from the computed policies and especially the accurate switch between verbal and non-verbal interactions. We chose a scenario that resembles to the cleaning soil-spots scenario from Section 5.4. The mission is composed of cleaning 5 soil-spots that have different positions in the environment as described in Figure 8.4. The figure shows a number for each position in the environment and a number for each task, the position of each task and the initial human position.

Table 8.2 shows a part of the interaction between the human and the robot during the mission and following the computed policy. We remind that $s^*_{in} = \underset{s_{in} \in S_{in}}{\text{argmax}}(b_{in}(s_{in}))$ and $s^*_{ep} =$

Available at `http://lmatigno.perso.info.unicaen.fr/research`

	Greeting at the beginning of the dialog:
1	$< to_do_by_robot, undecided, to_do_by_human, undecided, to_do_by_human >$
	ROBOT: Hello, how can I help you ? HUMAN: I'd rather you do the task 0.
2	$< to_do_by_robot, undecided, to_do_by_human, undecided, to_do_by_human >$
	ROBOT: Should I do the task 2? Human does task 2.
	The robot confirms all remaining (not_done) and ambiguous tasks:
3	$< to_do_by_robot, to_do_by_human, done, to_do_by_robot, to_do_by_human >$
	ROBOT: Should I do the task 4? HUMAN: No.
4	$< to_do_by_robot, to_do_by_human, done, to_do_by_robot, to_do_by_human >$
	ROBOT: Should I do the task 0? HUMAN: Yes.
5	$< to_do_by_robot, to_do_by_human, done, To_do_by_robot, to_do_by_human >$
	ROBOT: Should I do the task 3? HUMAN: Yes.
6	$< to_do_by_robot, to_do_by_human, done, to_do_by_robot, to_do_by_human >$
	ROBOT: Should I do the task 1? HUMAN: No.
	Executing the human's non-preferred tasks:
7	$< to_do_by_robot, to_do_by_human, done, to_do_by_robot, to_do_by_human >$
	Robot does task 0.
8	$< done, to_do_by_human, done, to_do_by_robot, to_do_by_human >$
	Robot does task 3.
9	$< done, to_do_by_human, done, done, to_do_by_human >$
	Robot waits.
	Checking if the human has changed his preferences:
10	$< done, to_do_by_human, done, done, to_do_by_human >$
	ROBOT: Which tasks should I do? HUMAN: I'd rather you do nothing.
	Executing nothing:
11	$< done, to_do_by_human, done, done, to_do_by_human >$
	Robot waits.
12	$< done, to_do_by_robot, done, done, to_do_by_robot >$
	Robot waits. INTUITIVE SYSTEM: *Change in human's intentions.*
	Detect a change of intention, checking the human's preferences:
13	$< done, to_do_by_robot, done, done, to_do_by_robot >$
	ROBOT: Should I do the task 4? HUMAN: Yes.
14	$< done, to_do_by_robot, done, done, to_do_by_robot >$
	ROBOT: Should I do the task 1? HUMAN: Yes.
	Executing the human's non-preferred tasks:
15	$< done, to_do_by_robot, done, done, to_do_by_robot >$
	Robot does task 1.
16	$< done, done, done, done, to_do_by_robot >$
	Robot does task 4.
17	$< done, done, done, done, done >$

Table 8.1: A dialog example between the human and the robot during the achievement of the mission (Unified Model).

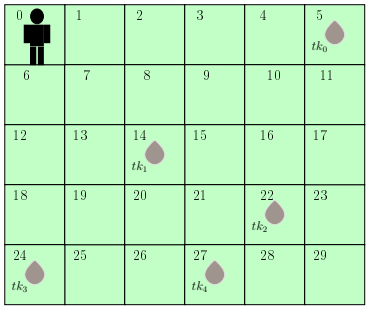


Figure 8.4: Cleaning area domain environment.

	Epistemic interaction:
1	$s^*_{ep} =< unknown, unkown, unkown, unkown, unkown >$ Robot: "Should I do task tk_3?" Human: "yes!"
2	$s^*_{ep} =< unkown, unkown, unkown, to_do_by_robot, unkown >$ Robot: "Should I do task tk_1?" Human: "no!"
3	$s^*_{ep} =< unkown, to_do_by_human, unkown, to_do_by_robot, unkown >$ Robot: "Should I do task tk_2?" Human: "yes!"
4	$s^*_{ep} =< unkown, to_do_by_human, to_do_by_robot, to_do_by_robot, unkown >$ Robot: "Should I do task tk_0?" Human: "any!"
5	$s^*_{ep} =< to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, unkown >$ Robot: "Should I do task tk_4?" Human: "yes!" $s^*_{ep} =< to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$
	Intuitive interaction:
6	$s^*_{in} =< 0, to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$ human: south
7	$s^*_{in} =< 6, to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$ human: south
8	$s^*_{in} =< 12, to_do_by_any, to_do_by_human, to_do_by_robot, to_do_by_robot, to_do_by_robot >$ human: south $s^*_{in} =< 18, to_do_by_any, unkown, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	Epistemic interaction:
9	$s^*_{ep} =< to_do_by_any, unkown, to_do_by_robot, unkown, to_do_by_robot >$ Robot: "Should I do task tk_3?" Human: "no!"
10	$s^*_{ep} =< to_do_by_any, unkown, to_do_by_robot, to_do_by_human, to_do_by_robot >$ Robot: "Should I do task tk_1?" Human: "yes!" $s^*_{ep} =< to_do_by_any, to_do_by_robot, to_do_by_robot, to_do_by_human, to_do_by_robot >$
	Intuitive interaction:
11	$s^*_{in} =< 18, to_do_by_any, to_do_by_robot, to_do_by_robot, to_do_by_human, to_do_by_robot >$ human: south
12	$s^*_{in} =< 24, to_do_by_any, to_do_by_robot, to_do_by_robot, to_do_by_human, to_do_by_robot >$ human: $do(tk_3)$
13	$s^*_{in} =< 24, to_do_by_any, to_do_by_robot, to_do_by_robot, done, to_do_by_robot >$ human: north $s^*_{in} =< 16, to_do_by_any, to_do_by_human, to_do_by_robot, done, to_do_by_robot >$
	Epistemic interaction:
14	$s^*_{ep} =< to_do_by_any, unkown, to_do_by_robot, done, to_do_by_robot >$

Table 8.2: Part of a cooperative verbal and non-verbal scenario between the human and the robot during the achievement of the mission (Disjoined Model).

