Towards socially intelligent robots in human centered environment

Amit Kumar Pandey

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Spécialité
Robotique

Présentée et soutenue par :
Amit Kumar PANDEY

Le 20 juin 2012

Titre :

Towards Socially Intelligent Robots in Human Centered Environment

Directeur de Thèse : Rachid ALAMI

Jury

Président
Peter Ford DOMINEY (CNRS - INSERM Lyon)

Rapporteurs
Michael BEETZ (Université technique de Munich)
Philippe FRAISE (Université Montpellier II)

Examinateurs
Rodolphe GELIN (Aldebaran Robotics)
Thierry SIMEON (LAAS-CNRS)
Rachid ALAMI (LAAS-CNRS)

École doctorale : *EDSYS*
Unité de recherche : *LAAS-CNRS*

LAAS-CNRS
(Laboratory of Analysis and Architecture of Systems -
National Center for Scientific Research)

THESIS

to obtain the title of

Ph.D.

of University of Toulouse delivered by

INSA (National Institute of Applied Sciences), Toulouse

Doctoral School : EDSYS

Specialization : ROBOTICS

Defended by

Amit Kumar PANDEY

Towards Socially Intelligent Robots in Human Centered Environment

Thesis Advisor

Rachid ALAMI

prepared at ROBOTICS AND INTERACTIONS (RIS) GROUP, LAAS-CNRS

defended on *20 June 2012*

Jury :

President

Peter Ford DOMINEY, *Research Director, Robot Cognition Laboratory, CNRS - INSERM Lyon*

Reviewers

Michael BEETZ, *Professor, Technical University of Munich (TUM)*
Philippe FRAISSE, *Professor, University of Montpellier*

Examiners

Rodolphe GELIN, *Research Director, Aldebaran Robotics*
Thierry SIMEON, *Research Director, LAAS-CNRS*
Rachid ALAMI, *Research Director, LAAS-CNRS*

Acknowledgment

A *life*, its four vibrant years (*2008-2012*), an excellent and caring supervisor (*Rachid Alami*), four challenging European (EU) projects (*CHRIS*, *Dexmart*, *URUS*, *SAPHARI*), three distinct robots (*HRP2*, *Jido*, *PR2*), five contribution aspects (*research*, *development*, *integration*, *testing*, *user studies*), a jury, comprising of a noted president (*Peter Ford Dominey*), two prominent reviewers (*Michael Beetz*, *Philippe Fraisse*), and two renowned examiners (*Rodolphe Gelin*, *Thierry Simeon*), a quality Doctoral System (*EDSYS*), its supportive secretaries (*Hélène*, *Sophie*).

All these make me to feel *privileged*, to achieve the important milestone of my life, *this thesis*.

The valuable remarks of the anonymous reviewers, the precious feedback of the EU projects reviewers *(Anne Bajart*, *Frank Pasemann*, *Brian Scassellati*,...), the encouraging discussions with the distinguished visitors (*Mohamed Chetouani*, *Steve Cousins*, *Dominique Duhaut*, *Alexandra Kirsch*, *Lynne Parker*, *Charles Rich*,...), the brainstorming sessions with the eminent researchers in the group (*Malik Ghallab*, *Daniel Sidobre*, *Félix Ingrand*,...), the continuous help and support from senior research engineers (*Sara Fleury*, *Matthieu Herrb*, *Anthony Mallet*,...), the appreciations from my past mentors (*Sanajy Goel*, *K Madhava Krishna*,...).

I feel *fortunate* to have their encouragements and feedback, elevating the quality of the thesis.

The friendly, scientific and technical supports from my colleagues (*Akin*, *Ali*, *Alhayat*, *Assia*, *Aurélie*, *Jean-Philippe*, *Jim*, *Ibrahim*, *Lavindra*, *Luis*, *Mamoun*, *Manish*, *Matthieu*, *Mokhtar*, *Naveed*, *Oussama*, *Raquel*, *Riichiro*, *Romain*, *Sabita*, *Samir*, *Séverin*, *Wuwei*, *Xavier*,...), the love and support from my *wonderful family*, the delighting presence of my *friends* and their fantastic families.

I feel like a *special* to have them, boosting a cheerful professional live with a joyful personal life in complete harmony.

Thanks to all, I gained a thesis, an expanded circle of marvelous friends and an enlightened direction for the path ahead...

Dedicated to my MOTHER and FATHER...

\- Amit Kumar Pandey

Abstract

Towards Socially Intelligent Robots in Human Centered Environment

Robots are no longer going to be isolated machines working in factory or merely research platforms used in controlled lab environment. Very soon, robots will be the part of our day-to-day lives. Whether it is street, office, home or supermarket, robots will be there to assist and serve us. For such robots to be accepted and appreciated, they should explicitly consider the presence of human in all their planning and decision making strategies, whether it is for motion, manipulation or interaction. This thesis explores various socio-cognitive aspects ranging from perspective-taking, social navigation behaviors, cooperative planning, proactive behaviors to learning task semantics from demonstration. Further, by identifying key ingredients of these aspects, we equipped the robots with basic socio-cognitive intelligence, as a step towards making the robots to co-exist with us in complete harmony.

In the context of socially acceptable navigation of a robot, it is a must that the robot should no longer treat us, the human, only as dynamic obstacles in the environment. For example, the robot should even decide to take a longer path, if it is satisfying the human's desire and expectation and not creating any confusion, fear, anger or surprise by its motion. This requires the robot to be able to reason about various criteria ranging from clearance, environment structure, unknown objects, social conventions, proximity constraints, presence of an individual or a group of people, etc.

Similarly, for the task when the robot has to guide a person from his/her current position to another place, it should support the person's activities and guide him/her in the way he/she wants to be guided. It is quite natural that there will be intentional or unintentional deviations in the person's motion from the path expected by robot. Further, because of person's behavior of leave-taking or temporary suspending the guiding process, if required, the robot should exhibit goal oriented approaching and re-engagement behaviors.

A human friendly robot should neither be over-reactive nor be simple wait and move machine.

On the other hand, when a robot has to explicitly work together with us in a cooperative Human-Robot Interactive manipulation scenario, it should be able to analyze various abilities and affordances of the person it is interacting with. Such capabilities of perspective taking is important for various decisions e.g. where to put an object so that human can reach it with least effort, where and how to show an object to the human, how to grasp an object so that human can also grasp it

for object hand-over tasks, etc. All these require the robot to reason beyond the stability of object's grasp and placement even for basic tasks such as show, give, hide make-accessible, put away, etc.

Capability to ground day-to-day interaction with the human, to ground the changes in the environment, which happened in the absence of the robot, to generate a shared plan for solving day-to-day tasks, such as clean the table, are some of the other important aspects for the existence of the robots in our day-to-day life. The grounding could be in terms of the object that the human is trying to refer, the agents and the actions, which might be responsible behind some changes, whereas the task planning could be deciding possible cooperation and help among different agents. All these requires the robot to reason at different levels for planning the task: at symbolic level to decide how to achieve the task and to assign roles to the agents; at geometric level to ensure the feasibility of the actions. Further, reasoning on the efforts and current state and desire of the agents should be taken into account to decide about the amount, extent and method of cooperation, and for grounding interaction and changes.

Another aspect of socio-cognitive interaction is behaving proactively, i.e. planning and acting in advance by anticipating the future needs, problems or changes.This demands the robots to be capable of reasoning about how to behave proactively, where to behave proactively to support ongoing interaction or task and so on.

Learning from demonstration of day-to-day tasks is an important aspect for the robot to efficiently perform the tasks. Even for basic tasks such as give, hide, make accessible, show, etc., depending upon the situation, the same task could be performed entirely differently. We should not expect that for each and every task, the robot will be provided with a situation-by-situation based example about how to perform that task. Hence, just imitating the actions of a demonstration is not sufficient. The robot should be able to understand the goal of the demonstration, i.e. what does the task mean in terms of desired effect. The robot should learn it autonomously at appropriate level of abstraction to be able to reproduce them, in diverse situations in different ways. It requires reasoning beyond the levels of trajectory and sub-actions.

This thesis focuses on these issues, which raise new challenges that cannot be handled appropriately by simple adaptation of state of the art robotics planning, control and decision making techniques. The thesis, first identifies such basic socio-cognitive ingredients from the child development and human behavioral psychology research and presents the general architecture for socially intelligent human-robot interaction. Next, we will present a generalized domain theory for Human Robot Interaction (HRI) and derive various research challenges under a unified framework. Further, we will introduce new terms and concepts from HRI point of view and develop frameworks for integrating them in robot's motion, manipulation and interaction behaviors. Implementation results on different types of real robots (PR2, HRP2, Jido,...) will show the proof of concept. This is a step towards Socially Intelligent Robots with the vision to build a base for developing more complex socio-cognitive robot behaviors for future co-existence of human and robot in complete harmony.

Keywords: *Human Robot Interaction (HRI), Theory of HRI, Socially Intelligent Robot, Reasoning about Human, Multi-State Perspective Taking, Mightability Analysis, Mightability Maps, Shared Attention, Situation Assessment, Agent State Analysis, Human-Robot Interactive Manipulation, Spatial Reasoning, Socially Aware Navigation, Social Robot Guide, Cooperative Robot, Proactive Behavior, Theory of Proactivity, Shared Plan, Affordance Graph, Grounding Interaction, Grounding Changes, Learning from Demonstration, Emulation Learning, Domestic Robots, Robot Assistant, Service Robot.*

Contents

Acknowledgment **i**

Abstract **iii**

1 Introduction **1**
1.1 Motivation: *Manava*, The Robot 1
1.1.1 Child Development Research 4
1.1.1.1 Visuo-Spatial Perspective Taking 4
1.1.1.2 Social Learning 5
1.1.1.3 Pro-social and cooperative behaviors 5
1.1.2 Human Behavioral Psychology Research 6
1.1.2.1 How do We Plan to Manipulate 6
1.1.2.2 Grasp Placement Interdependency 6
1.1.2.3 How do We Navigate 7
1.1.2.4 Social Forces of Navigation 7
1.2 Socially Intelligent Robot 8
1.2.1 Social Intelligence Embodiment Pyramid 8
1.2.2 Scope and Focus of the Thesis 9
1.2.3 Approach: Bottom-up Social Embodiment 11
1.3 Outline of the Thesis 11

2 Related Works, Research Challenges and the Contribution **13**
2.1 Introduction 13
2.2 Visuo-Spatial Perspective Taking, Situation Awareness, Effort and Affordances Analyses for Human-Robot Interaction 13
2.3 Social Navigation in Human Environment and Socially Aware Robot Guide 20
2.4 Manipulation in Human Environment 25
2.5 Grounding Interaction and Changes, Generating Shared Cooperative Plans 28
2.6 Proactivity in Human Environment 32
2.7 Learning Task Semantics in Human Environment 34

3 Generalized Framework for Human Robot Interaction **39**
3.1 Introduction 39
3.2 Environmental Changes are Causal 40
3.3 HRI Generalized Domain Theory 41
3.3.1 HRI Oriented Environmental Attributes 41
3.3.2 HRI Oriented General Definition of Environmental Changes . . 47
3.3.3 HRI Oriented General definition of Action 48

3.4 Development of Unified Framework for deriving HRI Research Challenges . . . 50
3.4.1 Task Planning Problem . . . 50
3.4.2 Constraint Satisfaction Problem . . . 51
3.4.3 Partial Plan . . . 52
3.4.4 Deriving HRI Research challenges . . . 52
3.4.4.1 Perspective Taking, Ability and Affordance Analysis 52
3.4.4.2 HRI Manipulation Task Planning . . . 52
3.4.4.3 HRI Navigation Task Path Planning . . . 54
3.4.4.4 Learning from Demonstration . . . 55
3.4.4.5 Predicting Future States . . . 56
3.4.4.6 Synthesizing Past State . . . 57
3.4.4.7 Grounding Interaction and Changes . . . 57
3.4.4.8 Synthesizing Proactive Behavior . . . 57
3.5 Switching among Different Representations and Encoding: State-Variable Representation . . . 58
3.6 Until Now and The Next . . . 60

4 Mightability Analysis: Multi-State Visuo-Spatial Perspective Taking 61
4.1 Introduction . . . 61
4.2 3D World Representation . . . 63
4.2.1 Discretization of Workspace . . . 64
4.2.2 Extraction of Support Planes and Places . . . 65
4.3 Visuo-Spatial Perspective Taking . . . 65
4.3.1 Estimating Ability *To See*: *Visible, Occluded, Invisible* . . . 65
4.3.1.1 For Places . . . 65
4.3.1.2 For Objects . . . 66
4.3.2 Finding Occluding Objects . . . 67
4.3.3 Estimating Ability *To Reach*: *Reachable, Obstructed, Unreachable* . . . 67
4.3.3.1 For Places . . . 67
4.3.3.2 For Objects . . . 68
4.3.4 Finding Obstructing Objects . . . 68
4.4 Effort Analysis . . . 69
4.4.1 Human-Aware Effort Analyses: Qualifying the Efforts . . . 70
4.4.2 Quantitative Effort . . . 72
4.5 Mightability Analysis . . . 72
4.5.1 Estimation of Mightability . . . 73
4.5.1.1 Treating Displacement Effort . . . 75
4.5.1.2 Mightability Map (MM) . . . 76
4.5.1.3 Object Oriented Mightability (OOM) . . . 79
4.5.2 Online Updation of Mightabilities . . . 79
4.6 Mightability as Facts in the Environment . . . 80

4.7 Analysis of Least Feasible Effort for an Ability 83
4.8 Visuo-Spatial Ability Graph 85
4.9 Until Now and The Next 85

5 Affordance Analysis and Situation Assessment 87
5.1 Introduction 87
5.2 Affordances 87
5.2.1 Agent-Object Affordances 89
5.2.2 Object-Agent Affordances 90
5.2.3 Agent-Location Affordances 91
5.2.4 Agent-Agent Affordances 91
5.2.4.1 Considering Object Dimension 96
5.3 Least Feasible Effort for Affordance Analysis 96
5.4 Situation Assessment 96
5.4.1 Agent States 97
5.4.2 Object States 103
5.4.3 Attentional Aspects 105
5.5 Until Now and The Next 106

6 Socially Aware Navigation and Guiding in the Human Environment 107
6.1 Introduction 108
6.2 Socially-Aware Path Planner 109
6.2.1 Extracting Environment Structure 109
6.2.2 Set of Different Rules 111
6.2.2.1 General Social Conventions (S-rules) 111
6.2.2.2 General Proximity Guidelines (P-rules) 112
6.2.2.3 General Clearance Constraints (C-rules) 113
6.2.3 Selective Adaptation of Rules 113
6.2.4 Construction of Conflict Avoidance Decision Tree 114
6.2.5 Dealing with Dynamic Human 116
6.2.6 Dealing with Previously Unknown Obstacles 116
6.2.7 Dealing with a Group of People 117
6.2.8 Framework to Generate Smooth Socially-Aware Path 117
6.2.9 Proof of Convergence 122
6.3 Experimental Results and Analysis 122
6.3.1 Comparative analysis of *Voronoi Path* vs. *Socially-Aware Path* vs. *Shortest Path* 122
6.3.2 Analyzing Passing By, Over Taking and Conflict Avoiding Behaviors 123
6.3.3 Qualitative and Quantitative Analyses of Generated Social Navigation with Purely Reactive Navigation Behaviors 129
6.4 Social Robot Guide 131
6.4.1 Regions around the Human 132

6.4.2 Non-Leave-Taking Human Activities . . . 133
6.4.3 Belief about the Human's Joint Commitment . . . 133
6.4.4 Avoiding Over-Reactive Behavior . . . 134
6.4.5 Leave-Taking Human Activity . . . 135
6.4.6 Goal Oriented Re-engagement Effort . . . 135
6.4.6.1 Prediction of Meeting Point . . . 135
6.4.6.2 Deciding Next Point towards Goal . . . 136
6.4.6.3 Deciding the set of points to deviate . . . 137
6.4.6.4 Generating smooth path to deviate . . . 137
6.4.7 Human Activity to be Re-engaged . . . 138
6.4.8 Searching for the Human . . . 140
6.4.9 Breaking the Guiding Process . . . 141
6.5 Experimental Results and Analysis . . . 141
6.6 Until Now and The Next . . . 145

7 Planning Basic HRI Tasks 147
7.1 Introduction . . . 148
7.2 How do we plan . . . 149
7.3 Problem Statement from HRI Perspective . . . 149
7.3.1 Components of a Placement . . . 150
7.3.2 Synthesizing Configuration . . . 150
7.3.3 Generating Trajectory . . . 150
7.3.4 Grasp-Placement inter-dependency . . . 150
7.3.5 A set of constraint classes . . . 150
7.4 Generation of Object Property Database . . . 151
7.4.1 Set of Possible Grasps . . . 151
7.4.2 Set of *To Place* in space orientations . . . 151
7.4.3 Set of *To Place* on plane orientations . . . 152
7.5 Realization of Key Constraints . . . 153
7.5.1 Constraint of Simultaneous Compatible Grasps . . . 153
7.5.2 Visuo-Spatial Constraints on 'To Place' Positions . . . 153
7.5.3 Object alignment constraints from the human's perspective . 153
7.5.4 Robot's wrist alignment constraint from the human's perspective 154
7.5.5 Collision free configuration constraint (CFC) . . . 154
7.5.6 Constraints on quantitative visibility . . . 155
7.6 Framework for Planning *Pick-and-Place* Tasks: Constraint Hierarchy based Approach . . . 155
7.7 Instantiation for Basic Tasks . . . 157
7.7.1 Show an object to the human . . . 159
7.7.2 Make an object accessible to the human . . . 159
7.7.3 Give an object to the human . . . 159
7.7.4 Hide an object from the human . . . 160
7.8 Experimental Results and Analysis . . . 160
7.8.1 Generalized system for different robots: JIDO, PR2, HRP2 . 160

7.8.1.1 Show Task . . . 161
7.8.1.2 Give Task . . . 162
7.8.1.3 Make-Accessible Task . . . 163
7.8.1.4 Hide Task . . . 166
7.8.2 Effect of constraints' parameters variations . . . 172
7.8.3 Convergence and Performance . . . 174
7.9 Until Now and The Next . . . 174

8 Affordance Graph: an Effort-based Framework to Ground Interaction and Changes, to Generate Shared Cooperative Plan 175
8.1 Introduction . . . 176
8.2 Incorporating Effort in Grounding and Planning Cooperative Tasks . 178
8.3 Decision on Effort Levels . . . 179
8.4 Taskability Graph . . . 180
8.5 Manipulability Graph . . . 183
8.6 Affordance Graph . . . 185
8.7 Computation Time . . . 188
8.8 Potential Applications . . . 189
8.8.1 Grounding Interaction, Agent, Action and Object . . . 190
8.8.2 Generation of Shared Cooperative Plan . . . 190
8.8.3 A remark on planning complexity . . . 196
8.8.4 Grounding Changes, Analyzing Effects and Guessing Potential Action and Effort . . . 198
8.8.5 Supporting High-Level Symbolic Task Planners . . . 202
8.9 Two Way Hand Shaking of Geometric-Symbolic Planners . . . 202
8.9.1 The Geometric Task Planner . . . 202
8.9.1.1 Layers of Geometric Planner . . . 202
8.9.2 The Symbolic Planner . . . 205
8.9.3 The Hybrid Planning Scheme . . . 205
8.9.3.1 System Demonstration . . . 207
8.10 Until Now and The Next . . . 209

9 Prosocial Proactive Behavior 211
9.1 Introduction . . . 211
9.2 Generalized Theory of Proactivity for HRI . . . 213
9.2.1 Proactive Action . . . 213
9.2.2 Proactive Action Planning Problem . . . 213
9.2.3 Spaces for Proactivity . . . 213
9.2.4 Proposed Levels of Proactive Behaviors . . . 215
9.2.4.1 Level-1 Proactive Behavior . . . 215
9.2.4.2 Level-2 Proactive Behavior . . . 216
9.2.4.3 Level-3 Proactive Behavior . . . 217
9.2.4.4 Level-4 Proactive Behavior . . . 218
9.3 Instantiation . . . 220

9.3.1 Objective of the hypothesized proactive behavior 221
9.3.2 Hypothesized Proactive Behavior for Evaluation 224
9.3.2.1 Proactive Reach Out to *Take* from the Human . . . 224
9.3.2.2 Proactively Suggesting 'where' to Place 224
9.3.3 Hypotheses about the effects of the human-adapted proactive behaviors in the joint task 224
9.3.3.1 Reduction in human partner's *confusion* 224
9.3.3.2 Reduction in human partner's *effort* 224
9.3.3.3 Effect on *perceived awareness* of the robot 224
9.3.4 Framework to Instantiate 'where' based Proactive Action . . 225
9.4 Illustration of the framework for different tasks 227
9.4.1 For "Give" task by the human: Proactively reaching out . . . 227
9.4.2 For "Make Accessible" task by human: Suggesting 'where' to place 230
9.4.3 Remark on convergence time 230
9.5 Experimental results 231
9.5.1 Demonstration of the proactive planner and analysis of human effort reduction in different scenarios 232
9.5.1.1 For proactive reach out for 'give' task by the human in different scenarios 232
9.5.1.2 Finding solution to proactively suggest the place for make accessible task in different scenarios 233
9.5.2 Validation of Hypotheses and Discoveries through User Studies 236
9.5.2.1 For "give" task by the user 236
9.5.2.2 For "make accessible" task by the user 240
9.5.2.3 Overall inter-task observations 244
9.6 Discussion on some complementary aspects and measure of proactivity 244
9.7 Until Now and The Next 246

10 Task Understanding from Demonstration 247
10.1 Introduction 248
10.2 Predicates as Hierarchical Knowledge Building 249
10.2.1 Quantitative facts: agent's least efforts 249
10.2.2 Comparative fact: relative effort class 250
10.2.3 Qualitative facts: nature of relative effort class 251
10.2.4 Visibility score based hierarchy of facts 251
10.2.5 Symbolic postures of agent and relative class 252
10.2.6 Symbolic status of objects 252
10.2.7 Object status relative class and nature 253
10.2.8 Human's hand status 253
10.2.9 Hand status relative class and nature 254
10.2.10 Object motion status and relative motion status class 254
10.3 Explanation based Task Understanding 255
10.3.1 General Target Goal Concept To Learn 256

10.3.2 Provided Domain Theory 256
10.3.3 m-estimate based refinement 257
10.3.4 Consistency Factor 258
10.4 Experimental Results and Analysis 260
10.4.1 Show an object 262
10.4.2 Hide an object 265
10.4.3 Make an object accessible 267
10.4.4 Give an Object 268
10.4.5 Put-away an object 269
10.4.6 Hide-away an object 270
10.5 Performance Analysis 271
10.5.1 Processing Time 271
10.5.2 Analyzing Intuitive and Learnt Understanding 272
10.6 Practical Limitations 274
10.7 Potential Applications and Benefits 274
10.7.1 Reproducing Learnt Task 274
10.7.2 Generalization to novel scenario 275
10.7.3 Greater flexibility to high-level task planners 276
10.7.4 Transfer of understanding among heterogeneous agents 277
10.7.5 Understanding by observing heterogeneous agents 277
10.7.6 Generalization for multiple target-agents 277
10.7.7 Facilitate task/action recognition and proactive behavior . . . 277
10.7.8 Enriching Human-Robot interaction 278
10.7.9 Understanding other types of tasks 278
10.8 Until Now and The Next 278

11 Conclusion 281
11.1 Main Contributions 281
11.2 Prospects 285
11.2.1 Immediate Potential Applications 285
11.2.2 Future Work 286
11.2.3 Future Technology Transfer Activities 287
11.3 Two Lines 288
11.4 One Line 288

A System Architecture 289
A.1 System Components 290
A.2 Perception of the World 290

B Human-Robot Competition Game 293
B.1 The Context and The Game 293
B.2 The Scenario 294
B.3 The Human's and The Robot's Explanations about the Observed Changes in the Environment and the Guessed Course of Actions . . 294

C Publications and Associated Activities 299
C.1 List of publications 299
C.2 Associated EU Projects 301
C.3 Associated Scientific Gathering Activities 301

Index 303

Bibliography 307

D Résumé en français 329

E Vers des robots socialement intelligents en environnement humain 331
E.1 Introduction 332
E.2 Pourquoi un robot social ? 334
E.2.1 Les ingrédients de l'intelligence sociale 334
E.2.2 Le robot social/sociable 335
E.2.3 Pyramide de l'incarnation de l'intelligence sociale 336
E.2.4 Notre approche de l'incarnation sociale 336
E.3 Travaux Connexes, Challenges et Contribution 337
E.4 Un cadre conceptuel pour l'Interaction Homme-Robot 337
E.5 Analyse de "Mightability": Prise de perspective spatio-visuel multi-états 338
E.5.1 Hiérarchie des efforts 338
E.5.2 Analyse de la Mightability 339
E.6 Analyse d'affordance et Evaluation de la situation 339
E.7 Navigation et Guidage socialement adaptés en environnement humain 342
E.7.1 Planificateur de trajectoire socialement acceptable 342
E.7.2 Robot guide 342
E.8 Planification de tâches basiques pour l'interaction homme-robot . . . 347
E.9 Graphe d'affordance: Un cadre basé sur les efforts pour établir l'interaction et la génération de plan partagée 348
E.9.1 Taskability Graph 348
E.9.2 Manipulability Graph 350
E.9.3 Affordance Graph 351
E.10 Comportement pro-social pro-actif 352
E.10.1 Proposition de niveaux de comportements pro-actifs 353
E.10.2 Instanciation de comportement pro-actifs 354
E.10.3 Etudes utilisateur 354
E.11 Compréhension de tâche par démonstration 355
E.11.1 Apprentissage via l'explication et l'utilisation d'un arbre d'hypothèses initiales 358
E.11.2 Facteur de cohérence 360
E.11.3 Bénéfices et applications possibles 364
E.12 Conclusion 365

CHAPTER 1

Introduction

Contents

1.1 Motivation: *Manava*, The Robot **1**
1.1.1 Child Development Research 4
1.1.2 Human Behavioral Psychology Research 6
1.2 Socially Intelligent Robot **8**
1.2.1 Social Intelligence Embodiment Pyramid 8
1.2.2 Scope and Focus of the Thesis 9
1.2.3 Approach: Bottom-up Social Embodiment 11
1.3 Outline of the Thesis **11**

1.1 Motivation: *Manava*, The Robot

The robot ***Manava*** *has been hired recently as an assistant in a Luxury Hotel. It is afternoon and rush hour to check-in. Mr. John, the manager, requested, "Please guide Mr. Smith to room number 108". Manava asks, "May I have the access key?" Interestingly while asking Manava does not stand still in his current posture, instead it plans where Mr. John could hand over the keys with least feasible effort and proactively stretches out its hand to take the key from him. Mr. John smiles and hands-over the key. Having the access key, Manava approaches Mr. Smith, greets him and starts to "take" him to the room. On the way in the lobby Mr. Kumar's family is coming. Manava "smoothly" adapts its path to politely pass by Mr. Kumar's family from their left sides. Manava deliberately did not pass amid them or from their right sides, hence did not create any confusion or discomfort for Mr. Kumar's family members. Now they are moving in a hallway, the robot is maintaining itself on the right half of the hallway, so that Ms. Leena smoothly passes by with her great smile without any discomfort or confusion. Down the hallway, Mr. Smith finds an interesting painting and stops for a while to take a look. Manava adapts its motion to support Mr. Smith's activity while showing destination oriented inclination. Further while passing through the lounge, Mrs. Amelia was moving slowly with a walker. Manava smoothly adapts his path to overtake Mrs. Amelia from her left side by maintaining appropriate proximity. Manava deliberately did not overtake from the right side of Mrs. Amelia, and she continues, as she does not* *notice anything uncomfortable. On the way, Mr. Smith sees his important client Mr. Lee and spontaneously reaches towards him. Manava does not terminate the task, instead it approaches Mr. Smith to again establish the guiding process from the expected meeting place. Again, the path to approach is inclined towards the next place to move to achieve the task of taking Mr. Smith to the destined room. As Mr. Smith is now comfortable with Manava, he predicts the next via place and moves ahead of Manava to reach there. Manava does not show any unnecessary reactive motion. Finally, they reach to the room number 108.*

Tired Mr. Smith asks for beer, Manava goes ahead to fetch the beer bottle. Interestingly when grasping the bottle Manava thinks about the associated task in terms of what to do with the bottle and where and how. Therefore, it deliberately grabs the bottle in such a way, which leaves sufficient space for Mr. Smith to take the bottle. Then it approaches towards Mr. Smith and gives the bottle at a place, which requires Mr. Smith to put least effort to see and take it. Intelligently while giving the bottle Manava maintains the front and top of the bottle visible from Mr. Smith's perspective. This makes Mr. Smith aware about the "object" he is taking. Happy Mr. Smith "rates" Manava by pressing the "rate me" button twice.

Manava now returns to the reception lobby. There is not much work, but as being a curious robot, it is observing the activities of people around. On the corner table while preparing the coffee, Sam asks her sister Ammy, "Can you make the sugar container accessible to me?". Ammy takes the container, puts it somewhere and runs away to play with the toys nearby. By observing the effect of Ammy's action Manava understands a new task "Make Accessible object X" as: "X should be easier to be reached and seen by the target-person". Manava is happy to learn a new task and could not resist itself from beeping spontaneously.

It's now the dinnertime, and Manava has been asked to assist at Mr. Kumar's dining table. Manava is fetching the items one by one. Mr. Kumar is searching for something. Manava looks for the items which are hidden from Mr. Kumar's perspective and hints most relevant item, "Are you looking for the salt, it is behind the Jug on your right". Manava deliberately does not reach to the salt to take and give it to Mr. Kumar, as it estimates that if Mr. Kumar will just lean forward, he can see and reach the salt container. Hence, Manava is interestingly able to analyze the ability to reach and see from Mr. Kumar's perspective not only from his current state but also from a virtual state: if he will lean forward.

In the kitchen, chief chef is making spicy chicken curry. Manava proactively anticipates the need of curry powder by the chef. It finds that curry powder container is not reachable by the chef from his current position but Manava can reach it from its current position. As being far from the chief chef, Manava requests the assistant chef, "can you please make this curry powder accessible to the chef" and gives the container to him. Interestingly Manava did not plan to go and make it accessible directly to the chief chef, as it finds an alternative plan with less overall time and effort. Further as chef is busy now, instead of giving the container in the hand of *chef, Manava plans to make the curry powder accessible to him, the make accessible task, which he has learnt newly. Manava is also intelligent enough to estimate the ability of assistant chef to make some object accessible to the chef and his ability to take some object from Manava with least effort. Surprisingly happy with Manava, the chef also rates it by pressing the "rate me" button thrice. And a happy Manava goes to recharge itself to take up the watchdog responsibility in the night.*

Manava is a kind of *intelligent social robot*, which supports the vision of this thesis:

"Human and robot should co-exist in complete harmony"

But, why *Manava* is Social? Because it is...

"...living or disposed to live in companionship with others or in a community, rather than in isolation..." (definition of social, [dictionary.reference.com])

Hence, we derive our motivation for this thesis: To explore various socio-cognitive building blocks as exhibited by *Manava*: perspective taking, proactivity, following social norms of navigation, reducing effort and confusion, learning from our day to day activity, planning cooperative tasks, etc. to design and develop algorithms and frameworks to equip the robots with such socio-cognitive abilities.

In fact, *Manava* is not far from being a reality. Robots are already entering into our day-to-day lives. They are expected to help and cooperate [Project], guide [Thrun 2000], or even play with us, teach us (see HRI survey [Goodrich 2007]) and that too with lifelong learning from our day-to-day activities [Pardowitz 2007].

When looked through the socio-cognitive window, the *AI (Artificial Intelligence)*, hence *artificial agents* should be able to take into account high level factors of other agents such as help and dependence, [Miceli 1995]. Here the agents' *social reasoning and behavior* is described as their *ability to gather information about others and of acting on them to achieve some goal.* Which obviously means such agents should not exist in isolation, instead must *fit* in with the current work practice of both people and other computer systems (agents), [Bobrow 1991]. While exploring this *'fit'*, works on social robots such as [Breazeal 2003], and survey of socially interactive robots such as [Fong 2003] altogether outline various types of social embodiment. This could be summarized as ***social interfaces*** to communicate; ***sociable*** robots, which engage with humans to satisfy internal social aims; ***socially situated*** robots, which must be able to distinguish between 'the agents' and 'the objects' in the environment; ***socially aware*** robots, situated in social environment and aware about the human; ***socially intelligent*** robots that show aspects of human style social intelligence.

And the *Manava* robot "dreamed" above is equipped with such basic socio-cognitive aspects to *fit* in our environment: reasoning from others' perspective, proactive behaviors, navigating by maintaining social norms, learning task semantics at human understandable symbolic level, performing day-to-day human interactive object manipulation task in the way accepted and expected by us, and so on.

As we will discuss next, the existence of basic socio-cognitive abilities become evident from the age of 12 months and as we grow, we acquire more complex socio-cognitive abilities and behaviors.

1.1.1 Child Development Research

1.1.1.1 Visuo-Spatial Perspective Taking

From the research of child development, visuo-spatial perception comes out to be an important aspect of cognitive functioning such as accurately reaching for objects, shifting gaze to different points in space, etc. Very basic forms of social understandings, such as following gaze and pointing of other's as well as directing other's attention by pointing, begins to reveal in children as early as at the age of 12 months, [Carpendale 2006]. At 12-15 months of age children start showing the evidence of an understanding of occlusion of others' line-of-sight [Dunphy-Lelii 2004], [Caron 2002]; and an adult is seeing something that they are not when looking to locations behind them or behind barriers [Deak 2000], for both: the places [Moll 2004] and the objects [Csibra 2008]. In [Flavell 1977] two levels of development of visual perspective taking in children have been hypothesized and further validated [Flavell 1981]. At earlier development, which Flavell calls as *level 1*, children starts to understand, which object the other person can see and later they develop *level 2*, that others can have different view of the same object when looking at it from different positions. Having developed such key cognitive abilities, the children could then show basic social interaction behaviors. For example, intentionally producing visual percept in another person by pointing and showing things and interestingly from the early age of 30 months, they could even deprive a person of a pre-existing percept by hiding an object from him/her [Flavell 1978]. Further studies such as [Rochat 1995], suggest that from the age of 3 years, children are able to perceive, which places are reachable by them and by others, as the sign of early development of allocentrism capability, i.e. spatial decentration and perspective taking. Evolution of such basic socio-cognitive abilities of visuo-spatial reasoning in children enable them to help, co-operate and understand the intention of the person they are interacting with.

Motivated from above evidences of basic socio-cognitive aspects, we will first equip the robot with such perspective taking capabilities of perceiving abilities to see and reach by self and others. Then based on these we will develop the frameworks to share the attention; produce visual percept, such as show an object; deprive visual percept, such as hide an object; facilitate reach by making an object accessible or directly giving it; deprive reaching by putting away.

1.1.1.2 Social Learning

From the perspective of social learning, which in loose sense is "A observes B and then 'acts' like B", in [Carpenter 2002], three components have been identified: Goal, Action and Result. Based on what is learnt there are basically three categories: *Mimicking*, *Emulation* and *Imitation*. Mimicking is just reproducing the action without any goal. *Emulation*, [Wood 1998], [Tomasello 1990], is bringing the same result, which might be with different means/actions than the demonstrated one. *Imitation* [Lunsky 1965], [Piaget 1945] is bringing the same result and with same actions. Here it is important to note that depending upon the level of abstraction the imitated action could be the movement, style, trajectory, and other details all the way down to which hand was used and the exact position of the fingers, etc. In one sense, we can say that *Emulation* involves reproducing the changes in the state of the environment that are the results of the demonstrator's behavior, whereas *Imitation* involves reproducing the actions that produced those changes in the environment.

Emulation is regarded as a powerful social learning skill, accounting for a large portion of social learning also among great apes [Tomasello 1990]. In fact, this also facilitates to perform a task in a different way. As studied in [Lempers 1977], children can show an object to someone in different ways: by pointing, by turning the object, by holding it so that other can see it. Similarly, it has been shown that the children are able to hide an object from another person in different ways, [Flavell 1978]: by placing a screen between the person and the object, by placing the object itself behind the screen from the person's perspective. These suggests that from the early developmental stages, a child is able to distinguish the desire effect and desired end state of a task from 'how' to achieve that task.

Motivated from these evidences, we also separate imitation and emulation parts of learning. Therefore, we equip our robots to perceive *effect* of a task/goal separately from the *action* and use it to develop a framework to understand the task's semantics independent from its execution. This facilitates task understanding in a 'meaningful' term as well as provides flexibility of planning alternatively for a task depending upon the situation.

1.1.1.3 Pro-social and cooperative behaviors

Apart from imitating and emulating, children also begin to demonstrate *prosocial* [Svetlova 2010], [Eisenberg 1998] and *cooperative* behaviors [Warneken 2007] from as early as the age of 14 months. Prosocial behaviors are aimed at acting on behalf of another agent's individual goal whereas cooperative behaviors are aimed toward achieving a shared goal. Such behaviors are not only core of complex social-cognitive behavioral coordination skills but also give rise to complex mind reading and communication capabilities, [Tomasello 2005].

Motivated from these core blocks of behaviors, we have developed frameworks, which facilitate the robot to generate shared plans for cooperatively achieving joint tasks, as well as to behave proactively to ease the achievement of the others' individual/joint tasks.

1.1.2 Human Behavioral Psychology Research

1.1.2.1 How do We Plan to Manipulate

On the other hand from our behavioral aspect, for performing pick and place task, we, the human, do posture based motion planning [Rosenbaum 1995], [Rosenbaum 2001]. Before planning a path to reach, we, the human, first find a single target posture. This target posture is found by evaluating and eliminating the candidate postures by prioritized list of requirements called *constraint hierarchy*: a set of prioritized requirements defining the task to be performed. Then a movement is planned from the current to the target posture. The Key motivational aspect is: the planning is not just a tradeoff between costs, but a constraint hierarchy and only the postures, for which the primary constraint is met, are further processed to test the feasibility of additional constraints.

Inspired from this we have also developed a framework, which first finds the final configuration of the robot and the human for performing basic human robot interactive manipulation tasks. And for doing so, the planner hierarchically introduces relevant constraints at different stages of planning. From the convergence of the task planning point of view this approach serves an important purpose of reducing the search space significantly before introducing the next constraint and hence the time for finding a solution.

1.1.2.2 Grasp Placement Interdependency

Further, to find the target-posture, we have to choose the target-grasp. Works such as [Zhang 2008], [Sartori 2011] show that how we take hold of objects depends upon what we plan to do with them. Further it has been shown that initial grasp configuration depends upon the target location from the aspect of task [Ansuini 2006], end state comfort [Rosenbaum 1992], [Zhang 2008], shape of the object [Sartori 2011], relative orientation of the object as well as on the initial and the goal positions [Schubö 2007].

Inspired from these studies, we have developed planning and decision making frameworks for performing human interactive manipulation tasks, by emphasizing interdependency nature of grasp and placement and introduction of hierarchical elimination of candidates based on task requirement, human's perspective, current environmental constraints, and so on.

We, the human, even tend to take hold of an object in an awkward way to permit a more comfortable, or more easily controlled, final position [Zhang 2008]. Therefore, we also allow the robot to autonomously select different grasp, even non-trivial one, by taking into account the effort, comfort, and needs not only of itself but also from the human perspective. A few examples of such needs are: minimize the human's effort to see or reach the object, to ensure the feasibility for the human to grasp the object if required, to ensure that the human can significantly see the object, its front, its top, and so on.

1.1.2.3 How do We Navigate

One the other hand when we move or interact, we prefer to maintain social or interaction distances, [Hall 1966]. Further there are private space of human, interpreted as territorial effect, [Liebowitz 1976], which plays an important role in human navigation pattern. The conflict in people avoidance behavior while walking in opposite direction is well known. It has been observed that there could be multiple failed attempts to break symmetry in such situation before a successful attempt to avoid and pass by. In [Helbing 1991], it has been proved mathematically that having an asymmetric probability of each individual to pass from a side, i.e. bias towards passing from a particular side will reduce the number of conflicting and failed attempts in avoidance behavior. Hence, it suggests a need of following a particular social or cultural norm of passing by, which could be from left side or right side depending upon the country. Further because of this bias, people stick to a particular side while passing through a walkway, forming a sort of virtual lane. This behavior reduces the frequency of situations of avoidance and corresponding delays. Further, in the situation where a person has to avoid another person, he/she does so by minimizing his/her deviation, hence he/she will pass another person along a tangent to the territory of another person.

Inspired from these, for a robot to be acceptable by its navigation strategy, we have equipped the robot to take into account such human-socio factors in its planning and decision making strategies, while avoiding, passing by and moving in human centered environment. This will further avoid conflicting and uncomfortable situations. Further to minimize the deviation as well as to avoid exerting any repulsive force onto the person, the robot plans a smooth deviation in its path and that too by trying to pass the person through a tangent point to the territory of that person. Moreover the robot treats people moving together as 'a group' and adapts its path accordingly.

1.1.2.4 Social Forces of Navigation

In [Helbing 1995], [Helbing 1991] it has been suggested that people motion exerts a kind of social force which in turn influences the other person's motion, decision and behavior. Such social forces are attractive or repulsive, which in turn can be used to push or pull a person. But at the same time, the attractive social force exerted by some other person or object [Helbing 1995] can sometime destruct or deviate a person from a joint task, such as guiding.