$\underset{s_{ep} \in S_{ep}}{\text{argmax}}(b_{ep}(s_{ep}))$. The scenario described in table 8.2 does not show the robot execution actions, however, it shows the epistemic interaction actions and how the intuitive sub-model is able to detect a change in the human's preferences by observing his actions. The HMM state includes the human position as a human variable. The HMM Q-values are based on a displacement MDP's and the value of a human displacement action toward reaching the position of each of the soil-spots. The described part of the interaction outlines various specific behaviors of the robot during the cooperation. The scenario starts at the beginning of the interaction, where the epistemic state is initialized with *unknown* preference for all tasks. The epistemic interaction verifies the human's preferences over all tasks in steps (1:5). Once all preferences are specified, the disjoined model switches to the intuitive sub-model. We note in step 8, the human action value (Q-value) was not in favor of any of his preferences; on the contrary, it was in favor of one of the robot's tasks (tk_3 in position 24). After applying the human action in step 8, the human position is set to position 18 and the belief state changes the belief over human's preference for tk_1 to value *unknown* and for tk_3 to value *to_do_by_human*. These changes will lead the ambiguity detector to detect ambiguity in the human's preferences (condition 2) and the disjoined model switches back to the epistemic interaction. The belief state for the epistemic interaction is initiated with *unknown* for tasks tk_1 and tk_3 where ambiguity is detected, all the other tasks hold the same unambiguous preferences. After querying the human in steps (9:10), which reveals his change in preference, the disjoined model switches to the intuitive sub-model, where the human moves towards tk_3 and achieves it. However, his actions lead to another ambiguity detection (condition 2) because his action to move north was not in favor of his preference to to task tk_4. The disjoined model switches back to the epistemic sub-model, etc.

The scenario shows that with a well defined library of Q-values, the intuitive sub-model is able to detect ambiguity (if it exists) between the queried preferences of the human and his preferences during the execution. Such a library will ensure switching between epistemic and intuitive sub-models when needed.

8.5.3 Comparison Between Unified and Disjoined Frameworks

Table 8.3 presents model size analysis and policy calculation time for missions with different numbers of tasks. For the unified and disjoined frameworks, the table shows the size of state, action, observation sets, and the needed time for calculating the unified POMDP policy and the epistemic POMDP policy. Experiments were done on Linux Intel 6 cores, with 20 Go Memory.

The epistemic state set size $|S_{ep}| = |\text{state_values}|^{|TK|} = 6^{|TK|}$. The action set includes a confirmation action for each task, therefore $|A_{ep}| = |TK|$. The observation set includes possible human answers and observations related to done tasks, therefore $|Z_{ep}| = 5 + |TK|$. The unified POMDP state set size $|S| = |\text{state_values}|^{|TK|} = 4^{|TK|}$. The action set includes a confirmation action for each task, an achieving action for each task in addition to $wait, ask$ and $greet$; therefore $|A_{ep}| = (|TK| * 2) + 3$. The observation set includes possible human answers (to ask and $conf$ queries) and observations related to done tasks, therefore $|Z_{ep}| = |TK| + 5 + |TK|$.

An important amount of calculation time in the table is used in creating the POMDP model, namely the transition function. However, our implementation did not focus on optimizing the time for creating the model that is the reason why the transition function had to calculate the transition value for each triple (s, a, s') which means $|S_{ep}| * |A_{ep}| * |S_{ep}|$ calculations ($5 * 10^{11}$ times for mission with 7 tasks).

	Disjoined (epistemic)				Unified			
$\|TK\|$	$\|S_{ep}\|$	$\|Z_{ep}\|$	$\|A_{ep}\|$	time (min)	$\|S\|$	$\|Z\|$	$\|A\|$	time (min)
4	1296	9	4	3	256	13	11	159
5	7776	10	5	7	1024	15	13	1200
6	46656	11	6	65	4096	17	15	-
7	279936	12	7	-	16384	19	17	-

Table 8.3: Mixed verbal and non-verbal frameworks: model complexity and policy calculation-time. " - ".

The unified framework was solved with an approximative topological solver [Dibangoye *et al.*, 2009]. This choice was motivated by the topological structure of the unified POMDP model. The epistemic POMDP was solved using the ZMDP solver. However, we notice that certain groups of actions affects only certain variables of the state (e.g. $confirm(t_1), do(t_1)$ are related to task status of task t_1) which is a motivation to use factored approaches to solve compact representations of structured POMDPs [Boutilier et Poole, 1996, Guestrin *et al.*, 2001, Bui *et al.*, 2010, Williams *et al.*, 2005].

We notice the unified framework did not detail the intuitive detection of the human's preferences. In order to do that using a unified POMDP, the number of observations will increase to include all possible human observed actions. This will lead to a leap in the complexity of solving the POMDP.

8.6 Discussion

In this chapter, we have presented the unified and disjoined frameworks. They allow an autonomous companion robot that cooperates with a human partner to infer his preferences and to switch accurately between verbal (epistemic system) and non-verbal (intuitive system) interactions.

We presented an example of a mission that shows how the robot switches between the epistemic and the intuitive systems when an ambiguity is detected. We aim to improve the epistemic model as discussed earlier, using factored approaches, to be able to solve missions with higher number of tasks; indeed, the epistemic model holds back the capability of the unified model to solve missions with higher task space. We also plan in future work on structuring the tasks with specific variables where preferences of the human can be generalized on several tasks

depending on the human's preferences over the tasks' variables. This will also help in overcoming the complexity of the epistemic POMDP model.

Part IV

Conclusion

Chapter 9

Conclusion and Perspectives

9.1 Conclusion

In this thesis, we investigated the question of what do we exactly expect from a companion robot. We proposed, consequently, different decision models that allows a companion robot to share the daily living activities of a human while being useful and regardful. Human needs during the day might be different and a companion robot should be able to detect these needs and act accordingly (cooperate, collaborate, assist). However, a human intention is part of his mental state and it cannot be directly observed by a robot. One theoretic model that would fit such problem is POMDPs.