Therefore, if the robot has to guide a person, it should not assume that the person would always follow the robot and that too by tracing its path. We have developed a framework, which could take into account natural deviation in the person's behavior/motion and provides the person with the flexibility to be guided in the way he/she wants. Further, in the case the person has deviated significantly, the framework tries to exert an attractive social force by its goal oriented approaching behavior as a re-engagement effort to influence/fetch/push/drag the person towards the goal.

1.2 Socially Intelligent Robot

We define a socially intelligent robot as follows:

"A socially intelligent robot is equipped with the key cognitive capabilities to understand and assess the situation and the environment; the agents and their capabilities; and exhibits behaviors, which are safe, human understandable, human acceptable and socially expected."

Hence, the definition includes all the characteristics of social interfaces, human awareness, socially situated, as discussed in the motivation section. This also provides latitude to incorporate a blend of expected socio-human factors like comfort, intuitiveness and so on.

Next, we will identify the hierarchy of cognitive and behavioral capabilities for an agent to be socially situated and socially intelligent, which we call Social Intelligence Embodiment Pyramid. Followed by that, we will explain the blocks, which are within the scope of this thesis.

1.2.1 Social Intelligence Embodiment Pyramid

As shown in figure 1.1, we have conceived a social intelligence embodiment pyramid by identifying a hierarchy of socio-cognitive abilities and behavioral aspects. This is based on exploring the studies of child development and human behavioral psychology and by analyzing about which ability or behavior serves for realizing which other ability or behavior. That is why, we have identified layers of various building blocks. We have identified and placed key cognitive and behavioral abilities at bottom layers. This includes perspective taking, affordance and effort analyses, basic situation assessment capabilities as key cognitive aspects. And we place basic navigation, manipulation, communication and attention aspects of oneself at key behavioral level. Note that the aspects of emotion, facial expression, could be placed as non-verbal aspects of communication. As already mentioned, such aspects are beyond the scope of the thesis, so we avoid placing them explicitly in the pyramid. Then the basic pro-social aspects have been identified, which require the key capabilities of the lower layers to further make an agent capable to co-exist socially. We attribute these two layers as pro-social because these are contrary to anti-social and further facilitate the existence of oneself in the society. (in fact the term pro-social has been created by social scientists as an antonym for antisocial, [Batson 2003] and *attributes to the aspects that benefit others'* [Eisenberg 2007], [psychwiki Prosocial] and even suggesting to have biological roots [Knickerbocker 2003]).More complex socio-cognitive abilities have been identified and placed above it, each of them again depends upon a combination of the basic blocks of layers below. For example, deciding to help proactively without asking for it, cooperate with someone to compete with someone else, negotiating by assessing situation, and aspects like these, which required abilities to reason by combining multiple blocks of lower layers.

Note that at every level there is a decisional component involved, only the level of abstraction will be different. Further a socially intelligent agent should take into account human factor, task oriented constraints at different layers in the analysis, decision-making and planning processes. And of course, all of these aspects could be learnt and refined lifelong. Hence, we place the *socio-human factors*, *task factors*, *decisional and planning aspects* and *learning* outside the pyramid, which in fact are equally important for a socially intelligent agent.

1.2.2 Scope and Focus of the Thesis

There have been works on social robots, with focus on facial expression [Bruce 2002], emotion [Breazeal 2002], verbal interaction, therapy, etc. See survey [Fong 2003] for related works on such aspects.

The focus of this thesis will be complementary to the above-mentioned aspects of social interface, facial expression, speech synthesis. In this thesis we will explore various human-socio aspects such as what a socially intelligent robot should infer about human, how should it move, how should it manipulate objects for human, how should it cooperate with humans, how should it behave proactively, and what does a task mean. We will develop frameworks to equip the robot with capabilities to take into account such human-socio aspects in its motion, manipulation, cooperation, and proactive behavior as well as to learn tasks at human understandable level.

We will instantiate key blocks of different layers by taking into account human factors, task oriented constraints and develop frameworks to autonomously deciding and planning one or another components of the decision and planning block of figure 1.1. We will push the socially intelligent agent's abilities and behavior up to a level from where more complex behavior could be developed in future. From the perspective of learning, we will focus on one key aspect: understanding of a demonstrated task independent of its execution, which has not been explored enough

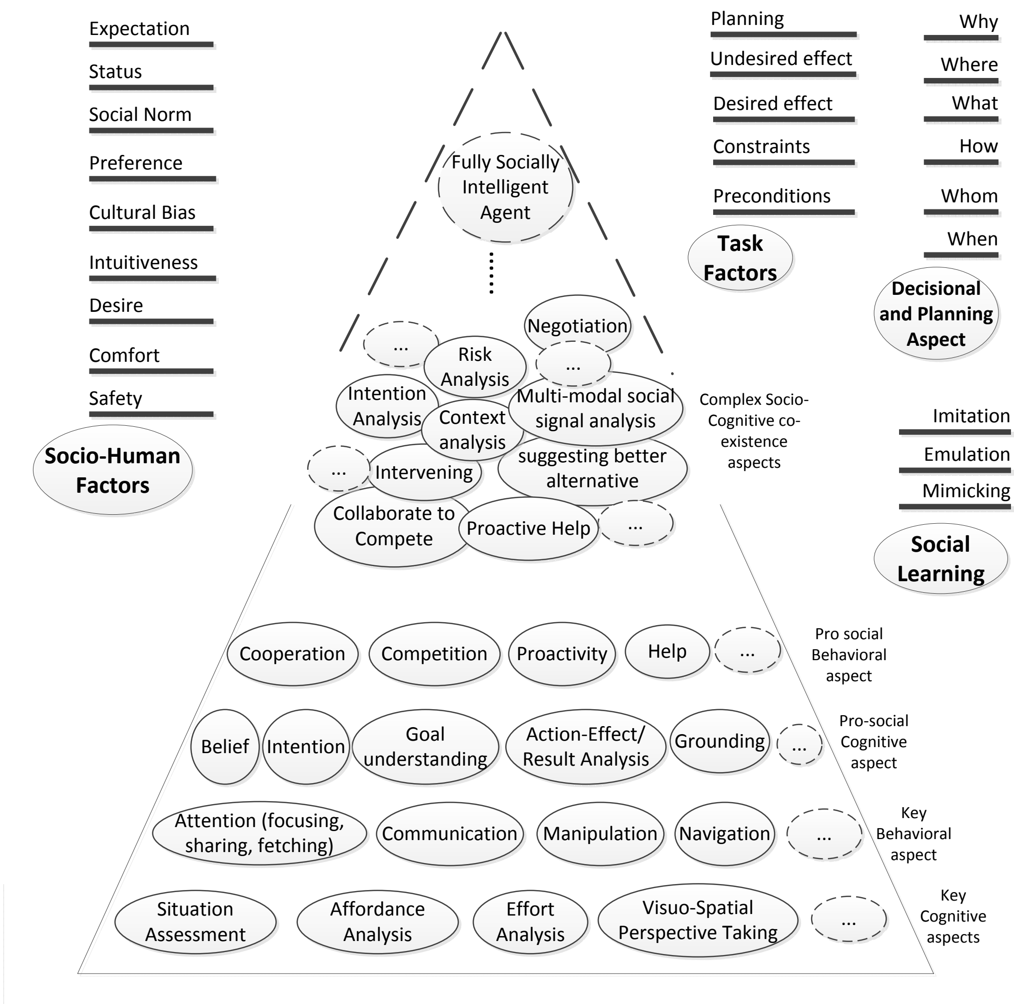


Figure 1.1: The ***Social Intelligence Embodiment Pyramid***, which we have constructed based on the evidence form psychology, child development and human behavioral research, as discussed in this chapter. The basic socio-cognitive abilities at lower layers lead to more complex socio-cognitive behaviors and eventually make an agent fully socially intelligent. Therefore, from Human-Robot Interaction (HRI) perspective, we propose the *bottom-up social embodiment approach*. For this, in this thesis, the pyramid and the different blocks at different layers will serve to develop frameworks and algorithms and introduce concepts from HRI perspective.

in robotics. This will serve another important aspect of a socially intelligent agent to understand the task at appropriate level of abstraction to "meaningfully" interact with human and to plan alternatively, based on situation, to achieve that task.

By equipping the robot with basic cognitive, behavioral and co-existence aspects, we will demonstrate the socio-cognitive behaviors by different robots: *HRP2*, *PR2* and *Jido*, and discuss that these basic abilities are in fact the building blocks for more complex socio-cognitive behaviors.

1.2.3 Approach: Bottom-up Social Embodiment

Inspired from child developmental research and emergence of social behaviors, we adapt the approach to grow the robot as "social" by developing basic key components, instead of taking 'a' complex social behavior and top down realizing the components for that behavior. Our choice of bottom up approach serves the objective of this thesis: *building a foundation for designing more complex socio-cognitive behaviors by exploring and realizing open 'nodes' to diversify and build upon.*

1.3 Outline of the Thesis

Next chapter (**chapter 2**) will present the state of the art, identify research challenges and outline the contribution of the thesis in terms of the blocks of figure 1.1.

Chapter 3 will present the first contribution of the thesis as a unified theory of HRI based on *causal nature of environmental changes*. We will present a generalized domain of HRI in terms of agent's state, abilities, affordances, and various other facts related to HRI. Altogether, they will serve as the attributes of the environment. Then, we will present a generalized notion of action and derive various research challenges of HRI within a unified framework of causality of environmental changes. We will take this as an opportunity to also incorporate the various scientific contributions of different chapters of the thesis within this framework.

Chapter 4 will present another contribution of the thesis, the concept of *Mightability Analysis*, which stands for "*Might be Able to...*". This enables the robot to reason on the agent's visuo-spatial abilities and non-abilities from multiple states the agent might attain, if he/she/it would put different levels of effort.

Chapter 5 will present the contribution of thesis in terms of enriched affordance analysis and rich situation assessments based on geometric reasoning on 3D world model obtained and updated in real-time. We will also introduce the concept of Agent-Agent Affordance and a framework to analyze such affordances.

Both, **chapter 4** and **chapter 5** will instantiate key environmental attributes of visuo-spatial ability, effort and affordances, as presented in generalized theory of HRI in chapter **3**. These in fact correspond to the bottom layer of the social embodiment pyramid, sketched in figure 1.1, which will serve a base for developing other contributions of thesis at higher levels of the pyramid in subsequent chapters.

Chapter 6 will present the contribution of the thesis from the navigational aspect of the robot. It will present framework to plan a socially expected and acceptable path as well as to guide a human in the way he/she wants to be guided. We will also compare the results with a purely reactive navigation behavior.

Chapter 7 will present the contribution of the thesis in terms of bridging the gap between Manipulation and HRI. It will identify the important property of grasp-placement inter-dependency and present a generic framework to plan basic human robot interactive manipulation tasks, such as show, give, hide, make-accessible by taking into account a hierarchy of constraints from the perspective of task, human and the environment.

Chapter 8 will present the contribution by introducing the concept of Affordance Graph, which will enrich the knowledge about various affordances and action possibilities between any pair of an agent and an object as well as between any pair of agents. This also facilitates to incorporate effort in grounding, decision-making and shared cooperative planning, and converts various decisional and planning aspects as graph search problem. Further, this chapter will introduce the link between symbolic level and geometric level planners as well as the concept of geometric task level backtracking to solve for a series of tasks.

Chapter 9 will contribute in presenting a generalized theory of proactivity, to "regulate" the allowed proactivity of an agent as well as to identify potential spaces for synthesizing proactive behaviors. Further, a framework to instantiate proactive behavior will be presented. Some results from preliminary user studies will be presented, advocating that carefully designed proactive behaviors indeed reduce human partner's effort and confusion and our framework is able to achieve that.

Chapter 10 will present the contribution of the thesis as an initiative to understand day-to-day tasks in terms of desired effects and that too at appropriate levels of abstractions. This is an important aspect of emulation learning, which could facilitate the robot to perform the same task in different ways in different situations.

Chapter 11 will conclude the thesis with a summary of the concepts and frameworks introduced in the thesis followed by the potential future work and application.

Chapter 2

Related Works, Research Challenges and the Contribution

Contents

2.1 Introduction 13
2.2 Visuo-Spatial Perspective Taking, Situation Awareness, Effort and Affordances Analyses for Human-Robot Interaction 13
2.3 Social Navigation in Human Environment and Socially Aware Robot Guide 20
2.4 Manipulation in Human Environment 25
2.5 Grounding Interaction and Changes, Generating Shared Cooperative Plans 28
2.6 Proactivity in Human Environment 32
2.7 Learning Task Semantics in Human Environment 34

2.1 Introduction

In this chapter, we will discuss the state of the art in robotics, related to the various blocks of socio-cognitive development as identified and discussed from the psychology, human behavioral and child development perspectives in the introduction chapter (chapter 1). We will discuss the related works, identify the research challenges and the system requirements for efficient human-robot interaction and highlight the contribution of the thesis. We will use figure 1.1 as reference and illustrate the contribution of the thesis in terms of both the research and the system development.

2.2 Visuo-Spatial Perspective Taking, Situation Awareness, Effort and Affordances Analyses for Human-Robot Interaction

Figure 2.1 shows the contribution of the thesis at key cognitive layer. The top right green block shows the contribution in terms of equipping the robot with basic visuo-spatial perspective taking abilities. Representation of reachable and ma-

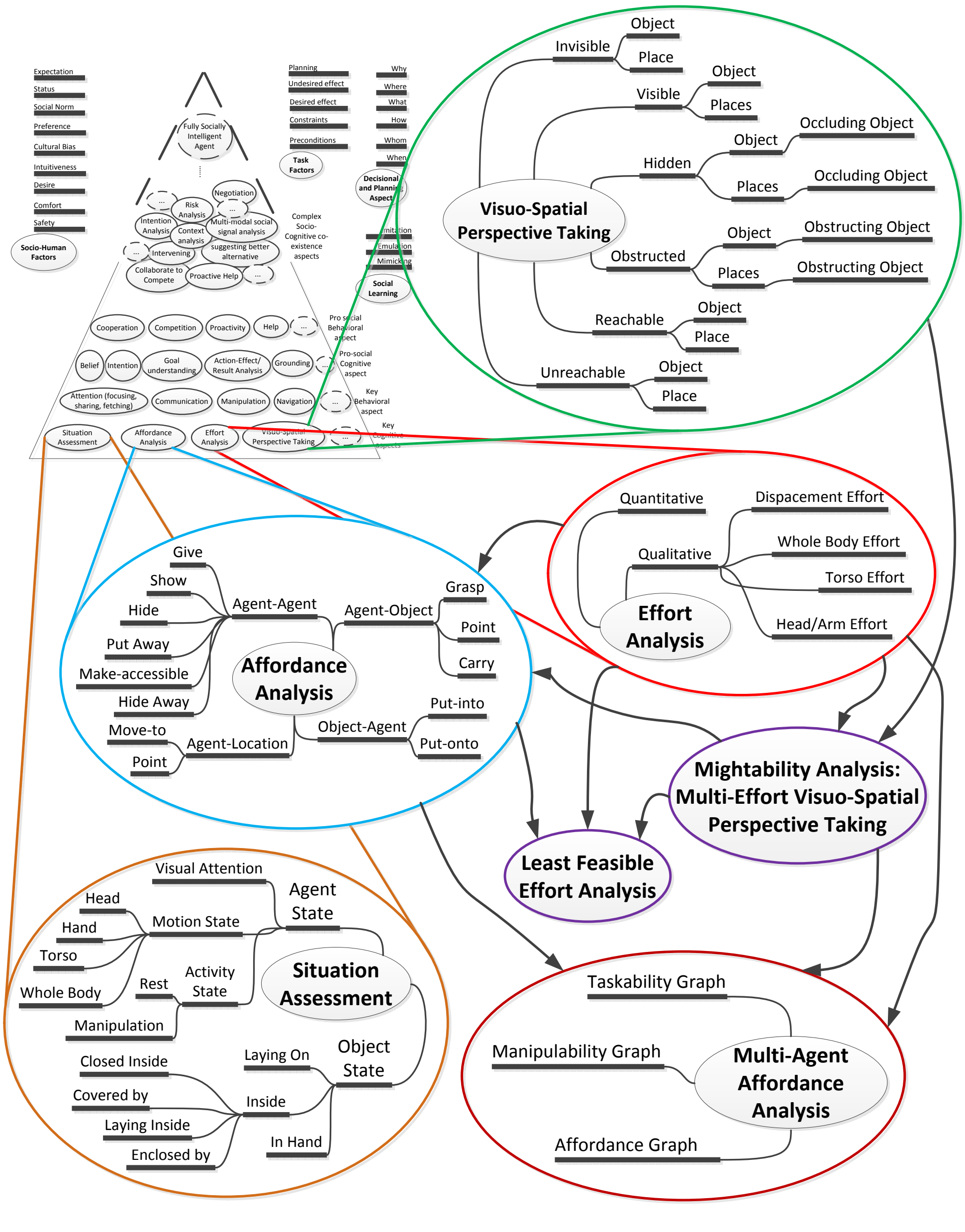


Figure 2.1: Contributions of the thesis in the *Key Cognitive components layer* of the *Social Intelligence Embodiment Pyramid*. An arrow, in this figure and other related figures in this chapter, shows the utilization of one component in developing the other component. For example *Visuo-Spatial Perspective Taking* and *Effort Analysis* contribute to develop the notion of *Mightability Analysis*, i.e. analyzing what an agent might or might not be able to see and reach, if he/she/it will put a particular effort.

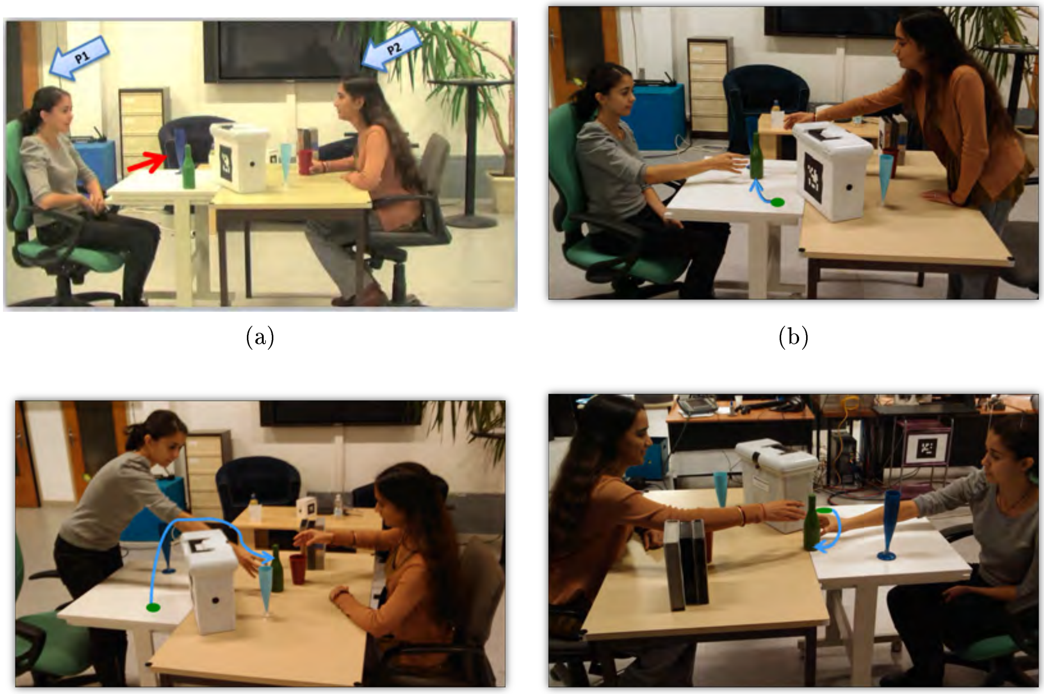
nipulable workspace has already received attention from various researchers. In [Zacharias 2007], the kinematic reachability and directional structure for the robot arm have been generated. Although, it is an offline process, such representation has been shown useful in generation of reachable grasp [Zacharias 2009]. In [Guilamo 2005], an offline technique for mapping workspace to the configuration space for redundant manipulator has been presented based on the manipulability measure. In [Guan 2006], a Monte Carlo based randomized sampling approach has been introduced to represent the reachable workspace for a standing humanoid robot. It stores the true or false information about the reachability of a cell by using the inverse kinematics. However, most of these works focus on *which places* are reachable in the workspace. Moreover, none of these works focus on such analysis with different postural and environmental constraint as well as they don't estimate such abilities of the human partner, which is one of the important aspect for decision making in a Human-Robot Interaction scenario.

Regarding the visual aspect of visuo-spatial reasoning, in the domain of Human-Robot Interaction (HRI), the ability to perceive what other agent is seeing has been embodied on various robots, to learn from ambiguous demonstration [Breazeal 2006], to ground ambiguous references [Trafton 2005a]. Such visual perspective taking has also been used in action recognition [Johnson 2005], for interaction [Trafton 2005b] as well as for shared attention [Marin-Urias 2009b]. However, most of such works answer to the question: *which object is visible?* They do not reason about the visible spaces in the environment, which in fact is a complementary issue.

We have equipped our robots with rich geometric reasoning capabilities to analyze not only which are the reachable and visible objects, but also which are the reachable and visible places, that too in the 3D space and on horizontal support planes. This facilitates the robots to autonomously find places in different situations for performing various tasks for the human: *give*, *show*, *hide*, etc. Further, we have equipped the robots to reason on the *non-abilities* of the agents. The robots can find out, which are not reachable and not visible places from an agent's perspective. We will show that such capabilities facilitate the robots to autonomously find places in different situations for competitive tasks and games: *hide*, *put away*, etc. as well as for grounding interaction and changes. The robots are further able to find the objects, which are obstructing and occluding another object or some place from an agent's perspective. This enriches the robots' knowledge about *why* an agent is deprived from reaching and seeing something and help in reasoning on *how* to 'aid' him/her/it for reaching and seeing that object.

Further, the state of the art on perspective taking focuses on analyzing agent's abilities to see or reach an object or place from the current state of the agent. This is not sufficient for the robots to live in human-centered environment, as will be clear from the following example.

Let us consider a common task in Human-Human Interaction (HHI): make an object accessible to a person, which is currently invisible and/or unreachable for that



(a) (b) (c) (d)

Figure 2.2: (a) Initial scenario for the task of making the green bottle (indicated by red arrow) accessible to person *P2* by person *P1*. *P1* puts the bottle so that it will be visible and graspable by *P2* if she will: (b) stand up, lean forward and stretch out her arm; (c) just stretch out the arm; (d) lean forward and stretch out the arm from the sitting position. In (b) *P1* is trying to reduce self-effort, in (c) she is trying to reduce *P2*'s effort, whereas in (d) she is trying to balance the mutual effort. This suggests the need of reasoning from other's perspective from multiple effort levels, for day-to-day interaction, task planning as well as understanding the task semantics from demonstration.

person. In figure 2.2(a), person *P1* has to make green bottle accessible to person *P2*. Depending upon the current mental/physical state, desire and relation, *P1* could prefer to perform the task by putting the bottle at different places, figures 2.2(b), 2.2(c) and 2.2(d). Here, the interesting point is, for taking the decision about where to place the object for different requirements such as to reduce self-effort (figure 2.2(b)), to reduce other's effort (figure 2.2(c)) or to balance mutual effort (figure 2.2(d)), *P1* is able to infer from *P2*'s perspective, the feasible placement of the object. *P1* is able to reason that if *P2* will stand up, lean forward, and stretch out her arm, she can get the bottle (figure 2.2(b)), whereas in the case of figure 2.2(c), *P2* will be just required to stretch out the arm. In figure 2.2(d), *P1* leans forward and puts the bottle at a place, which requires *P2* to lean and stretch out the arm to take it. This indicates that we, the human, do not only know what an agent would be able to see and reach from his current position, but also what he/she can see and reach if he/she will put different efforts, which plays an important role in our decision making and planning a task for others. The task was same in these three cases, only *where* to perform the task has been changed, based on different mutual effort requirements.

Above example suggests that the robot should be able to perform the *perspective taking* not only from an agent's current state but also from different states the agent *might* attain. For this, first we have developed a qualitative notion of effort hierarchy as shown in the *Effort Analysis* block of figure 2.1. Then, based on this we have introduced the concept of *Mightability Analysis*, which fuses the *effort analysis* with *visuo-spatial perspective taking* to analyze agent's ability to see or reach from multiple states achievable by the agent. *Mightability* stands for *Might be Able to...* and it enriches the robot's knowledgebase with the facts like "*the human1 who is currently sitting might be able to see the object2 if he will stand up and lean forward*". This type of multi-state perspective taking is absolutely important for efficient day-to-day human robot interaction and reasoning on effort, which is currently missing in state of the art robotics systems. **Chapter** 4 will present the contribution of the thesis on visuo-spatial perspective taking, effort analysis, Mightability analysis and least feasible effort ability analysis, as shown in figure 2.1.

Figure 2.1 also shows the contribution of the thesis in terms of elevating and enriching the affordance analysis from HRI perspective. In cognitive psychology, Gibson [Gibson 1986] refers affordance as what an object offers. He defined affordances as all action possibilities, independent of the agent's ability to recognize them. Whereas, in Human Computer Interaction (HCI) domain, Norman [Norman 1988] defines affordance as perceived and actual properties of the things, that determines how the things could be possibly used. He tightly couples affordances with past knowledge and experience. In robotics, affordances have been viewed from different perspectives: agent, observer and environment; hence, the definition depends upon the perspective, [Şahin 2007]. Irrespective of the shifts in the definitions, affordance is another important aspect for a socially situated agent for performing day-to-day cooperative human-robot interactive manipulation tasks. Affordance itself could be learnt [Gibson 2000] as well as could be used to learn action selection [Lopes 2007].

In this thesis, we have proposed a more general notion of affordances, which combines the definitions from diverse disciplines as well as elevates the notion of affordances to other agents, by incorporating inter-agent task performance capabilities in addition to agent-object affordances. Our notion of affordance includes what an agent can do for other agents (give, show, ...); what an agent can do with an object (take, carry, ...); what an agent can afford with respect to places (to move-to, ...); what an object offers (to put-on, to put into, ...) to an agent, as shown in affordance analysis block of figure 2.1. Affordance have been used in robotics for tool use [Stoytchev 2005], for traversability [Ugur 2007] for the robot, but rich geometric reasoning based *what* an agent offers to another agent (give, show, hide, make accessible, ...) and *where*, with *which effort level*; *what* an object offers to an agent (to put something *on*, to put something *inside*, ...) and *where* in a given situation, have not been seen in state of the art robotics systems from human robot interaction point of view. **Chapter 5** will present the contribution of the thesis in terms of this rich affordance analysis.

Further, we have incorporated the effort analysis, *Mightability Analysis* and affordances to equip the robot with rich reasoning of agent's capabilities, as shown in *Multi-Agent Affordance Analysis* block of figure 2.1. We have introduced the concept of *Taskability Graph*, which will encode what each agent could do for all other agents and with which levels of mutual efforts; *Manipulability Graph*, which will encode what each agent could do with all the objects and with which effort level; and fuse them to construct *Affordance Graph*, which will encode different possible ways in which an object could be manipulated among the agents and across the places, along with the corresponding effort levels. This will serve as a basis for addressing a range of HRI problems, such as grounding interaction, grounding the agent, action, effort and object to the environmental changes, generating shared cooperative plan, within a unified framework based on graph search. **Chapter 8** will present this contribution of the thesis. The *Taskability Graph*, which basically encodes the agent-agent affordance is conceptually different and even complementary to the *Interpersonal Map*, presented in [Hafner 2008]. There, the idea was to use affordances to model the relationship between two robots and common representation space to allow robots to compare their behavior to that of others. Whereas, in the Taskability Graph, the idea is to encode different action possibilities between two agents, such as to give, show, hide, etc.

Situation Awareness, the ability to perceive and abstract important information from the environment [Bolstad 2001], is an important capability for the people to perform tasks effectively [Endsley 2000]. From the practical requirements of efficient human-robot interactive manipulation, we have equipped the robot to analyze various states of the agent, his/her/its visual attention and the states of the objects, as show in figure 2.1. The physical states include facts like head turning, hand moving, hand manipulating object, and so on.

Further, to provide the robot with explicit understanding about what will be effect of manipulating a container object *obj2*, on another object *obj1*, which is found to be inside *obj2*, we have categorized different states for *obj1* such as *closed inside*, *covered by*, *laying inside* and *enclosed by*.

All such analyses are done by using a rich 3D model of the environment and the human, which are updated online (see appendix B for the description), and a set of facts are produced in real time for a real human-robot interactive scenario. These serve the purpose of planning, monitoring and executing basic cooperative tasks in a typical human robot interactive scenario for our high-level task planner [Alili 2009] and the robot supervision system [Clodic 2009]. **Chapter 5** will present the contribution of the thesis, which equips the robot with such situation assessment capabilities.

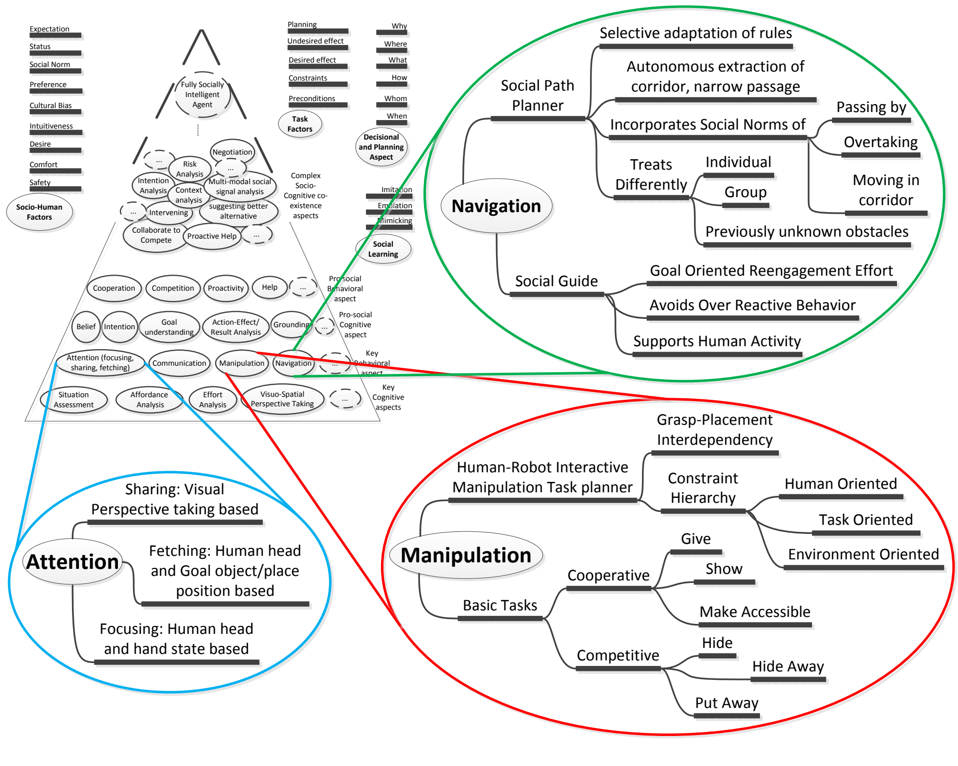


Figure 2.3: Contribution of the thesis in the *Key Behavioral component layer* of Social Intelligence Embodiment Pyramid.

System development contribution in the *attention* component has been shown in figure 2.3. Based on rich geometric reasoning of situation assessment and visuo-spatial perspective taking we have equipped the robot to: share the attention by looking at the object, the other agent is looking at; fetch the attention of the other agent by first looking at him and then looking at the place or object of interest; focus the attention of the robot itself on human activities, if his hand has been detected as manipulating something. Here it is important to note that there are complementary aspects of attention based on saliency, [Ruesch 2008], or by modeling artificial curiosity [Luciw 2011] or intrinsic motivation [Oudeyer 2007], which is beyond the scope of the thesis. **Chapter 5** will briefly show few results of such attentional behaviors, which in fact have been integrated in different interaction scenario presented throughout the thesis and basically serve to our supervision system [Clodic 2009] for activity monitoring and action execution.

As being a social robot, it should take into account a hierarchy of constraints and preferences associated with us, the human, in its navigation and manipulation planning strategies. Next two sections will describe the contribution of the thesis at key behavioral level, as summarized in figure 2.3.

Taking into account the human, in robot's navigation and manipulation strategies, has already been addressed in various ways from different aspects. Works, such as [Sisbot 2008], takes into account the human's comfort and visibility aspects in cost grid for path planning to navigate and manipulate, assuming a static human. In [Kruse 2010], these aspects have been further incorporated in optimistic planning, which returns a solution which might require other agent to move or clear the path, while respecting the visibility and comfort criteria. Whereas [Kirby 2009] incorporates human like walking in hallway in cost grid based framework. In [Marin-Urias 2009a] the human's perspective has been taken into account in the placement planning of the robot. This thesis will be complementary to these works, where we will develop frameworks, which will explicitly reasons on the environment structure, motion of the humans present in the environment, spaces around the humans, social norms of navigation and manipulation at symbolic level along with rich geometric reasoning, and *decides* to behave in a 'particular' way based on the situation. This also makes the robot 'aware' about its own behavior or decision. Below we will discuss in detail the existing navigation and manipulation works in HRI and outline the contribution of the thesis.

2.3 Social Navigation in Human Environment and Socially Aware Robot Guide

As robots will be required to navigate around us for various reasons: following [Gockley 2007], passing [Pacchierotti 2005], accompanying [Hoeller 2007], guiding [Martin 2004] a person or a group of people [Martinez-Garcia 2005], it is apparent that various aspects ranging from safety, reasoning about spaces around human to social norms and expectations should be reflected in the robots' motion.

As shown in figure 2.4, we have identified different aspects of navigation, which a robot should take into account while navigating in the human centered environment.

- **Physically Safe**: Physical safety is one of the most important aspects. The robot should avoid collision with other entities (Agents and Objects) in the environment. Fraichard presents a guideline about the motion safety in terms of collision avoidance, [Fraichard 2007].
- **Perceivable Safe**: Because of the presence of human, the robot should not only avoid physical collision, but also try to make the human feel safe. One way to achieve this type of perceived safety is to signal its intention at appropriate instance of time and space. For example, studies in [Pacchierotti 2005], [Pacchierotti 2006a], indicates that the robot should start avoiding maneuver at a particular signaling distance so that the human will feel safe and comfortable. Similarly, the human should not feel unsafe by evading motion [Shi 2008].
- **Comfortable**: The robot motion should not cause any discomfort to the

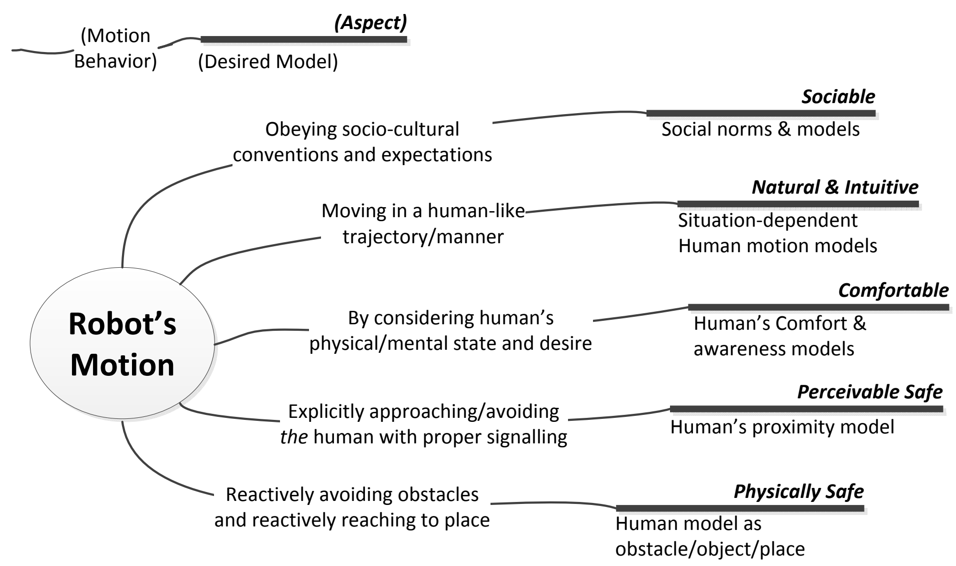


Figure 2.4: We have categorized various factors and qualified the motion aspects, which the robot is expected to take into account while navigating in the human centered environment.

people in the environment. The notion of comfort is wide ranging starting from maintaining a proper distance to considering mental state and awareness of the human. For example, in [Sisbot 2007a], [Kirby 2009], [Lam 2011], [Tranberg Hansen 2009], [Huang 2010], [Svenstrup 2010], comfort has been modeled as maintaining proper distance around human. Towards elevating the notion of comfort beyond the aspect of maintaining a physical distance, [Martinson 2007] takes into account the noise generated by the robot motion itself and presents an approach to generate an acoustic hiding path while moving around a person. Whereas, in [Tipaldi 2011], by avoiding the robot to navigate in the areas causing potential interference with others, while performing the tasks like cleaning the home, the "do not disturb" aspect of comfort has been addressed.

- **Natural & Intuitive**: If the robot would move in a human like pattern, it would be more predictable and the human would find the robot's motion as natural and intuitive. Again, there are various aspects of being natural and intuitive, such as moving in a smooth trajectory, minimize jerk [Arechavaleta 2008], direction following [Kirby 2007] to follow a person in a natural manner, to make the robot move along with the people who are moving in the same direction towards the goal of the robot, as an attempt to exhibit human-like motion behavior in highly populated environments, [Müller 2008].

- **Sociable motion**: We regard *sociable motion* as executing a path, which is

  planned by considering the socio-cultural expectations, influences and favors, the agents (the humans and the robots) can exchange in the social environment.

  A very generic definition of being social could implicitly incorporate the aspects of safety, comfort, and naturalness, but one can be safe and comfortable for someone by maintaining a very large distance from him/her, but perhaps will not be considered social. Therefore, the sociable motion should exploit the fact that the humans are social being, therefore, would have some expectations from others beyond safety and comfort and the same could be expected from him/her as well. Using this idea, some researchers are trying to fulfill such expectations of the human by the robot's motion, whereas others are trying to exploit the expectations from the humans while planning the motion.

  The model for pedestrian behavior by Helbing [Helbing 1991] includes a bias towards a preferred side in the cases of conflict, hence breaking symmetry. In a related way, pedestrians can often be observed to walk in virtual lanes in corridors. Which side to prefer is a cultural preference, a norm that varies between cultures. In [Helbing 1991], [Helbing 1995], it has been suggested that human motion exerts a kind of social force that influences the motions of other people. Hence, the robot can use this model to predict as well as to influence the motion of humans.

  In [Kirby 2009], a cost grid based framework is used to assign higher cost on the right side of the person, hence biasing the robot to pass by from the left.

  Several publications try to exploit the idea that people, as being social agents, adapt to the environment and other agents in a favorable manner, so the robot may use that knowledge about humans to pursue its navigation goals. For example, a person who stands in the way of a robot may very well move aside without discomfort if approached by the robot who wants to pass, [Kruse 2010], [Müller 2008], moving humans may themselves adapt their motion to avoid collision with the robot [Trautman 2010].

In the context of Human-Robot Co-existence with a better harmony, it is necessary that the Human should no longer be on the compromising side. The Robot should 'equally' be responsible for any compromise, whether it is to sacrifice the shortest path to respect social norms or to negotiate the social norms for physical comfort of the person. In [Clodic 2006], we evaluated the long-term performance of our tour guide robot, which suggests that navigating in a human centered environment by considering a person only as a mobile object is neither enough nor accepted. In this context, it is also important that robot should be able to do a higher-level reasoning for planning its path based on the local structure of the environment, clearance around human, intended motion of the human and obviously the social-cultural conventions of the country or the place it is 'working' in. In [Althaus 2004] the robot tries to behave human like by maintaining 'proper' orientation and distance, while approaching and joining a group of people. In [Shi 2008], robot tries to adjust its velocity around the human. In [Sisbot 2007b], the robot takes into account human's visibility and hidden areas, whereas in [Krishna 2006], the robot considers unknown dynamic objects from the hidden zones while planning the path to generate a proactively safer velocity profile. In [Paris 2007], virtual autonomous pedestrians extrapolate their trajectories in order to react to potential collisions.

However, most of these approaches lack in some of the basic socio-cultural aspects such as to pass by or overtake a person from the correct side, proactively maintain itself to a particular side while moving in a narrow passage like corridor, avoid passing through a group of people moving together. All such aspects are necessary for avoiding conflicts and exhibiting socially expected behaviors as discussed in section 1.1.2. Also, the existing approaches either assume that the environment topological structures like corridor, door, hall, etc. are known to the robot or no obvious link between the robot motion behavior with the local environment structure has been shown. Further, not all of these approaches consider the smoothness of the path, which is important for exhibiting natural and predictable motion, as discussed earlier.

Our goal is to develop a mobile robot navigation system which:

(i) Autonomously extracts the relevant information about the global structure and the local clearance of the environment from the path planning point of view.

(ii) Dynamically decides upon the selection of the social conventions and other rules, which needs to be included at the time of planning and execution in different sections of the environment.

(iii) Plans and re-plans a smooth path by respecting social conventions and other constraints.

(iv) Treats an individual, a group of people and a dynamic or previously unknown obstacle differently.

We will present a *via-points* based framework to plan and modifying smooth path of the robot by taking into account static and dynamic parts of the environment, the presence and the motion of an individual or group as well as various social conventions. It also provides the robot with the capability of higher-level reasoning about its motion behavior as exhibited by *Manava*, such as passing and overtaking a person from a correct side. The robot selectively adapts reactive and proactive behaviors depending upon the environment part (wide space, narrow passage, door, ...) as an attempt to avoid conflict as well as to maintain least feasible length of path. This contribution is summarized in navigation block of figure 2.3. First part of **chapter 6** will present the contribution of the thesis in terms of a framework to generate socially acceptable path in human-centered dynamic environment.

On the other hand, if the navigation task is more than just reaching to a goal, other kinds of social aspects become more prominent. Guiding a person to a goal place is one of such scenarios, where the robot has to coordinate motion not just to avoid discomfort, but also to achieve a joint goal. Here, the context of guiding is different from guiding a visually challenged person [Kulyukin 2006] in the sense, the human will not simply follow the robot by some physical means. It also differs from the wheel-chair guiding [Gulati 2008], as robot and human both can take decisions independently.

In [Clodic 2006], we have evaluated the long-term performance of our tour guide robot Rackham. It revealed that in the context of guiding, it is necessary that robot should no longer treat human as a dynamic entity quietly following the robot. The simple stop-and-wait model of the joint task of guiding based on presence and re-appearance of the person to be guided is neither enough nor appreciated. The robot should explicitly consider the presence of human and his/her natural behavior in all its planning and control strategies. In this context, assuming the human to be a social entity, the robot should not expect that the person to be guided would exactly and always trace the path of the robot or always follow the robot. The person could show various natural deviations in his/her path and behavior, perhaps by different social forces imposed by the environment and other agents. The person can slow down, speed up, deviate or even suspend the process of being guided for various reasons. And as being a social robot, the robot should not stop the guiding process, it should try to support the person's activities and re-engage the person if required. This poses challenges for developing a robot's navigation behavior, which is neither over-reactive nor ignorant about the person's activities.

In [Martinez-Garcia 2005], a scenario of multiple robots guiding a group of people is presented. In [Martin 2004], the scenario of guiding a visitor to the desired staff member has been addressed, but from the viewpoint of reliable person tracking. In [Pacchierotti 2006b], an office guide robot has been implemented, but the focus of the motion control module is on people passing maneuver. In [Zulueta 2010], multiple robots guide a group of people, but they focus on the strategy to make a formation that would restrict people to leave the group or to minimize the work done to bring the left people back. Our focus on the complementary issues of supporting the person's activity and to reason on the joint-task and final-goal oriented deviations in the robot's path.

We argue that a social robot should allow and support the natural deviations of the person and avoid showing unnecessary reactive or forcing behavior. Further, in case the human has deviated significantly the robot should exhibit re-engagement efforts by exerting social forces (see section 1.1.2 of the introduction chapter (chapter 1)) by its motion. We have developed an approach for social robot guide, which monitors and adapts to the human's commitment on the joint task of guiding and shows appropriate goal oriented re-engagement efforts, while providing the human with the flexibility to be guided in the way he/she wants, as summarized in *Navigation* block of figure 2.3. To our knowledge, it is the first work in the context of guiding from

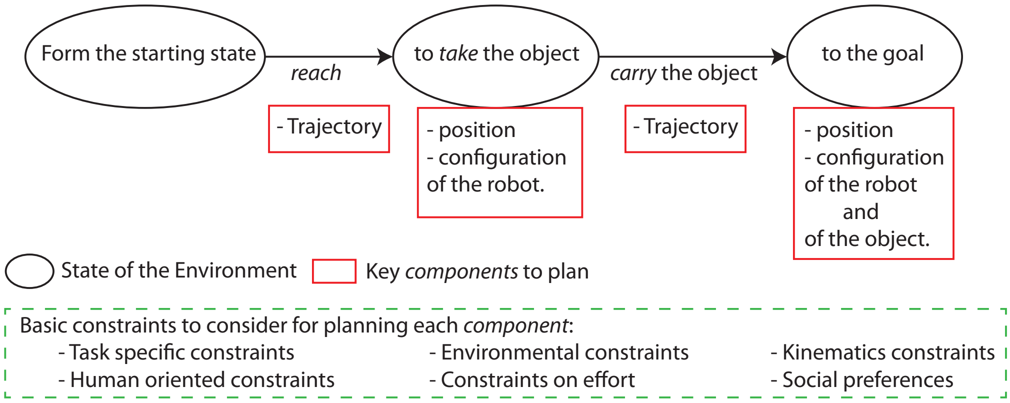


Figure 2.5: Typical planning *components* of an object manipulation task. We have identified various *constraints* from HRI perspective, while planning for each of the components. In chapter 7, we have instantiated it from the perspective of *pick-and-place* type HRI tasks (figure 7.2), exploited inter-dependencies of some of these *components* and presented a framework to incorporate a hierarchy of such *constraints* while planning for a set of basic tasks.

the viewpoint of monitoring and adapting to the human commitment on the joint task as well as verifying and carrying out appropriate goal oriented re-engagement attempts, if required. Second part of **chapter 6** will present this contribution of the thesis of socially aware robot guide.

2.4 Manipulation in Human Environment

In a typical day-to-day HRI, the robot needs to perform various tasks for the human, hence should take into account various human oriented and social aspects. As shown in figure 2.5, we have separated the key components for planning a typical object manipulation task, which involves "*From the starting state, reach to take the object and carry it to the goal*". Here, the goal could be partially provided, or specified in terms of various constraints, as will be clear in chapter 3, where we will present the generalized HRI theory. From the figure we can identify three complementary aspects:

(i) Trajectory Planning (to move and/or to manipulate)

(ii) Placement Planning (position and orientation of the robot and of the object)

(iii) Configuration Planning (of the whole body and of the object)

From the perspective of planning basic human robot interactive object manipulation tasks, different components such as trajectory to reach, trajectory to carry, position and configurations of the robot and the objects are influenced by the presence of human. For example, works such as [Sisbot 2007b], [Sisbot 2010], [Mainprice 2011] take into account human factors such as comfort in *planning the path or trajectory*. Works such as [Marin-Urias 2009a], reason about the human for *planning the placement position* of the robot's base to perform the task for the human. Here, we are essentially interested in the complementary aspect of planning the configuration of the robot and configuration and position of the object for performing basic human-robot interactive object manipulation tasks, such as to give, to show, to hide, etc. In this context, reasoning about the human's abilities, effort, selection of a 'good' grasp and synthesis of a 'good' placement of the object with respect to the human, turn out to be prominent factors to reason about. And various constraints identified in the figure 2.5 influences the choice of grasp and placement. Hence, in this context it is not sufficient that the robot selects grasp and placement of the object from the stability point of view only, as it will be clear from the discussion below.

Figure 2.6 shows two different ways to grasp and hold an object to show it to someone. In both cases, the grasp is valid and the placement in space is visible to the other human, but in figure 2.6(a) the object will be barely recognized by the other person, because the selected grasp to pick the object and the selected orientation to hold the object are not good for this task. We would rather prefer to grasp and hold the object in a way, which makes it significantly visible and also tries to maintain the notions of top and front from other person's perspective, as shown in figure 2.6(b). Similarly for other tasks, such as to give or to make something accessible to the human, there will be a different set of constraints and preferences and will require a different set of information (e.g. grasp possibility, reachability of the other human) for behaving in a socially acceptable and expected way.

In the context of Human-Robot Interaction, study of a human handing-over an object to a robot [Edsinger 2007] shows that the human instinctively controls the object's position and orientation to match the configuration of the robot's hand. Whereas in [Cakmak 2011], a study on a robot handing-over an object to human shows preferences on object's goal position and orientation. A similar study was performed on the Robonaut [Diftler 2004] to grasp the tool handed by a human. Basic human-robot interactive tasks "taking", "giving" or "placing" and incorporating the symbolic constraint of maintaining object upright have been addressed in [Bischoff 1999]. In [Kim 2004], the robot takes into account human's grasp for hand-over task.

However, these works assume that either the grasp or to place position and orientation are fixed or known for a particular task, [Berenson 2008], [Xue 2008]. In addition, either it is assumed that the human grasps the same surface as the robot grasping sites and just shifts the robot grasp site accordingly [Kim 2004] or it learns that there should be enough space for the human to grasp [Song 2010]. These approaches do not synthesize simultaneous grasps by the human and the robot for object of different shapes and sizes. However, works such as [Adorno 2011] begin to represent a cooperative task in terms of relative hand configurations of the human and the robot. However, most of the above-mentioned works still lack the incor-

(a)

(b)

Figure 2.6: The person on the left is showing an object to the other person. Notice the key role of how to grasp and place. In both the cases, the grasp is valid and the placement in the space is visible to the other person, but (a) is *not the good way* to show as the hand occludes object's features from the other person's perspective, whereas (b) is the *better way* to show, as the object's top is maintained upright, features are not occluded and the object is recognizable as a cup to the other person. This suggests the necessity of incorporating various human-oriented symbolic constraints, beyond the stability aspects of grasp and placement, in day-to-day HRI tasks (chapter 7).

poration of some of the key complementary aspects from the human's visuo-spatial perspective about reachability, visibility and on different effort levels, which the human partner can put, while planning for a task. In addition, the set of the tasks considered from HRI perspective are limited: hand-over or to place, [Cakmak 2011], [Bischoff 1999]. Also, the notion that selecting a particular *grasp* restricts potential *placement* and feasibility of the task and vice-versa has not been explicitly considered in the planning frameworks from the HRI tasks perspective.

In this thesis, first we will identify the key constraints for basic human-robot interactive manipulation tasks. Then, we will identify the importance of considering grasp and placement inter-dependency, hence the need of planning for pick and place components together. Then, we will present a generic human robot interactive manipulation tasks planner, which could plan for a set of manipulation tasks by incorporating various constraints and considering the grasp-placement inter-dependency. To our knowledge, it is the first planner to consider this type of rich human-oriented constraints and grasp-placement inter-dependency for planning object manipulation tasks for HRI context. In the framework, the task is modeled as a set of constraints from the perspective of the agents involved. The framework can autonomously decide upon the grasp, the position to place and the placement orientation of the object, depending upon the task, and the human's perspective while ensuring least effort of the human partner. This contribution is summarized in the *Manipulation* block of figure 2.3 and presented in **chapter 7**.

2.5 Grounding Interaction and Changes, Generating Shared Cooperative Plans

One might wonder about the inclusion of interaction and changes grounding and generating shared cooperative plan into a single section. However, we have done it purposefully, because we are essentially interested here in the common aspect of analyzing affordances and effort based planning.

Based on the key cognitive components, the robot is further equipped to analyze the basic pro-social cognitive components as shown in figure 2.7. We have equipped the robot to analyzes the effect of a demonstrated action, in terms of changes in various facts. This contribution, which will be presented in first part of chapter 10, will be compared with state of the art and discussed in more detail in section 2.7 from the point of view of learning task semantics.

The grounding block of figure 2.7 shows the contribution of the thesis in terms of grounding interaction and changes to the objects, with the possible actions and to the agents involved. The problem of symbol grounding, [Harnad 1990], and the subproblem of anchoring, [Coradeschi 2003] are basically establishing the link between the symbols in one's knowledgebase to some input (verbal, sensory-motor) subsymbols, which could be manipulated and/or reasoned about. In [Harnad 1990],

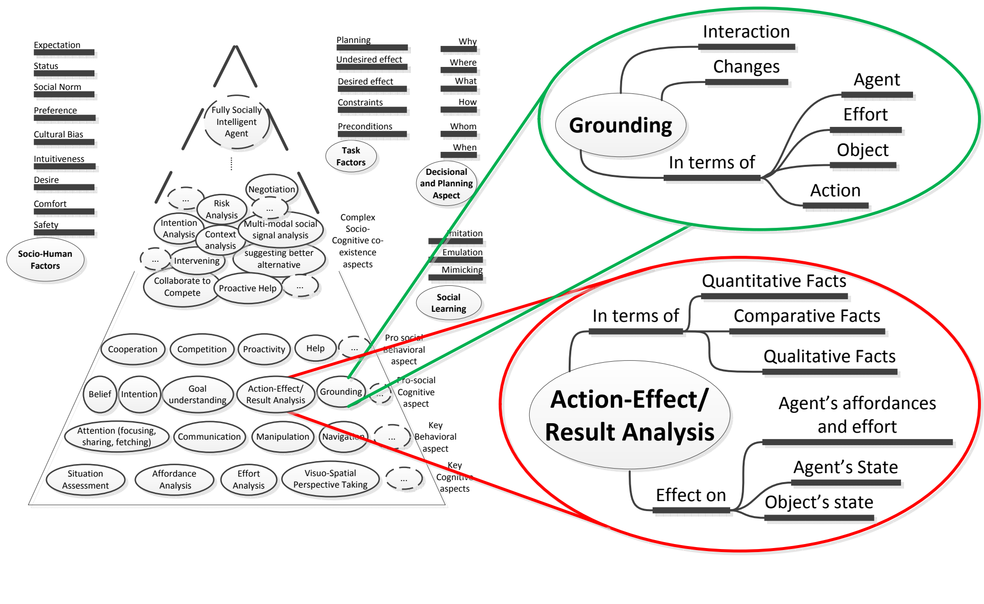


Figure 2.7: Contribution of the thesis in *Pro-social cognitive component layer* of the Social Intelligence Embodiment Pyramid.

discrimination and *identification* have been seen as two important aspects in the grounding process. For example, categorizing the objects as bottles is *identification*, whereas distinguishing between two bottles based on some criteria is *discrimination*. In the context of Human-Robot verbal interaction this discrimination for grounding could be seen as disambiguating the object referred [Trafton 2005a], [Trafton 2005b], [Lemaignan 2011c], [Lemaignan 2011b]. A part of the approach to disambiguate depends upon the perspective taking based mechanism, which was limited in two main aspects: the notion of effort was missing, the interaction scenario was between two agents, one human and a robot. In this thesis we will enrich such grounding capabilities by overcoming those limitations.

In MACS project [Rome 2008] and the related works [Lörken 2008], the notion of using affordances for robot control and for grounding planning operators have been presented in the context of robot interacting with the environment having objects. They present an interesting aspect of using affordances within the planning problem. Because of its domain of interest, the notion of affordance was limited to action possibilities of the robot with respect to the objects, such as the *liftable* affordance of a cylinder, with the planning operator *lift*. In this thesis we are interested in a rich notion of affordance analysis mechanism, which not only reasons about agent-object action possibilities but also agent-agent task performance capabilities.

In addition, very often robot and human have to work cooperatively. Either it is to give something to a third person or to clean the table by putting the objects in the

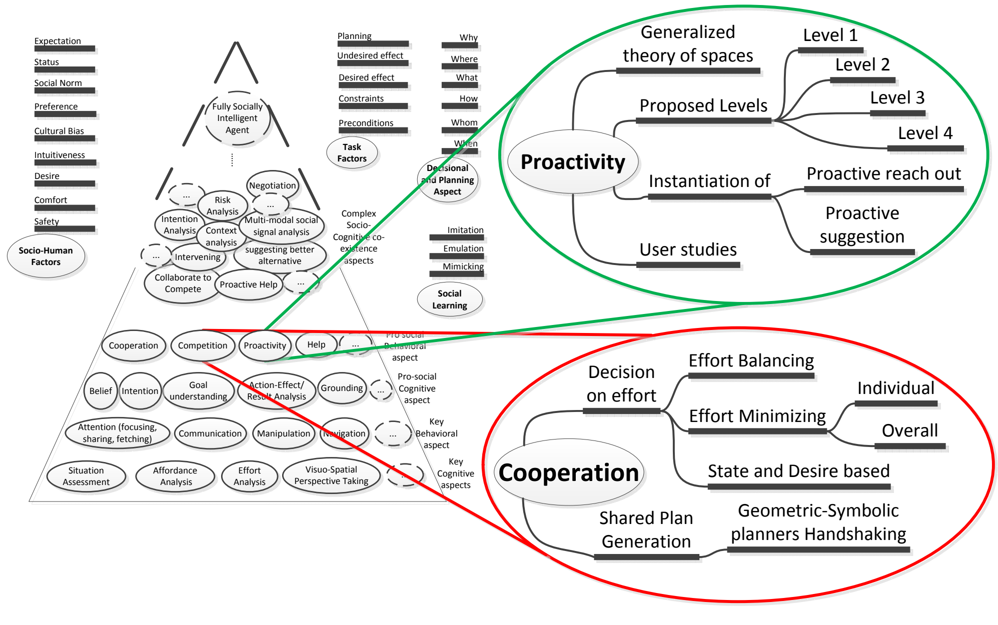


Figure 2.8: Contribution of the thesis in *Pro-social behavioral component layer* of the Social Intelligence Embodiment Pyramid.

trashbin, the robot should be able to generate a set of actions not only by planning for itself but also for all the agents in the environment including the humans.

As long as the robot reasons on the current states of the agents, the complexity as well as the flexibility of cooperative task planning is bounded in the sense, if the agent cannot reach an object from current state, it means that agent cannot manipulate that object, similarly if the agent cannot give an object to another agent it means he/she/it will not do so. But thanks to Mightability Analysis, our robot is equipped with rich reasoning of agents' ability from multiple states/efforts. This introduces another dimension: effort in the grounding and cooperative task planning, as theoretically every agent would be able to perform a task, only the effort to do so will vary.

We are interested in elevating such grounding and shared task planning capabilities by incorporating a rich set of affordances, by incorporating the notion of effort and by enlarging the domain to multi-agent context. By doing so, a subset of grounding problems becomes the planning problem among different agents with different efforts. For example, assume there are three agents (*human1*, *human2* and *robot1*) sitting around a table, and there are bottles placed at different locations on the table. If *human1* asks *robot1*, "*please give me the bottle*," then the problem of grounding 'which bottle' *human1* needs involves various affordances planning, such as who can and cannot see and reach which of the bottles and with what levels of efforts; who can or cannot give which of the bottles, to whom and with what levels of mutual efforts.

We will introduce the concept of *Taskability Graph*, *Manipulability Graph* and fuse them to construct *Affordance Graph*, which will encode different possible ways an object could be manipulated among the agents and across the places, as shown in Mightability based affordance analysis block of figure 2.1. We will show its application for grounding interaction, changes as well as for generating shared plan. Cooperation block of figure 2.8 shows contribution of the generation of shared plan by reasoning about effort of multiple agents. This contribution of the thesis will be presented in the first part of the **chapter 8**. In addition, we will show that the similar mechanism could be used to ground changes in the environment, in terms of agents, efforts, objects and actions, assuming that during the course of those changes the robot was not monitoring the environment, as shown in grounding block of figure 2.7.

On the other hand, to solve a complex task that requires a series of actions by different agents, a close interaction between high-level task planner and the low-level geometric planner is required. It is now well known that while symbolic task planners have been drastically improved to solve more and more complex symbolic problems, the difficulty of successfully applying such planners to robotics problems still remains. Indeed, in such planners, actions such as "navigate" or "grasp" use abstracted applicability situations that might result in finding plans that cannot be refined at the geometrical level. This is due to the gap between the representation they are based on and the physical environment (see the pioneering paper [Lozano-Perez 1987]). Earlier we have proposed in [Cambon 2009] a general framework, called *AsyMov*, for intricate motion, manipulation and task planning problems. This planner was based on the link between a symbolic planner running Metric FF [Hoffmann 2003] with a sophisticated geometric planner that was able to synthesize manipulation planning problems [Alami 1990], [Siméon 2004]. The second contribution of AsyMov was the ability to conduct a coordinated search of the symbolic task planner and its geometric counterpart.

In this thesis, we extend this approach and apply it to the challenging context of human-robot cooperative manipulation. We propose a scheme that is still based on the coalition of a symbolic planner and a geometric planner but which provides a more elaborate interaction between the two planning environments. We have developed a *two-way handshaking* framework, which facilitates such interaction between the planners and allows to take into account different effort based affordances as well as various social, personal, and situation based constraints. The idea is that the two planners should backtrack at their levels and inform each other about feasibility, constraints and alternatives for performing a task or sub-task as summarized in the *task factor* part of figure 2.9. We have elevated the geometric counterpart of such frameworks from the typical trajectory or path planner to a far richer geometric task planner and then we have introduced the notion of *geometric task level backtracking*. This reduces the burden of the symbolic planner to worry about the

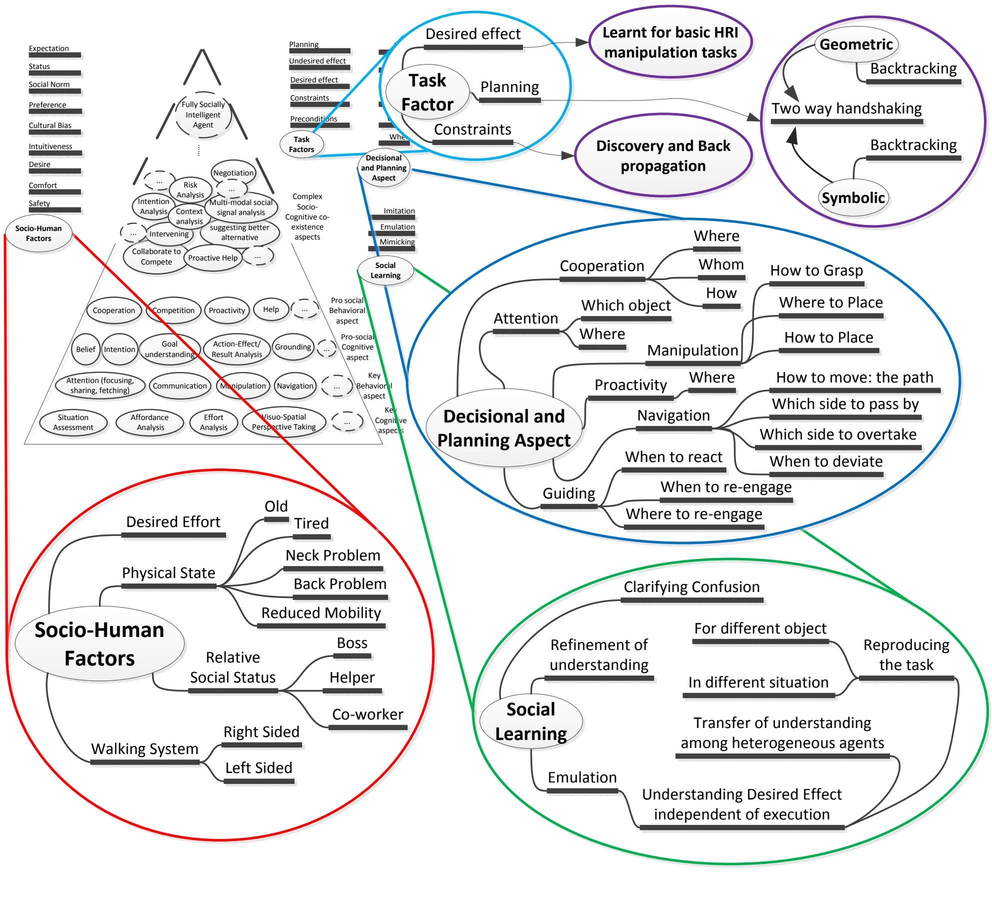


Figure 2.9: Contribution of the thesis in various *Global components* of the Social Intelligence Embodiment Pyramid.

geometric parameters and the constraints of the task as well as avoid flooding the symbolic planner with unnecessary fail reports, which could be handled at geometric level itself by backtracking. This contribution of the thesis will be presented in the second part of the **chapter** 8.

2.6 Proactivity in Human Environment

A social agent is expected to behave proactively. For a robot to be co-operative and socially intelligent, it is not sufficient for it to be active or just reactive. Behaving proactively in a human centered environment is one of the desirable characteristics for social robots [Cramer 2009], [Salichs 2006].

Proactive behavior has been studied in robotics but there is a clear lack of a unified theory to formalize the spaces to synthesize such behaviors. Proactive behavior, i.e. taking the initiative whenever necessary to support the ongoing interaction/task is a mean to engage with the human, to satisfy internal social aims such as drives, emotions, etc., [Dautenhahn 2007]. Proactive behavior could be at various levels of abstractions and could be exhibited in various ways ranging from simple interaction [L'Abbate 2007], to proactive task selection [Schmid 2007], [Kwon 2011], [Schrempf 2005], [Buss 2011]. In [Schmid 2007], [Schrempf 2005], the robot estimates what the human wants and selects a task using probability density function. In [Hoffman 2010], a cost based anticipatory action selection is done by the robot to improve joint task coordination. In [Kwon 2010], temporal Bayesian networks are used for proactive action selection for minimizing wait time. In [Carlson 2008], the robot wheelchair takes control when handicapped human needs it. In [Cesta 2007], activity constraints violation based scheduler is used to remind human. In [Duong 2005], switching hidden semi-Markov model is used to learn house occupant's daily activities and to alert the caregiver in case of abnormality.

But most of these existing works assume 'a' particular kind of proactive behavior and instantiate or validate them. There exists no comprehensive analytical framework to reason about what are the potential spaces in which an intelligent artificial agent could autonomously synthesize proactive behaviors depending upon the specifications of task, context and situation. This is important for life-long adaptivity and evolvability of an autonomous agent, by diminishing behavior feeding on case-by-case basis.

We identify three different aspects of proactivity:

(i) *Autonomous synthesis* of the type of proactive behavior, i.e. how to behave proactively such as speak, suggest, reach out, warn, etc. It is basically synthesizing the operators or actions, which perhaps are not completely grounded.

(ii) The situation based *instantiation* of that type of proactive behavior (what to speak, where to reach out), grounding the actions.

(iii) *On time execution* of that behavior, so that it would be regarded as proactive and does not seem to be reactive.

As shown in *Proactivity* block of figure 2.8, to address the point (i) as mentioned above, we will present generalized theory of proactivity, based on the potential spaces and influence of the proactive behavior on ongoing interaction or on the planned course of actions and categorize different levels of proactivity. This will provide a mean to regulate the "allowed proactivity" of a robot with different levels of autonomy from the perspective of HRI. For the point (ii), we will adapt the framework of our HRI task planner to instantiate various human-robot interactive object manipulation related proactive behaviors. Aspect (iii) is complementary to this thesis and being explored by other contributors in our group. However, we will provide pointers our robot supervisor software, which is responsible to execute and control the robot with such proactive behaviors based on the situation.

In addition, we have conducted a set of user studies to validate a couple of hypothesized proactive behaviors. The results suggest that proactive behaviors are indeed important aspect of being socially situated. This is based on our finding that proactive behaviors reduce the *confusion* of the human partner and if such behaviors are also human-adapted, they further reduce the *effort* of the human partner. Further, for the users, the robots seem to be more *supportive* and *aware* in the cases the robots behaved proactively. **Chapter 9** will present this contribution of the thesis.

2.7 Learning Task Semantics in Human Environment

One of the main challenges in 'natural' and 'cooperative' existence of the robots with us is, the robots should be capable to understand the semantics of day-to-day tasks independent from their executions. Further, such understanding should be at the level of abstraction comprehensible by the human and could be scaled to diverse environment. This will also facilitate the achievement of the same task in different ways depending upon the situation.

Various researchers have addressed many aspects of robot learning through demonstration, see [Argall 2009] for a survey. In [Gribovskaya 2011], trajectories for *pick-and-place* type tasks have been learnt by the robot with constraints on orientations. In [Muhlig 2009], the task of *pouring* by a human performer has been adapted at trajectory level by the robot for maintaining collision free movement. In [Calinon 2009], [Dragan 2011], learning of the trajectory control strategies has been presented from the point of view of adapting to modified scenarios. In [Ye 2011], configuration and landmarks based motion features have been encoded in the learnt trajectory to avoid novel obstacles and to maintain critical aspects of the motion. Such approaches are in fact complementary to learning the symbolic description of the task: what does the task mean and how (at non-trajectory level) to perform the task. This will help to generalize the learnt skill for diverse scenarios as well as to facilitate the transfer of learning among heterogeneous robots. Further, such symbolic level understandings will support natural human-robot interaction.

At symbolic primitives level, the task is mainly learnt in two forms:

(i) *Sub-action based*: The task is learnt based on the sequence of sub-actions.

(ii) *Effect based*: The task is learnt based on the effect in terms of changes in the environment.

In the sub-action learning approaches, the task, *place an object next to another object* would be inferred as *reach*, *grasp* and *transfer_relative*, [Chella 2006]. *Take a bottle out of the fridge* would be sub-symbolized as *Open the fridge*, *Grasp the bottle*, *Get the bottle out*, *Close the fridge* and *Put the bottle on the table in a stable position*, [Dillmann 2004]. In [Pardowitz 2007], incremental learning of the task precedence graph, for the tasks of *pouring the bottle* and *laying the table*, has been presented. In [Kuniyoshi 1994], the robot grounds the task of *assembling a table* by a human in terms of *reach*, *pick*, *place* and *withdraw*, and tries to learn the dependencies to facilitate reordering and adapting for different initial setups. In [Ogawara 2003], a hybrid approach tries to represent the entire task in a symbolic manner but also incorporates trajectory information to perform the task.

However, most of these approaches actually reason on *actions*, i.e. trying to represent a task in sub-tasks/sub-actions from the point of view of execution. There is no explicit reasoning on the semantics of the task independent of the execution. As mentioned earlier in this thesis, our focus will be on task understanding from the effect point of view, i.e. to emulate the task. Recognizing the effect of actions, based on initial and resulting world states, has been discussed as an important component of causal learnability, and a complementary aspect for reasoning action level, i.e. how to generate that effect, [Michael 2011].

As mentioned in section 1.1.1, from the perspective of social learning, which in a loose sense is, *A observes B and then 'acts' like B*, *Emulation*, is regarded as a powerful social learning skill. This is related to understanding the effect or changes of the task, which in fact facilitates to perform a task in a different way. For successful *Emulation* (i.e. bringing the same result, which might be with different means/actions than the demonstrated one), understanding the "*effect*" of the task is an important aspect.

From the aspect of analyzing effects in terms of the task driven changes, the robot tries to learn the effect through dialogue or by observation. In [Cantrell 2011], through dialogue, the task *to follow* a person will be understood as *to remain within 1 meter* of the person. From the perspective of learning interactive object manipulation tasks by observing human demonstrations, in [Ekvall 2008], the effect of *pick-and-place* type tasks have been analyzed by using predicates such as *holding object*, *hand empty*, *object at location*, etc. In [Montesano 2007], the robot performs different actions such as *grasp*, *touch* and *tap* on different objects to analyze the effects; once learnt could be used to select the appropriate action for achieving a particular effect [Lopes 2007]. However, the effects of each action on the object were described in terms of velocity, contact and object-hand distance. In [Tenorth 2009], a first order knowledge representation and processing system KnowRob is presented. It represents the knowledge in action centric way and learns the action models of real world *pick-and-place domain*, coupled with object and its properties. In [Schmidt-Rohr 2010], an approach has been presented to learn abstract level action selection from observation. In this, the *position*, the *orientation*, and the symbolic interpretations of the performer's body movement, such as *bow*, *pick object* are considered.

However, in all these approaches, the effects from the perspective of changes in target-agent's (the agent for whom the task is being performed) abilities have not been exploited, which is one of the basic requirement even for a set of basic yet key tasks in a typical human-human interactive manipulation scenario: give, make accessible, show, hide, put-away, hide-away. One common effect of such tasks is to enable and/or disable the actions or abilities of the *target-agent*. For example, *make accessible* enables the *target-agent* to take the object whenever he/she wants. *Hide* deprives the target-agent from the ability to see the object. Hence, reasoning on the effect of a task from target-agent's perspective is a must for understanding such tasks.

Let us look back to our example scenario of figure 2.2 from the learning point of view. Assume that the robot is observing the task as performed in figure 2.2(c), and learns just by reasoning on the actions, in terms of symbolic sub-tasks such as grasp bottle, carry bottle and put bottle at 'x' distance from the person *P2* or put the bottle reachable by *P2*'s current position. In this case, it will not be able to identify that the tasks performed in situations as shown in figures 2.2(b) and 2.2(d) are the same tasks. This is because of two main reasons: (i) what the robot has learnt actually is how to perform the task, (ii) it did not reason at correct level of abstraction required for such tasks. In this example, the more appropriate understanding of the task should be: *the object should become 'easier' to be seen, reached and grasped by the target-agent*. This is only possible when the robot will also reason on the aspect complementary to reasoning on actions, which is analyzing the effect. Further, the robot should be able to infer the facts at a level of abstractions, which are not directly observable, such as comparative facts: easier, difficult, etc. and use them in learning process.

In [Michael 2011], two desirable capabilities of an autonomous causal learnability have been discussed as: (a) Ability to infer the indirect facts, which could be obtained by ramifications of the action's effects. (b) Build a hypothesis that the agent can use to make predictions of effect-based resultant world state from a novel initial state, which has not been observed before.

The main contribution of the thesis is to deal with the above-mentioned two components in the following manner:

(i) *Hierarchical Knowledge building*: Enriching the robot's knowledge with a set of hierarchy of facts. By reasoning on the multi-state visuo-spatial perspective of the agent, we enable the robot to infer comparative facts such as *easier*, *difficult*, *maintained*, *reduced*, etc. as well as qualitative facts such as *supportive*, *non-supportive*, etc. The robot's knowledge has been further enriched with hierarchy of facts related to the object's state. In our knowledge such facts have neither been generated nor been used in the context where the robot is trying to understand human-human or human-robot interactive object manipulation tasks from demonstrations. The social learning block of figure 2.9, shows this contribution of the thesis, presented in first part of **chapter 10**.

(ii) *Learning Situation and Planning-Independent Task's Semantics*: We present an explanation based learning (EBL) framework to learn effect-based tasks' semantics by building a hypothesis tree. Further, we have incorporated *m-estimate* based reasoning to find consistency based relevant predicates for a task. The framework autonomously learns at the appropriate level of abstractions. We show that such understanding successfully holds for novel scenarios as well as facilitates transfer of task's understanding to heterogeneous robots. Second part of the **chapter 10** presents this contribution of the thesis.

The high-level *socio-human* block of figure 2.9 gives a global idea about the various socio-cognitive factors, a sub-set of which could be incorporated in the various frameworks and algorithms developed in this thesis. Further, the *decisional and planning* block shows various aspects, which the presented frameworks and algorithms enable the robot to autonomously decide.

Next, chapter (**chapter 3**) will first present the contribution of the thesis by providing a generalized domain theory of Human-Robot Interaction. This is a step towards developing a unified framework in which the above-mentioned socio-cognitive components could be incorporated and which could lead towards realizing different behavioral aspects discussed with reference to the Social Intelligence Embodiment Pyramid (figure 1.1) constructed in the introduction chapter. The chapters afterward will present the rest of the contributions of the thesis.

Chapter 3

Generalized Framework for Human Robot Interaction

Contents

3.1 Introduction . . . 39
3.2 Environmental Changes are Causal . . . 40
3.3 HRI Generalized Domain Theory . . . 41
3.3.1 HRI Oriented Environmental Attributes . . . 41
3.3.2 HRI Oriented General Definition of Environmental Changes . 47
3.3.3 HRI Oriented General definition of Action . . . 48
3.4 Development of Unified Framework for deriving HRI Research Challenges . . . 50
3.4.1 Task Planning Problem . . . 50
3.4.2 Constraint Satisfaction Problem . . . 51
3.4.3 Partial Plan . . . 52
3.4.4 Deriving HRI Research challenges . . . 52
3.5 Switching among Different Representations and Encoding: State-Variable Representation . . . 58
3.6 Until Now and The Next . . . 60

3.1 Introduction

Research in Human Robot Interaction (HRI) has begun to guide the direction of future of personal, domestic and service robotics. It is a domain incorporating diverse disciplines, see the survey [Goodrich 2007] for some of such interesting pointers. However, we still lack a general formal description of Human Robot Interaction domain, which could be used to identify the spaces for HRI research as well as could provide a guideline to design and develop various components for HRI. There have been attempts to generalize the Human-Robot Interaction, [Scholtz 2003], but it discussed HRI along different dimensions: roles (supervisor, peer, ...), the physical nature of robots (mobile platform on ground, fixed base, unmanned systems in the air, ...), the number of systems a user may be required to interact with simultaneously, and the environment in which the interactions occur. And a similar taxonomy is presented in [Yanco 2004] by incorporating human-robot physical proximity.

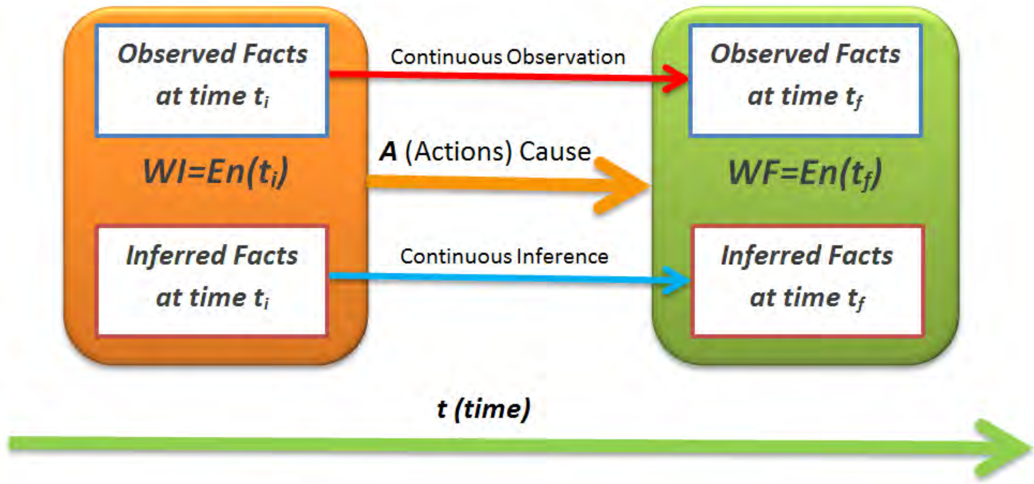


Figure 3.1: <*WI, A, WF*> triplet, showing Causal Nature of Environment Change, a sequence of actions A on initial world *WI* at time t_i results into a final world *WF* at time t_f.

In this chapter, we will present a theory for HRI, along a complementary dimension: *Causality of Changes in the Environment*, so that most of the HRI challenges could be represented in a unified framework of Planning. For this, we will first present a generalized description of *Environmental Attributes*, *Agent* and *Action* from HRI perspective and then we will derive various challenges of HRI in a formal way, which will also link the contributions in the rest of the chapters within this unified framework.

3.2 Environmental Changes are Causal

In the context of HRI, we adapt the typical relations of task, agent, action and environment; see [Ghallab 2004], [Michael 2011], [Kakas 2011], [Novak 2011]. We define, a task T can be achieved by a series of actions A by a set of agents Ag, causing some changes C in the environment En, see figure 3.1. As [Michael 2011], we also postulate that changes could be values of the directly observable facts DF e.g. for the fact variable *objects visible to a human*, and values of the inferred facts IF, e.g. *least feasible effort requires to see an object*. Note that we call them as *fact variables* because they are not ground atoms (in fact when the environment is represented in state variable notation, see [Ghallab 2004], these *fact variables* will be similar to *state variable* with some unground parameters). Further, we ramify that observation/inference could be based on a single time instant, for example, *box is on table*, or based on a course of time, such as *ball is moving*. We define the set F of all such fact variables as:

$$F = DF \cup IF \tag{3.1}$$

Let L be the set of all possible values of all the fact variables F in the environment. Hence, at a particular instance of time t_i, the state s of the environment will be a subset of L, i.e. $s \subseteq L$.

We will adapt the notions of *class*, *type variable* and *constant* from [Ghallab 2004], for our current discussion in HRI context. We partition HRI domain into various classes. The minimal set of classes consists of: Robots, Humans, Objects, Locations and the classes related to their attributes. These classes define the type variables of the domain. Note that type variables could be a class itself such as variable type *Obj* of class *Object*. Similarly, variable types *Rob* of *Robots*, Hum of $Humans$, *Loc* of *Locations* as well as union of classes such as variable type Ag, which stands for *agents*, and consists of classes *Robot* and $Human$, i.e. $Ag \in Robots \cup Humans$. Similarly, we define type variable *Et* which stands for *entity*, such that $Et \in Agent \cup Objects$. Instances of these type variables are the constant symbols, such as $Human1$ as an instance of Ag, which exists in the environment.

We define, the set of all the agents AG and the set of all objects OBJ constitute to the set of entities ET in the environment, i.e.

$$ET = AG \cup OBJ \tag{3.2}$$

Agents are the active entities in the environment, who can act upon another *Agents* and *Objects*, where *Objects* are passive entities in the environment.

Here, we are particularly interested in identifying those attributes of environment, which constitute to the set of environmental facts from HRI aspect. Hence, below we will mainly identify HRI oriented entities and their attributes.

For the rest of the discussion, to get rid of time suffix, we will use *WI* for initial environment and *WF* for final environment as shown in figure 3.1.

3.3 HRI Generalized Domain Theory

In this section we will present a generalized domain theory for HRI, by identifying the *attributes*, and then providing the generalized definitions of *action* and *changes*.

3.3.1 HRI Oriented Environmental Attributes

We define the *state space for agent variable* Ag as follows:

$$\begin{aligned} S_{Ag} = Geometrical\_State_{Ag} \times Physical\_State_{Ag} \times Mental\_State_{Ag} \\ \times Spatial\_Relation_{Ag} \times Proxemics\_Relation_{Ag} \end{aligned} \tag{3.3}$$

Similarly, we define the *state space for object variable Obj* as follows:

$$S_{Obj} = Geometrical\_State_{Obj} \times Physical\_State_{Obj} \times Spatial\_Relation_{Obj} \\ \times Intrinsic\_Affordance_{Obj} \quad (3.4)$$

For a particular instance $ag \in AG$ and a particular instance $ob \in OBJ$, the states will be s_{ag} and s_{ob} receptively, where $s_{ag} \in S_{Ag}$ and $s_{ob} \in S_{Obj}$.