Motivated by an intelligent companion robot that is able to understand the human intention and act accordingly, we contributed with the following decisional models:

- We proposed to estimate human intentions by observing his actions. This estimation is based on simulating human models for achieving different tasks and create a library of human action values (Q-values). Those values are integrated in the observation function of an augmented POMDP model that will allow the robot to evaluate the observed human actions toward each of the possible human intentions. We showed that the model was able to highly estimate the real human intention. We also showed the inability of this model to solve large size problems with large environment spaces.

- A coactive decision model that allows a robot to incite his partner's action for a collaborative task. This contribution shows that robots can be able to reason about their behavior according to the progress of the mission.

- A model for human-robot multi-type interaction. This model uses an HMM, that integrates the Q-values in its observation function, to estimate the human need and act accordingly. The robot is able to switch between different kinds of interactions: cooperation, collaboration or assistance. This model is able to solve large size problems and can integrate a set of coactive policies for collaborative tasks in its precalculated database of policies.

- A Mixed verbal and non-verbal interactions model. This contribution shows that a robot can infer a change in the human's preferences without questioning him continuously. We proposed a mixed model of verbal and non-verbal interaction that allows a robot to question the human and, while executing tasks, it observes the human actions to infer a change in his preferences.

9.2 Perspectives

- Social studies to prove the effectiveness of the coactive decision model in addition to different coactive models for other tasks are our perspective for the near future. We will also improve the principal model by adding another hidden state representing the level on which the human is annoyed. This level increases when the robot tries to incite the human to collaborate and decreases when the human reacts positively to the incitement.

- Currently we are motivated by learning human models in order to create the Q-values instead of simulating rational human. This will generate Q-values that are more personalized to the accompanied human.

- In this thesis, we did not concentrate on the scalability in solving some POMDP models, such as the epistemic model in the mixed verbal and non-verbal interaction and the augmented POMDP. We mentioned earlier that the epistemic model has certain properties that make it interesting to use factored POMDP solvers.

- We would like to relate the epistemic POMDP to the approaches dedicated to preference elicitation to take advantage of some theoretic properties as for minimizing the needed number of queries to infer the human's preferences. Moreover, epistemic POMDP solving can be improved if we assume some preference structure between the tasks.

- We would like to relate this work to other domains in robotics such as: image processing, speech processing to design an architecture with the proposed decision models and multimodal interaction (speech, gesture, ...) in order to carry the contributions of this thesis to a higher applicable level and to test its feasibility in actual scenarios.

Bibliography

[Adlam *et al.*, 2009] Adlam, T., Carey-Smith, B. E., Evans, N., Orpwood, R., Boger, J. et Mihailidis, A. (2009). Implementing Monitoring and Technological Interventions in Smart Homes for People with Dementia - Case Studies. *In The Book Manufacturers Institute (BMI) Book*, pages 159–182. 60

[Alami *et al.*, 1998] Alami, R., Chatila, R., Fleury, S., Ghallab, M. et Ingrand, F. (1998). An Architecture for Autonomy. *Internatonal Journal of robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming*, 17(4). 105

[Alili *et al.*, 2009] Alili, S., Warnier, M., Ali, M. et Alami, R. (2009). Planning and Plan-execution for Human-Robot Cooperative Task achievement. *In Proceedings of the* 19^{th} *International Conference on Automated Planning and Scheduling: Workshop on Planning and Plan Execution for Real World Systems (ICAPS)*. 68, 106

[Allen *et al.*, 1999] Allen, J., Guinn, C. et Horvtz, E. (1999). Mixed-Initiative Interaction. pages 14–23. 33

[Armstrong-Crews et Veloso, 2007] Armstrong-Crews, N. et Veloso, M. (2007). Oracular Partially Observable Markov Decision Processes: A Very Special Case. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2

[Asimov, 1950] Asimov, I. (1950). *I, Robot*. New York: Doubleday & Company. 32

[Babaian *et al.*, 2002] Babaian, T., Grosz, B. J. et Shieber, S. M. (2002). A Writer's Collaborative Assistant. *In Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 7–14. 40

[Bellman, 1957] Bellman, R. (1957). A Markovian Decision Process. *Indiana University Math. J.*, 6:679–684. 5, 51, 52

[Bernstein et Zilberstein, 2000] Bernstein, D. et Zilberstein, S. (2000). The Complexity of Decentralized Control of MDPs. *In Proceedings of the Conference in Uncertainty in Artificial Intelligence (UAI)*. 57

[Boger *et al.*, 2005] Boger, J., Poupart, P., Hoey, J., Boutilier, C., Fernie, G. et Mihailidis, A. (2005). A Decision-Theoretic Approach to Task Assistance for Persons with

Dementia. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1293–1299. 1, 25, 36, 67, 70

[Bonet, 2002] Bonet, B. (2002). An epsilon-Optimal Grid-Based Algorithm for Partially Observable Markov Decision Processes. *In Proceedings of the International Conference on Machine Learning (ICML)*, pages 51–58. 55

[Bouguerra et Karlsson, 2005] Bouguerra, A. et Karlsson, L. (2005). PC-SHOP: A Probabilistic-Conditional Hierarchical Task Planner. *Journal of Artificial Intelligence*, 2(4): 44–50. 56

[Boutilier *et al.*, 1999] Boutilier, C., Dean, T. et Hanks, S. (1999). Decision-Theoretic Planning: Structural Asumptions and Computational Leverage. *Journal of Artificial Intelligence Research*, 1:1–93. 10, 77

[Boutilier et Poole, 1996] Boutilier, C. et Poole, D. (1996). Computing Optimal Policies for Partially Observable Decision Processes Using Compact Representations. *In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI/IAAI)*, volume 2, pages 1168–1175. 139

[Boutilier,] Boutilier, G. Sequential Optimality and Coordination in MultiAgents Systems, booktitle = Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), year = 1999, owner = akarami, timestamp = 2011.05.05. 57

[Breazeal, 2004] Breazeal, C. (2004). Social Interactions in HRI: the Robot View. volume 34, pages 181–186. 33, 41