Below we explain each of the above constituting attributes.

Geometrical state of an entity $e \in AG \cup OBJ$ is a tuple:

$$Geometrical\_State_e = \langle position, orientation, configuration \rangle \quad (3.5)$$

Spatial relation is defined as the relative position of an entity $e_i \in AG \cup OBJ$ with respect to any other entity $e_j \in AG \cup OBJ$, where $e_i \neq e_j$. It is a tuple of the form $\langle e_i, e_j, sr \rangle$. Where $sr \in SpRel$ and $SpRel$ is set of all possible spatial relation types defined in the domain:

$$SpRel = \{On, In, Left, Far, Adjacent, ...\} \quad (3.6)$$

Note that there might exist more than one types of spatial relation for a given pair of entities $\langle e_i, e_j \rangle$, for example, an object could be *Adjacent* to an agent and could also be on the *Left* side of the agent. Therefore, there will be set of such tuples representing all the spatial relations between the entity pair, which is denoted as:

$$SR_{e_i}^{e_j} = \{\langle e_i, e_j, sr \rangle\} \quad (3.7)$$

At a given instance of time, for a particular entity $e \in AG \cup OBJ$, there will be a set of all the spatial relations between e and all other entities $e_j \in AG \cup OBJ$ as follows:

$$Spatial\_Relation_e = \bigcup_{\substack{e_j \in AG \cup OBJ \\ e_j \neq e}} SR_e^{e_j} \quad (3.8)$$

Proxemics relation is defined as the proxemics zone in which an agent $ag_i \in AG$ is belonging with respect to any other agent $ag_j \in AG$, where $ag_i \neq ag_j$. It is a tuple of the form $PR_{ag_i}^{ag_j} = \langle ag_i, ag_j, pxr \rangle$. Where $pxr \in PxrSpc$ and $PxrSpc$ is set of all possible proxemics spaces defined in the domain:

$$PxrSpc = \{Intimate, Personal, Social, Public\} \quad (3.9)$$

Note that, there will be only one type of proxemics relation for a given pair of agents' positions.

It is worth mentioning that the $PxrSpc$ contains the spaces defined by [Hall 1966], however the ranges of these zones should be adapted in HRI based on the shape and size of the agents and various other factors.

At a given instance of time, for a particular agent $ag \in AG$, there will be a set of proxemics relations between ag and all other agents $ag_j \in AG$ as follows:

$$Proxemics\_Relation_{ag} = \bigcup_{\substack{ag_j \in AG \\ ag_j \neq ag}} PR_{ag}^{ag_j} \tag{3.10}$$

We define *physical state space* of agent variable Ag as:

$$\begin{aligned} Physical\_State_{Ag} = Attention\_physical_{Ag} \times Posture_{Ag} \times Hand\_state_{Ag} \\ \times Hand\_mode_{Ag} \times Motion\_status_{Ag} \end{aligned} \tag{3.11}$$

where for a particular agent $ag \in AG$,

$$Attention\_physical_{ag} = \langle looking\_at_{ag}, pointing\_at_{ag} \rangle \tag{3.12}$$

$looking\_at_{ag}$ and $pointing\_at_{ag}$ are set of all the entities and locations, ag is looking at and pointing at in the given time instance.

The *posture* of a particular agent $ag \in AG$ is:

$$Posture_{ag} \in \{standing, sitting, ...\}, \tag{3.13}$$

Further, for the agent variable Ag, we define the *hand state space* as:

$$Hand\_state_{Ag} = \prod_{i=1}^{N_h^{Ag}} (hand\_occupancy\_status_{Ag}^{i}) \tag{3.14}$$

where, N_h^{Ag} is number of hands of the Ag type. This representation facilitates to incorporate agents of different types having different number of hands.

For a particular $ag \in AG$, $hand\_state_{ag}$ is set of N_h^{Ag} number of tuples of the form $hand\_occupancy\_status = \langle ht, ov \rangle$, where $ht \in HandType$ and $HandType$ is set of all the possible hand types in the domain. And $ov \in OccVal$, where $OccVal$ is the set of all the possible *occupancy status of the hand.* We define below the minimal required elements of these sets from HRI perspective:

$$HandType = \{Right\_hand, Left\_hand\} \tag{3.15}$$

$$OccVal = \{Free\_Of\_Object\} \cup \{\langle Holding\_Object, \{Object\_Names\} \rangle\} \tag{3.16}$$

For a particular agent ag of class $humans$, a valid hand state $hs_{ag} \in Hand\_state_{ag}$ could be

$(\langle Right\_hand, Free\_Of\_Object \rangle, \langle Left\_hand, \langle Holding\_Object, \{glass\} \rangle \rangle)$.

From HRI perspective, for an agent it is important to distinguish the *mode* of the hand, is it in the mode to do something, such as to point, waiting to take, to give, etc., which we term as *manipulation mode*, or it is in the *rest mode*. Therefore, we define the set of *hand mode* types $HandMode$ as follows:

$$HandMode = \{\langle Rest\_Mode, Rest\_Mode\_type\rangle\} \cup \{Manipulation\_Mode\} \tag{3.17}$$

where $Rest\_Mode\_type$ can be:

$$Rest\_Mode\_type = \{Rest\_by\_Posture\} \cup \{\langle Rest\_on\_Support, Support\_Name\rangle\} \tag{3.18}$$

$Rest\_by\_Posture$ corresponds to the situations when the hand is in rest mode identified as rest postures. $Rest\_on\_Support$ corresponds to the situations when the hand is resting on some support. For example, someone sitting on a chair and the hand is on a table in front or on the armrest of the chair.

Based on the relative posture of the arm with respect to shoulder and torso, the spatial relation of hand with respect to object in contact and with the knowledge about the whole body rest-posture of the agent, such modes can be inferred by geometric reasoning. We will present the results of such reasoning at geometric level in the next two chapters.

We define for the agent variable Ag, the *hand mode space* as:

$$Hand\_mode_{Ag} = \prod_{i=1}^{N_h^{Ag}} (hand\_pos\_mode^i_{Ag}) \tag{3.19}$$

For a particular $ag \in Ag$, $Hand\_mode_{Ag=ag}$ is the set of N_h^{Ag} number of tuples of the form $hand\_post\_mode = \langle ht, hm\rangle$. $ht \in HandType$ as defined earlier and $hm \in HandMode$ defined above.

For the agent variable Ag, we define the motion status space as:

$$Motion\_status_{Ag} = \prod_{bp \in BodyPart_{Ag}} BdPtMotSt^{bp} \tag{3.20}$$

$BdPtMotSt^{bp}$ is a set of tuples of the form $\langle bp, mst\rangle$, where $bp \in BodyPart_{Ag}$ and $mst \in MotSt$. $BodyPart_{Ag}$ is the set of symbols to represent different body parts of the agent class to which Ag belongs. For HRI domain, we define the following minimal set of *body parts*:

$$BodyPart_{Ag} = \{whole\_body, torso, head\} \cup \left\{ \bigcup_{i=1}^{N_h^{Ag}} hand \right\} \tag{3.21}$$

$MotSt$ is the set of possible symbols in which the *motion status* could be qualified. For HRI domain, we define the following minimal set as:

$$MotSt = \{not\_moving, moving, turning\} \tag{3.22}$$

For a particular instance of $ag \in AG$, the *physical state* will be $ps_{ag} \in Physical\_State_{Ag}$. An example physical state ps_{ag} could be:

$$\Bigg\{ \Bigg(\overbrace{\langle \underbrace{\{box, red\_bottle\}}_{\text{looking\_at}}, \underbrace{\{red\_bottle\}}_{\text{pointing\_at}} \rangle}^{\text{Attention\_Physical}} \Bigg), \Bigg(\overbrace{standing}^{\text{Posture}} \Bigg),$$
$$\Bigg(\overbrace{\langle Right\_hand, \langle Holding\_Object, \{blue\_bottle\}\rangle\rangle, \langle Left\_hand, Free\_Of\_Object\rangle}^{\text{Hand\_state}} \Bigg),$$
$$\Bigg(\overbrace{\langle Right\_hand, Manipulation\_Mode\rangle, \langle Left\_hand, \langle Rest\_on\_Support, Table1\rangle\rangle}^{\text{Hand\_mode}} \Bigg),$$
$$\Bigg(\overbrace{\{\langle whole\_body, not\_moving\rangle, \langle torso, not\_moving\rangle, \langle head, turning\rangle,}^{\text{Motion\_status}}$$
$$\underbrace{\langle Right\_hand, moving\rangle, \langle Left\_hand, not\_moving\rangle\}}_{\text{Motion\_Status}} \Bigg) \Bigg\} \tag{3.23}$$

Physical state space of object variable Obj is:

$$Physical\_State_{Obj} = \{MotSt\} \tag{3.24}$$

where $MotSt$ is defined in eq. 3.22.

Mental state of a particular agent $ag \in Ag$ consists of tuple:

$$Mental\_state_{ag} = \langle Belief_{ag}, Emotional\_state_{ag}, Attention\_mental_{ag}\rangle \tag{3.25}$$

Belief could include agent's awareness about the situation, the task, etc. Works such as [Gspandl 2011], [Hoogendoorn 2011] could be used to provide the robot with the belief management capabilities of the agents in the environment.

Emotional state of a particular agent $ag \in Ag$ could be:

$$Emotional\_state_{ag} \subseteq \{Happy, Angry, Sad, ...\} \tag{3.26}$$

Intrinsic_ Affordance of object are the functionality it could provide or support:

$$\begin{aligned} Intrinsic\_Affordance = \{&to\_put\_on, to\_grasp, to\_put\_into, to\_carry, \\ &to\_push, to\_lift, ...\} \end{aligned} \tag{3.27}$$

Note that this notion of affordance is similar to [Gibson 1986], in the sense, it defines affordances as action possibilities, independent of the agents. However, from the HRI perspective, in this thesis we will enrich the notion of affordance (**chapter 5**) with agent-object and agent-agent action possibilities. That is why to avoid any confusion, we use the term *Intrinsic_Affordance*.

Ability oriented facts requires the capability to analyze self-ability and abilities of others, which is a key for any autonomous and cooperative agent. Inferring and grounding a variety of environmental changes expressed in terms of the agents abilities, e.g. "a change in environment state, which could result into the loss of an agent's ability to reach some object," would be possible in the unified framework if we appropriately incorporate *ability* as attribute to infer the facts such as "loss of reach-ability". Therefore, we assimilate the basic abilities of an agent into the attributes of the environment, as will be explained next.

We define AB_{Ag}, the set of basic *abilities* for agent variable Ag as a set Ab_{Ag}, where, each Ab_{Ag} is a tuple:

$$Ab_{Ag} = \langle T_{ab}, P_{ab}, EC_{ab} \rangle \tag{3.28}$$

where $T_{ab} \in TypeAb$ is the type of the ability:

$$TypeAb = \{speak, see, reach, grasp, ...\} \tag{3.29}$$

P_{ab} is the parameters of the ability type. Depending upon T_{ab}, P_{ab} can be NULL, ordered list of entities, words (sentence), etc.

EC_{ab} is the *enabling condition*, which if will be met, the feasibility of T_{ab} will hold for the particular agent in a given state of the environment. This enabling condition depends upon the given instance of environment, and hence differs from the typical notion of pre-conditions of an action. In this context, it is important to equip the robot with the capabilities of analyzing agents' abilities, not only from the current state of the agents but also from a set of different states attainable by the agents. This enabling condition is an ordered list of ec_i, where ec could be an action (definition of which, from the HRI perspective, we will adapt in the next section), an effort (defined in chapter 4), an instance of agent's state defined in eq. 3.3, an instance of the environment state itself, etc. This notion of enabling condition facilitates the reasoning beyond the current state of an agent, which is desirable from HRI perspective. For example, it is not sufficient to know that an agent could not reach an object from his/her/its current state. The robot should be also able to figure out the agent's state and/or actions in which the agent might reach the object.

This facilitates the robot to estimate that the $human1 \in AG$ will be able to reach the cup (currently unreachable), if he will achieve a state by *standing_up* and then *leaning_forward* from his current state. In this case, the enabling condition will be $\langle stand\_up, lean\_forward \rangle$ and an instance of the human's ability will be:

$$(reach, cup, \{\langle stand\_up, lean\_forward \rangle\}) \in \ ability_{Human1} \tag{3.30}$$

Theoretically, finding these enabling conditions, based on the environment, could be viewed as a planning problem in a sub-domain, as we have a given state, and we want to know the resulting state, in which the effects of the ability is satisfied. Hence, it depends upon the domain and the requirements of the HRI context, to decide about the different types of abilities to be pre-computed as the facts of the environment.

As defined in the beginning of the chapter, F the set of all fact variables. For the HRI domain, these fact variables could be the attributes of the entities, and abilities as defined above or could be a derived fact such as "*places* where agent $ag1 \in AG$ could give object *ob* to agent $ag2 \in AG$, *places* which an agent can reach with a particular *effort* eft, and so on. Hence, the set of all the fact variables F, mentioned in section 3.2, which defines the attributes of the environment is actually a superset of all the attributes defined above. One way to represent such facts is to use parameterized state variable, as will be outlined in section 3.5. In the next section, based on F, we will define what does a change in the environment mean.

3.3.2 HRI Oriented General Definition of Environmental Changes

The state space of an environment En is defined as:

$$S_{En} = \prod_{f \in F} V_f \tag{3.31}$$

where V_f is the set of all possible values the fact variable f could take. As we defined in the beginning of the chapter, L as the set of all possible values of all the facts in the environment, we can say that:

$$L = \bigcup_{f \in F} V_f \tag{3.32}$$

If a fact variable f has been assigned a single value at any instance, it is said to be *grounded*, otherwise f is said to be *ungrounded*. At any instance t, the state of the environment, denoted as $s_i \in S_{En}$ will be the grounded values of all the facts:

$$s_i = \bigcup_{f \in F} v_f \tag{3.33}$$

where, $v_f \in V_f$ is the value of the fact variable f at that instance. We say there is a change in two instances of environment, s_i and s_j, if the value of at least one fact variable $f \in F$ is different in both of the instances:

$$change(s_i, s_j) \longrightarrow \exists f | v_f^i \in s_i \wedge v_f^j \in s_j \wedge v_f^i \neq v_f^j \tag{3.34}$$

Let us denote two instances of the environment as the initial and the final states s_{init} and s_{fin}. Change in the environment, denoted as $C_{s_{init}}^{s_{fin}}$ is a set of tuples:

$$C_{s_{init}}^{s_{fin}} = \{\langle f, v_f^{init}, v_f^{fin} \rangle | f \in F \wedge v_f^{init} \in V_f \wedge v_f^{fin} \in V_f\} \tag{3.35}$$

where f, is the fact variable, v_f^{init} and v_f^{fin} are the values of the fact variable in initial and final states.

This notion of environmental changes together with our domain of HRI facilitates to incorporate making changes in the agent's mental state within the unified framework of planning, as will be clear from our discussion about *action* in the next section.

3.3.3 HRI Oriented General definition of Action

As mentioned earlier, we will use typical notion of intention behind an action: an action a is an act, which cause changes in the environment.

$$\begin{aligned} a = action \rightarrow \exists En_{init}, \exists En_{fin} | \left(apply\left(a, En_{init}\right) \ results\_into \, En_{fin}\right) \wedge \\ \left(C_{En_{init}}^{En_{fin}} = NOT\_NULL\right) \end{aligned} \tag{3.36}$$

The dictionary definition of 'action' incorporates expressing by means of attitude, voice and gesture, [merriam webster.com a]. Further, it is important for a human-robot interactive system to be multi-modal. Hence, to facilitate the reasoning on generalized multi-modal space for proactive actions, we adapt a broader delineation of action, which includes verbal and non-verbal acts of the agent:

$$type\_action\left(a\right) \subseteq \{verbal, gaze, gesture, motion, manipulation, ...\} \tag{3.37}$$

For the changes caused by non-agent, terms such as tendency (for falling due to gravity, etc.) [Rieger 1976], event (corresponds to internal dynamics of the system) [Ghallab 2004] have been used. We assume that such events or tendencies could in fact be triggered by an action of the agents. For example, an agent's action might trigger an intentional (to drop something into the trashbin) or accidental free fall (unknowingly hitting something placed on the table's edge) of an object.

We define an action as a tuple:

$$a = \langle name, parameters, preconditions, effect\rangle \tag{3.38}$$

For most of the discussion, we will omit some of elements of the tuple and represent an action as a or $a(parameters)$.

An action can cause changes in any of the environmental facts, which includes attribute's values of an agent, such as agent's mental state. Hence, saying, *"How are you?"* also falls into our definition of an action if its intention is to change the fact related to the emotional state of the agent from *sad* to *happy*, hence, $\langle Emotional\_state, Sad, Happy\rangle \in C_{s_{init}}^{s_{fin}}$.

Saying *"hey..."* is also an action if its intention is to fetch visual or mental attention i.e. changing facts related to the attentional part of the agent's state. Verbal action could also change the belief about what, when, how, where, etc. about the situation, task, etc. Actions could be to confuse or to clarify 'something' depending upon the

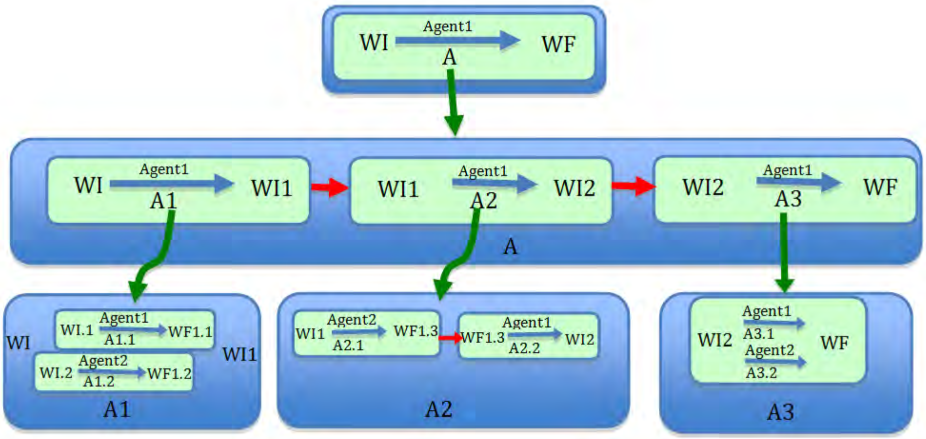


Figure 3.2: An action can be further decomposed into sub-actions and there could be different kinds of dependence relations among them. Note that A is an action and Ai, where suffix $i \in \{1, 2, 1.1, 2.1..\}$, indicates sub-action.

need of the game or task: co-operation or competition. An action could cause change in the agent's self-mental and physical states e.g. looking around to update own knowledge about the environment. Our representation of action contains its name/type, the performing agent, and the parameters of the action, but unless necessary, we will avoid their explicit mention.

Similar to [Novak 2011], we also allow an action to be recursively subdivided into (sub)actions as long as the basic characteristics of an action: causing change in the environment is respected. This facilitates to reason at different levels of abstraction and to plan using hierarchy of abstraction spaces [Sacerdoti 1974], [Alili 2009]. Hence, at different levels of abstraction, an action could be of single agent such as grasp, put, etc., or could be combined act of multiple agents, such as hand-over, carry together a heavy object or push a car together. Depending upon the level of decomposition, an action can be co-operative action by multiple agents, e.g. *clean_table* or it can be a micro action e.g. *move_joint*. Therefore, the symbolic level task, *clean the room*, could also be treated as an action at appropriate level of abstraction, because it satisfies the definition of an action: intended to cause changes in the world state.

An action can be assigned to an agent or a group of agents. Even if an action has been assigned to an agent, when decomposed into sub-actions by the planner or by the agent, it can involve actions of other agents also, see figure 3.2. For example, if the robot has to perform the action "clean the room", at the highest level the agent for this action is robot, but while decomposing it into sub-actions, it can ask human partner to clean one of the tables in the room (Type1: independent sub-actions) or ask human to open cabinet so that it can clean it (Type2: dependent sub-actions) or ask human to hold and carry together a heavy object to place it properly in the room (Type3: tightly coupled concurrent sub-actions), see figure 3.3. In figure 3.2,

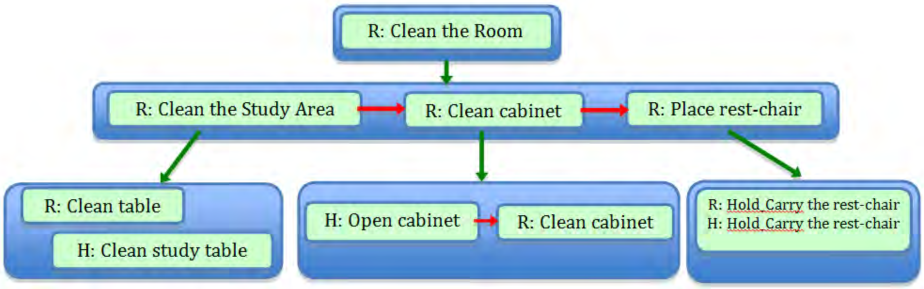


Figure 3.3: An instantiation of action decomposition.

action A itself is Type 1 at highest level of abstraction. Whereas at the next level of decomposition $A1$ is again Type1 but $A2$ and $A3$ are Type 2 as they depend upon $A1$ and $A2$ respectively. Similarly, in next level of decomposition $A1.1$ and $A1.2$ are Type 1 as could be executed independent of each other. But $A3.1$ and $A3.2$ are Type3 as both will be required to be performed simultaneously.

3.4 Development of Unified Framework for deriving HRI Research Challenges

In this section we will derive various research aspects of HRI addressed in this thesis. Above mentioned domain of HRI and the notion of environment and action, facilitate to address a wide range of HRI issues, which are linked to the changes in the environment. Under the assumption that environmental changes are causal, we will be able to bring together various HRI aspects, under the unified framework of planning problem.

3.4.1 Task Planning Problem

To represent the causality of environmental changes, we use the typical general model of the planning domain $\Sigma = (\mathcal{S}, \mathcal{A}, \mathcal{E}, \gamma)$, which is independent of any particular goal or initial state. Where $\mathcal{S}$ is set of states, $\mathcal{A}$ is set of actions, $\mathcal{E}$ is set of events and γ is state transition function. We define a planning problem as:

$$\mathcal{P} = (\Sigma, s_0, g, F\_in, A\_in, F\_av, A\_av,) \tag{3.39}$$

s_0 is initial state of the environment represented in eq. 3.33, g is set of expressions of the requirements a state must satisfy in order to be a goal state. Here, we are deliberately avoiding to give an expression for g, because it will depend upon the representation of planning domain. If it is set theoretic representation, it will be a subset of all the propositions, if it is state variable representation it will be a set of grounded as well as ungrounded state-variable expressions. However, depending upon g, there could be a set of goal states:

$$S^g_{En} = \{s_i \in S_{En} | s_i \; satisfies \; g\} \tag{3.40}$$

It is important to note that we relax the assumption of restricted goal of classical planning problem by explicitly mentioning other elements in the planning problem tuple. This is because of the fact, that in HRI domain controlling the system requires more complex objectives than just giving a final goal state. For example, the system should go through a set of states and actions, the system should avoid a set of states and actions, a set of facts should always be maintained and so on. Extended goal could be represented in different ways, such as temporal logic, utility function or by utilizing other planning under uncertainty frameworks. The detail about representation of such extended goal is beyond the scope of the current discussion, which depends upon the type of extended goal we want to incorporate. However, to facilitate the discussion with extended goal, we have explicitly incorporated F_av, F_in, A_in and A_av in the planning problem defined above. $F\_in = \{\langle precond, f\_in \rangle\}$ is a set of expressions, which tells about the facts to be maintained during the intermediate states of the plan. $F\_av = \{\langle precond, f\_av \rangle\}$ is a set of expression, which tells about the facts to be avoided during the intermediate states of the plan. Where $precond = \{v^i_f\}$ is set of preconditions in terms of grounded fact, i.e. $precond \subseteq L$. If $precond$ is not NULL then f_in or f_av should be considered to be maintained or avoided, only when the $precond$ is getting satisfied. If $precond$ is NULL, we assume that f_in or f_av should be maintained or avoided always. A_av is the set of actions, which should be avoided to be incorporated in the plan and A_in is the set of actions, which should be incorporated in the plan. We assume that even if the elements of these sets are not directly provided, the system is able to deduce them and populate g, F_in, F_av if they are provided in the form of constraints. Next, we will briefly outline the constraint satisfaction problem.

We assume that given an instance of planning problem, a plan A is produced which is a sequence of actions:

$$A = \langle a_1, a_2, ..., a_k \rangle \tag{3.41}$$

3.4.2 Constraint Satisfaction Problem

Constraint satisfaction problem (CSP) in general is: given a set of variables and their domains, and the set of constraints on the compatible values that the variables may take, the problem is to find a value for each variable within its domain such that these values meet all the constraints, (see [Ghallab 2004]). From HRI perspective, we define a constraint c_j restricts the possible values of a subset of fact variables, $\{f_k\} \subseteq F$. A constraint can be specified explicitly by listing the set of all allowed values or by the complementary set of forbidden values or by using relational symbols. We will basically use this notion of CSP to restrict the solution space for a task, by a set of constraints $Ctrs = \{c_j\}$.

3.4.3 Partial Plan

We adapt the definition of a partial plan from [Ghallab 2004], as a tuple:

$$\pi = \langle A^{\sqsubset}, \prec, B, L^{\rightarrow} \rangle \tag{3.42}$$

where, $A^{\sqsubset} = \{a_1, a_2, ..., a_k\}$ is a set of partially instantiated actions, $\prec$ is a set of ordering constraints on $A^{\sqsubset}$ of the form $(a_i \prec a_j)$, B is the set of binding constraints on the variables of action in $A^{\sqsubset}$, $L^{\rightarrow}$ is the set of causal links of the form $\langle a_i \rightarrow a_j \rangle$.

3.4.4 Deriving HRI Research challenges

Using the above representation of planning problem, and how much and which type of information is provided, below, we will derive various HRI research challenges for a variety of sub-domains: affordance analysis, manipulation and motion task planning, learning, proactive behavior, prediction, grounding interaction and changes, etc. This will also present the various contributions of the thesis into the unified theoretical framework.

3.4.4.1 Perspective Taking, Ability and Affordance Analysis

As discussed earlier, our HRI domain incorporates abilities of an agent as attributes of the environment state. This requires that the robot should be able to perform such analyses for all other agents in the environments, which is termed as perspective taking. Further, our definition of ability (eq. 3.28) allows to incorporate enabling condition for an ability. This could enrich the decision-making, planning and affordance analysis capabilities of the robot. However, it imposes the need of reasoning about the abilities of the agent beyond the current state of the agent. A sub-problem of analyzing such abilities is to find the feasibility of an ability of an agent, from a virtual state attainable by the agent, if he/she/it would put a particular effort. Further, such abilities, inheriting the notion of effort could serve for enriched affordance analysis. For example, the robot would be able to find the feasibility of picking an object with the effort involved and feasibility of giving an object to another agent with the criteria of balancing mutual effort, and so on. In **chapter 4** and **chapter 5**, we will focus on such ability and affordance analysis, which will serve as the basis for other contributions of the thesis.

3.4.4.2 HRI Manipulation Task Planning

Consider an instance of eq. 3.39, for the task to *show* an object *obj* by agent *ag1* to agent *ag2*. If the planning problem is expressed in terms of the constraints on the desired goal state that the object should be visible to the *ag2*, then this provides greater flexibility of synthesizing the plan A. There will be different types of decisions, the planner will be required to take: where to perform the task, i.e. reasoning on the goal state; how to perform the task, i.e. reasoning on A. Depending upon the situation and other constraints, the task planner can result into various plans:

(i) $A = \langle grasp(ag1, obj), carry(ag1, obj), hold(ag1, obj, at(P))\rangle$, i.e. grasping, carrying and holding the object at a place to make it visible to $ag2$.

(ii) The plan could involve to displace another object $obj2$, which is potentially occluding the object obj from the agent $ag2$'s current perspective.

(iii) The plan could even involve third agent $ag3$ by giving the object to him and asking to show the object to the $ag2$.

(iv) Even the plan could involve a verbal action by agent $ag1$ to enhance the knowledge of $ag2$ about obj and a set of actions for the $ag2$ to see the object. For example, $A = \langle say(\text{"}Obj\ is\ behind\ the\ box\text{"}), stand\_up(ag2), lean\_forward(ag2)\rangle$.

However, for each of these plans, the question of deciding a goal state has to be addressed. Now assume that a partial plan (see eq. 3.42) is also provided to the task planner in terms of partially grounded ordered sub-actions, e.g. $\langle grasp(ag1, obj, use\_grasp(GSP)), carry(ag1, obj, to(P)), hold(ag1, obj, at(P))\rangle$. Further, assume that each of these sub-actions could further be decomposed only into $move\_hande$ sub-action. Then this left the planner with the trajectory finding problem in the workspace. In this case, the planner will have less flexibility to plan alternatively, however it will still have flexibility of planning different trajectories. Moreover, if the parameters of these sub-actions, such as the grasp GSP, the place P are not grounded by the planning problem specification, the planner would still have latitude to decide about the final state, by grounding the not-grounded fact variables of the final environmental state, denoted as s_f. While deciding the s_f, the planner could incorporate a set of constraints from the perspective of the task, the agents, the environment, etc. Hence, the constraint satisfaction problem can be solved to get the search space S_R, in which s_f would lie.

In fact, the problem of finding final world state s_f incorporates a reasoning mechanism, which will take into account already partially specified goal state g, the set of constraints $Ctrs$, the set of desired and undesired facts F_{in}, F_{av} and the ungrounded parameters of the set of desired and undesired actions A_{in}, F_{av}. In **chapter 4**, we will present the frameworks to ground the values of one of the important parameter of most HRI tasks, "the places" and then in **chapter 7**, will exploit the aspect of planning by instantiating the final environmental state with a set of constraints for a set of basic HRI tasks, assuming the A is already provided in terms of partial plan of $Pick$ and $Place$ type sub-actions, with some ungrounded parameters.

In general, different types of constraints at the time of planning decide the search space for finding a solution as well as could influence the possibility of different plans for the task. For example, consider the same task of showing the object with constraints that the object should be at the right side of the agent $ag2$ on the plane of the table $tab1$ and change in the $ag2$'s $Geometrical\_State$ is undesirable. Depending upon s_0, the plans (i) discussed earlier, which involves displacing the occluding object may no longer be obtained. Also the plan (iv) would not be found as $ag1$ could not ask human to perform some action. In addition, the flexibility of selecting the places about where to perform the task, which in fact could lead different sub-actions including involving third agent, will be more restricted.

This decision of synthesizing the action, the environment state and selecting the agents and parameters of the action could be performed and refined during planning as well as execution of a task. In fact, there is a fuzzy boundary between the symbolic task planner, which plans a task by deciding the high-level actions A and the geometric task planner, which tries to ground the final environmental state and finds a feasible solution for basic actions. Also the constraints on agent, action, final world states will be accumulating and evolving during the course of planning, execution and interaction. In **chapter 8**, we will try to identify these aspects and try to establish a link between both the planners to better converge towards a plan for a high-level goal.

3.4.4.3 HRI Navigation Task Path Planning

Generally, the robots navigating in human centered environment need to find a path, which satisfies a set of safety, comfort and social constraints. We have already relaxed the notion of restricted goal in the planning problem in eq. 3.39, which facilitates to incorporate various undesired facts during the intermediate states of the plan. Further, we can adapt a form of satisfiability problem, see [Ghallab 2004] to constraint the planning during a particular step.

From the navigation point of view, the goal state could be in terms of the fact on the final position of the robot. A *fluent*, fl_i is defined as a grounded fact that describes state of the environment at a given step i of planning (and during execution also to monitor the need of re-planning). For a path or trajectory planning problem, step depends upon the resolution used to discretize of space or time or spacing between the via-points in the topological map. We can constraint the planner by proving a set of facts to avoid F_{av}, which could also be incorporated into the set fluents that must hold at step i of planning: $\bigwedge fl_i^+ \wedge \bigwedge fl_i^-$. Where, $fl_i^+ \in FL_i^+$ is the set of facts that should hold at step i and $fl_i^- \in FL_i^-$ is the set of facts that should not hold at step i. For example, if the robot should not enter into the personal space of the human on the way and should pass by from the left side of the human throughout the path to the goal, then for each relevant human, h, $\langle robot, h, Left\rangle \in FL_i^+$ and $\langle robot, h, Personal\_Space\rangle \in FL_i^-$. Note the criteria of whether a human is relevant to consider at a particular step in the planning strategy depends upon various factors, such as the distance, prediction of potential future relative positions, the task, the local structure of the environment and so on. In **chapter 6** we will discuss on this aspect. There could be other types of constraints if the current step of planning corresponds to a particular environmental state such as *the robot is in* *corridor*. In this case the constraint could be to maintain a particular side in the corridor. Hence, there could also be a set of preconditions for a particular constraint to be applied.

Similarly, if the task is to guide a person to the goal position, the description of final environment state could be same as earlier. However, a new set of constraints at each state of planning and execution will be emerged to incorporate a set of social behavior. For example, the robot should not go out of the social region of the person to be guided and so on.

In **chapter 6**, we will present various constraints as a set of different groups of rules, the notion of selective adaptions of such rules based on the preconditions. Then we will present algorithms to plan a path based on the initial and desired goal states, while maintaining these sets of rules.

3.4.4.4 Learning from Demonstration

Various aspects of learning from demonstration could also be achieved within the framework of the planning domain and the planning problem described earlier. Depending upon which element of the planning domain Σ, as defined earlier, is observable and/or provided, the robot could learn various parameters for decision-making and planning in Human-Robot interaction. Such learning could involve understanding task semantics in terms of effect, learning trajectory preference based on agent and situation, learning to select actions and agent for a particular task in a particular situation, etc. The accuracy and resolution of the learning will depend upon those of observed parameters of the planning problem.

By comparing the two environmental states $WI = s_i$ and $WF = s_f$, the robot could find the changes in the environment C_{WI}^{WF}, as defined in eq. 3.35. This will facilitate to find the effect of a task in terms of changes on the facts of the environment. This in fact helps in emulation aspect of social learning, by knowing the task semantics in terms of *what* to achieve for the task. Whereas, by observing the course of actions A, the robot could learn *how* to perform the task. Depending upon the abstraction space of the action, the robot could learn the task at the trajectory level or at sub-action level. However, even if only one element from the tuple $\langle WI, A, WF \rangle$ was observable, the robot could learn something. For example, if something has been demonstrated to the robot and only WI was observable, then the robot could learn at-least the preconditions of the task with repeated demonstrations.

Learning space of a task semantics in terms of effect could be at the level of directly observable changes/non-changes in the environmental state as well as at the level of changes/non-changes of the inferred facts, which could be built upon comparing two values of a particular fact. For example, easiest visibility *maintained*, reachability becomes *easier*. In **chapter 10**, we will identify the key facts from learning basic HRI tasks, present a hypothesis space and then an explanation based learning framework to learn task semantics in terms of the desired effect to achieve.

Information about a task T	***What could be learnt for task T***
WI	Preconditions, initial world state
A	Trajectory, sub-actions
WF	Post-conditions, desired world state
WI, A	Trajectory, sub-actions, initial world state based preferences about selection of actions, sub-actions and trajectory.
A, WF	Trajectory, sub-actions, desired world state based preferences about which, how and where for actions, sub-actions and trajectories to perform T.
WI, WF	***Effect based semantics****, desired, undesired and side effects and changes, preferences about where to perform ***T***.
WI, A, WF	Preconditions, effects, initial situation and desired world state based preferences about which, how and where for actions, sub-actions and trajectories to perform ***T***.

Figure 3.4: Observation and Learning components correlation. The aspect of effect based task understanding, marked by * (important for emulation learning) will be one of the contributions of the thesis.

Figure 3.4 shows the possible components, which could be learnt based on what is observable or provided to the robot.

3.4.4.5 Predicting Future States

If s_0 and the plan in terms of the sequence of action A is known, the final environmental state space S^f_{En} could be constructed by $\gamma(s_0, A)$. Depending upon which assumption of the classical planning domain is relaxed, S^f_{En} could be a single state, or a set of states or probabilistic representation of the states.

From HRI perspective, this capability could be achieved by simulating the actions and the triggered events in the given state, which could be related to level 3 of situation awareness [Endsley 2000], which corresponds to ability to project from the current state, events and dynamics to anticipate future events/actions A_{future} and their implications, the S^f_{En}. The accuracy and resolution of predicted S^f_{En} will depend on those of s_0 and A. Such prediction could be also used to behave proactively in HRI as well as for HRI task planning in advance many steps ahead.

This will be illustrated in the **chapter 7** and **chapter 8**, when planning in future is done for various reasons.

3.4.4.6 Synthesizing Past State

As opposed to the problem of prediction, where s_0 and A are used, if the final environment s_f and A are known, S^0_{En} could be synthesized, by removing the effects of A and any event E (observed or provided) from s_f. As A could be composed of sub-actions and different agents, again depending upon how much and at which level of abstraction, the parameters of A is known, S^0_{En} could be a single state or partially grounded state, in the sense some of the facts are not grounded. Even sub-actions of A could be "guessed".

3.4.4.7 Grounding Interaction and Changes

As the presented HRI domain incorporates agent's abilities, affordances coupled with situation assessment, the robot could ground the interaction as well as environmental changes by using the same planning domain, in which, one or the other element is not grounded. For example, if there are two humans and a robot sitting around the table and one human asks the robot to give the cup, the robot could ground "which" cup, based on the cup which is "easily" reachable to the robot than the other agents.

Further, if some object has been displaced by an agent and the robot was oblivious of that, then it can also ground the change by reasoning about the agent and the probable action. This could help the robot to ground *what*, *how*, *who*, *where* like facts about a change, which happened in the absence of the robot's attention. **Chapter 8** will present an affordance graph based framework to demonstrate such abilities of the grounding objects, changes and agents.

3.4.4.8 Synthesizing Proactive Behavior

Dictionary definition of the term *proactive* is: "*Acting in anticipation of future problems, needs and changes.*" [merriam webster.com b]. Hence, any action defined in section 3.3.3 is proactive if it satisfies the additional characteristics mentioned above. Proactive actions by an autonomous intelligent agent could be synthesized in different spaces depending upon "how much" and "which parts" of the currently planned or being executed actions/roles of all the agents and the outcomes will be altered. For synthesizing proactive behavior, we need to incorporate the notion of partial plan, so that the proactive planner can reason on the search space of partial plan to come up with proactive behaviors. For this, we assume that the proactive planner is also provided with a partial plan (see eq. 3.42) of the planning problem. This partial plan could even be provided by the human partner during the course of interaction, such as "I will *give* this bottle to you", or even could be inferred by the robot. Moreover, the robot itself could obtain a partial plan, based on the specification of the planning problem of the task.

Once the partial plan is known, which could also be a NULL plan, the robot could proactively reason about how to completely ground the plan by instantiating or binding the variables of the plan. The robot partially or fully synthesizes a solution for an ongoing interaction and the task and proactively communicates it through different actions, which in fact will be the proactive action A^{pro}. In **Chapter 9** will develop a general framework for representing different spaces for synthesizing different level of proactive behavior. This is based on which elements of the planning problem described in eq. 3.39 and the partial plan if any, are being altered and what were the actual status (grounded/not grounded) of those elements.

3.5 Switching among Different Representations and Encoding: State-Variable Representation

Until this point, we have used set theoretic representations to describe the HRI domain and to derive different research aspects within the framework of a planning problem.

However, depending upon the requirements, the description of the planning problem can vary and the domain could be represented into one or the other form, see [Ghallab 2004] for different representations, *set-theoretic*, *classical* and *state-variable* and their comparison. In particular, state-value representation is especially useful for representing domains in which a state is a set of attributes that ranges over finite domains and whose value changes over time, which in fact is the case for most of the attributes of our HRI domain described earlier. Therefore, next, we will briefly illustrate the feasibility of converting the HRI domain into state-value representation and outline the equivalent planning problem.

For the continuity, we briefly describe the ingredients of state-variable representation (see [Ghallab 2004] for detail):

Constant Symbols: A domain consists of a set of constants. For our HRI domain, it will be names of all the agents, objects, locations, etc. e.g. *Human1*, *PR2_Robot*, *Grey_ Tape*, *Room1*, and so on.

Classes of Constant: Constant symbols could be partitioned into disjoint classes, such as robots, humans, locations, objects, etc.

Item Variables: Typed variable ranging over a class or union of classes of constants, e.g. $Agent \in Robots \cup Humans$. Note that in [Ghallab 2004], it is termed as *Object Variable*, which is qualified here as *Item Variable* to distinguish it from the explicit and widely practiced notion of objects in the environment in the HRI domain. Each item variable v ranges over a set of constants, D_v.

Item Symbol: We will name an instance of item variable as *item symbol*. These in fact are *constant* within the domain, e.g. *Human2*, *Robot1*, *Room5*, *Grey_ Tape*, etc.

Term: A *term* is either an item variable or a constant i.e. item symbol.

State Variable: Functions from the set of states and sets of constants (sets of constants could be null also) into a set of constants. A *k-ary state variable* is an expression of the form:

$$x(tr_1, tr_2, ..., tr_k) \tag{3.43}$$

where x is the state variable symbol and tr_i is a *term* as defined earlier.