[Broz *et al.*, 2011] Broz, F., Nourbakhsh, I. et Simmons, R. (2011). Designing POMDP models of socially situated tasks. *In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 39 –46. 2

[Broz *et al.*, 2008] Broz, F., Nourbakhsh, I. R. et Simmons, R. G. (2008). Planning for Human-Robot Interaction Using Time-State Aggregated POMDPs. *In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1339–1344. 2

[Bui, 2003] Bui, H. H. (2003). A General Model for Online Probabilistic Plan Recognition. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1309–1318. 2, 62

[Bui *et al.*, 2002] Bui, H. H., Venkatesh, S. et West, G. A. W. (2002). Policy Recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research (JAIR)*, 17:451–499. 2, 62

[Bui *et al.*, 2010] Bui, T. H., Zwiers, J., Poel, M. et Nijholt, A. (2010). Affective dialogue management using factored POMDPs. *In* Babuska, R. et Groen, F. C. A., éditeurs

: *Interactive Collaborative Information Systems*, volume 281 de *Studies in Computational Intelligence*, pages 209–238. Springer Verlag, Berlin. 139

[Cassandra *et al.*, 1994] Cassandra, A. R., Kaelbling, L. P. et Littman, M. L. (1994). Acting Optimally in Partially Observable Stochastic Domains. *In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1023–1028. 6, 53, 55, 66

[Cirillo *et al.*, 2009a] Cirillo, M., Karlsson, L. et Saffiotti, A. (2009a). A Human-Aware Robot Task Planner. *In Proceedings of the International Conference on Automated Planning and Scheduling: Workshop on Planning and Plan Execution for Real World Systems (ICAPS)*. 1, 25, 36

[Cirillo *et al.*, 2009b] Cirillo, M., Karlsson, L. et Saffiotti, A. (2009b). Human-Aware Task Planning for Mobile Robots. *In Proceedings of the International Conference on Advanced Robotics (ICAR)*, Munich, DE. 60, 69, 70

[Clark, 1996] Clark, H. H. (1996). *Using Language*. Cambridge University Press. 106

[Clodic, 2007] Clodic, A. (2007). *Supervision pour un robot interactif : Action et Interaction pour un robot autonome en environnement humain*. Thèse de doctorat, University of Toulouse. 106

[Clodic *et al.*, 2007] Clodic, A., Alami, R., Montreuil, V., Li, S., Wrede, B. et Swadzba, A. (2007). A Study of Interaction Between Dialog and Decision for Human-Robot Collaborative Task Achievement. *In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 913–918. 106

[Clodic *et al.*, 2008] Clodic, A., Cao, H., Alili, S., Montreuil, V., Alami, R. et Chatila, R. (2008). SHARY: A Supervision System Adapted to Human-Robot Interaction. pages 229–238. 106

[Cohen et Levesque, 1991] Cohen, P. R. et Levesque, H. J. (1991). Teamwork. *Nous*, 25(4): 487–512. 36, 106

[Dautenhahn, 2007] Dautenhahn, K. (2007). Socially intelligent robots: dimensions of human-robot interaction. *Hilosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704. 34

[Dautenhahn *et al.*, 2005] Dautenhahn, K., Woods, S., Kaouri, C., Walters, M., Koay, K. et Werry, I. (2005). What is a Robot Companion - Friend, Assistant or Butler? *In Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems*, pages 1488–1493. 68

[de Silva et Padgham, 2004] de Silva, L. et Padgham, L. (2004). A Comparison of BDI Based Real-Time Reasoning and HTN Based Planning. *In Australian Conference on Artificial Intelligence*, pages 1167–1173. 57

[Dehais *et al.*, 2011] Dehais, F., Sisbot, E. A., Alami, R. et Causse, M. (2011). Physiological and Subjective Evaluation of a Human-Robot Object Hand-Over Task. *Applied Ergonomics*, In Press, Corrected Proof:–. 106

[Dibangoye *et al.*, 2009] Dibangoye, J. S., Shani, G., Chaib-draa, B. et Mouaddib, A.-I. (2009). Topological Order Planner for POMDPs. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1684–1689. 7, 54, 55, 139

[Diuk et Littman, 2009] Diuk, C. et Littman, M. L. (2009). Hierarchical reinforcement learning. *In Encyclopedia of Artificial Intelligence*, pages 825–830. 52

[Doshi et Roy, 2008] Doshi, F. et Roy, N. (2008). Spoken Language Interaction with Model Uncertainty: an Adaptive Human-Robot Interaction System. *Journal of Connection Science - Language and Robots*, 20(4):299–318. 2, 65, 125

[Duong *et al.*,] Duong, T. V., Phung, D. Q., Bui, H. H. et Venkatesh, S. Efficient Duration and Hierarchical Modeling for Human Activity Recognition. *Journal of Artificial Intelligence Research (JAIR), year = 2009, volume = 173, pages = 830-856, number = 7-8, file = :files/duong-2009.pdf:PDF, keywords = SHSMM, intention.* 1, 2, 25, 36, 62, 65

[Feil-Seifer et Matarić, 2009] Feil-Seifer, D. J. et Matarić, M. J. (2009). *Human-Robot Interaction*, pages 4643–4659. Springer New York. 32, 34

[Ferguson et Allen, 2007] Ferguson, G. et Allen, J. F. (2007). Mixed-Initiative Systems for Collaborative Problem Solving. *AI Magazine*, 28(2):23–32. 33

[Fern *et al.*, 2007] Fern, A., Natarajan, S., Judah, K. et Tadepalli, P. (2007). A Decision-Theoretic Model of Assistance. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1879–1884. 2, 41, 60, 61, 63, 65, 66, 70, 91

[Fong, 2001] Fong, T. (2001). *Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation.* Thèse de doctorat, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. 34

[Gallese et Goldman, 1998] Gallese, V. et Goldman, A. (1998). Mirror Neurons and the Simulation Theory of Mind-Reading. 2(12):493–501. 8, 60

[Ghallab *et al.*, 2004] Ghallab, M., Nau, D. et Traverso, P. (2004). *Automated Planning: Theory and Practice.* 45