A state variable denotes an element of a state-variable function. Further, a state variable is intended to be a characteristic attribute of the state of the environment. Hence, to represent the attribute $Motion\_status$ presented in eq. 3.20, we could define a state variable function $AgMotStatus$ as follows:

$$AgMotStatus : Agent \times Body_{Part} \times \mathcal{S} \rightarrow Motion_{Type} \tag{3.44}$$

where $Motion_{Type}$ and $Body_{Part}$ are item variables, which are ranging over sets of constant item symbols $\{moving, not\_moving, turning\}$ and $\{whole\_body, torso, head, \bigcup_{i=1}^{N_h} hand\}$ respectively. N_h is another constant symbol, which is maximum number of hands an agent can have in the domain. This encodes the possibility of having a robot with more than two hands. $\mathcal{S}$ is the set of all the possible grounded states. Then by instantiating this for each agent and each body part from a particular state $s \in \mathcal{S}$, we can realize the attribute $Motion\_status$.

Similarly, rest of the attributes of the HRI domain presented earlier could be converted into parameterized state-value representation.

A state variable of eq. 3.43 is *grounded* if each tr_i is a constant. A state variable is *ungrounded* if at least one tr_i is item variable, as defined above.

Let, X be a set of all *grounded* state variable, i.e. if $x \in X$ is a k-ary state variable, then at any time instance t_i, the state of the environment s includes a syntactic expression of the form $x(b_1, b_2, ..., b_k) = d_l$, where d_l is the value of the state variable and each b_i being a constant, where $i = 1, 2, ..., n$.

$$En(t_i) = s = \bigcup_{x \in X} \{x(b_1, b_2, ..., b_k) = d_l\} \tag{3.45}$$

Relation Symbols: The rigid relations on the item symbols (constants) which are always the same irrespective of the state of the environment for the given domain, e.g. $inside(Robotics_{Lab}, BuildingH)$

Planning Operator: It is a tuple:

$$o = \langle identification(o), precondition(o), effect(o) \rangle \tag{3.46}$$

where,

$identification(o)$ consists of the name n of the operator and all the item variables relevant to that operator, and expressed as $n(u_1, ..., u_k)$.

$precondition(o)$ consists of (i) set of expressions on state variables and (ii) rigid relations.

$effect(o)$ is a set of assignment of values to state variables.

Note that there are two parts of precondition of an operator. In this representation, if an instance of operator o meets the rigid relations of the operator's preconditions, then it's identification is qualified as an action a. If for an operator, there is no rigid relation in the precondition, then each instance of it will be an action. For example, $give(robot1, human1, grey\_tape)$, is an action provided there was no rigid relation in the precondition. In the extended form of this representation, we assume that parameters of an action could have ungrounded variables. Hence, our HRI oriented definition of action could also be well incorporated for state-variable representation based planning and adapted to encode various HRI problems discussed above.

A planning problem in state value representation is $\mathcal{P} = (\Sigma, s_0, g)$. s_0 is an initial state and the goal g is a set of expressions on the state variables. The goal g may contain ungrounded expressions and could contain a set of goal states. Hence, in its extended form it could incorporate the constraints and the planning problem could be represented as satisfiability and constraint satisfaction problem, [Ghallab 2004].

To focus on the algorithmic aspects, in rest of the chapters we will avoid repeating the theoretical formulations as done above for different problems unless it is really required, such as in **chapter 9** where we derive spaces and theory for synthesizing proactive behaviors. For most of the chapters, we will stick with the notations, which will better help in illustrating the core aspects of the problem and the algorithm.

A "truly intelligent" robot should "wire" most of the interpretative abilities from the presented theory of causality nature of environmental changes grounded from the perspective of HRI. Recent attempts are trying to link agents, actions and goals in dynamic environment [Novak 2011], integrating planning and learning during execution to dynamically enhance and refine them all [Agostini 2011].

3.6 Until Now and The Next

In this chapter, we have identified and presented a rich and general description of HRI domain and action, incorporated various HRI aspects into unified theory of *causality of environmental changes* and derived various HRI research challenges under a unified theoretical framework of planning domain. Next two chapters will present the contribution of the thesis in terms of the novel frameworks, algorithms and concepts to instantiate some of the key attributes of HRI domain presented in this chapter. This will lead us to instantiate the applications of the presented framework interpreted above, in the subsequent chapters.

CHAPTER 4

Mightability Analysis: Multi-State Visuo-Spatial Perspective Taking

Contents

4.1 Introduction **61**

4.2 3D World Representation **63**

4.2.1 Discretization of Workspace 64

4.2.2 Extraction of Support Planes and Places 65

4.3 Visuo-Spatial Perspective Taking **65**

4.3.1 Estimating Ability *To See*: *Visible, Occluded, Invisible* 65

4.3.2 Finding Occluding Objects 67

4.3.3 Estimating Ability *To Reach*: *Reachable, Obstructed, Unreachable* 67

4.3.4 Finding Obstructing Objects 68

4.4 Effort Analysis **69**

4.4.1 Human-Aware Effort Analyses: Qualifying the Efforts 70

4.4.2 Quantitative Effort 72

4.5 Mightability Analysis **72**

4.5.1 Estimation of Mightability 73

4.5.2 Online Updation of Mightabilities 79

4.6 Mightability as Facts in the Environment **80**

4.7 Analysis of Least Feasible Effort for an Ability **83**

4.8 Visuo-Spatial Ability Graph **85**

4.9 Until Now and The Next **85**

4.1 Introduction

Interestingly humans are able to maintain rough estimations of visibility, reachability and other capabilities of not only themselves but of the person, they are interacting with. Moreover, it is not sufficient to know which objects are visible or reachable, but also which are the visible and reachable places. For example if we need to find place in 3D space to show or hide something from others. As discussed in section 1.1.1 of

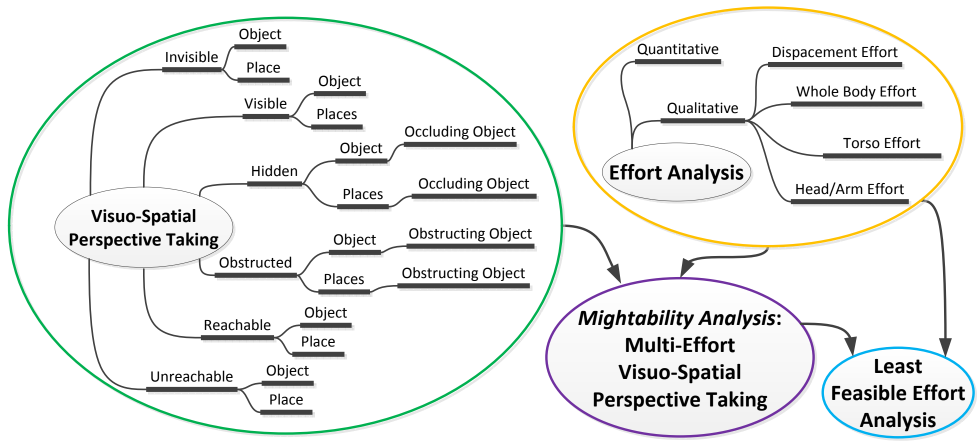


Figure 4.1: Contribution of this chapter. Rich visuo-spatial perspective taking, which not only analyzes what is visible and reachable, but also what is not and why. Effort analysis from a different perspective will also be presented, by developing a set of qualifying effort types and effort-hierarchy. This will facilitate the robot to reasoning on the effort in human understandable way. Further, we will developed the concept of *Mightability Analysis*, derived by fusing visuo-spatial perspective taking and effort analysis, which further facilitates to analyze the least feasible effort.

motivation chapter 1, studies in neuroscience and psychology suggest that from the age of 12-15 months children start to understand the occlusion of others line-of-sight and from the age of 3 years they start to develop the ability, termed as perceived reachability for self and for others. As such capabilities evolve in the children, they start showing cooperative, intuitive and proactive behavior by perceiving various abilities of the human partner. Inspired from such studies, which suggest that visuo-spatial perception plays an important role in Human-Human Interaction, we equip our robot with the capabilities to maintain various types of reachabilities and visibilities information of itself and of the human partner in the shared workspace.

We identify three complementary aspects about the ability to see or reach an object or place x by an agent Ag:

(i) **Direct**: Given the current environment and the state of the agent Ag, x is directly reachable or visible.

(ii) **Within range, could be enabled**: Given the current state of the agent, x could be made reachable or visible to an agent Ag, if there will be some change in states of other agents or objects in the environment. Basically, this corresponds to the situations, in which something is otherwise within the reach range or field of view of Ag, but Ag could not reach or see it because of other agents or objects.

(iii) **Beyond range, inevitable self engagement**: Given the current environment, x could be made visible or reachable only if the state of the agent Ag or the state of x itself will change. This corresponds to the situations, in which x is outside the reach range or field of view of Ag, and manipulating other agents and objects will not be sufficient to make x visible or reachable to Ag.

For the ability to see, these points correspond to:

- *visible* (directly)
- *occluded* (by some object or agent)
- *invisible* (need some action by the agent itself)

For the ability to reach, these points correspond to:

- *reachable* (directly)
- *obstructed* (by some object or agent)
- *unreachable* (need some action by the agent itself).

This chapter will present the contribution to equip the robot with such reach visuo-spatial perspective taking abilities. First, the visuo-spatial perspective taking for a given environment will be presented. Then the robot's ability to analyze the effort of the agents will be presented. Then we will derive the concept of *Mightability Analysis*, which stands for *Might be Able to...*, and elevates the robot's capability of perspective taking from multiple states of the agent. Figure 4.1 shows the contribution and scope of this chapter. It also shows that we equip the robot not only to reason about something is obstructed or occluded, but also the obstructing or occluding object from an agent's perspective.

This enriches the robot's knowledge about the world state, facilitates rich human-robot interaction, as well as elevates the decision-making and planning capabilities about how to facilitate the ability to see or reach an object or a place x for an agent Ag. In the case of *occluded* or *obstructed*, it could be achieved by making changes in the other parts of the environment (such that displacing the obstructing or occluding object or agent), without involving/disturbing the Ag and x. Whereas, in the case of *invisible* and *unreachable*, it would be necessary to change the current state/position of the Ag or x.

Next, we will present the detail about how to achieve such viso-spatial perspective taking abilities and derive the concepts discussed above.

4.2 3D World Representation

The robot uses 3D representation and planning platform Move3D [Simeon 2001] to reason on 3D world. Through various sensors, the agents and objects are updated in this system. Figure 4.2(a) shows a real world scenario of Human and HRP2 robot sitting in a face-to-face interacting situation. Figure 4.2(b) shows its real

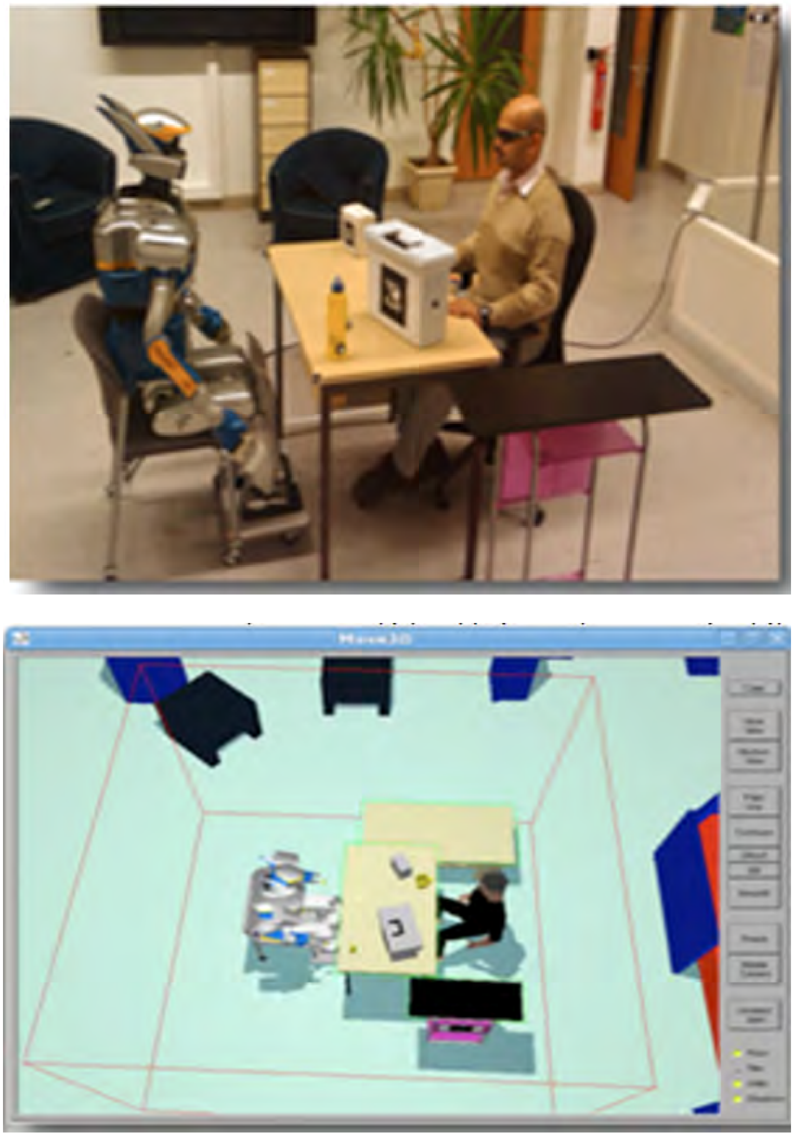


Figure 4.2: Real world and its real time 3D representation in Move3D (see appendix A for detail). The red bounding box shows the current workspace used construct and update the *Mightability Maps* in real time.

time 3D representation in Move3D (see appendix A for the detail). Move3D further facilitates the robot to check self and external collisions of all the agents and objects.

4.2.1 Discretization of Workspace

For reasoning on the spaces, the robot constructs a 3D workspace (red box in figure 4.2(b), dimension of $3m \times 3m \times 2.5m$ for current scenario) and discretizes it into cells, each of dimension $5cm \times 5cm \times 5cm$. Note that the dimension and position of this bounding box for workspace can be decided upon the interest and requirement of the human-robot interaction scenario and context. For most of the discussion in this chapter, we will discuss in the context of human-robot interactive object manipulation tasks, with the objects on the tables. So, we define the workspace which is centered at the middle of the central table and large enough to cover all the object and agents of interest. Such bounding box of the workspace facilitates to achieve the goal of online updation of various facts related to places, such as visible and reachable places from different agents' perspectives. Further, each cell in the workspace is marked as occupied or free of obstacles, and in the case of occupied, the name of the corresponding object or the agent is associated to the cell.

4.2.2 Extraction of Support Planes and Places

In Move3D, the object's shape is modeled as a polyhedron. We have developed an approach to autonomously extract all possible support planes on which some object could be placed. For this, first all the facet having vertical normal vectors are extracted. All such facets belonging to same object are merged together. Then a symbolic name is provided to the support name based on the object.

Further, to find visible and reachable places (cells) on table or any other support plane, the cells belonging to planner tops are extracted and further the information about the object belonging to that support plane is stored as supporting object.

This equips the robot to place an object on the top of a table plane, on the top of any other object such as box. So, no external information about supporting surfaces is provided. The robot autonomously finds and updates the places where it could put "something", depending upon the environment.

4.3 Visuo-Spatial Perspective Taking

In this section, first we will describe calculation of places visible, reachable, occluded and obstructed from an agent's perspective. Then we will present such calculations for the objects, further the calculation of occluding and obstructing object will be presented.

4.3.1 Estimating Ability *To See*: *Visible, Occluded, Invisible*

4.3.1.1 For Places

For calculating the visibility, from a given position and yaw and pitch of the head, robot finds the plane perpendicular to the axis of field of view. Then that plane is uniformly sampled to the size of a cell of the 3D grid of the workspace. Then as shown in figure 4.3, a ray is traced from the eye/camera position of the agent to

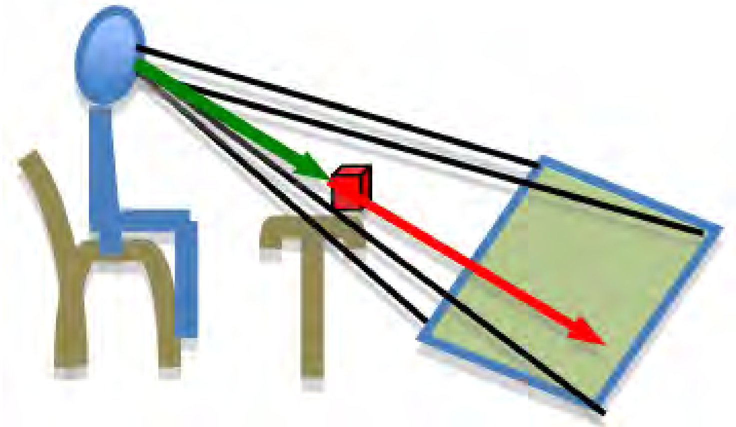


Figure 4.3: Ray tracing based calculation of an agent's Visibility from a particular physical state of the agent. Red small box is an object. The points on the green ray are said to be visible, whereas the points on red ray are said to be invisible. The red object is said to be occluding object.

each such sample points on the plane. All the cells on the ray until an obstacle cell (if any) are marked as *Visible*, as shown by green arrow. And all the cells from the obstacle cell until the plane (red arrow) are marked as *Occluded*. Let the set of all the cells in the environment's 3D grid is G. The set of visible cells for a particular agent for a particular environment is V and that of occluded cells is O, then we define the set of *Invisible* cells I as:

$$I = G - \{V \cup O\} \tag{4.1}$$

Here it is important to note that these places are estimated for a given posture of the agent for a given head orientation.

4.3.1.2 For Objects

We use two levels of object visibility calculation: *Cell based* for a rough but fast estimation and *Pixel based* for finding precise percentage of how much the object is visible. For cell based object visibility calculation, as the robot has the information about the visible cells and to which object the cell belong, an object is said to be visible if at least one cell belonging to that object is visible. Further, to estimate "how much" an object is visible, a *visible area* VS is found for an object *obj* from an agent Ag perspective as:

$$VA_{obj}^{Ag} = NC_{obj} \times 2 \times cell_{length} \tag{4.2}$$

where, NC_{obj} is number of visible cells which is multiplied to the area of one face of the 3D cell to get the total visible area.

For pixel based visibility information, the robot uses the projected image of the field of view of the agent and calculates total number of pixels belonging to the object of interest in that image. In case of pixel based estimation, we further define a visibility score VS of an object obj from an agent Ag perspective as:

$$VS_{obj}^{Ag} = \frac{N_{obj}}{N_{FOV}} \tag{4.3}$$

where, N_{obj} is number of pixels of the object in the image of agent's field of view and N_{FOV} is total number of pixels in that image.

Depending upon the level of accuracy required, *VA* or *VS* will be used to find whether an object obj is occluded or invisible from an agent Ag perspective. If obj is inside the solid angle formed by field of view of Ag and *VA* or *VS* is zero, the object is said to be *Occluded*. If obj is outside the solid angle formed by field of view of Ag, the object is said to be *Invisible*.

4.3.2 Finding Occluding Objects

The robot not only estimates that an object is occluded, but also finds the objects, which is occluding that object from the agent's perspective. For this, from each cell belonging to the occluded object *Obj*, a ray R is traced back to the eye of the agent Ag and a set S of cells satisfying following criteria is extracted on the ray: (a) cell is occupied (b) cell does not belong to current object of interest, *Obj*. Then elements of S are grouped based on the corresponding objects to which the cells belong. Further, these objects are sorted in reverse order based on which cell appeared first in the ray R. Hence, not only the objects, which are occluding an object is found but also the relative order from the agent's perspective is obtained.

4.3.3 Estimating Ability *To Reach*: *Reachable, Obstructed, Unreachable*

4.3.3.1 For Places

Although one can choose to calculate reachability of an agent using inverse kinematics (IK) approaches. But these approaches are expensive and take hours to calculate and update [Zacharias 2007] in a changing human robot interactive environment. We chose to postpone such expensive calculations until the last stage of actual movement planning. As a first step to perceive reachability of an agent, we adapt from how we perceive reachability. From the studies in [Carello 1989], [Bootsma 1992], [Rochat 1997] the general agreement is that the prediction to reach a target with the index finger depends on the distance of the target relative to the length of the arm and plays as a key component in actual movement planning. Therefore, we will also use the length of the arm to estimate the reachability boundary for the given posture of the agents. Hence, a cell will be marked as reachable from a particular posture of the agent if: (i) it is within a distance of arm length from the shoulder joint position and (ii) there is no occupied cell on the line joining the shoulder joint and the cell. If (i) is not satisfied, then the cell is marked as *Unreachable*. If (i) is satisfied but (ii) is not satisfied, then the cell is marked as *Occluded*. The joint limits of shoulders of agents are used to restrict the directions vector from the shoulder to calculate the reachable points by a particular hand.

Here it is important to note that in calculating this reachability, all the joints except belonging to the arm of interest of the agent is assumed to be fixed. It is similar to estimating: given this posture of the agent, if he/she/it will stretch out his left/right hand, which are the places he can reach. It is the calculation of Mightability, which we will introduce later on in this chapter, where robot activates one or another joints of the agents by applying some virtual actions of symbolic efforts, such as lean forward, turn around, to estimate reachablity in different postures.

An agent can show reaching behavior to touch, grasp, push, hit, point or take some object from inside some container, etc. Hence, having a perceived maximum extent of the agent's reachability even with some over estimation will be acceptable as the first level of estimating the ability, which could be further filtered by the nature of the task as well as more rigorous kinematics and dynamics constraints.

4.3.3.2 For Objects

As already mentioned an agent can show reaching behavior to touch, grasp, push, hit, point, take out or put into something from a container object, etc., precise definition of reachability of an object depends on the purpose. So, at first level we chose to have a rough estimate of reachability based on the assumption that if at least one cell belonging to the object is reachable, then that object is *Reachable*. Further, the total number of reachable cells belonging to that object is also stored. Note that if required, this reachability is further refined based on the task requirement at later stages of planning and decision-making. But again to facilitate online estimation and updation, we prefer to avoid performing more expensive whole body generalized inverse kinematics based reachability testing until the final stages of task planning, where it is really required.

An object is said to be *Obstructed* if no cell of the object is reachable and at least one cell of the object is obstructed. If an object is neither reachable, nor obstructed, it is said to be *Unreachable* if the agent will stretch out his/her/its hand from a given posture.

4.3.4 Finding Obstructing Objects

The robot not only estimates that an object is obstructed to be reached by an agent from a given posture, but also finds the objects, which in fact are obstructing

Figure 4.4: Taxonomy of reach actions studied in human movement and behavioral psychology research, [Gardner 2001], [Choi 2004]:(a) arm-shoulder reach, (b) arm-torso reach, (c) standing reach. We have adapted and enriched this taxonomy to develop the human-aware effort analysis table as shown in figure 4.5(a).

that object from the agent's perspective. For this, an approach similar to finding occluding objects in section 4.3.2 has been used. The difference is from each cell belonging to the obstructed object *Obj*, a ray *R* is traced back to the shoulder joint of the agent *Ag*. And similarly the robot not only finds the objects, which is obstructing but also finds the relative order from the agent's perspective to reach.

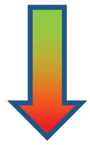
Until now, we have discussed how we perform visuo-spatial perspective taking of the agent from a given state. We have also discussed that how we extract information about finding occluding or obstructing objects. This provides the information about "what" is depriving an agent to see or reach something (place or object), which should otherwise be visible and reachable from a given state of the agent. This information can help in deciding "what" changes should be made in the environment to enable the agent to see and reach without any additional effort by the agent itself. However, as discussed earlier, there are objects and places, which are not visible or reachable because they are beyond the field of view or reachability range of the agent. This requires agent to put some effort to see or reach such places/objects provided the environment is not altered. Below, we will first discuss our proposed hierarchy of efforts and then we will present the concept of the *Mightability Analysis*, which performs effort based visuo-spatial perspective taking.

4.4 Effort Analysis

Perceiving the amount of effort required for a task is another important aspect of a socially situated agent. It plays roles in effort balancing in a co-operative task as well as provides a basis for offering help pro-actively. A socially situated robot should be able to perceive the effort quantitatively as well as qualitatively in a '*meaningful*' way understandable by the human. An accepted taxonomy of such '*meaningful*' symbolic classification of effort could be developed by taking inspiration from the research of human movement and behavioral psychology, [Gardner 2001],

Effort to Reach	Effort to See
No_Effort	*No_Effort*
Arm_Effort	*Head_Effort*
Arm_Torso_Effort	*Head_Torso_Effort*
Whole_Body_Effort	*Whole_Body_Effort*
Displacement_Effort	*Displacement_Effort*
No_Possible_Known_Effort	*No_Possible_Known_Effort*

(a)

Effort Level
Minimum: 0

Maximum: 5

(b)

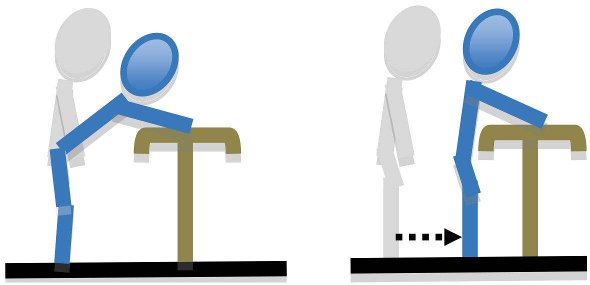
Figure 4.5: Human-aware effort analysis and effort hierarchy (motivated from the studies of human movement and behavioral psychology, [Gardner 2001], [Choi 2004] (see figure 4.4): (a)**Human-Aware Effort Analysis**: Qualifying efforts *to see* and *to reach* some object or place in the human understandable levels of abstraction. (b)**Human-Aware Effort Hierarchy**: One possible way of comparative effort analysis. Such analysis facilitates to ground, compare and reason about efforts in a meaningful and human-understandable way for day-to-day human-robot interaction.

[Choi 2004], where different types of reach actions of the human have been identified and analyzed. Figure 4.4 shows taxonomy of such reaches involving simple arm-shoulder extension (arm-and-shoulder reach), leaning forward (arm-and-torso reach) and standing reach. This suggests us a way to qualify human effort in terms of main body joints involved. Inspired from this we also equipped our robots to analyze and reason on the efforts of all the agents at a human understandable level.

4.4.1 Human-Aware Effort Analyses: Qualifying the Efforts

We have conceptualized a symbolic set of effort based on the body parts involved in performing an action. Let us assume that an agent Ag is currently sitting on a chair. From this current state Ag can put different efforts to attain different states to see or reach something or to perform some task. From this current state if the agent has to just turn his/her/its head to see an object or place, we term it as *Head_Effort*. If he/she/it has to turn torso, it is *Torso_Effort*, if agent is required to stand up, it is *Whole_Body_Effort*, if required to move, it is *Displacement_Effort*. Similarly, if the agent has to just stretch out his/her/its arm (to point, to reach, ...) an object it is *Arm_Effort*, if he/she/it has to turn around or lean, it is again *Torso_Effort* to reach and so on. The robot further associates descriptors like left, right. For example, the robot could further distinguish the arm-torso efforts to reach, which is turning left and reaching by right hand from another arm-torso effort, which might be turning right and reaching by left hand, and so on. This effort analysis has been shown in figure 4.5(a).

Associating a level of effort to such qualifying labels could further facilitate the



(a) (b)

Figure 4.6: Reaching to a place on the table with different types of efforts (a) *Arm_Torso_Effort* and (b) *Displacement_Effort*. Depending upon the individual's desired, situations, state and constraints, one or the other effort type could be preferred or said to be requiring relatively less effort.

comparative analysis of efforts. One intuitive levels of effort has been shown in 4.5(b). For most of the human-robot day-to-day interaction situations, we can reasonably use this to compare different efforts. In this thesis, wherever we talk about such human-aware effort analysis by also incorporating the effort levels, we will use the term *human-aware effort hierarchy*. Note that such effort hierarchy may not always hold strictly, or there might exist a fuzzy boundary depending upon the situation and individual preferences. For example. figure 4.6 shows an agent is reaching to a place on the table with two different types of efforts. In both cases, the categorization of the effort as shown in figure 4.5(a) holds, and the robot would be able to distinguish between the *Arm_Torso_Effort* and *Displacement_Effort*. However, the interpretation of the relative level of effort might vary. Depending upon the criteria to measure effort, one or the other effort type could be said to be requiring less effort. The studies of musculo-skeletal kinematics and dynamics models such as [Khatib 2009], [Sapio 2006], combined with the time and distance could be used to find a measure of relativeness of the efforts in such situations.

The significance of such effort analyses includes:

- **Grounding Effort**: It can be used to describe an effort to a meaningful i.e. human understandable symbols, hence enriching the robot's grounding capabilities in human-robot interaction. The robot can further ground the agent's movement to a meaningful effort.
- **Constraining planning and decision making**: Another direct advantage of such effort levels is that we can directly incorporate different constraints related to the desire and physical state of an agent, in decision-making and cooperative task planning. For example, if the agent is having back or neck pain, we can exclude his efforts associated with the torso or head movement. Someone who faces challenge in standing up or have reduced mobility, the robot can directly restrict his maximum effort level as torso effort and so on.

- **Regulating effort levels**: Similarly, current situation and preferences could also be used to restrict the maximum allowed effort level or to exclude some effort. For example, if someone is tired and sitting on a chair, the robot can restrict his/her effort in planning for a cooperative task, such as the agent would not prefer to stand up or move, hence restricting his/her effort to *Arm_Effort*.
- **Incorporating social preferences**: Further, such levels of effort can be used to plan a cooperative task based on the relative social status of the agents. For example if the agents are friends, the mutual efforts could be balanced, so that both will lean forward for an object hand-over task. If one agent is boss, another agent can plan to perform the task so that boss will be required less effort, by standing and giving the object to the boss so that boss will require only arm-effort to take it, and so on.

4.4.2 Quantitative Effort

As the robot reasons on 3D model of the agents with the rich information of joints, it is further able to compare two efforts of same symbolic level, i.e. capable of intra-level quantitative effort measures, based on how much the joint is required to move/turn or how much the agent is required to move. However, as mentioned earlier, the studies of musculo-skeletal kinematics and dynamics models such as [Khatib 2009], [Sapio 2006], could be used to assign a quantitative measure to different effort types presented in figure 4.5(a).

4.5 Mightability Analysis

By fusing the effort-based analysis with visuo-spatial perspective taking, we have developed the concept of *Mightability Analysis*, which stands for "*Might be Able to...*". The idea is to analyze various abilities of an agent such as ability to see, ability to reach, not only from the current state of the agent, but also from a set of states, which the agent might achieve from his/her/its current state.

For performing Mightability Analysis, the robot applies, $A_V = [a_1, a_2, ..., a_n]$, an ordered list of virtual actions, to make the agent virtually attain a state and then estimates the abilities by respecting the environmental and postural constraints of the agent. Currently, the set of virtual actions are:

$$a_i \in \left\{ A_V^{head}, A_V^{arm}, A_V^{torso}, A_V^{posture}, A_V^{displace} \right\} \tag{4.4}$$

Visibility for Human and Robot	**Reachability for Human and Robot**
From **Current** head orientation, (C)	Arm-shoulder reach from **Current** position, (C)
Virtually **Turning** head **Straight**, (T_H_S)	Arm-torso reach by virtually **Leaning Torso** until collision or waist's pitch joint limit is reached, (L_T)
Virtually **Turning** the **Head** around, **Left** and **Right**, until neck's yaw joint limit is reached, (T_H_L,), (T_H_R)	Arm-shoulder reach by virtually **Turning Torso** around, **Right** and **Left**, until collision detected or waist's yaw joint limit is reached, (T_T_L), (T_T_R)
Virtually **Turning** the **Torso Left** and **Right**, until collision or waist's joint limits are reached and then turn **Head** until neck's yaw joint limit is reached, (T_T_H_L), (T_T_H_R)	Arm-torso reach by virtually **Turning Torso** around, **Left** and **Right**, and **Leaning** until collision detected or waist's yaw and pitch joint limits are reached, (T_T_L_L), (T_T_R_L)
Virtually **Standing** and applying same actions as above, (S_C), (S_T_H_S), (S_T_H_L), (S_T_H_R), (S_T_T_H_L), (S_T_T_H_R)	Virtually **Standing** and applying same actions as above, (S_C), (S_L_T), (S_T_T_L), (S_T_T_R), (S_T_T_L_L), (S_T_T_R_L)

Figure 4.7: A subset of virtual states from all possible attainable States of the agents, which is used to proactively calculate and update the Mightabilities. This is to make the robot more 'aware' during the course of Human Robot Interaction.

where,

$$A_V^{head} \subseteq \{Pan\_Head, Tilt\_Head\} \tag{4.5}$$

$$A_V^{arm} \subseteq \{Stretch\_Out\_Arm\,(left/right)\} \tag{4.6}$$

$$A_V^{torso} \subseteq \{Turn\_Torso, Lean\_Torso\} \tag{4.7}$$

$$A_V^{posture} \subseteq \{Make\_Standing, Make\_Sitting\} \tag{4.8}$$

$$A_V^{displace} \subseteq \{Move\_To\} \tag{4.9}$$

The robot performs Mightability Analyses by taking into account collision as well as the joint limits. The robot uses kinematic structures of the agents and performs various virtual actions until the joint limits of the neck and/or torso are reached or the collision of the torso of the agent with the environment is detected.

4.5.1 Estimation of Mightability

For maintaining a rich knowledge about the agents' abilities, we have chosen a set of virtual actions for which Mightability is to be computed and updated throughout the course of interaction.

Figure 4.7 summarizes different virtual states for which the robot calculates and continuously updates the Mightability.

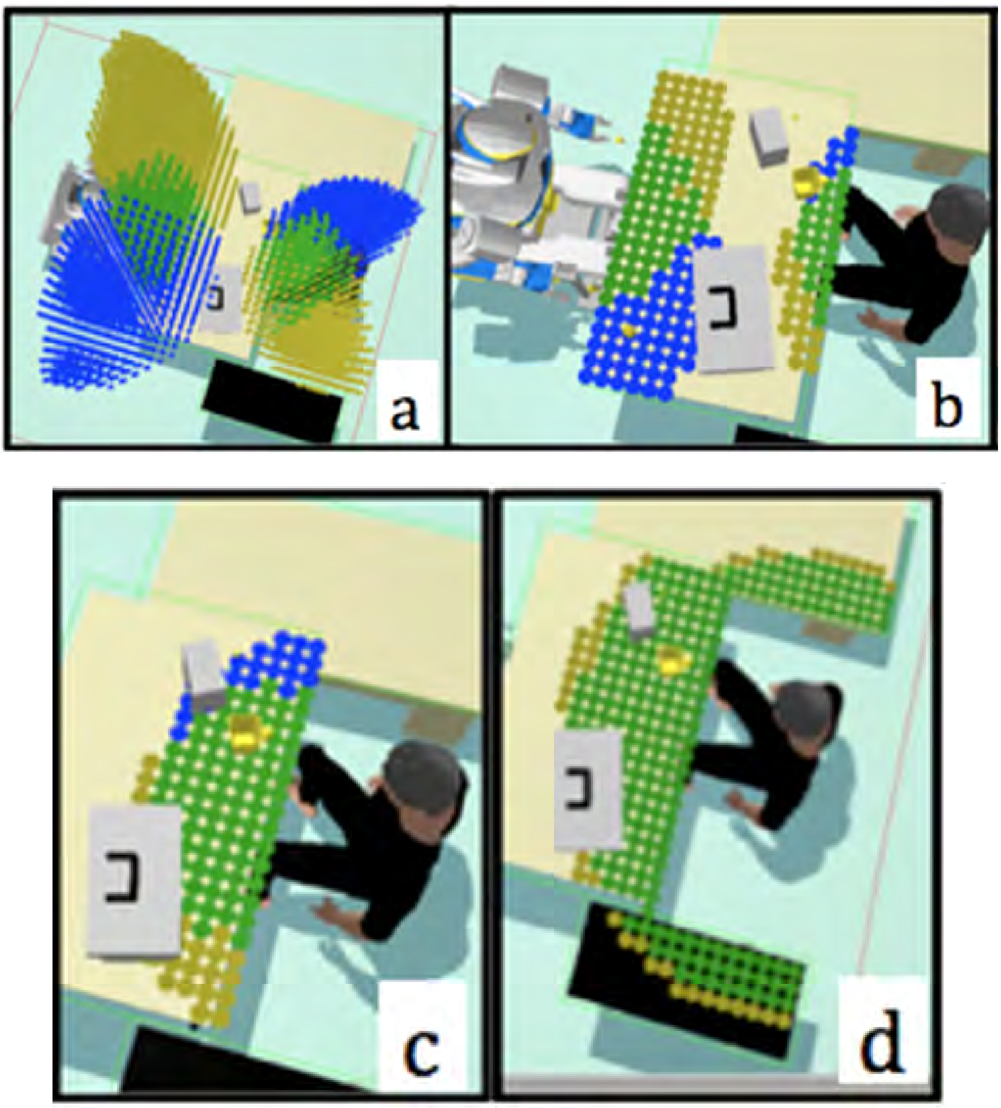


Figure 4.8: Mightability Maps of reachability for the Human and the HRP2 robot corresponding to the real world scenario of figure 4.2. (a) and (b) show the *Arm_ Effort* reachability from the current states of the agent in 3D grid (a) and on table-top (b). It also distinguishes the reachability by the left hand only (yellow), by the right hand only (blue) and by both hands (green) of an agent. (a) and (b) also show that there is no common reachable place if neither of the agents will put any further effort. (c) Shows the places, the human might reach, if he will maximally possible lean forward, an action associated with *Arm_ Torso_ Effort*. The human can reach more places as compared to (b). (d) Shows the reachable places if the human will turnaround and leaning, other actions associated with *Arm_ Torso_ Effort*. The human might reach some parts of the tables of different heights on his both sides.

Note that depending upon the requirements the robot could apply a different set of virtual actions from expression 4.4 to calculate the Mightability of an agent from a different virtual state.

The robot first calculates the arm-shoulder reach. For this, the robot stretches the hand of the 3D model of the agent by permissible limit of each shoulder's yaw and pitch joints and performs the *to-reach* perspective taking as explained in section 4.3.3. Then the robot virtually leans the agent's model by its torso incrementally (by an angular step of 5 degrees in current implementation) until there is collision with the upper torso or the maximum limit of waist pitch joint has been reached. And from each of these new virtual positions of the agent, the robot again performs the visuo-spatial perspective taking as explained in section 4.3.3. Next, the robot turns the torso of the agent's model at its current position until collision or maximum limit of human waist yaw is reached and again performs the visuo-spatial perspective taking. Similarly, to-reach visuo-spatial perspective taking of other states are performed, such as virtually changing the posture of the agent from standing to sitting or from sitting to standing. Similarly, the robot performs *to-see* perspective taking as explained in section 4.3.1. First, it finds from the current head orientation of the agent. Then it turns the head, towards left and right, until the neck joint limit. Then it turns the torso left and right until collision or waist yaw limit is reached. Such analyses are done for each agent in the environment, including the robot itself. Since the system is generic to perform Mightability Analysis for any type of agent in the environment, depending upon the kinematics structure of the agent, some of the virtual states might not be feasible for that agent. For example for PR2 robot there is no degree of freedom for the torso joint to lean forward.

4.5.1.1 Treating Displacement Effort

As already mentioned, the robot continuously maintains and updates visuo-spatial abilities of all agents upto the *Whole_Body_Effort Level*. The estimation of *Displacement_Effort* level based ability to see or reach is calculated only when it is required. For this, first the space around the object/place is uniformly sampled in a co-centric circular manner with increasing radius. And the agent is virtually placed at each such position, if there is no collision with the environment. From this new virtual position, the ability to see and reach is calculated. If still not reachable or visible, the agent is virtually leaned-forward by angular steps until collision or waist joint limit. If still the object is not reachable, next sampled place around the object/place is tested. The maximum radius of the circle to sample the places around is limited by the total length of the arm and the torso to shoulder length, with the assumption that agent's ability to lean forward completely is the maximum effort he can put to reach/see something from a position. Of course, if still the agent is not able to see or reach, depending upon the situation or requirement, the further subset of virtual action could be applied to the new position of the agent. In section 4.7, we will show the example of calculated *Displacement_Effort* to reach an object.