[Gray *et al.*, 2005] Gray, J., Breazeal, C., Berlin, M., Brooks, A. et Lieberman, J. (2005). Action Parsing and Goal Inference using Self as Simulator. *In Proceedings of Fourteenth IEEE Workshop on Robot and Human Interactive Communication (Ro-Man)*, pages 202–209. IEEE. 8, 60, 61

[Green *et al.*, 2008] Green, S., Billinghurst, M., Chen, X. et Chase, J. (2008). Human-Robot Collaboration: a Literature Review and Augmented Reality Approach in Design. 5:1–18. 35

[Guestrin *et al.*, 2001] Guestrin, C., Koller, D. et Parr, R. (2001). Solving Factored POMDPs with Linear Value Functions. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) workshop on Planning under Uncertainty and Incomplete Information*. 139

[Guestrin *et al.*, 2003] Guestrin, C., Koller, D., Parr, R. et Venkataraman, S. (2003). Efficient Solution Algorithms for Factored MDPs. *Journal of Artificial Intelligence Research (JAIR)*, 19:399–468. 10, 77

[Guestrin *et al.*, 2011] Guestrin, C., Koller, D., Parr, R. et Venkataraman, S. (2011). Efficient Solution Algorithms for Factored MDPs. *Computer Research Repository (CoRR)*, abs/1106.1822. 10, 77

[Hauskrecht, 2000] Hauskrecht, M. (2000). Value-Function Approximations for Partially Observable Markov Decision Processes. *Journal of Artificial Intelligence Research (JAIR)*, 13:33–94. 54, 55

[Hauskrecht et Kveton, 2003] Hauskrecht, M. et Kveton, B. (2003). Linear Program Approximations for Factored Continuous-State Markov Decision Processes. *In Proceedings of the Neural Information Processing Systems (NIPS)*. 55

[Hoey *et al.*, 2010] Hoey, J., Poupart, P., von Bertoldi, A., Craig, T., Boutilier, C. et Mihailidis, A. (2010). Automated Handwashing Assistance for Persons with Dementia using Video and a Partially Observable Markov Decision Process. *Computer Vision and Image Understanding*, 114(5):503–519. 2, 60, 65, 70, 125

[Hoffman et Breazeal, 2007] Hoffman, G. et Breazeal, C. (2007). Cost-Based Anticipatory Action Selection for Human-Robot Fluency. *Proceedings of the IEEE Transactions on Robotics*, 23(5):952–961. 2, 60, 65, 67, 69, 70

[Hoffman et Breazeal, 2008] Hoffman, G. et Breazeal, C. (2008). Achieving Fluency through Perceptual-Symbol Practice in Human-Robot Collaboration. *In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 1–8. 1, 25, 36

[Howard, 1960] Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA. 5, 52

[Hui et Boutilier, 2006] Hui, B. et Boutilier, C. (2006). Who's Asking for Help?: a Bayesian Approach to Intelligent Assistance. *In Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 186–193. 2, 40, 62

[J. Pineau et Thrun, 2003] J. Pineau, G. G. et Thrun, S. (2003). Point-Based Value Iteration: An Anytime Algorithm for POMDPs. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1032. 2

[Jenkins *et al.*, 2007] Jenkins, O. C., Serrano, G. G. et Loper, M. M. (2007). Tracking Human Motion and Actions for Interactive Robots. *In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 365–372. 41

[Jensen, 2001] Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. New York: Springer. 46

[Johnson *et al.*, 2010] Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Sierhuis, M. et van Riemsdijk, B. (2010). Toward Coactivity. *In Proceeding of the* 5^{th} *ACM/IEEE international conference on Human-robot interaction (HRI)*, HRI '10, pages 101–102, New York, NY, USA. ACM. 2, 11, 25, 34, 94

[Kaelbling *et al.*, 1998] Kaelbling, L. P., Littman, M. L. et Cassandra, A. R. (1998). Planning and Acting in Partially Observable Stochastic Domains. *Journal of Artificial Intelligence Research (JAIR)*, 101(1-2):99–134. 7, 53, 66

[Kaelbling *et al.*, 1996] Kaelbling, L. P., Littman, M. L. et Moore, A. W. (1996). Reinforcement Learning: A Survey. *Computer Research Repository (CoRR)*, cs.AI/9605103. 52

[Kamar *et al.*, 2009] Kamar, E., Gal, Y. et Grosz, B. J. (2009). Modeling User Perception of Interaction Opportunities for Effective Teamwork. *In Proceedings of the International Conference on Computational Science and Engineering (CSE)*, pages 271–277. 2, 64

[Karami *et al.*, 2009] Karami, A.-B., Jeanpierre, L. et Mouaddib, A.-I. (2009). Partially Observable Markov Decision Process for Managing Robot Collaboration with Human. *In Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 518–521. 73

[Karami *et al.*, 2010] Karami, A.-B., Jeanpierre, L. et Mouaddib, A.-I. (2010). Human-Robot Collaboration for a Shared Mission. *In Proceedings of the* 5^{th} *ACM/IEEE International Conference on Human Robot Interaction (HRI)*, pages 155–156. 73

[Karami et Mouaddib, 2011] Karami, A.-B. et Mouaddib, A.-I. (2011). A Decision Model of Adaptive Interaction Selection for a Robot Companion. *In Proceedings of the* 5^{th} *European Conference on Mobile Robots (ECMR)*, pages 83–88. 109

[Kaupp *et al.*, 2010] Kaupp, T., Makarenko, A. et Durrant-Whyte, H. F. (2010). Human-Robot Communication for Collaborative Decision Making - A Probabilistic Approach. *Robotics and Autonomous Systems*, 58(5):444–456. 64

[Kelley *et al.*, 2008] Kelley, R., Tavakkoli, A., King, C., Nicolescu, M. N., Nicolescu, M. et Bebis, G. (2008). Understanding Human Intentions via Hidden Markov Models in Autonomous Mobile Robots. *In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 367–374. 2, 62, 65, 70

[Kiesler et Hinds, 2004] Kiesler, S. et Hinds, P. (2004). Human-Robot Interaction. *Special Issue of Human-Computer Interaction*, 19:1–8. 32