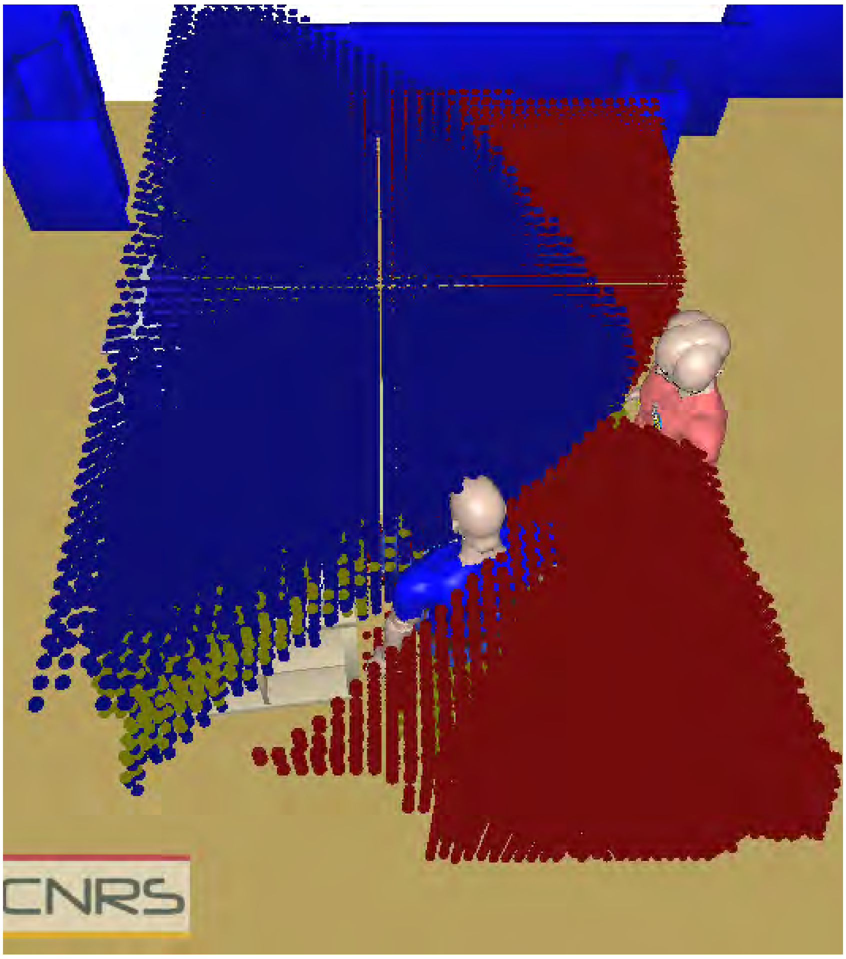


Figure 4.9: Mightability Maps of visibility for the Human on the right with *Head_Effort*. The blue cloud shows currently visible places, and the red cloud shows the places, which the human can see if he will look around only by turning head.

4.5.1.2 Mightability Map (MM)

When such Mightability analyses are performed at the levels of cells of the discretized 3D workspace, we term it as *Mightability Maps (MM)*.

Mightability Maps encode, which places an agent might be able to see and reach, if he/she/it will put a particular effort or perform an action. This can be used for a variety of purposes. For example, finding the candidate places where an agent can perform a task for another agent with a particular effort level, or where an agent can potentially hide an object from another agent with maximum possible effort level, so that the agent can reason about potential places to search for.

Mightability Maps for the human and the humanoid Robot HRP2 from their current states to reach have been shown in 3D, figure 4.8(a), and on table plane, figure 4.8(b).

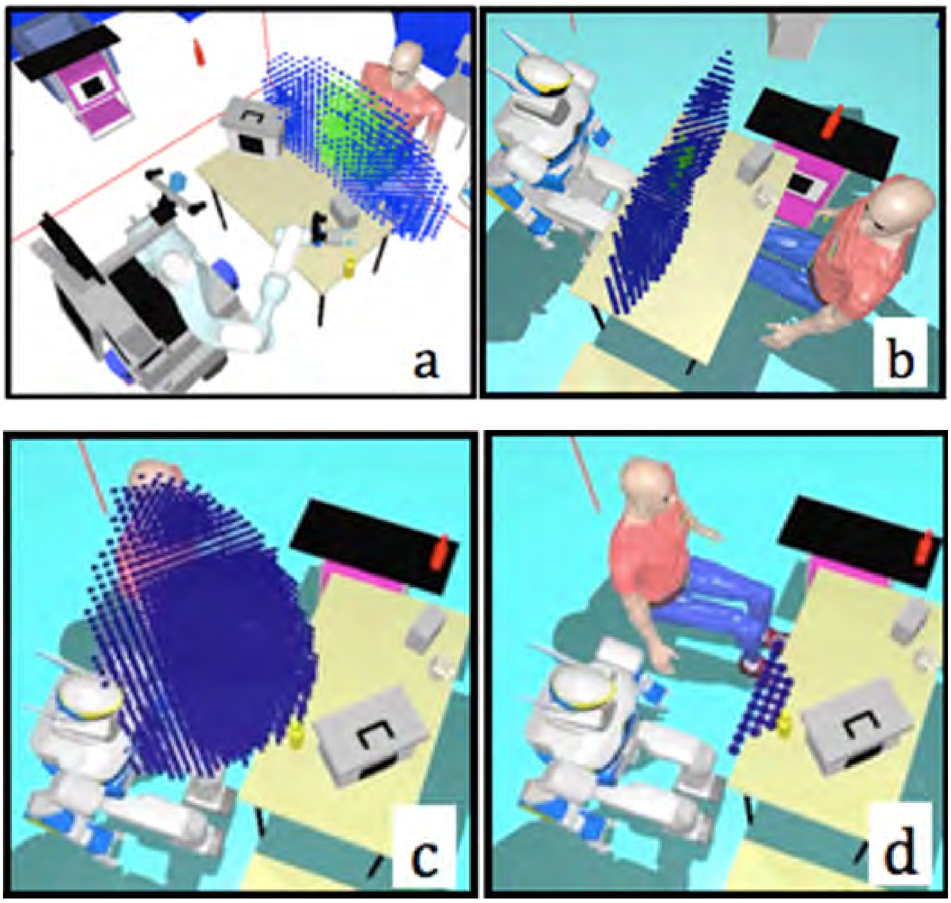


Figure 4.10: Common reachable regions: (a) for human and JIDO robot, (b) for HRP2 and lean forward effort of Human, (c) for HRP2 and Human from their current state in 3D and (d) on the table plane.

Robot also distinguishes among the cells, which could be reached only by left hand (yellow), right hand (blue) and by both hands (green). The robot could use this information to conclude that there is no common reachable region if neither of them will lean forward. Figure 4.8(c) shows reachability of human on table with maximum possible leaning forward. The robot also perceives that if human will turn around and lean he might be able to reach parts of the side-by tables as well, as shown in figure 4.8(d).

Figure 4.9 shows the visibility Mightability Maps for the human sitting on the right. The red cloud shows the currently visible places for him, whereas the red cloud shows the places which the human can see if he will put *Head_ Effort* and look around.

As such Mightability Analysis could be performed for different types of agents, figure 4.10(a) shows the common reachable region in 3D obtained by intersection operation on reach Mightability Maps of Human and another single-arm robot Jido from their current states. This in fact could serve as candidate place where Jido can hand over

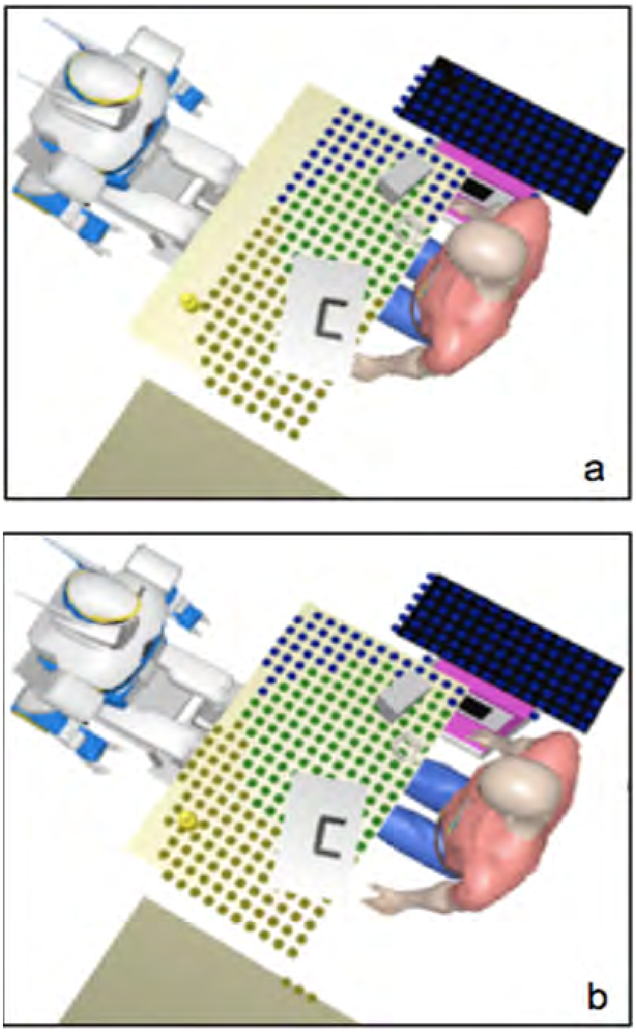


Figure 4.11: An interesting fact encoded in *Mightability Maps* because of environmental constraints on possible virtual actions. Figures show the reachability of the human on the table surface by Mightability Analysis for torso effort to attain the state of maximal possible lean forward. As the human closer to the table could lean less compared to sitting away from the table. Hence, even if the human is sitting away from the table he can reach more parts of the table (see reachable regions in (b)) compared to sitting very close to the table (see reachable regions in (a)).

an object to the human. As shown in figure 4.8(a) there was no common reachable region from the current states of Human and HRP2, but as shown in figure 4.10(b), HRP2 is able to estimate that if the human puts effort to lean forward then there might exist a common reachable region. Figure 4.10(c) and (d) show the common reachable region in 3D and on table plane from the current states of the Human and HRP2 in a different setup where both are sitting side-by-side. These regions respectively could serve as the candidate places to give an object and to put an

3D grid creation and initialization (one time process)	**1.6**
3D visual Mightability Maps for Human	0.146
3D visual Mightability Maps for HRP2 Robot (excluding making the robot virtually standing)	0.089
3D spatial Mightability Maps for Human	0.128
3D spatial Mightability Maps for Robot (excluding making the robot virtually standing)	0.083
Total Time for all Mightability Maps' calculation (excluding one time 3D grid creation and initialization process) = **0.446 seconds**	

Figure 4.12: Initialization and Calculation times for Mightability Maps for a typical scenario as shown in figure 4.2. Hence, by choosing to update only those parts, which have been affected by the changes in the environment, we achieve to maintain Mightability Maps updated in real time.

object for the human to take.

Figure 4.11 shows an interesting observation about leaning forward reach. The reachable region by leaning forward in figure 4.11(a) is less compared to that of figure 4.11(b), even the human is closer to the table in the former case. This is because, as mentioned earlier our approach respects the postural and environmental constraints, and in the former case the human is very close to the table edge, hence, could lean less as compared to the latter case where there is sufficient gap between human torso and the table to lean more without collision.

4.5.1.3 Object Oriented Mightability (OOM)

When the Mightability analysis is performed for the object in the environment, we call it *Object Oriented Mightabilities (OOM)*.

Object Oriented Mightability encodes, which objects an agent might be able to see and reach, if he/she/it will put a particular effort and perform an action. This can be used for variety of decision-making and planning purpose. For example if robot knows different effort levels to see and reach same object, it can generate a plan to perform a shared task by taking into account time and effort. It could assign a sub-task to an agent who can perform it with least effort.

4.5.2 Online Updation of Mightabilities

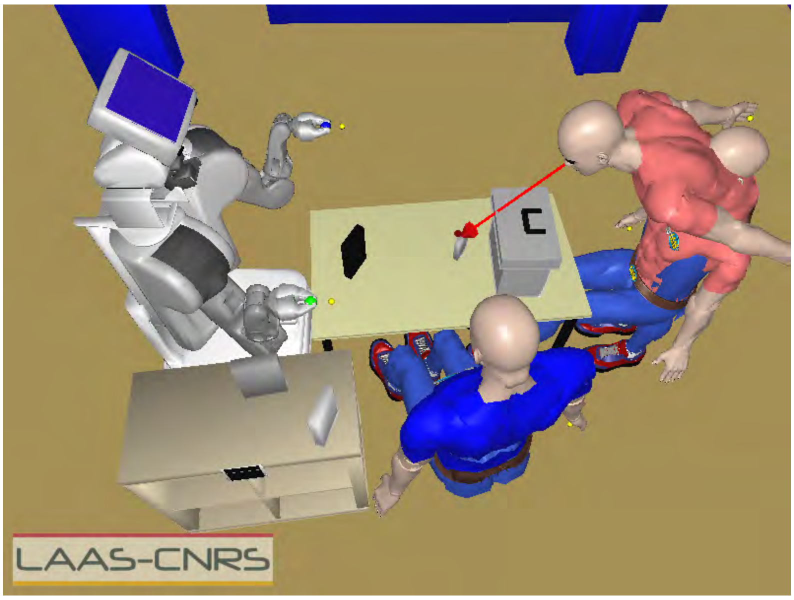
Figure 4.12 shows time for calculating various Mightability Maps for the human and the HRP2 humanoid robot sitting face-to-face as shown in figure 4.2(a). It also shows the time for one time process of creating and initializing cells of the 3D grid to discretize the workspace with various information like cells which are

Figure 4.13: Example scenario with two humans, and the PR2 robot. There are different objects, reachable and visible by different agents with different effort levels.

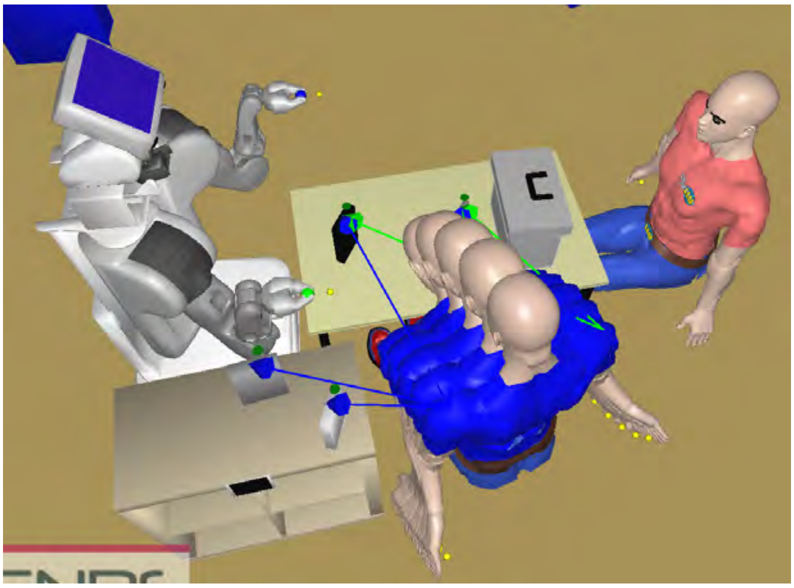
obstacle free, which contains obstacles, which are the part of the horizontal surfaces of different tables, etc. Note that it took 1.6 seconds to create and initialize 3D grid consisting of 180000 $(60 \times 60 \times 50)$ cells, each of dimension $5cm \times 5cm \times 5cm$, hence, 0.000009 seconds for a single cell. Figure 4.7 also shows that for a typical scenario as shown in figure 4.2 it takes about 0.446 seconds to calculate all the Mightability Maps for the human and the robot, once the 3D grid is initialized. As these are the calculation time for all the virtual states, for all the agents for all the cell, and as practically the changes in the environment will affect a fraction of the 3D grid, the Mightability Map set are updated online. For this, we have carefully devised rule to update only those parts and those information, which are getting affected by the change in the environment. For example due to movement of objects on the table, the information about the cells belonging to the object's old and current positions need to be updated in 3D grid and then the visibility and reachability of the agents. Similarly, if an agent is looking around, only the visibility Mightability Map of that agent and that too only of his/her/its current state should be changed as the position of the agent has not changed.

4.6 Mightability as Facts in the Environment

As discussed in section 3.3.1 of chapter 3, we have incorporated abilities of different agents as the attributes of the environment. This facilitates to reason about the



(a)



(b)

Figure 4.14: Least feasible effort analysis. For the current scenario of figure 4.13, based on Mightability Analysis, the robot is able to find: (a) The least effort to see the small tape by the right human. It successfully finds that the human will not only be required to stand up but also to lean forward to see the small tape, which is currently behind the box from the human's perspective. (b) Least effort to reach the black tape by the middle human, which is estimated to be lean forward effort.

environmental changes in terms of the facts associated with agents' abilities. We have defined in eq. 3.28, ability of an agent as a set of tuple $Ab_{Ag} = \langle T_{ab}, P_{ab}, EC_{ab} \rangle$,

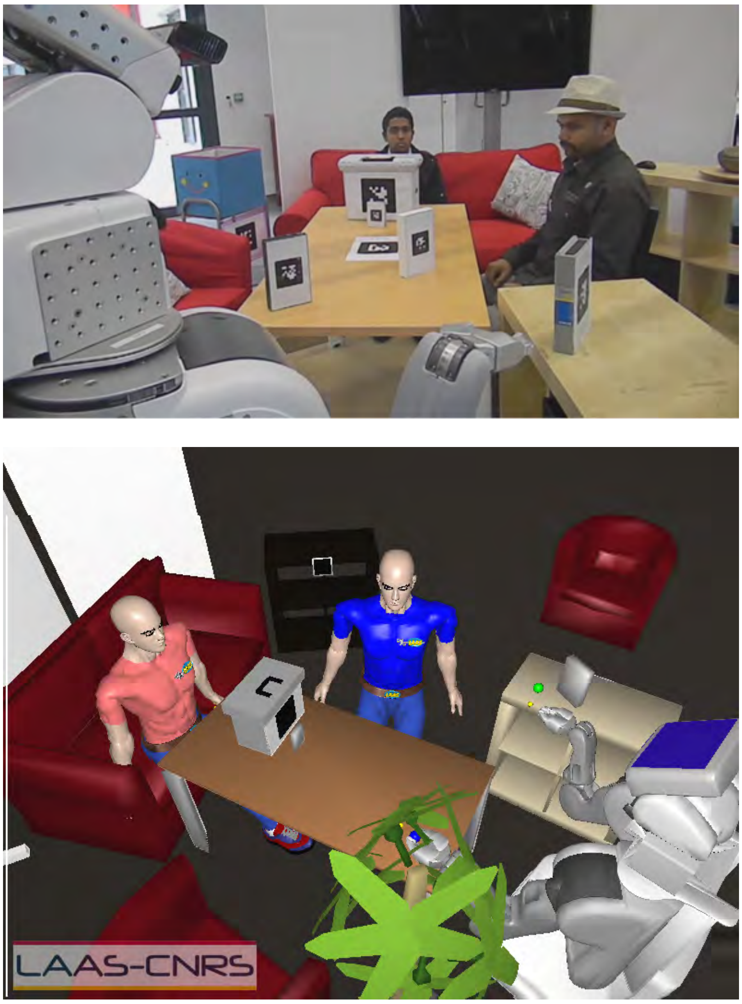


Figure 4.15: Human-Human-Robot interactive scenario (Top). And its 3D model constructed and updated online (Bottom).

where T_{ab} is the type of ability, P_{ab} is the parameter of the ability, EC_{ab} is the enabling condition of the ability, which could be anything ranging from a state, to an action of effort. Hence, we can easily represent the Mightability Maps and Mightability Analysis in this form of environmental fact. For example, for $Ag = human1$, $f = Ab_{human1} = \langle see, object1, Head\_Effort \rangle$ will be a fact $f \in F$ of the environment, which will constitute to determine the state $s \in S$ of the environment. Hence, it facilitates to state a task planning problem discussed in chapter 3 in an enriched way, e.g. find a plan so that the goal state will require more effort of the *human1* to see *object1*, or find a plan so that the goal state consists of the fact: *object1* is reachable by *human2* with *Whole_Body_Effort*.

Figure 4.16: *Least Effort Analysis* for the human currently sitting on the sofa to reach the object on the right of the robot. The robot not only estimates that the human will be required to move but also the possible positions to reach the object; hence, need to put *Displacement_Effort*, followed by leaning forward.

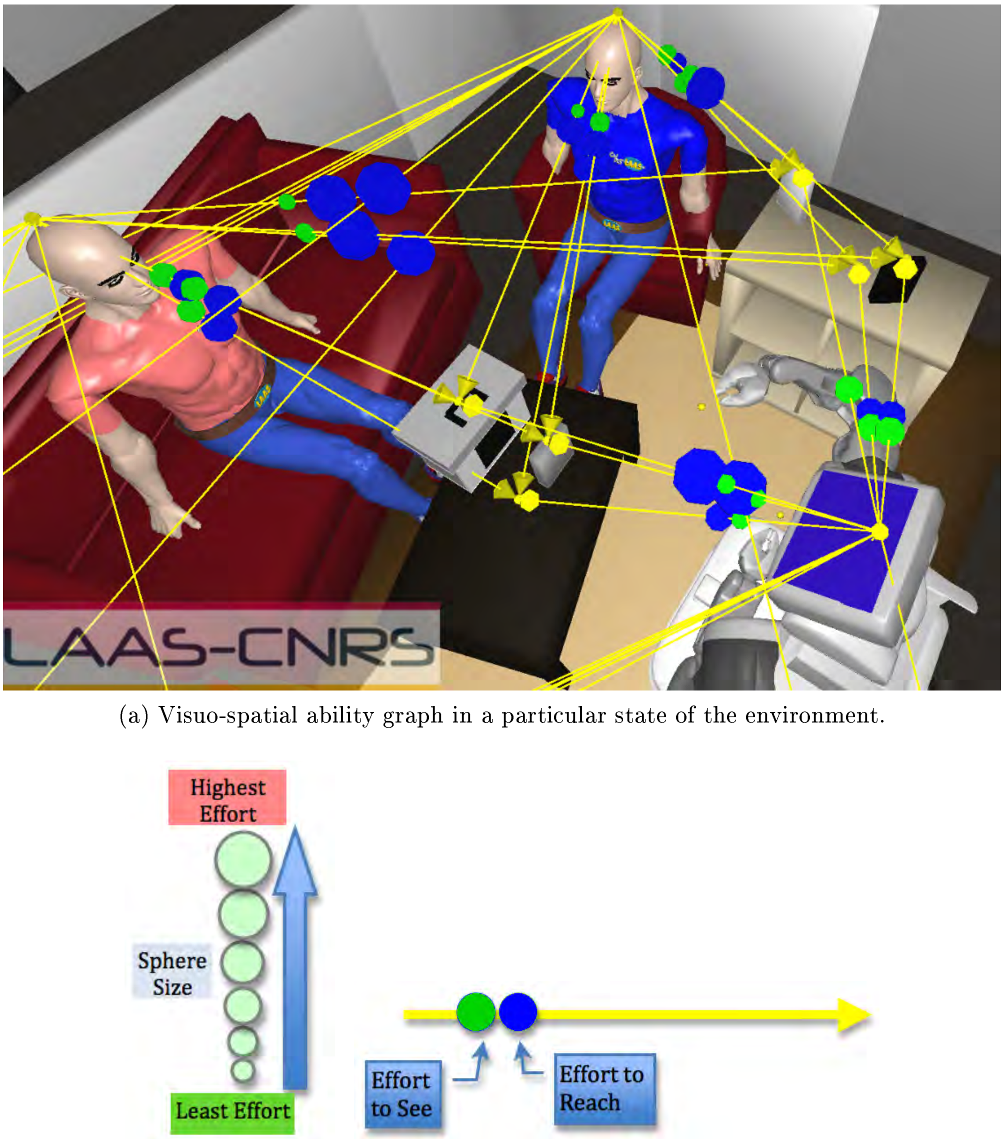
4.7 Analysis of Least Feasible Effort for an Ability

Using the Mightability Analysis, for a given scenario the robot is able to find the multi-effort ability (see, reach, ...). From those efforts, then it can extract the least feasible effort state from the current state of the agent, which makes an object visible and reachable from the agent's perspective.

Figure 4.13 shows one of the example scenarios, with two humans and the PR2 robot. The robot constructs and updates, in real time, the 3D model of the world by using *Kinect* based human detection and tag based object localization and identification through stereo vision. In the current situation, the robot not only knows that the object, small tape, is currently neither visible nor reachable to the human on the right, but also able to estimate the least effort state to see it and reach it.

As shown in figure 4.14(a), the robot estimates that the human on the right will be at least required to stand up and lean forward to see the small tape object, which corresponds to *Whole_Body_Effort*. Similarly, the robot estimates that if the human on the middle has to reach the black tape, he will be required to at least put *Torso_Effort*, as he is required to lean forward, figure 4.14(b).

Figure 4.15 shows another example scenario with the corresponding 3D model, which



(a) Visuo-spatial ability graph in a particular state of the environment.

(b) Effort Sphere (c) Edge Description

Figure 4.17: Visuo-spatial ability graph and an edge description. Each edge encodes the least feasible effort to see and reach an object by an agent. Note that for a same *agent-object* pair both the efforts could be different, which has been captured successfully by the Mightability Analysis.

is constructed and updated online. Figure 4.16 shows that the robot is able to estimate that the least effort of the human sitting on the sofa will be required to put *Displacement_Effort*, to reach the object, which is on the right of the robot. It also estimates that the human will not only be required to move but also will be required to lean forward to reach the object. It further shows the possible positions and postures of the human to reach the object. Note that at the symbolic level of effort, all such postures correspond to *Displacement_ Effort*. These could further be ranked based on the path length to move to the location and the amount of leaning forward required.

4.8 Visuo-Spatial Ability Graph

We store the facts of least effort related to Object-Oriented Mightability in a graph, which we termed as *visuo-spatial ability graph*. It is a directed graph VSA_G:

$$VSA\_G = (V\,(VSA\_G)\,, E\,(VSA\_G)) \tag{4.10}$$

$V\,(VSA\_G)$ is set of vertices representing entities $ET = AG \cup OBJ$ (AG is the set of agents and OBJ is set of objects in the environment as discussed in chapter 3):

$$V\,(VSA\_G) = \{v\,(VSA\_G)\,|v\,(VSA\_G) \in AG \vee v\,(VSA\_G) \in OBJ\} \tag{4.11}$$

$E\,(VSA\_G)$ is set of edges between an ordered pair of agent and object:

$$\begin{aligned} E\,(VSA\_G) = \{e\,(VSA\_G)\,|e\,(VSA\_G) = \langle v_i\,(VSA\_G)\,, v_j\,(VSA\_G)\,, \\ \langle S_{ef}, R_{ef}\rangle\rangle \wedge v_i\,(VSA\_G) \in AG \wedge v_j\,(VSA\_G) \in OBJ\} \end{aligned} \tag{4.12}$$

where S_{ef} is the least feasible effort to see and R_{ef} is the least feasible effort to reach. Hence, each edge in the graph is directed edge from an agent to an object in the environment and shows the effort to see and reach the object. Figure 4.17 shows the visuo-spatial graph of the current state of the environment and it also describes what does an edge revels. The bigger the side of the sphere, greater is the effort. Note that different effort levels to see and reach different object by all the agents have been successfully encoded in the graph.

4.9 Until Now and The Next

In this chapter, we have presented the concept of the *Mightability Analysis*, which stands for "*Might be able to...*". This elevates the perspective taking ability of the robot, which in fact is an essential capability for any social agent, by facilitating to reason about visuo-spatial abilities from multiple achievable states of an agent. We have shown that, such computations could be achieved online. Further, we have equipped the robot to find the least feasible effort to see and reach some object or place and encoded them in a graph. All these will serve as an important component throughout the thesis, such as for planning basic Human Robot Interactive manipulation tasks, in generating shared plans, in learning effort based effect from task demonstration, in deciding where to behave proactively and so on.

In the next chapter we will present the concepts and contributions in terms of analyzing affordance and assessing situation. The Mightability Analysis presented in this chapter will also serve in such analyses.

Chapter 5

Affordance Analysis and Situation Assessment

Contents

5.1 Introduction **87**

5.2 Affordances **87**

5.2.1 Agent-Object Affordances 89

5.2.2 Object-Agent Affordances 90

5.2.3 Agent-Location Affordances 91

5.2.4 Agent-Agent Affordances 91

5.3 Least Feasible Effort for Affordance Analysis **96**

5.4 Situation Assessment **96**

5.4.1 Agent States 97

5.4.2 Object States 103

5.4.3 Attentional Aspects 105

5.5 Until Now and The Next **106**

5.1 Introduction

This chapter will give an overview of the instantiation of the various attributes of the environment presented in chapter 3 related to agents and object status based on their 3D models perceived and updated online. We have enriched the notion of affordance by including inter-agent task performance capability apart from agent-object affordances. Our notion of affordance includes what an agent can do for other agents (give, show...); what an agent can do with an object (take, carry...); what an agent can afford with respect to places (to move-to...); what an object offers (to put-on, to put into, ...) to an agent. Figure 5.1 summarizes the contribution of this chapter.

5.2 Affordances

As mentioned earlier, we have assimilated different notions of affordances as well as added the notion of "what an agent can do for another agent" to develop the

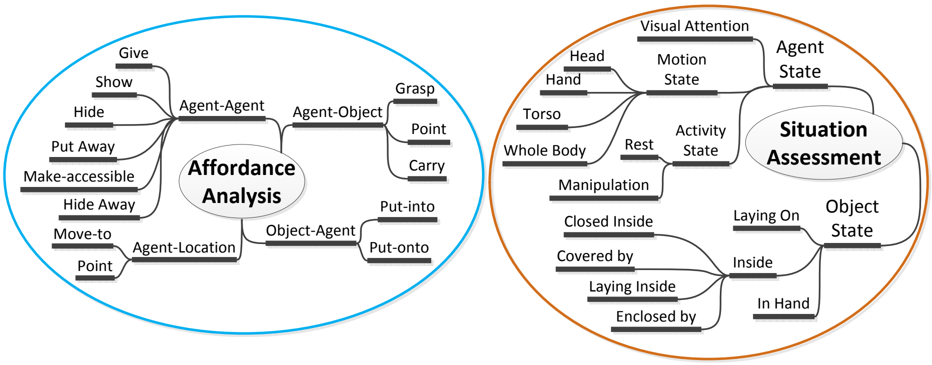


Figure 5.1: Contribution of this chapter in terms of enriched affordance analysis and geometric level situation assessment.

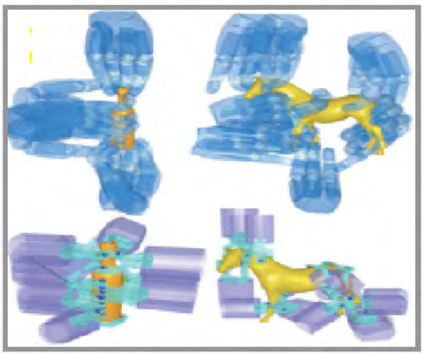


Figure 5.2: Subset of generated grasp set for objects of different shapes, for anthropomorphic hand (top) and for robot's gripper (bottom). (see [Saut 2012])

concept of affordance, as shown in figure 5.1. We conceptualize four categories of affordance analysis from HRI point of view:

(i) ***Agent-Object***: This suggests what an agent could potentially do to an object in a given situation and state.

(ii) ***Object-Agent***: This type of affordance suggests what an object offers to an agent in a given situation.

(iii) ***Agent-Location***: This type of affordance analysis suggests what an agent can afford with respect to a location.

(iv) ***Agent-Agent***: This type of affordance analysis suggests which agent can perform which task for which other agent.

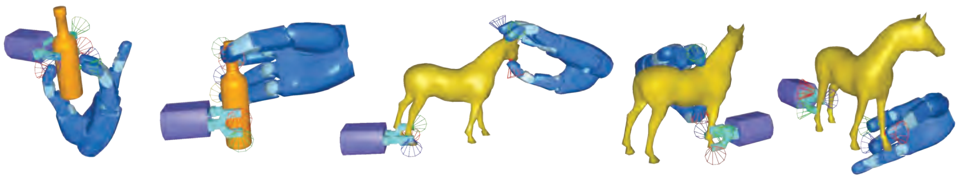


Figure 5.3: Reasoning on the possibilities of the simultaneous grasps of different objects by two agents for the tasks requiring object hand-over.

5.2.1 Agent-Object Affordances

Currently the robot is equipped to find affordance to *Take*, *Point* and *Carry*. We are using a dedicated *grasp* planner, developed in-house (see [Saut 2012]), which could autonomously find sets of possible grasps for 3D object of any shape and rank them based on stability score. Figure 5.2 shows the subset of generated grasps for different objects for the robot's arm gripper and anthropomorphic hand used to test feasibility of grasp by the human. We have used this grasp generation module to equip the robot with reasoning on possibilities to take an object based on situation.

An agent can either take an object that is lying on a support or from the hand of another agent. For the first case of taking an object lying on a support, the existence of collision free grasp for that object is tested. Therefore, existence of at least one collision free grasp, along with the fact that the object is reachable and visible from a given state of the agent, serves as the criteria for the ability to *take* the object lying on the support. For the case where an agent has to take some object from the hand of another agent, we have equipped the robot to reason on the existence of simultaneous grasps by both the agents. As shown in figure 5.3, the robot is able to reason on, for a particular way of grasping an object by the robot, how the human could grasp the object. This ability serves for planning or testing feasibility of the tasks requiring object hand-over. Therefore, the existence of at least one pair of the collision free simultaneous grasp, serves as a criteria for analyzing the *take* object ability from another agent.

Another agent object affordance is to *point* to an object. In the current implementation, an object is said to be point-able by an agent if it is not hidden and not blocked. Something is *blocked* or not is perceived in similar way as done for something is *obstructed* as explained earlier in visuo-spatial perspective taking section 4.3 of chapter 4. The only difference is the test, whether or not the object is within the reach of the agent, is relaxed. An agent can *carry* an object if there exist a collision free grasp and the weight of the object is within acceptable range. Currently the weight information is already provided as the object property.

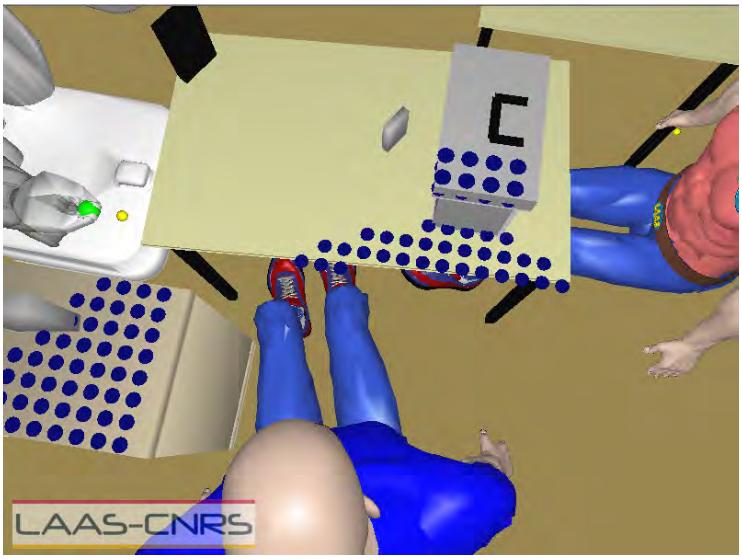


Figure 5.4: Object-Agent Affordance to *Put-onto*: The robot autonomously extracts all the possible supporting objects in the environment. In this scenario, it found that some part of the tabletop as well as top of the box offers the human in the middle to put something *onto* from his current position.

5.2.2 Object-Agent Affordances

We have equipped the robot with the capability to autonomously find the *horizontal supporting facet* and *horizontal open side*, if exist, of any object. For this, the robot extracts planar top by finding the facet having vertical normal vector from the convex hull of the 3D model of the object. The planner top is uniformly sampled into cells and a virtual small cube (currently used of dimension of $(5cmx5cmx5cm)$ is placed at each cell. As the cell already belongs to a horizontal surface and is within the convex hull of the object, so, if the placed cube collides with the object, it is assumed to be a cell of support plane. Otherwise, the cell belongs to an open side of the object from where something could be put inside the object. With this method the robot could find, which object offers to put something onto it and which offers to put something inside as well as which are the places on the object to do these. This reduces the need of explicitly providing the robot with the information about supporting objects such as table or the container objects such as trashbin.

Figure 5.4 shows the automatically extracted places where the human in the middle can put something onto. Note that the robot not only found the table as the support plane, but also the top of the box. Similarly, in figure 5.5 the robot autonomously identified the pink trashbin as a container object having horizontal open facet. And it also found the places from where the human on the right can put something inside this pink trashbin. In these examples, analysis has been done for the human's effort level of *Arm_Effort*. (see section 4.4.1 of chapter 4 for effort hierarchy.)

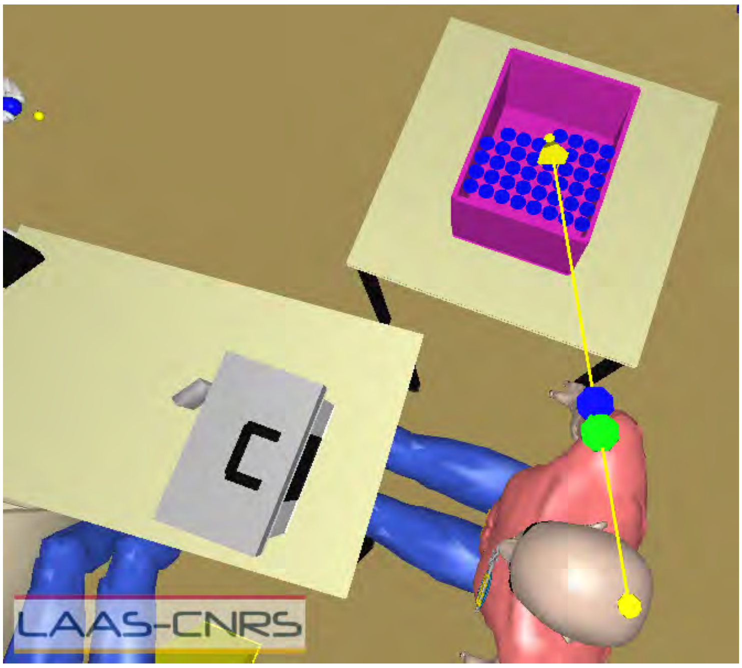


Figure 5.5: Object-Agent Affordance to Put-into: The robot autonomously extracts all the possible container objects having open sides. Hence, it finds that there is a possibility to put something into the trashbin. Further, it finds the places from the human's perspective from where he can put something *into* from his current position.

5.2.3 Agent-Location Affordances

Currently there are two such affordances: can the agent move to a particular location and can the agent point to a particular location. For move-to, the agent is first placed at that location, tested for collision free placement and then existence of a path is tested. For point-to a location, similar approach is used as point to an object, as explained in section 5.2.1.

5.2.4 Agent-Agent Affordances

This aspect of affordance analysis is to *find the feasibility of performing a particular task, T by one agent $Ag1 \in AG$ to another agent $Ag2 \in AG$.* In this context a task T is provided as a tuple:

$$T = \langle name, parameters, constraints \rangle \tag{5.1}$$

Currently the robot is equipped to analyze a set of basic Human-Robot Interactive manipulation tasks denoted as BT.

$$BT = \{Give, Show, Hide, Put\_Away, Make\_Accessible, Hide\_Away\} \tag{5.2}$$

Parameter of a basic task is:

$$parameter = \langle performing\_agent \in AG, target\_agent \in AG, target\_object \in OBJ \rangle \quad (5.3)$$

Performing agent performs the task for a target agent for a target object.

Constraints denoted as $Ctrs$, where $Ctrs = \{c_i | i = 1...n\}$, is a set of expressions c_i, which describes the candidate solution space of the task. Hence, finding the solution space of a task becomes the modified form of constraints satisfaction problem as discussed in section 3.4.2 of chapter 3. For the current discussion, for the agent-agent affordance we restrict candidate space as the places to perform the task, therefore, the set of constraints will be related to the places. However, in chapter 7, where we will present framework to find a feasible executable solution for a task, we will introduce a richer set of constraints. There the candidate space will be the Cartesian product of multiple parameters of the task, such as $place \times grasp \times orientation$. (The set of $Ctrs$ is treated as conjunction of the constraints. However, we do not put restriction on how the actual constraints are specified. We have implemented a basic logical interpreter, which converts the constraints represented in terms of basic logical expressions into logical conjunction.)

For the current discussion each c_i is of the form:

$$c_i = \langle agent, effort, ability, val \in \{true, false\} \rangle \quad (5.4)$$

In the current implementation, for the agent-agent affordance $ability \in \{see, reach\}$ and effort as the element of effort hierarchy presented in section 4.4.1 in chapter 4.

The set of constraints could be provided by the high-level symbolic task planner, such as ours [Alili 2009], or could even be learnt, as we will show in chapter 10 of learning task semantics.

Depending upon the task name, the set of constraints requires to tests for existence of commonly reachable and/or commonly visible places or the places, which are reachable and visible for one agent but invisible and/or unreachable for another agent. For this it uses the *Mightability Maps* of the agents (presented in Mightability Analysis chapter 4) for a given effort level and solves the constraint satisfaction problem by performing set operation on Mightability Maps, to get the following set of candidate points:

$$P_{place}^{obj,Cnts} = \{p_j | p \equiv (x, y, z) \wedge j = 1 \ldots n \wedge (p_j \; holds \; \forall c_i \in \{Cnts\})\} \quad (5.5)$$

n is the number of places.