[Klein *et al.*, 2004] Klein, G., Woods, D. D., Bradshaw, J. M., Hoffman, R. R. et Feltovich, P. J. (2004). Ten Challenges for Making Automation a "Team Player" in Joint Human-Agent Activity. *IEEE Intelligent Systems*, 19(6):91–95. 2, 26, 41

[Koay *et al.*, 2007] Koay, K. L., Sisbot, E. A., Syrdal, D. A., Walters, M. L., Dautenhahn, K. et Alami, R. (2007). Exploratory Study of a Robot Approaching a Person in the Context of Handling Over an Object. *In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposia*, Palo Alto, CA, USA. 106

[Koller et Parr, 2000] Koller, D. et Parr, R. (2000). Policy Iteration for Factored MDPs. *In Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (UAI), Stanford University, Stanford, California, USA*, pages 326–334. 10, 77

[Krauthausen et Hanebeck, 2009] Krauthausen, P. et Hanebeck, U. D. (2009). Intention Recognition for Partial-Order Plans Using Dynamic Bayesian Networks. 2, 62

[Levesque *et al.*, 1990] Levesque, H. J., Cohen, P. R. et Nunes, J. H. T. (1990). On Acting Together. *In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 94–99. 36, 41

[Li et Littman, 2005] Li, L. et Littman, M. L. (2005). Lazy Approximation for Solving Continuous Finite-Horizon MDPs. *In Proceedings of the National Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference*, pages 1175–1180. 55

[Li, 2005] Li, X. (2005). Active Affective State Detection and User Assistance With Dynamic Bayesian Networks. *In Proceedings of the IEEE International Conferenec on Systems, Man and Cybernetics (SMC), Part A:Systems and Humans*, pages 93–105. IEEE. 48

[Littman *et al.*, 1995] Littman, M. L., Cassandra, A. R. et Kaelbling, L. P. (1995). Learning Policies for Partially Observable Environments: Scaling Up. *In Proceedings of the International Conference on Machine Learning (ICML)*, pages 362–370. 54

[Matignon *et al.*, 2010] Matignon, L., Karami, A. et Mouaddib, A. I. (2010). A Model for Verbal and Non-Verbal Human-Robot Collaboration. *In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Fall Symposium Technical Reports on Dialog with Robots*, pages 62–67. 123, 126

[Mihajlovic et Petkovic, 2001] Mihajlovic, V. et Petkovic, M. (2001). Dynamic Bayesian Networks: A State of the Art. DMW-project. 48

[Montreuil *et al.*, 2007] Montreuil, V., Clodic, A., Ransan, M. et Alami, R. (2007). Planning Human Centered Robot Activities. *In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2618–2623. 68, 106

[Mouaddib *et al.*, 2010] Mouaddib, A.-I., Zilberstein, S., Beynier, A. et Jeanpierre, L. (2010). A Decision-Theoretic Approach to Cooperative Control and Adjustable Autonomy. *In Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 971–972. 33

[Myers *et al.*, 2007] Myers, K. L., Berry, P., Blythe, J., Conley, K., Gervasio, M. T., McGuinness, D. L., Morley, D. N., Pfeffer, A., Pollack, M. E. et Tambe, M. (2007). An Intelligent Personal Assistant for Task and Time Management. *AI Magazine*, 28(2):47–61. 60

[Nair et Tambe, 2005] Nair, R. et Tambe, M. (2005). Hybrid BDI-POMDP Framework for Multiagent Teaming. 23:367–420. 57, 67

[NASA, 2011] NASA (Retrieved 17 august 2011). 94

[Natarajan *et al.*, 2007] Natarajan, S., Tadepalli, P. et Fern, A. (2007). A Relational Hierarchical Model for Decision-Theoretic Assistance. *In Proceedings of the International Conference on Inductive Logic Programming (ILP)*, pages 175–190. 2, 60, 62, 66, 70

[Nguyen *et al.*, 2005] Nguyen, N. T., Phung, D. Q., Venkatesh, S. et Bui, H. H. (2005). Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 955–960. 2, 41, 60, 62, 70

[Pineau *et al.*, 2006] Pineau, J., Gordon, G. J. et Thrun, S. (2006). Anytime Point-Based Approximations for Large POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 27: 335–380. 7, 54

[Pineau *et al.*, 2003] Pineau, J., Montemerlo, M., Pollack, M. E., Roy, N. et Thrun, S. (2003). Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Systems*, 42(3-4):271–281. 1, 25, 36, 94, 125

[Pollack *et al.*, 2003] Pollack, M. E., Brown, L. E., Colbry, D., McCarthy, C. E., Orosz, C., Peintner, B., Ramakrishnan, S. et Tsamardinos, I. (2003). Autominder: an Intelligent Cognitive Orthotic System for People with Memory Impairment. *Robotics and Autonomous Systems*, 44(3-4):273–282. 2, 60, 62, 63, 69, 70

[Poole *et al.*, 1998] Poole, D., Mackworth, A. et Goebel, R. (1998). *Computational Intelligence: A Logical Approach*. New York: Oxford University Press. 33

[Poppe, 2010] Poppe, R. (2010). A Survey on Vision-Based Human Action Recognition. *Journal of Image and Vision Computing*, 28(6):976–990. 8, 74

[Puterman, 1994] Puterman, M. L. (1994). *Markov Decision Process: Discrete Stochastic Dynamic Programming*. John Wiley & Sons Inc. 10, 76

[Rabiner, 1989] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *In Proceedings of the IEEE*, pages 257–286. 62

[Rabiner et Juang, 1986] Rabiner, L. R. et Juang, B. H. (1986). An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16. 4, 49, 50, 62

[Rameson et Lieberman, 2009] Rameson, L. T. et Lieberman, M. D. (2009). Empathy: A Social Cognitive Neuroscience Approach. pages 94–110. 8, 60

[RoboCup@home, 2011] RoboCup@home (2011). http://www.ai.rug.nl/robocupathome, link = http://www.ai.rug.nl/robocupathome/. 16, 110, 119

[Rosenthal et Veloso, 2011] Rosenthal, S. et Veloso, M. M. (2011). Modeling Humans as Observation Providers using POMDPs. *In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2, 64, 66