For example, if the task is to give an object by the robot $R1$ to the human $H1$, the planner knows that the abilities to see and reach the candidate places by the performing and the target agents should be true for the desired effort level. Further assume that the desired effort levels to see and reach the places are set as Arm_Torso_Effort for $H1$. Whereas for $R1$, the desired effort to

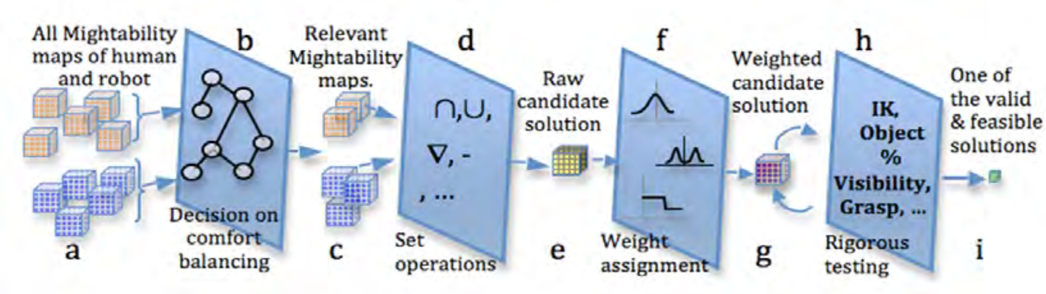


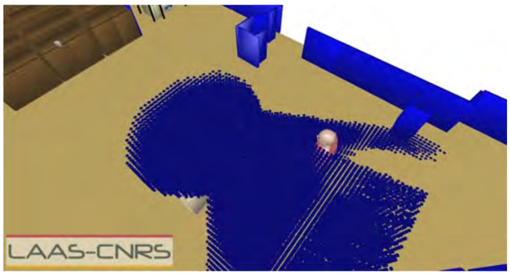
Figure 5.6: Steps for extracting candidate places for Agent-Agent Affordances and further finding a feasible solution if required. (a) Initial Mightability Maps, (b) decision-making on relevant Mightability Maps depending on task and required comfort level of agents, (c) relevant Mightability Maps, (d) task specific set operations, (e) raw candidate solution set, (f) weight assignment based on spatial preferences, (g) set of weighted candidate points, (h) applying rigorous and expensive tests on reduced search space, (i) the feasible solution of highest weight.

see is *Head_Effort* and to reach is *Arm_Effort*. Then the set of constraints will be: $Ctrs = \{c_1, c_2, c_3, c_4\}$, where $c_1 = \langle H1, Arm\_Torso\_Effort, see, true\rangle$, $c_2 = \langle H1, Arm\_Torso\_Effort, reach, true\rangle$, $c_3 = \langle R1, Head\_Effort, see, true\rangle$, $c_4 = \langle R1, Arm\_Effort, reach, true\rangle$.

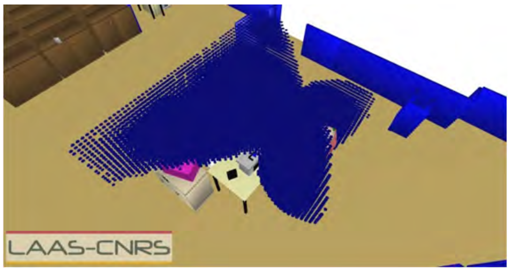
Hence, the robot could find the places for hand-over task, places to put object for hide task, etc. with particular effort levels of the agents. If *obj*, which is the name of the object for which the task is to be performed is not provided, an object of dimension of a cell is assumed. However, if the object is provided, then before finding the candidate places, the corresponding Mightability Maps are grown or shrunken as will be later explained in section 5.2.4.1. If eq. 5.5 results into NULL set, then agent-agent affordance for that task for the given level of effort is not possible. If it is NOT NULL then eq. 5.5 will return the set of candidate places where the task could be performed.

Figure 5.6 shows the main steps of finding the candidate places. Let us assume that the task is to give an object to the human by the PR2 robot, for the initial scenario as shown in figure 5.7(a). From the initial set of all the Mightability Maps for the robot and for the human, the planner extracts the relevant Mightability Maps based on the task and the desired efforts of the agents, in step *b* of figure 5.6. For the current example, maximum desired effort for the human has been assumed to be *Torso_Effort*, i.e. he is willing to lean forward at the most. As the task requires a hand-over operation so the relevant Mightability Maps obtained in step *c* is corresponding to the reach and visibility of both the agents, as shown in figures 5.7(b) and 5.7(c) for the robot and for the human respectively. Then the planner performs set operations in step *d* to obtain the raw candidate points in step *e* of figure 5.6. For the current task, set operation is finding the intersection of reachable and visible places by both the agents. Figure 5.7(d) shows the resultant candidate points

(a)



(b)



(c)

(d)

(e)

Figure 5.7: (a): Initial scenario for finding candidate places for the PR2 robot's affordance for the *give* task to the human. (b)-(e): Illustration of some of the steps for finding the significantly reduced candidate weighted search space for this task.

Figure 5.8: HRP2-Human face-to-face scenario for performing basic human robot interactive tasks.

Initial Total Number of Cells in Workspace = **144000**	
Task (HRP2 for Human)	**Significantly Reduced Search Space (final number of candidate cells)**
Make bottle accessible	8
	33
Show bottle	414
Hide bottle	42

Figure 5.9: Significant reduction in candidate search space for performing a set of tasks in the scenario of figure 5.8

obtained in step e of figure 5.6, which in fact is commonly reachable and visible by both the agents, for the given effort levels. Further, based on various criteria such as comfort, preferences, etc. weights are assigned to the raw candidate points in step f of figure 5.6 to obtain weighted candidate points in step g. Figure 5.7(e) shows the weighted candidate points, red cells are least preferable and the green cells are most preferable. In fact, eq. 5.5 returns this candidate point cloud. Then depending upon the task and constraints, various other tests could be performed in this space to find a feasible solution for basic human robot interactive tasks, which will be presented in chapter 7. However, at this point it is interesting to note that the search space has been significantly reduced as compared to entire workspace, for performing expensive feasibility tests. Table in figure 5.9 shows the significant reduction in search space for a variety of tasks by HRP2 robot for the human in the initial scenario shown in figure 5.8.

In step h of figure 5.6, each candidate cell is iteratively tested for feasibility in the order of highest to lowest weight until a solution is found. For finding a feasible solution, various task dependent constraints are introduced. Such tests would have been very expensive if done for entire workspace.

For the sake of maintaining the agent-agent affordances online, we avoid performing expensive tests in the last block until planning to actually perform the task. This last block will be explained in detail in chapter 7. We stop at the step of weight

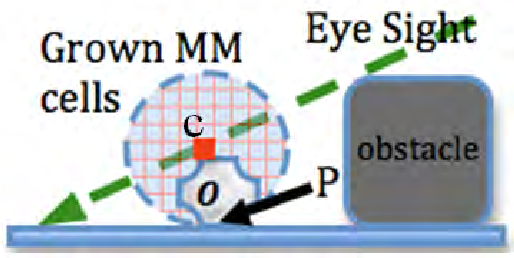


Figure 5.10: Growing Mightability Maps based on object's dimension

assignment to get a set of weighted candidate places to perform the task.

5.2.4.1 Considering Object Dimension

The candidate place obtained earlier could be shrunken or grown, depending upon the nature of the task (cooperative: give, show,...) or competitive (Hide, put-away,...) if the object is known. For example, for cooperative tasks the robot grows the corresponding Mightability Maps by a sphere of radius $2 \times l$, where l is the longest dimension of the bounding box of the object. This avoids the ruling out of the places from where the object will be partially visible or reachable. Figure 5.10 shows one cell c belonging to the original visibility Mightability Map of the agent, which has been expanded for object O. Now the position P is the part of grown Mightability Map, hence, the robot could find P as valid position where if the object O would be placed, agent can partially see it, even if P is not directly visible to the agent. Similarly, it facilitates to find the positions to hand-over an object even if there is no commonly reachable place.

5.3 Least Feasible Effort for Affordance Analysis

Similar to visuo-spatial perspective taking which could be done for different effort levels, the affordance analysis is also done for different effort levels. As for a given scenario the robot is able to find the multi-effort affordance (give, take, pick, show, ...), so from these efforts it can then extract the least feasible effort.

5.4 Situation Assessment

In this section, we will identify those aspects of situation assessment, which serve as key for developing a smooth and better decision-making capabilities for HRI. The concepts and the system developed in this section are in fact serving to our high-level planner HATP [Alili 2009] as well as to our high-level robot supervision system SHARY [Clodic 2009] for plan execution and monitoring.

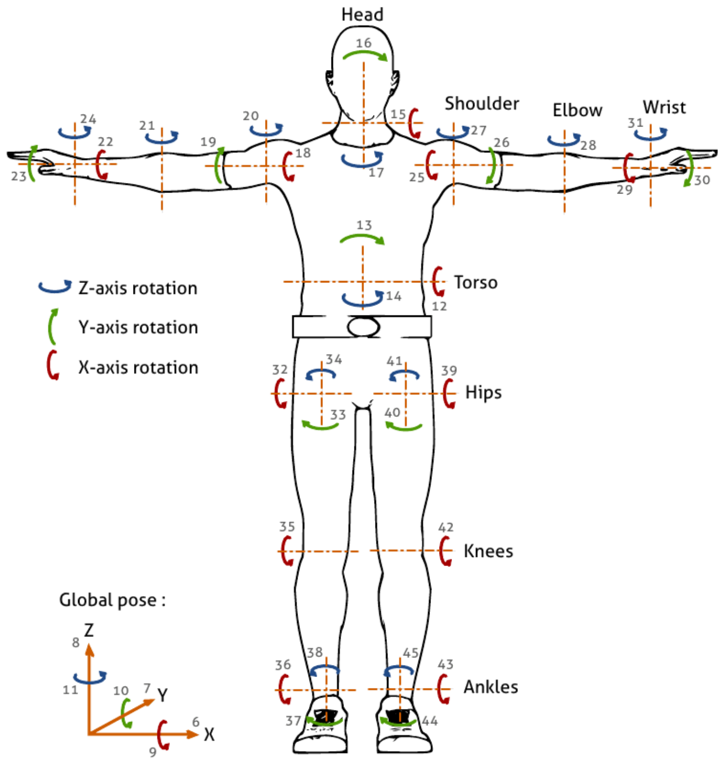


Figure 5.11: Joints of the 3D human model in our 3D representation and planning platform Move3D [Simeon 2001]. (Drawing courtesy to Séverin Lemaignan, LAAS-CNRS)

5.4.1 Agent States

We have equipped the robot to infer a set of facts related to the state of an agent and the states of various body parts of the agent. This analysis is done on rich 3D model of the human and the environment. Figure 5.11 shows the joints of the human model used in our 3D representation and planning platform Move3D [Simeon 2001]. This model of the human, and the corresponding models of other agents are updated online through various sensors of the robot. See appendix A for detail. By analyzing the values of the joints, various facts about the agent states are inferred in real time. Based on the requirement of our HRI domain, currently the following facts are calculated (see eq. 3.13 - 3.22):

$$Posture = \{Standing, Sitting\}$$
$$Hand\_Occupancy = \{Free\_Of\_Object\} \cup \{\langle Holding\_Object, \{Object\_Names\}\rangle\}$$
$$Hand\_Mode = \{\langle Rest\_Mode, Rest\_Mode\_type\rangle\} \cup \{Manipulation\_Mode\}$$
$$Rest\_Mode\_type = \{Rest\_by\_Posture\} \cup \{\langle Rest\_on\_Support, Support\_Name\rangle\}$$
$$Body\_Part = \{whole\_body, torso, head, right\_hand, left\_hand\}$$
$$\forall bp \in Body\_Part\ Motion\_Status^{bp} = \{not\_moving, moving, turning\}$$

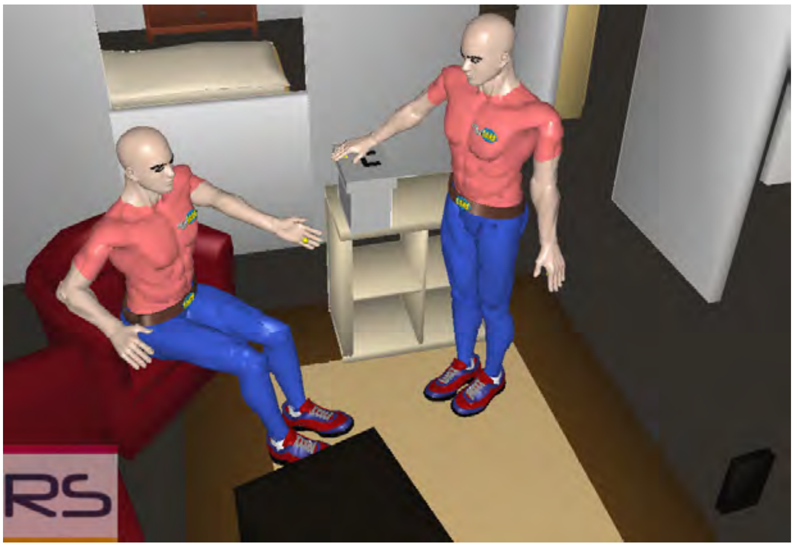


Figure 5.12: Two agents in the environment with different postures and modes of the hands. The system autonomously finds out that the posture of the human on the left is *sitting* and that of the human on the right is *standing*. Further, it returns the facts about the agents' hand state: For the left human sitting on the sofa: $\langle Righ\_Hand, \langle Rest\_On\_Support, Sofa\rangle\rangle$, $\langle Left\_Hand, Manipulation\_Mode\rangle$; for the right human: $\langle Righ\_Hand, \langle Rest\_On\_Support, Box\rangle\rangle$, $\langle Left\_Hand, Rest\_by\_Posture\rangle$.

For finding the posture of the agent, based on the values of the hip joints (joint 32 & 39) and the knee joints (joint 35 & 42), an agent is said to be sitting or standing. We found a set of thresholds of such joints based on a reference sitting position, similar to one of the human on the left in figure 5.12. Hence, the left human in figure 5.12 is detected by the system to be sitting and the right is autonomously detected to be standing. We classified *occupancy status* of a hand of the agent into *Free_ Of_ Object* or *Holding_ Object*. This is also found by analyzing the 3D model of the world. If any object *Obj* is within a threshold distance from any of the hand, (this threshold is very small ($\sim 2\ cm$) and tried to incorporate sensor noise) or there is a collision detected between an object *obj* and the hand, the object is said to be contact with the hand. Currently, we assume that the object in contact is the object being hold by the hand, which turns out to be sufficient and fast enough for our HRI experiments. If there is no object in contact, hand is said to be free of object.

An agent's hand is said to be in *rest mode* if (i) either the arm is straight downward as we stand or sit, (ii) or its relative position and orientation are not changing with respect to the body frame, and it is found to be in contact with some object obj, and obj is in contact with some other supporting object $obj2$ or the ground. A hand is in *manipulation mode*, if it is not in the *rest mode* within some threshold. Further, a hand can be in manipulation mode with holding or carrying some object, or without some object (e.g. waiting for someone to give something, pointing to something, part

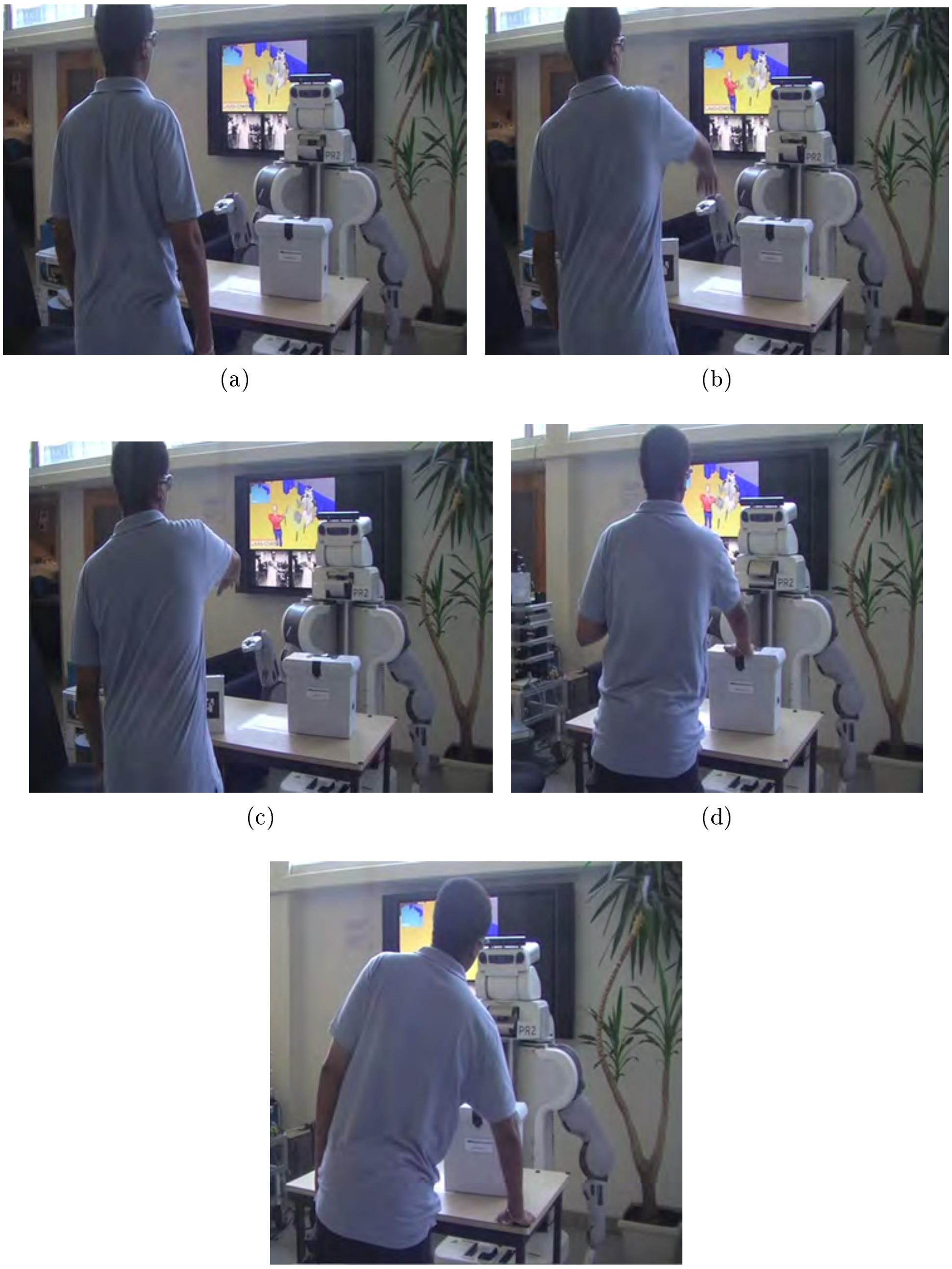
(a) Categorization of hand mode in different sitting postures of an agent. Left posture: hand in *rest mode*, rest mode type: *by posture*. Middle three postures: hand in *rest mode*, rest mode type: *by support*, because the hand is lying on a support, armrest, table, lap. Right most posture: hand in *manipulation mode*.

(b) Categorization of hand mode in different standing postures of an agent. Left posture: hand in *rest mode*, rest mode type: *by posture*. Middle posture: hand in *rest mode*, rest mode type: *by support*, because the hand is lying on a table. The same posture will be categorized as manipulation mode if it would have been without any support as in the right most figure. Right posture: hand in *manipulation mode*.

Figure 5.13: A subset of different postures of an agent, which we have equipped the robot to infer. For illustration, hand is drawn in green. Classification of hand mode into *in rest* and *in manipulation*. Such classification is required for a variety of purpose, such as to focus the attention at the hand, which is in manipulation mode and might be trying to point, give or take something.

of some gesture, etc.). Figure 5.13 shows a subset of rest and manipulation modes of the hand, which our system is currently able to infer by analyzing the 3D model of the world. See the figure's caption for the detail.

Following is the output of the hand modes of both the agents of the figure 5.12: For the left human sitting on the sofa: $\langle Righ\_Hand, \langle Rest\_On\_Support, Sofa\rangle\rangle$, $\langle Left\_Hand, Manipulation\_Mode\rangle$. For the right standing human:



(a) (b) (c) (d) (e)

Figure 5.14: Online hand mode analysis for an agent's action. The key facts generated by the system related to the right hand of the agent during the course of action are: (a) Hand in *Rest mode*, rest mode type: *by posture*, (b) hand *Moving*, (c) hand in *Manipulation mode*, hand *free of object*, (d) hand in *Rest mode*, rest mode type: *by support*, support name: *Box*, (e) hand in *Rest mode*, rest mode type: *by support*, support name: *Table*.

(a) (b)

Figure 5.15: Online hand state and mode analyses for another agent's action. The key facts generated by the system related to the left hand of the agent during the course of action are: (a) Hand in *Rest mode*, rest mode type: *by support*, support name: *Human*, (b) hand in *Manipulation mode*, hand *holding object Grey_ Tape*.

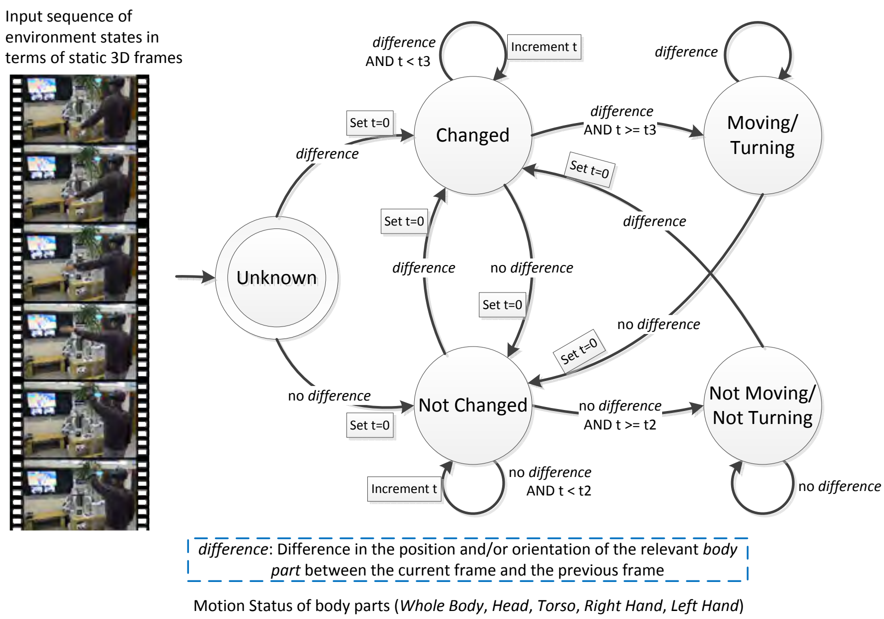


Figure 5.16: State transition diagram for agent and agent's body parts' motion status analyses. The similar transition diagram is used for different body parts.

$\langle Righ\_Hand, \langle Rest\_On\_Support, Box\rangle\rangle$, $\langle Left\_Hand, Rest\_by\_Posture\rangle$.

As, the calculations are online, figure 5.14 and figure 5.15 show updating of the facts as the humans' hand move. See the captions of the figure for the description.

Further, from the robot supervision point of view, such as [Clodic 2009], it is important to detect whether the agent's hand is moving (perhaps carrying something, perhaps required to track, etc.), static (perhaps pointing something, perhaps waiting to hand-over something, etc.) or just the position has changed from the previously observed one; whether the human head is turning (perhaps looking around, searching for something, etc.) or static (looking at something, etc.) of just changed from the previous observed orientation (indicating some change in human's belief, knowledge, etc.). All such pieces of information are required to monitor the human activity and to take decision related to execution and/or re-planning of actions, such as when to give something, where to look, when to suspend the execution of current plan and request to re-plan for the task because of change in human's attention, commitment, etc.

We have implemented a state machine based on geometric information of the world to provide as the basic tool to facilitate such reasoning. This provides geometric level inference about whether some part of the body is moving and/or turning or not. As practically, the 3D representation of the world is updated at a particular frequency ($\sim 5-10$ frames/sec) based on the input from various sensors, the problem is to perceive *motion* from a series of static images (snapshot of the 3D world model) with time stamps. Further, we want to distinguish the notion that something has *changed* only, from the notion of something is *moving/turning*. Therefore, our state transition diagram is based on the logic: *continuous changes suggest motion* and *continuous non-changes suggest stationary*. Figure 5.16 shows a general state transition system used for any body parts or for the whole body. It is clear from the diagram that the system avoids to conclude whether something is moving/turning, until it observes a series of changes in its position/orientation for some time $t3$. However, it can figure out starting from the second image itself if the position/orientation of something has changed. Similarly, the system avoids to conclude whether something is static, until it observes a series of non-changes in its position/orientation for some time $t2$. The *change* is found geometrically by analyzing whether the difference between the current value and the previous value is beyond a threshold (to incorporate sensors' noise) or not. Note that the system based on this state transition diagram serves for the basic practical requirement to distinguish something is moving from the cases when only the position or orientation of something has changed. Further, it distinguishes that something is static (not-moving) from the cases when the position or orientation of something has not changed only in previous couple of frames. By setting the values of $t2$, $t3$, which we term as *assurance window*, we can change the threshold of how much to wait before asserting about something is moving or static.

Such rich knowledge about the agent's hand state, hand's mode, body and body part motion status, altogether facilitate the supervisor SHARY [Clodic 2009] with various online and on time decision-making processes including re-planning and engagement. Further, it could be used in understanding task semantics and execution from demonstration, which will be discussed in chapter 10.

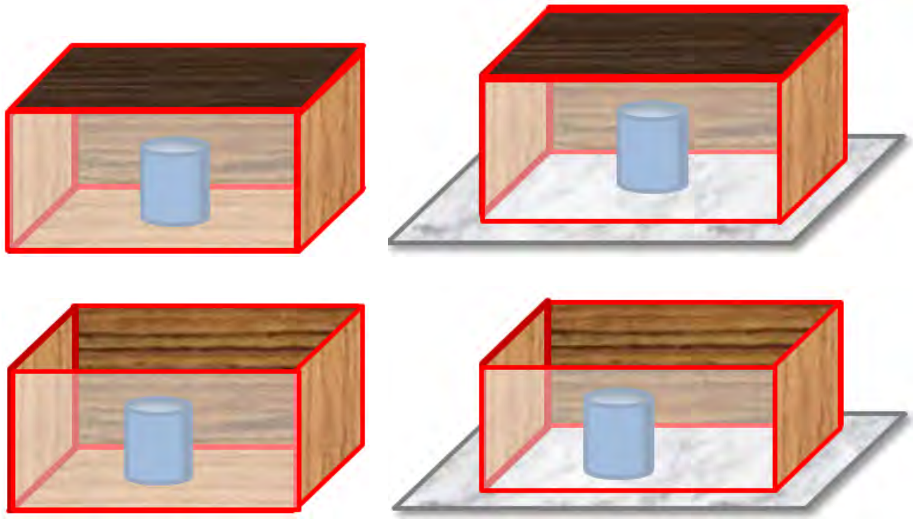


Figure 5.17: Subcategory of "inside" relation: blue cylinder is (a) closed inside; (b) covered by; (c) lying inside; (d) enclosed by; the box. This enables to the robot to explicitly reason on different effects on the object, which is 'inside' if the container object (the box) will be manipulated.

5.4.2 Object States

We have equipped our robots with a 'meaningful understanding' of the scenario. Based on reach 3D model of the objects in the environment, the robot is able to distinguish among the situations where an object is:

- inside
  - closed inside
  - covered by
  - lying inside
  - enclosed by
- lying on a support «support name»
- floating in air

For finding some object *obj1* is inside some container object *obj2*, all the vertices of the convex hull of *obj1* is checked to be inside the convex hull of *obj2*. Further, we have sub-categorized "inside" in four different situation, figure 5.17. If from all directions the *obj1* is surrounded only by the walls of *obj2*, *obj1* is said to be closed inside *obj2*, figure 5.17(a). An object *obj1* is said to be covered by another objects *obj2*, if *obj1* is lying on a support plane, which does not belong to *obj2*, as shown in figure 5.17(b). An object *obj1* is said to be lying inside if it is surrounded by the walls of *obj2* except one face, and it is supported on the one of the facet of *obj2*, figure 5.17(c). If the *obj1* is not supported by any of the facet of *obj2* and also there is an open side of obj2, obj1 is said to be enclosed by obj2, as shown in

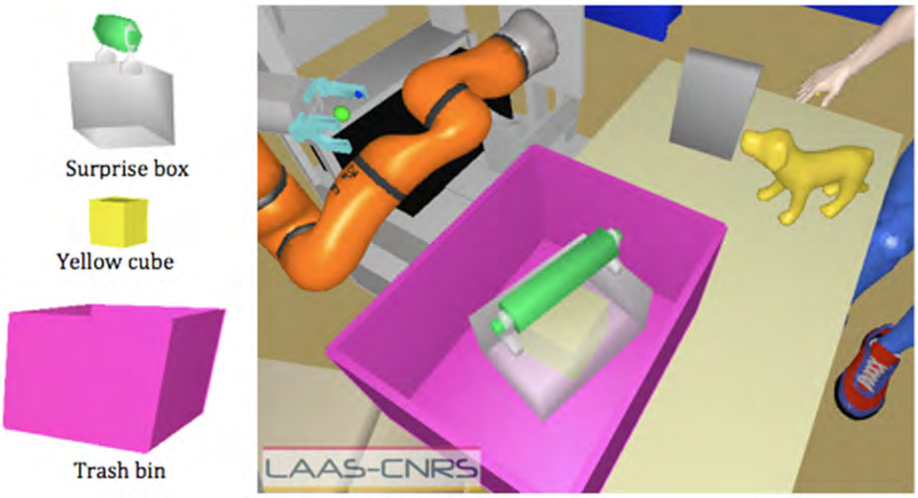


Figure 5.18: A scenario to demonstrate inter-object spatial situation assessment.

figure 5.17(d).

In fact, the motivation behind this categorization is to provide the robot with explicit understanding about what will be effect of manipulating the container object, *obj2*, on *obj1*, which is found to be *inside obj2*. If *obj1* is *covered by* a container object, *obj2*, lifting *obj2* will not move *obj1* but might change the visibility or reachability of *obj1* from some agent's perspective. In case of *obj1* is *closed inside obj2*, manipulating *obj2* will also move *obj1*. Further, in both cases, without manipulating *obj2*, one cannot physically act upon *obj1*. In case of *obj1* is *lying inside obj2*, manipulating *obj2* will affect *obj1* global position, but *obj1* could also be manipulated without physically acting upon *obj2*. In case of *obj1* is just *enclosed by obj2*, there are possibilities to manipulate both the objects independently.

Out approach to geometrically categorize whether *obj1*, which has been already found to be inside *obj2*, is covered by, closed inside, lying inside or enclosed by, *obj2* is as follows: First *obj1* is virtually moved up and down along vertical. Let us assume that while moving down the first collision is detected with *obj3*, whereas while moving up the first collision is detected with *obj4*. If $obj2 = obj3 = obj4$, then *obj1* is said to be closed inside *obj2*. If $obj2 \neq obj3$ but $obj2 = obj4$, then *obj1* is said to be covered by *obj2*. If $obj2 = obj3$ and $obj4 = NULL$, then *obj1* is said to be lying inside. If $obj2 \neq obj3$ and $obj4 = NULL$, then *obj1* is just said to be enclosed by *obj2*. Below we present the partial output of robot's understanding of the scenario of figure 5.18:

- Yellow cube is *covered by* Surprise box
- Yellow cube is *lying on* support: Trash bin
- Yellow cube is *lying inside* Trash bin
- Surprise box is *lying on* support: Trash bin

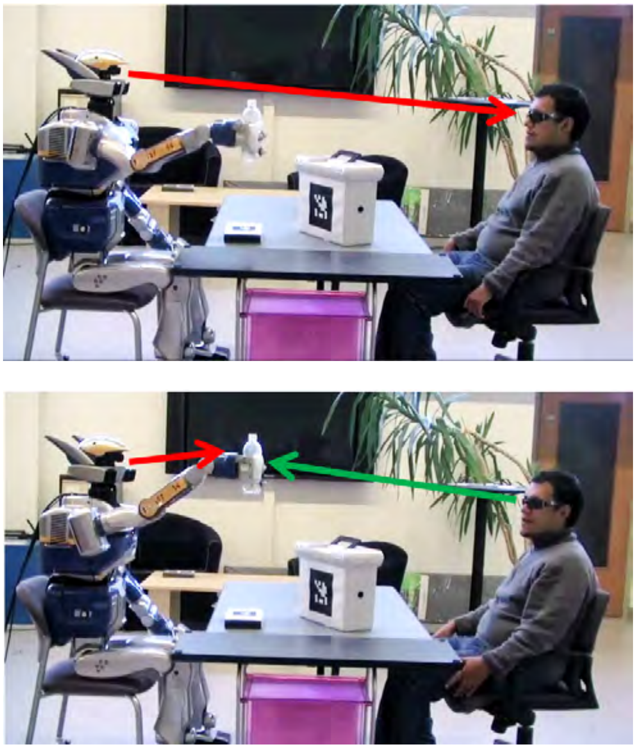


Figure 5.19: The HRP2 robot fetches the human partner's attention in the task of holding and showing an object to the human. (a) While performing the task, the robot first looks at human to *engage* him, then (b) at the object to *draw* his attention.

- Surprise box is *lying inside* Trash bin
- Trash bin is *lying on* support: Table
- Toy Dog is *lying on* support: Table
- Grey Tape is *lying on* support: Table

Hence, the robot is able to explicitly understand that yellow cube is *covered by* surprise box.

5.4.3 Attentional Aspects

Based on situation assessment and geometric reasoning, we have equipped the robot to show following basic attentional behaviors for any human-robot interactive scenario:

- Share Attention: Look at, where the human is looking.
- Fetch Attention: Look at agent to engage him/her then look at object or place of interest to draw his/her attention.
- Focus attention: Look at the human's hand if it is in Manipulation State.

As mentioned earlier these attentional components are based on rich geometric reasoning and aimed to facilitate 'natural' and 'informing' human-robot interaction. This is complementary to higher level reasoning on attention based on saliency, [Ruesch 2008], or curiosity [Luciw 2011] or intrinsic motivation [Oudeyer 2007]. Currently these components are used as requests with the desired parameters in various human-robot interactive scenarios by the robot supervisor module SHARY [Clodic 2009] as well as throughout various experiments in this thesis. For example, fetching attention while showing some object by holding it, (chapter 7), and proactively suggesting a place to put something (chapter 9). Figure 5.19 demonstrates the robot's attempt of fetching the attention of the human while performing the task of showing an object by grasping and holding it.

5.5 Until Now and The Next

In this chapter, we have presented the approaches to realize some of important attributes and facts of the generalized HRI domain presented in chapter 3. We took this opportunity to identify different types of affordances and introduce the concept of agent-agent affordance and a framework to analyze that. We have shown the practical results of obtaining these facts in real environment. In our architecture, these facts also serve as input to various other high-level decision-making modules and planning modules developed by other contributors in our group, such as our robot supervisor SHARY, high-level task planner HATP, ontology based knowledge management system ORO and so on. See appendix A for an overview of the overall system contributing to LAAS robot architecture.

Until now, we have achieved the realization of the basic blocks of key-cognitive level presented in our social intelligence embodiment pyramid of figure 1.1 along with some new concepts from HRI perspective such as Mightability Analysis, Agent-Agent Affordance and so on, as summarized in figure 2.1. Equipped with such key cognitive aspects, now we are ready to use them and move a level up in the pyramid to realize some of the key behavioral aspects. We will begin this by first presenting in the next chapter, frameworks for the navigation aspects incorporating human-aware and social constraints, which will be followed by the manipulation aspects in the subsequent chapter.

Chapter 6

Socially Aware Navigation and Guiding in the Human Environment

Contents

6.1 Introduction . . . 108

6.2 Socially-Aware Path Planner . . . 109

6.2.1 Extracting Environment Structure . . . 109

6.2.2 Set of Different Rules . . . 111

6.2.3 Selective Adaptation of Rules . . . 113

6.2.4 Construction of Conflict Avoidance Decision Tree . . . 114

6.2.5 Dealing with Dynamic Human . . . 116

6.2.6 Dealing with Previously Unknown Obstacles . . . 116

6.2.7 Dealing with a Group of People . . . 117

6.2.8 Framework to Generate Smooth Socially-Aware Path . . . 117

6.2.9 Proof of Convergence . . . 122

6.3 Experimental Results and Analysis . . . 122

6.3.1 Comparative analysis of *Voronoi Path* vs. *Socially-Aware Path* vs. *Shortest Path* . . . 122

6.3.2 Analyzing Passing By, Over Taking and Conflict Avoiding Behaviors . . . 123

6.3.3 Qualitative and Quantitative Analyses of Generated Social Navigation with Purely Reactive Navigation Behaviors . . . 129

6.4 Social Robot Guide . . . 131

6.4.1 Regions around the Human . . . 132

6.4.2 Non-Leave-Taking Human Activities . . . 133

6.4.3 Belief about the Human's Joint Commitment . . . 133

6.4.4 Avoiding Over-Reactive Behavior . . . 134

6.4.5 Leave-Taking Human Activity . . . 135

6.4.6 Goal Oriented Re-engagement Effort . . . 135

6.4.7 Human Activity to be Re-engaged . . . 138

6.4.8 Searching for the Human . . . 140

6.4.9 Breaking the Guiding Process . . . 141

6.5 Experimental Results and Analysis . . . 141

6.6 Until Now and The Next 145

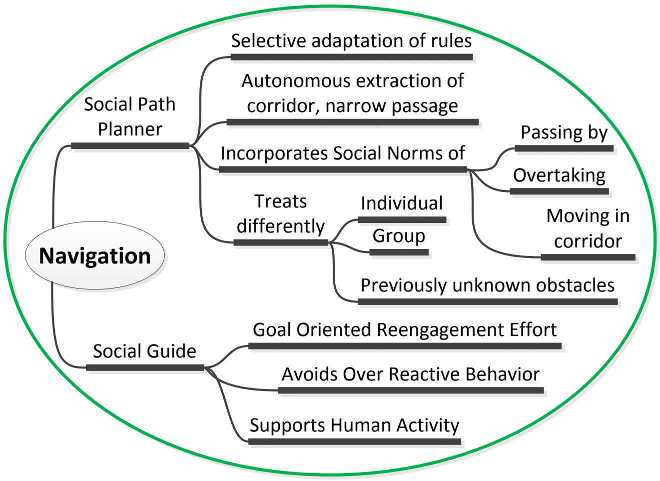


Figure 6.1: Contribution of this chapter, in terms of development of a socially-aware path planner and a social robot guide framework.

6.1 Introduction

In the context of Human-Robot Co-existence with a better harmony, it is necessary that the human should no longer be on the compromising side. The robot should 'equally' be responsible for any compromise, whether it is to sacrifice the shortest path to respect social norms or to negotiate the social norms for physical comfort of the person or to provide the human with the latitude in the way he/she wants to be guided. As discussed in section 1.1.2, it has been proved that social bias to pass a person from a particular side, or to move in a lane like manner in corridor are essential for reducing conflicts, confusion and failed attempts in avoidance behavior. Further, as discussed in section 2.3, from the robot navigation point of view the social norms and reasoning about the spaces around the human should be reflected in the robot's motion. Moreover, as discussed in section 1.1.2, an agent motion exerts different kinds of so-called non-physical *social forces*: attractive and repulsive, which in turn could be used to push, pull or attract other person.

In this chapter, we will develop a framework, which takes into account various social norms of moving around and plans a smooth path by selective adaptations of rules depending upon the dynamics and structure of the local environment. Further, we will present a framework, which takes into account natural deviation of the human to be guided by the robot, and avoid showing unnecessary reactive behaviors. And in the case the human suspends the joint task of guiding, the robot tries to approach him/her in a goal directed manner, to exert a kind of social force to re-engage him/her towards the goal. The contribution of this chapter has been summarized in figure 6.1.

The framework presented in this chapter basically plans/re-plans a smooth path by interpolating through a set of milestones (the points through which the robot must pass). The key of the framework is the provision of adding, deleting or modifying the milestones based on static and dynamic parts of the environment, the presence and the motion of an individual or group as well as various social conventions. It also provides the robot with the capability of higher level reasoning about its motion behavior.

6.2 Socially-Aware Path Planner

The goal of this section is to develop a mobile robot navigation system which: (i) autonomously extracts the relevant information about the global structure and the local clearance of the environment from the path planning point of view, (ii) dynamically decides upon the selection of the social conventions and other rules, which needs to be included at the time of planning and execution in different sections of the environment, (iii) re-plans a smooth deviated path by respecting social conventions and other constraints, (iv) treats an individual, a group of people and a dynamic or previously unknown obstacle differently.

Next sections will describe our approach to extract the path planning oriented environment information. Then the set of social conventions, proximity guidelines and the clearance constraints will be described. Subsequently the selective adaptation of rules and their encoding in a decision tree will be discussed. Then the strategies for dealing with the humans and previously unknown obstacles will be followed by our algorithm to produce the smooth path.

6.2.1 Extracting Environment Structure

One of the important aspects of autonomous navigation oriented decision-making is to know the local clearance in the environment like door, narrow passage, corridor, etc. In our current implementation, we are using *Voronoi diagram*, which has been shown to be useful by us [Van Zwynsvoorde 2001] and by others [Friedman 2007], [Thrun 1998], for capturing the skeleton of the environment. For this we define the followings:

- **Voronoi Diagram**: Since we are constructing the Voronoi diagram at discrete level of grid cells, we define it as the set of cells in the free space that have at least two different equidistant cells in the occupied space. Figure 6.2 shows

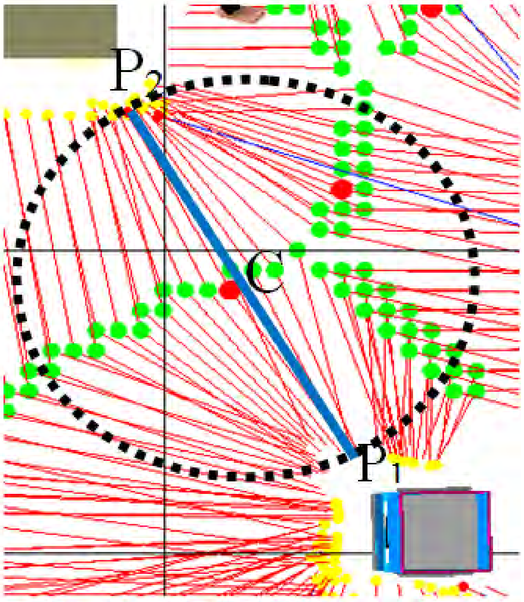


Figure 6.2: Voronoi Diagram based environment clearance analysis. Interesting cell (IC) C and Interesting Boundary Line (IBL) P_1P_2.

different Voronoi cells (green circles) and the red lines connecting them to the corresponding nearest occupied cells.