[Roy *et al.*, 2000] Roy, N., Pineau, J. et Thrun, S. (2000). Spoken Dialogue Management Using Probabilistic Reasoning. *In Proceedings of the* 38^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*. 65

[Russell et Norvig, 2003] Russell, S. J. et Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey: Prentice Hall, 2nd édition. 33

[Scerri *et al.*, 2004] Scerri, P., Sycara, K. et Tambe, M. (2004). Adjustable Autonomy in the Context of Coordination. *In AIAA* $3^{rd} Unmanned Unlimited Technical Conference, Workshop and Exhibit, Invited Paper.$ 33

[Schmid et al.*, 2006]* Schmid*, A.,* Wörn*, H.,* Schrempf*, O. et* Hanebeck*, U. (2006). Towards Intuitive Human-Robot Cooperation.* Proceedings of the 2nd International Workshop on Human-Centered Robotic Systems (HCRS)*, pages 7–12. 64*

[Schmid et al.*, 2007]* Schmid*, A. J.,* Weede*, O. et* Wörn*, H. (2007). Proactive Robot Task Selection Given a Human Intention Estimate.* In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 726–731. 2, 62, 69, 70*

[Schmidt et al.*, 1978]* Schmidt*, C. F.,* Sridharan*, N. S. et* Goodson*, J. L. (1978). The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence.* Journal of Artificial Intelligence (JAIR)*, 11(1-2):45–83. 40*

[Schmidt-Rohr et al.*, 2008a]* Schmidt-Rohr*, S. R.,* Knoop*, S.,* Lösch*, M. et* Dillmann*, R. (2008a). Reasoning for a Multi-Modal Service Robot Considering Uncertainty in Human-Robot Interaction.* In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 249–254. 2, 41, 64, 70*

[Schmidt-Rohr et al.*, 2008b]* Schmidt-Rohr*, S. R.,* Lösch*, M. et* Dillmann*, R. (2008b). Human and Robot Behavior Modeling for Probabilistic Cognition of an Autonomous Service Robot.* In Proceedings of the 17^{th} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2*

[Schrempf et al.*, 2007]* Schrempf*, O. C.,* Albrecht*, D. et* Hanebeck*, U. D. (2007). Tractable Probabilistic Models for Intention Recognition based on Expert Knowledge.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1429–1434. 2, 63*

[Schrempf et Hanebeck, 2005] Schrempf*, O. C. et* Hanebeck*, U. D. (2005). A Generic Model for Estimating user Intentions in Human-Robot Cooperation.* In Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 251–256. 2, 62, 70*

[Schrempf et al.*, 2005]* Schrempf*, O. C.,* Hanebeck*, U. D.,* Schmid*, A. J. et* Wörn*, H. (2005). A Novel Approach to Proactive Human-Robot Cooperation.* In Proceedings of the 14^{th} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 64*

[Schut et al.*, 2002]* Schut*, M. C.,* Wooldridge*, M. et* Parsons*, S. (2002). On Partially Observable MDPs and BDI Models.* In Foundations and Applications of Multi-Agent Systems*, pages 243–260. 57*

[Shani et al.*, 2007]* Shani*, G.,* Brafman*, R. I. et* Shimony*, S. E. (2007). Forward Search Value Iteration for POMDPs.* In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2619–2624. 7, 54*

[Simari et Parsons, 2006] Simari*, G. I. et* Parsons*, S. (2006). On the Relationship Between MDPs and the BDI Architecture.* In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1041–1048. 57*

[Sisbot et al.*, 2010]* Sisbot*, E. A.,* Marin-Urias*, L. F.,* Broquere*, X.,* Sidobre*, D. et* Alami*, R. (2010). Synthesizing Robot Motions Adapted to Human Presence.* International Journal of Social Robotics*, 2(3):329–343. 1, 25, 106*

[Smith, 2005] Smith*, T. (2005). ZMDP Software for POMDP and MDP Planning. 7, 54, 84*

[Smith et Simmons, 2004] Smith, *T. et* Simmons, *R. (2004). Heuristic Search Value Iteration for POMDPs.* In Proceedings of the Conference in Uncertainty in Artificial Intelligence (UAI). *7, 54, 56*

[Smith et Simmons, 2006] Smith, *T. et* Simmons, *R. G. (2006). Focused Real-Time Dynamic Programming for MDPs: Squeezing More Out of a Heuristic.* In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI). *56*

[Sohrabi et al.*, 2009]* Sohrabi, *S.,* Baier, *J. A. et* McIlraith, *S. A. (2009). HTN Planning with Preferences.* In Proceedings of the 21^{st} International Joint Conference on Artificial Intelligence (IJCAI), *pages 1790 –1797, Pasadena, CA, USA. 45*

[Sondik, 1971] Sondik, *E. (1971).* The Optimal Control of Partially Observable Markov Decision Processes. *Thèse de doctorat. 53*

[Sondik, 1978] Sondik, *E. (1978). The Optimal Control of Partially observable Markov Decision process over Finite Horizon.* In Operational Research, *pages 24:282–304. 7, 54*

[Sutton et Barto, 1998] Sutton, *R. et* Barto, *A. (1998).* Reinforcement Learning: An Introduction. *MIT Press/Bradford Books. 52*

[Szepesvari, 2010] Szepesvari, *C. (2010).* Algorithms for Reinforcement Learning, Morgan and Claypool. *Morgan and Claypool, Cambridge, MA, USA. 52*

[Taha et al.*, 2008]* Taha, *T.,* Miró, *J. V. et* Dissanayake, *G. (2008). POMDP-Based Long-Term user Intention Prediction for Wheelchair Navigation.* In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), *pages 3920–3925. 2, 41, 60, 66, 70*

[Tambe, 2008] Tambe, *M. (2008). Electric Elves: What Went Wrong and Why.* AI Magazine, *29(2):23–27. 40*

[Walker et al.*, 2002]* Walker, *W.,* Lamere, *P. et* Kwok, *P. (2002). FreeTTS - A Performance Case Study. Rapport technique TR-2002-114. 135*

[Walker et al.*, 2004]* Walker, *W.,* Lamere, *P.,* Kwok, *P.,* Raj, *B.,* Singh, *R.,* Gouvea, *E.,* Wolf, *P. et* Woelfel, *J. (2004). Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Rapport technique. 135*

[Weiss, 1999] Weiss, *G., éditeur (1999).* Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. *MIT Press, Cambridge, MA, USA. 43*

[Williams, 2006] Williams, *J. D. (2006). Scaling POMDPs for Dialog Management with Composite Summary Point-based Value Iteration.* In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop for Spoken Dialogue Systems. *2, 125*

[Williams et al.*, 2005]* Williams*, J. D.,* Poupart*, P. et* Young*, S. (2005). Factored Partially Observable Markov Decision Processes for Dialogue Management.* In Proceedings of the 4^{th} Workshop on Knowledge and Reasoning in Practical Dialog Systems*, pages 76–82. 139*

[Williams et Young, 2007] Williams*, J. D. et* Young*, S. (2007). Scaling POMDPs for Spoken Dialog Management.* IEEE Transactions on Audio, Speech & Language Processing*, 15(7): 2116–2129. 64*

[Wooldridge, 2002] Wooldridge*, M. (2002).* An Introduction to Multiagent Systems*. John Wiley & Sons. 67*

[Yokoyama et Omori, 2010] Yokoyama*, A. et* Omori*, T. (2010). Modeling of Human Intention Estimation Process in Social Interaction Scene.* In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1–6. 62*

[Yorke-Smith et al.*, 2009]* Yorke-Smith*, N.,* Saadati*, S.,* Myers*, K. L. et* Morley*, D. N. (2009). Like an Intuitive and Courteous Butler: a Proactive Personal Agent for Task Management.* In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 337–344. 40, 64*

[Young et al.*, 2010]* Young*, S.,* Gašić*, M.,* Keizer*, S.,* Mairesse*, F.,* Schatzmann*, J.,* Thomson*, B. et* Yu*, K. (2010). The Hidden Information State Model: A Practical Framework for POMDP-Based Spoken Dialogue Management.* Journal of Computer Speech and Language*, 24(2):150–174. 2, 125*

[Zhou et Hansen, 2001] Zhou*, R. et* Hansen*, E. A. (2001). An Improved Grid-Based Approximation Algorithm for POMDPs.* In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 707–716. 54, 55*

[Zhu et Sheng, 2011] Zhu*, C. et* Sheng*, W. (2011). Wearable Sensor-Based Hand Gesture and Daily Activity Recognition for Robot-Assisted Living.* Proceedings of the IEEE International Conferenec on Systems, Man and Cybernetics (SMC), Part A*, 41(3):569–573. 8, 74*

Modèles Décisionnels d'Interaction Homme-Robot

Résumé : Nous étudions des Modèles Décisionnels pour l'Interaction Homme-Robot basés sur des Processus Décisionnels Markoviens. Premièrement, nous proposons un modèle de décision augmenté du robot compagnon afin d'agir en tenant compte de l'intention estimée de l'être humain. Ce modèle traite le problème d'estimation de l'intention de l'être humain en observant ses actions. Nous avons proposé de simuler le comportement de l'être humain afin de construire une bibliothèque de valeurs de ses actions par rapport à ses intentions possibles. Ces valeurs sont intégrées dans un Processus Décisionnel Markovien Partiellement Observable (POMDP). Nous parlerons alors de POMDP augmenté. Deuxièmement, nous avons présenté un modèle de décision qui permet au robot en collaboration avec un être humain de choisir son comportement selon l'avancement de la réalisation de la tâche partagée. Ce modèle est basé sur un POMDP augmenté et permet au robot d'être coactif afin d'inciter l'action de l'humain pour réaliser la tâche en harmonie avec lui. Troisièmement, nous avons aussi défini un modèle unifié pour diffèrent types d'interaction homme-robot où le robot analyse les besoins de l'humain et agit en conséquence. Afin de contourner la complexité des POMDPs, le modèle unifié sépare le problème en deux partie, une première responsable d'estimer les intention de l'humain avec une chaîne de Markov Cachée (HMM) et une deuxième responsable de choisir le type d'interaction correspondant (collaboration, assistance, coopération) avec un Processus Décisionnel Markovien (MDP). Finalement, nous proposons un modèle qui alterne entre interaction verbale afin d'inférer les préférence de l'humain et interaction non-verbale où les préférences sont estimés en observant les actions de l'humain. Ce modèle permet de revenir à l'interaction verbale quand une ambiguïté dans les préférences est détectée.

Mots-clés : Processus de Décision Markovien Partiellement Observable, Modèles formels d'Interaction Homme-Robot, Intelligence Artificiel, Robots compagnons.

Decisional Models for Human-Robot Interaction

Abstract: This thesis is focused on decision models for human-robot interaction based on Markovian Decision Processes. First, we propose an augmented decision model that allows a companion robot to act considering estimated human intentions. This model addresses the problem of estimating the intention of the human by observing his actions. We proposed to simulate the behavior of a human to build a library of human action values toward his possible intentions. These values are integrated into the augmented Partially Observable Markov Decision Process (POMDP). Second, we present a coactive decision model that allows a robot in collaboration with a human to choose his behavior according to the progress of the shared task. This model is based on an augmented POMDP and allows the robot to act coactively to encourage the human actions and to perform the task in harmony with him. Third, we also propose a unified model for different types of human-robot interactions where the robot analyzes the needs of the human and acts accordingly. To overcome the complexity of POMDPs, the unified model divides the problem into several parts, the first estimates the human intention with a hidden Markov model (HMM) and another is responsible for choosing the corresponding type of interaction (collaboration, assistance, cooperation) using a Markov Decision Process (MDP). Finally, we propose a model that alternates between verbal interaction to infer the preference of the human using queries and non-verbal interaction in which preferences are estimated by observing the human actions. This model switches back to the verbal interaction when an ambiguity about the preferences is detected.

Keywords: Partially Observable Markov Decision Processes, Formal models for Human-Robot Interaction, Artificial Intelligence, Companion robots.

Discipline : Informatique et applications
GREYC, CNRS UMR 6072
Université de Caen Basse-Normandie BP 5186
14032 Caen Cedex, FRANCE