- **Interesting Cell (IC)**: We define the term *'Interesting Cell' (IC)* as the Voronoi Cell: (a) which is equidistant from exactly two cells in the occupied space and, (b) both the equidistant points are on the opposite sides on the diameter of the circle centered at that Voronoi cell. In figure 6.2, the Voronoi cell C is such as $\angle P_1CP_2 \approx 180\,degrees$, hence, it is an *IC*.
- **Interesting Boundary Line (IBL)**: We name the line joining both the equidistant points of IC as the *'Interesting Boundary Lines' (IBL)*, P_1P_2.
- **Local Clearance**: The length of the *IBL* will be the 'clearance' of that local region, in the absence of any dynamic obstacle and human. Later on we will show that based on the presence of any human or previously unknown obstacles, the planner modifies this information dynamically.

By setting a threshold on this clearance, the robot decides whether it is a *narrow passage* or *wide region*. Figure 6.3 shows the local clearance of a part of the map of our lab, captured by this approach. The thin blue line with a red circle at the middle shows one *IBL*. Note that, as shown in figure 6.3, in case of corridor or long but narrow passage, we will get a set of approximately parallel *IBLs*.

Hence, the robot has a clearance and topological information of the environment in terms of door, corridor, narrow passage, wide region, etc. Below we will identify different set of rules, which should be incorporated based on this information as well as by the presence of the human in the environment.

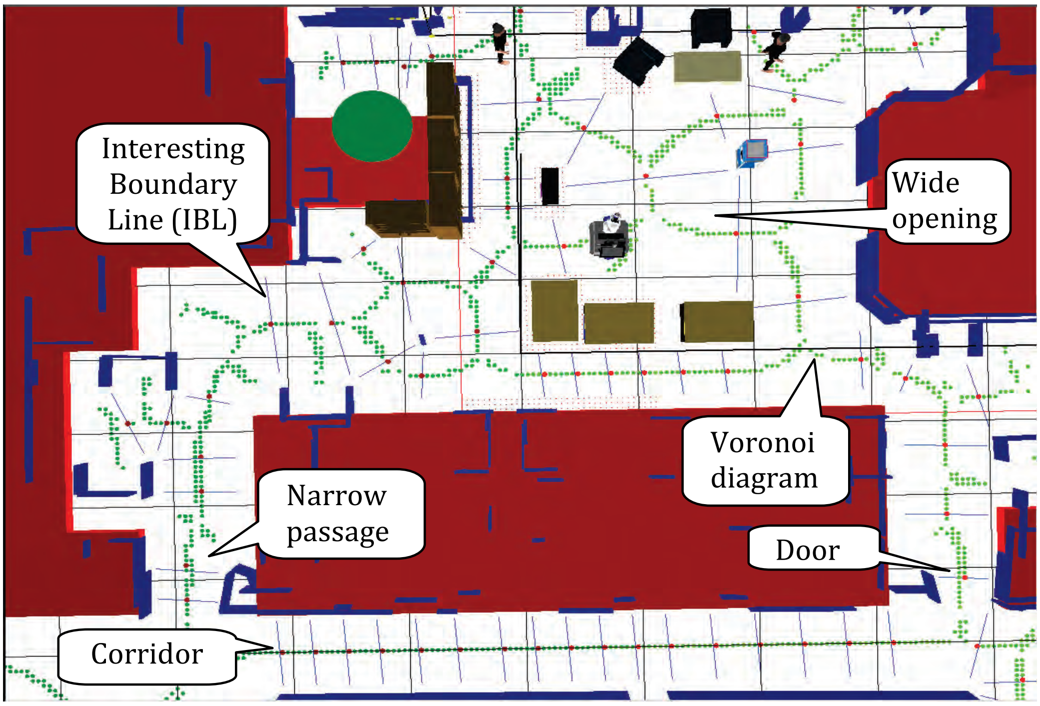


Figure 6.3: Voronoi Diagram based capturing local clearance of the part of LAAS Robotics Lab environment. The thin blue line with a red circle at the middle shows one *Interesting Boundary Line (IBL)*. In the regions of a corridor or a long but narrow passage, we get a set of approximately parallel *IBLs*.

6.2.2 Set of Different Rules

Based on the norms of the human navigation to avoid conflict and confusion as discussed earlier, in the current implementation, we chose to incorporate following set of rules:

6.2.2.1 General Social Conventions (S-rules)

- (S.1) Maintain right-half portion in a narrow passage like hallway, door or pedestrian path.
- (S.2) Pass by a person from his left side.
- (S.3) Overtake a person, from his left side.
- (S.4) Avoid very close sudden appearance from behind a wall.

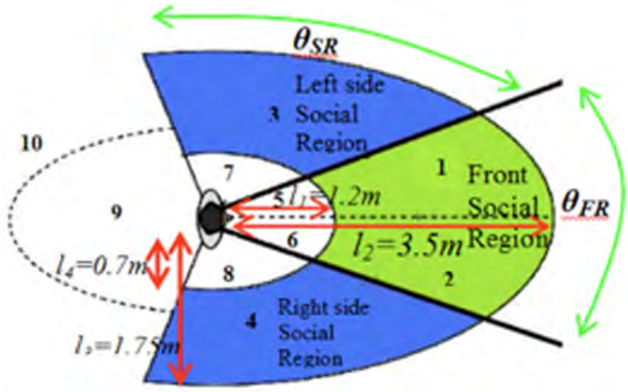


Figure 6.4: Construction of regions around a human, based on proximity and relative position with respect to the human's front.

6.2.2.2 General Proximity Guidelines (P-rules)

From the point of view of safety and physical comfort, the robot should always maintain an appropriate distance from the human. Given that proxemics plays an important role in Human-Human interaction, proxemics literatures [Hall 1966] typically divide the space around a person into 4 zones:

(i) Intimate
(ii) Personal
(iii) Social
(iv) Public

Several user studies and experiments [Pacchierotti 2005], [Yoda 1997] have been conducted, to establish and/or verify these spatial distance zones from the viewpoint of Human-Robot interaction. Their results comply with the hypothesized minimum social distance of *1.2 m* and maximum social distance of *3.5 m* in front of a person for a typical human sized robot. Whereas the lateral passing distance of more than *0.7 m* from the side of the person, makes him feel physically comfortable, where the range of the human and the robot speeds are *1 m/s* to *1.5 m/s* and *0.5 m/s* to *1 m/s*. Based on analysis of the results from such user studies we construct a set of parameterized semi-elliptical regions around the human as shown in figure 6.4. Note that the angular spread of the accompanying span is slightly beyond *90 degrees* from the human axis on both sides. This is because sometimes even as an accompanying person, the human may want to move slightly ahead of the robot. Although these distance values will serve as reference in our current implementation for the speed range of *0.5 m/s* to *1 m/s* for the human and the robot, one should not consider them as fixed. Studies suggest, these parameters vary from children to adult, context and the task [Yamaoka 2008], and depend upon environment, agent's speed and size, and even with the personality of the person [Walters 2005]. Hence, we have implemented our framework so that these values are parameter to the planner and the robot could adjust them online if required, depending upon the situations.

The set of proximity rules, which we are presently using, are:

- (P.1) Do not enter into intimate space until physical interaction is needed.
- (P.2) Avoid entering into personal space if no interaction with the human is required.
- (P.3) Avoid crossing over the person if the robot is already within the outer boundary of side-social regions numbered as 3 and 4 in figure 6.4, instead pass by the human from his nearest side.

One can notice that in some situations rule *(P.3)* can cause conflict with the social rule *(S.2)*, but we choose *(P.3)* to dominate because the robot will be in close proximity of the human. Rules *(P.1)* and *(P.2)* also serve another purpose of ensuring physical safety of the human.

6.2.2.3 General Clearance Constraints (C-rules)

The clearance analysis takes care of spacious sufficiency to compromise with other types of rules. The set of clearance rules used are:

- (C.1) Avoid passing through a region around the human if it has a clearance less than *d1*.
- (C.2) Maintain a minimum distance *d2* from the walls and obstacles.
- (C.3) Do not pass through an *Interesting Boundary Line (IBL)*, if its length is less than *d3*.

Currently the values of *d1*, *d2*, and *d3* depend upon the robot's size only.

We will use the term *milestone*, as a point through which the path of the robot must pass. Our framework performs one of the following actions for each of the rules mentioned above:

(i) Inserts a new set of milestones in the list of existing milestones.
(ii) Modifies the positions of a subset of existing milestones.
(iii) Verifies whether a particular rule is being satisfied on the existing set of milestones or not.

6.2.3 Selective Adaptation of Rules

From the path-planning point of view, we will globally divide the rules into two categories: (i) Those that need to be included at the time of initial planning, taking into account the static obstacles and structure of the environment. (ii) And those that will be included at the time of path execution as the humans or unknown obstacles will be encountered. *S-rules (S.1)* & *(S.4)* and *C-rules (C.2)* & *(C.3)* fall into first category. Rules *(S.1)* & *(S.4)* are due to the obvious reasons to avoid conflicting situation in narrow passages as well as to avoid collision and the feelings of surprise or fear in the human. Similarly, *(C.2)* & *(C.3)* are to avoid moving very close to obstacle or being stuck in a too narrow passage. Other rules fall into the second category.

This selective adaptation of rules is an attempt to balance the tradeoffs between the path that minimizes the time of flight and the path that avoids conflicting, reactive and confusing situations in a human-centered environment.

6.2.4 Construction of Conflict Avoidance Decision Tree

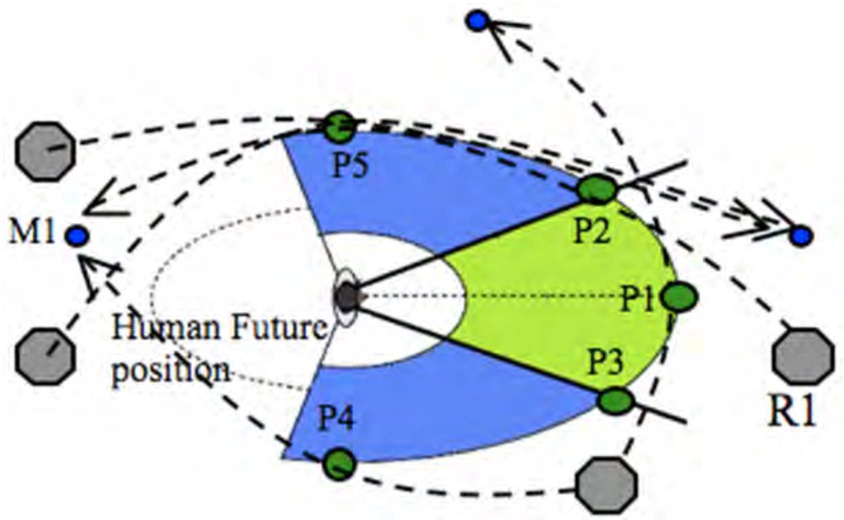
We have constructed a rule based decision tree based on different possible cases for the relative positions of the human, next milestone in the current path and the clearance of different regions around the human. In case of conflicts, the clearance constraints and the proximity guidelines have been given preference over the social conventions. The robot uses this decision tree to perform higher-level reasoning, for dealing with the dynamic human. A capable robot could also learn or enhance such decision tree based on user studies or demonstration. We define following two functions to query the decision tree:

$$\begin{aligned}(side, valid\_regions) = get\_side\_regions(&R\_pos, H[i]\_\ pos,\\ &M\_next, left\_min\_clearence, right\_min\_clearance)\end{aligned} \tag{6.1}$$

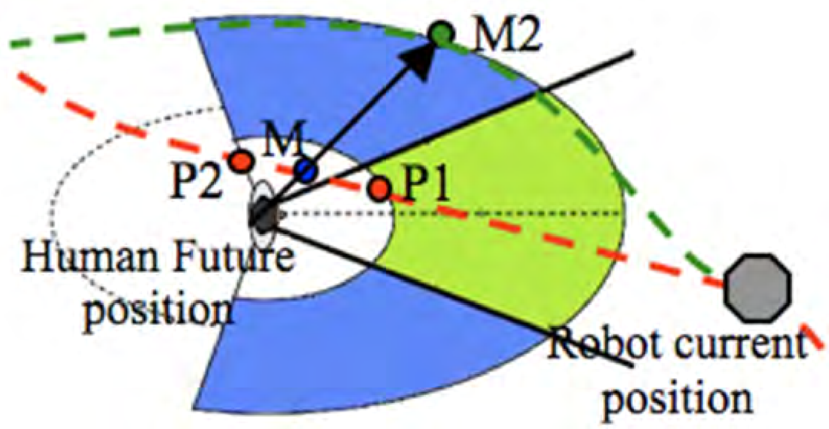
$$(milestones) = get\_milestone\,(R\_pos, H[i]\_\ pos, M\_next, side, valid\}) \tag{6.2}$$

where *R_pos* is the current position of the robot, *H[i]_pos* is the predicted position and orientation of the human *i, M_next* is the immediate next milestone in the robot's current path, $left\_min\_clearance$ and $right\_min\_clearance$ are the minimum lengths of *Interesting Boundary Lines (IBLs)* on left and right sides of the human predicted position. Function 6.1 returns, the side of the human (left/right), through which the robot should ideally pass and the set of acceptable regions (among 1-10, marked in figure 6.4) around the human, through which the robot may pass.

In figure 6.5(a), a subset of the decision tree, in form of different combinations of the robot positions (gray) and positions of the next milestone (blue), has been shown. Function 6.2 returns an ordered list of points to pass through as the intermediate milestones, from the set of points *(P1, P2, P3, P4, P5)* of figure 6.5(a). For example, if the robot is at *R1*, the next milestone to pass through is *M1*, then function 6.1 will return *(left, (1, 2, 3))* as the preferred side and acceptable regions in which the robot could navigate around the human while satisfying various rules. By taking the output of function 6.1, function 6.2 will return $\langle P2, P5\rangle$ as an ordered list of intermediate milestones, through which the path of the robot should preferably pass. But if there are some obstacles on the left side of the human such that *left_ min_ clearance* is not sufficient, functions 6.1 and 6.2 will return *(right, 2, 4)* and $\langle P3, P4\rangle$ respectively.



(a)



(b)

Figure 6.5: Different ways to get milestones to find deviated path to avoid a person. (a) By using decision tree: Avoiding a person by using decision tree for getting milestones. Different combination of the robot's position (gray polygon) and next milestone of the robot's path (blue circle) relative to the human predicted position result into different set of points around the human (green circles) treated as new milestones for modified path, through which the robot should pass. (b) By calculating new milestones: Another way of avoiding a person by calculating new milestone. Initial path is shown in red and the modified path in green. The segment *P1P2* of the initial path, which intersects the personal space of predicted human future position, is found and its midpoint *M* is projected to point *M2* (treated as new milestone) till the social boundary of the human.

6.2.5 Dealing with Dynamic Human

As soon as a human becomes visible to the robot and falls within some distance range, the robot has to decide whether or not to initiate the human avoidance process. For this the robot finds the minimum clearance around the human's predicted future position by constructing a separate set of *Interesting Boundary Lines (IBLs)*, as explained in the section 6.2.1. The robot also predicts a series of future positions for every visible human, just by extrapolating their previous positions and speeds (studies and works on human walking pattern like [Arechavaleta 2008], [Paris 2007], could help in better prediction). Then the robot checks, whether any segment of its current path is falling inside any of the regions from 1-9 of figure 6.4 or not. If not, then the robot will not show any reactive behavior assuming it will be far from the human and its motion behavior will not influence the human. Otherwise, there will be two cases: the path segment falls inside the personal space (5-8) or only inside the social space around the human (1-4). In the first case, the robot decides to smoothly deviate from its path by re-planning, even if there may not be any point-to-point collision with the human. This will serve the purpose of maintaining a comfortable social distance from the human as well as to signal the human about its awareness and intention well in advance. In the second case, the robot first queries the decision tree through function 6.1, *get_side_regions()*, and checks whether the passing-by side returned by the function is same as the passing by side while following the current path or not. If not, only then the robot will decide to re-plan.

Once the robot has decided to deviate, it needs to find a set of intermediate points (milestones) around the human through which the deformed path should pass. Figure 6.5(b) shows a situation in which the current path of the robot (red line) enters into the personal space of the human predicted position at *P1* and exits at *P2*. The robot first finds the mid point of the line *P1P2* and projects it to the outer ellipse of social space, at *M2*, from the viewpoint of the human predicted future position. If side of *M2* complies with the values returned by function 6.1, *get_side_regions()*, the robot accepts it as the milestone to pass through. Otherwise the robot uses function 6.2, *get_milestones()*, to get the milestones, for deviation, from the fixed set of points around the human.

6.2.6 Dealing with Previously Unknown Obstacles

The obstacles, which were previously unknown or are at changed positions; need to be dealt dynamically by the robot. For this, the robot first updates the Voronoi diagram in a window of width w around that obstacle. Then for avoiding such obstacles, the rules, which have been discussed in section 6.2.3, for planning using static environment, will be used to add or modify milestones for re-planning the smooth deviated path.

6.2.7 Dealing with a Group of People

In the current implementation, we assume that if people form a group, then each person should be within personal space of at least one other human. And if the group is moving, the difference of the speeds and orientations or each individual should be within some threshold. Once the robot detects a group, it finds the orientation *Th_ G*, and center *C_ G*, of the group by simply averaging the positions and orientation of every human of that group. For avoiding a group, the robot again constructs a similar set of elliptical regions, but with respect to the center of the group and with a different set of values for parameters, based on the spread of the group. The robot modifies the major axis of ellipse of the social region, which is actually responsible for signaling distance, by adding the distances of the farthest human from the center *C_ G* to it. But the minor axis, which is responsible for passing by distance from side, is modified by adding the distance of the farthest human of the side region only. This will ensure sufficient space in front region and only required space in side region, while avoiding. After dynamically adjusting the parameters of region for avoiding a group of people, the same algorithm presented above will generate the socially acceptable path for the robot to avoid the group.

6.2.8 Framework to Generate Smooth Socially-Aware Path

For the current discussion, the task of the robot is to reach to a goal place from its current location. The algorithm to generate the smooth path has been shown in algorithm .1. The first iteration flag is to ensure that the robot will pass through those regions and boundaries through which the shortest path is passing, by taking into account the static environment. This will ensure that just to avoid dynamic objects and humans, the robot should not take a longer path through entirely different regions. Wherever merging has been mentioned, it is done by following analysis: between which two successive boundaries of *CP* a particular point is falling and in the case of conflict the nearer one to the robot is put first in the merged list.

Figure 6.6 illustrates different steps of the algorithms. The dotted blue line shows the shortest path from start point S to the goal point G, generated by cost grid based A^* approach. The initial Voronoi Diagram of the environment generated by taking into account the static obstacles only, has been shown as skeleton of green points. The thin red lines are the *Interesting Boundary Lines (IBLs)*. Reader should not be confused with the rectangular tiles on the floor with *IBLs*. The blue circles show the set of initial milestones *CP*, extracted at the first iteration, *steps 1 - 7*. Now, to realize the social rule and clearance constraints selected to be used at the initial planning state as discussed in section 6.2.3, a process of refinement on the milestone along the line of minimum clearance i.e. *IBL* will be performed. *Steps 9 - 14* perform these refinements on the milestones. For the realization of rule *(S.1)*, the refinement process is to shift the milestones, which are of a corridor, a door or a narrow opening, towards the middle of the right half portion, based on the expected

Algorithm .1: ALGORITHM TO GENERATE A SOCIALLY-AWARE PATH.

Input: En:Environment 3D model, S:Start position, G:Goal Position

Output: Socially-aware path

1 $FIRST\_ITERATION = true$, $FM = [S, G]$, $FM\_D = NULL$; `//` FM `and` FM_D `are ordered list of fixed milestones and milestones due to dynamic environment respectively.`

2 $tmp\_FM = merge(FM\_D, FM)$; `// Merging two ordered list`

3 $SP = find\_path(tmp\_FM)$; `// Considering static obstacles only, find` A^* `based shortest path using all the ordered milestones`

4 Extract $CBP = [\langle cb, cp\rangle]$; `// Ordered list of tuple consisting of the boundary` $cb \in IBL$`, which the path` SP `crosses and the corresponding crossing point` cp

5 **if** $FIRST\_ITERATION = true$ **then**

6   Label_Crossing_Boundaries(En, SP, CBP); `// Subroutine (algorithm .2) to label crossing boundaries as corridor, wide opening.`

7     $FIRST\_ITERATION = false$;

8 $CP\_M = NULL$; `// To store list of modified crossing points`

9 **foreach** $\langle cb, cp\rangle \in CBP$ **do**

10   **if** $label(cb) \neq PROCESSED$ **then**

11   cp_m=Apply(SR_P, on $\langle cb, cp\rangle$) ; `// Get modified crossing point by applying` SR_P`, the set of rules selected to be used considering the static part of the environment.`

12   **if** $cp\_m \neq cp$ **then**

13   insert(CP_M, cp_m), replace(cp by cp_m in $\langle cb, cp\rangle$);

14   label($cb, PROCESSED$);

15 **if** $CP\_M == NULL$ **then**

16   ***Goto*** step 20;

17 **else**

18     $tmp\_FM = NULL$, $tmp\_FM = merge(FM, CP\_M)$;

19   ***Loop*** from step3;

20 $tmp\_FM = NULL$, $tmp\_FM = merge(FM, CP)$; `//` CP `is the ordered list of crossing points stored in` CBP

21 IP=Get_Interpolated_Path(tmp_FM) ; `// Generate spline path through interpolation among the milestones of` tmp_FM.

22 FM_D=Treat_Dynamic_Environment_Part(En, IP) ; `// Subroutine (algorithm .3) to extract information about unknown obstacles, individual and group, and apply relevant rules.`

23 **if** $FM\_D \neq NULL$ **then**

24   ***Loop*** from step 2;

25 **else**

26   ***return*** IP;

Algorithm .2: ALGORITHM TO LABEL CROSSING BOUNDARIES OF A PLANNED PATH.

Input: *En*:Environment 3D model, *SP*:Planned Path, *CBP*:List of tuple of crossing boundaries and the corresponding crossing points.

Output: Labeled crossing boundaries as narrow passage, corridor entry, corridor exit, wide opening.

$Topo = extract\_topological\_info(SP)$ `// Extract environment topological information along the path` SP.

foreach $\langle cb, cp \rangle \in CBP$ *and* $label(cb) \neq INACTIVE$ **do**
  if $cb \in narrow\_passage$ *or* $cb \in door$ **then**
    label(cb, NARROW); `// Label corresponding cb as narrow region`

foreach $\langle cb, cp \rangle \in CBP$ *and* $label(cb) \neq INACTIVE$ **do**
  if $cb \in corridor$ **then**
    $C\_Enter = cb$, $C\_Exit = extract\_exit(C\_Enter, SP)$;
    forall the *crossing boundaries,* cb_i *between* C_Enter *and* C_Exit **do**
      label(cb_i, INACTIVE); `// Will be not used for finding path in subsequent iterations.`

foreach $\langle cb, cp \rangle \in CBP$ **do**
  if $label(cb) \neq INACTIVE$ *and* $label(cb) \neq NARROW$ **then**
    label(cb, WIDE);

orientation at crossing points. The green milestones at boundaries 1, 5, 6 and 7 of figure 6.6 are obtained by shifting such blue milestones. The refinement associated with other rules are, if the distance of the crossing point is less than a required minimum distances from the nearest end of the corresponding *IBL*, then shift away the crossing points along the *IBL* until middle of the *IBL* is reached or the desired distance is achieved. These rules resulted into the green milestones at boundary 3 and 4 by shifting away the corresponding blue milestones. All the milestones, which will be refined by the initial social rules, will be treated as the fixed milestones for the next iterations. *Steps 15 - 19* assure the shortest path between two fixed milestones, because as few milestones have been shifted, the other milestones may no longer fall on the probable shorter path. For example, the blue milestones of boundaries 2 and 8 have been shifted to the green milestones in the second iteration of the algorithm.

Then the control reaches to step 21 to find the smooth path by interpolating through all the milestones obtained so far. Then in step 22, this path is used to check any conflict or violation of different rules on the dynamic or previously unknown part of the environment. For avoiding any previously unknown entity: obstacles, objects, human or group of people, in the current implementation, we chose to plan to avoid

Algorithm .3: ALGORITHM TO EXTRACT THE INFORMATION ABOUT THE DYNAMIC AND UNKNOWN PARTS OF THE ENVIRONMENT: PREVIOUSLY UNKNOWN OBSTACLES, INDIVIDUAL OR GROUP OF PEOPLE. TEST FOR SOCIAL AND PROXIMITY RULES.

Input: En: Environment 3D model, IP: Planned smooth interpolated path considering static environment

Output: Ordered list of new milestones because of the presence of previously unknown entities.

Update list of visible Humans H;
$FM\_D = NULL$;
HG=Extract_Groups(H) ; `// Find set of humans moving or standing in groups`
$HI = H - HG$; `// Set of individuals, not belonging to any group`
Extract_New_Obstacles(O) ; `// Find set of obstacles which were previously unknown`
LE=merge(HG, HI, O) ; `// Obtain the list of all the potential entities, an individual, a group or an obstacle, to be avoided, by merging them in order`
foreach *entity* $e \in LE$ **do**
    if $e \in HG$ **then**
        Construct_Regions_Around_Group(e) ; `// e is a group of people. Construct a single elliptical region around the group, parameters of which depend on the spread of the group`
        if *Need_Group_Avoidance(e, IP) == TRUE* **then**
            FM_D=Avoid_Group(e) ; `// Apply the group avoidance rules and extract the new ordered list of milestones`
            return FM_D
    if $e \in HI$ **then**
        if *Need_Individual_Avoidance(e, IP) == TRUE* **then**
            FM_D=Avoid_Individual(e) ; `// Apply the avoidance rules for an individual and extract the new ordered list of milestones`
            return FM_D
    if $e \in O$ **then**
        if *Need_Obstacle_Avoidance(e, IP) == TRUE* **then**
            FM_D=Avoid_Obstacle(e) ; `// Apply the avoidance rules for avoiding the obstacle and extract the new ordered list of milestones`
            return FM_D

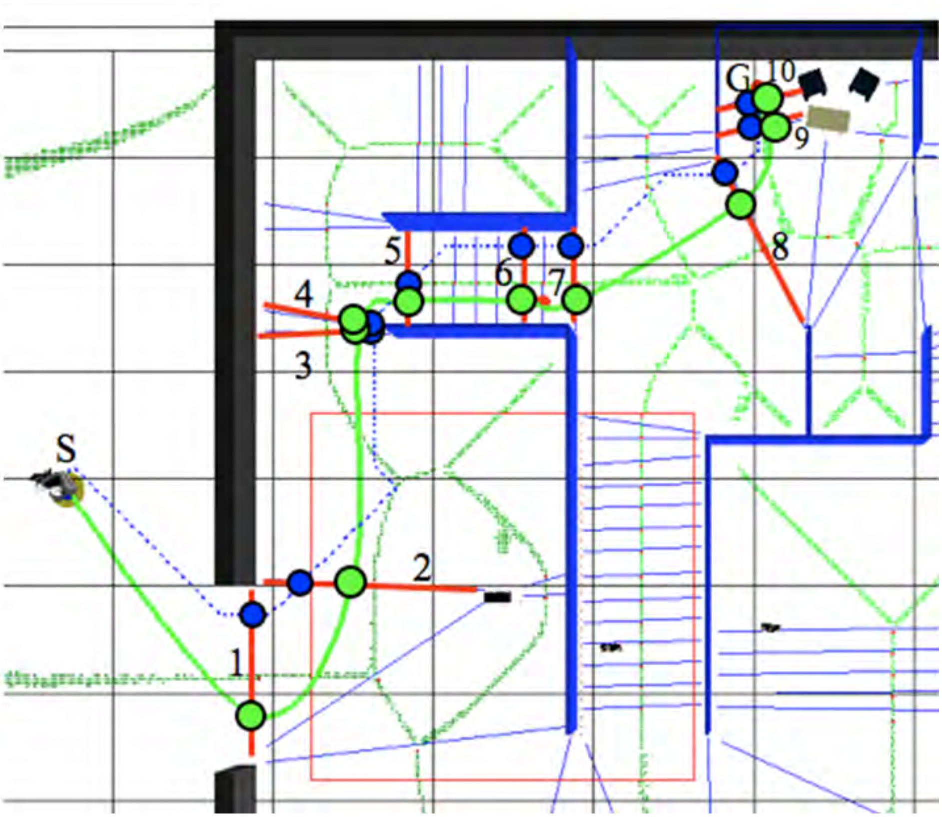


Figure 6.6: Steps of iterative refinements on the path to incorporate social conventions and clearance constraints at planning stage. Blue dotted path from S to G is the initially found A^* based shortest path. Green path from S to G is the obtained smooth and socially-aware path. Different rules have been incorporated in the different segments of the path by accordingly manipulating the milestones.

them in piece-wise manner. This means first plan to avoid the nearest object, human or group, which is conflicting with the constraints to be maintain. And if the new plan is still conflicting with some other entity, then append the set of milestones to avoid that entity also and so on. That is why algorithm .3 returns as soon as it finds a new set of milestones for the first group, individual or object that is conflicting. This choice has been made with the assumption that avoiding the nearest entity might have changed the path so that the existing conflict with other entity might not be valid any more. However, this choice of looking one conflict in advance could be altered and one could decide to plan to avoid all the currently conflicting entities, which could be required if the environment is crowded.

After getting a set of milestones through which the robot should pass, the robot solves Hermite cubic polynomial for continuity constraint on velocity and acceleration at boundaries to piece-wise connect the milestones. The green curve in figure 6.6 shows the final smooth path generated by using the final set of milestones for planning the initial path.

6.2.9 Proof of Convergence

The convergence of the algorithm lies in the fact that, after each iteration there will be a set of fixed milestones, which will not change in next iterations, as they will already be satisfying the rules. Hence, eventually the *step 15* will result into an empty set of modified milestones, CP_M. Further, eventually the smooth path generated in *step 21* will not be required to be altered because at some point of iteration it will incorporate the milestones due to all the conflicting dynamic parts. Hence, FM_D obtained in *step 22* will be $NULL$, resulting into the termination of the algorithm. In all our test runs, in 2 to 3 iterations the algorithm has converged, hence facilitating the algorithm to run online. However, the speed of convergence and efficiency of the re-planning will depend upon, how much crowded and dynamic the environment is.

6.3 Experimental Results and Analysis

For testing our framework, the models of environment, the robot and the human is fed and updated into our developed 3D representation and planning software Move3D. Figure 6.7 is the part of a big simulated environment of dimension $25m \times 25m$; S and G are start and goal position for the robot. The blue lines are the *Interesting Boundary Lines (IBLs)* extracted by our proposed approach.

6.3.1 Comparative analysis of *Voronoi Path* vs. *Socially-Aware Path* vs. *Shortest Path*

The Voronoi diagram has been shown as green skeleton of points. A^* shortest path has been shown as blue dotted path. The green curve is the smooth social path generated by the robot by our proposed algorithm. Note that the robot autonomously inferred that it is in a corridor and shifted the path to the right side of the corridor until the autonomously found exit of the corridor. In literature [Victorino 2003], [Garrido 2006], Voronoi diagram itself has been used as the robot's path. However, one could discover that the planned path by presented approach avoids unnecessary route of Voronoi diagram in the wider regions, e.g. the region enclosed by blue ellipse. Moreover, in the regions where all the constraints are satisfied, our algorithm provides a path segment close to the shortest path by A^* planner, e.g. the region enclosed by the red ellipse. But if there is no sufficient clearance, our algorithm shifts the crossing points to the middle of the *IBLs*, hence following the Voronoi diagram in that region for assuring maximum possible clearance. Hence, our algorithm inherits the characteristics of A^* and Voronoi diagram based paths at the places where they perform better while globally maintaining the social conventions and smoothness of the path.

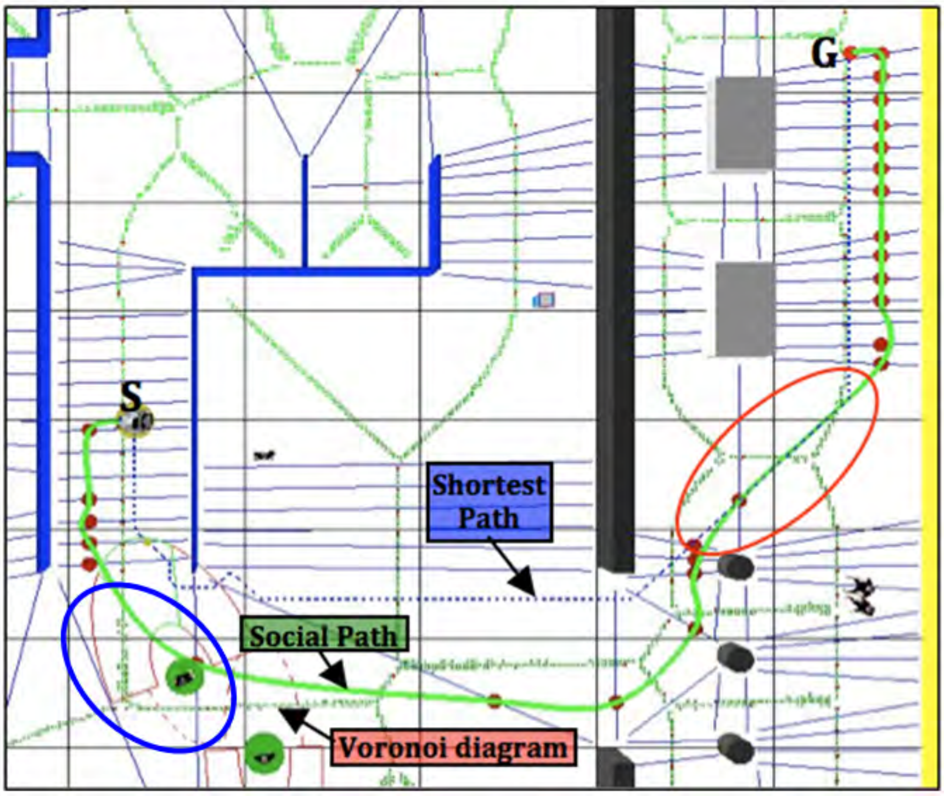


Figure 6.7: S and G are start and goal positions. Thick Green path is the smooth and socially acceptable path planned by our approach. Dotted Blue path is the shortest path planned by cost grid based A^* planner. Green skeleton of points is the Voronoi diagram. The planned socially-aware path avoids unnecessary long route of Voronoi diagram, for example, in the segment enclosed by blue ellipse. In addition, wherever feasible, the socially-aware path follows the shortest path, for example, the region enclosed by red ellipse. Whereas, in the case of insufficient clearance, the planned social path autonomously seems to be following the Voronoi diagram, to assure maximum possible clearance around.

6.3.2 Analyzing Passing By, Over Taking and Conflict Avoiding Behaviors

Figure 6.8(a) shows the robot is passing by a person in the corridor without creating any conflicting situation. Figures 6.8(b) and 6.8(c) show the detection of a group of people based on their relative speeds and positions, and avoiding the group from the left side. Note that the initial path in figure 6.7 has been smoothly modified in figure 6.8(b) at the predicted passing by place.

We have implemented our presented framework on our mobile robot Jido. It uses vision based tag identification system for detecting dynamic objects like trash bin, table, etc., and markers based motion capture system for reliable person detection.

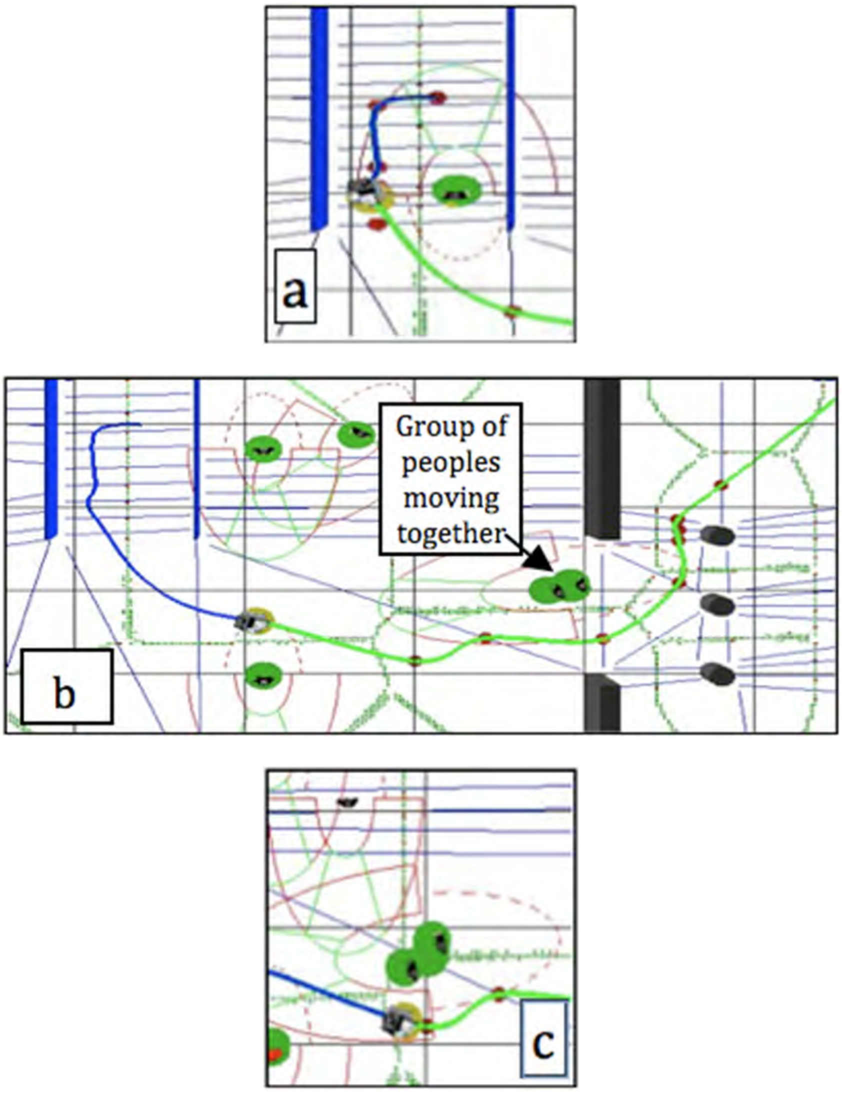


Figure 6.8: (a) The robot is smoothly passing by a person in the corridor, (b) planning a smooth deviation in the path to the avoid group of people with sufficient signaling distance at the expected passing by place (see figure 6.7 for initial path), (c) smoothly and without any conflict, passing by the group from the left.

Figure 6.9 shows the sequence of images where the robot has predicted that even if there is no direct collision with the human, it might enter into the personal space of the human hence modifies its path to smoothly avoid the person from her left side.

Figure 6.10 shows the case when the robot has planned the path, shown as red arrow, to smoothly cross the standing person, to reach the goal, while maintaining the proximity constraints *(P.1)* & *(P.2)* around the person. Figure 6.11 shows the results of avoiding previously unknown obstacles, for which the robot updates the Voronoi diagram to extract new clearance information and our presented algorithm

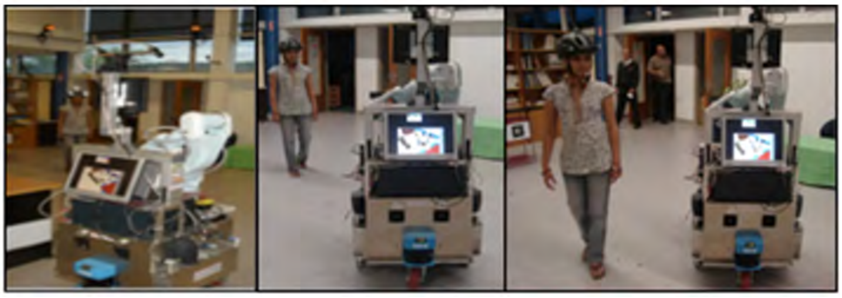


Figure 6.9: The Jido robot avoiding the person by maintaining the social convention of passing by from her left side.

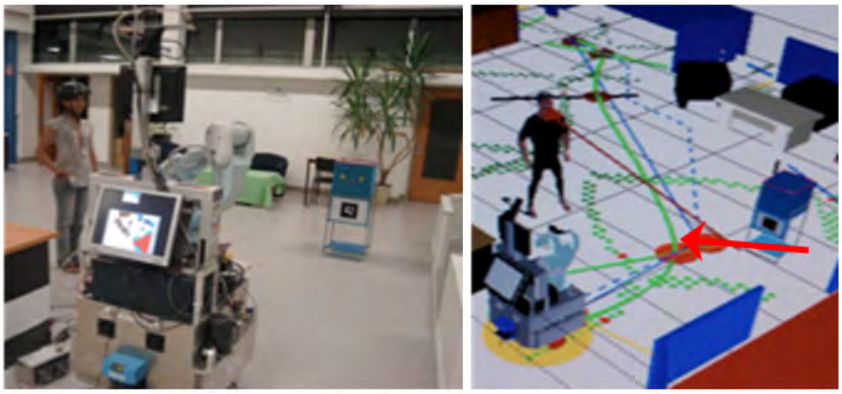


Figure 6.10: The robot crosses a standing person by avoid to enter into her personal space, because no interaction is required. The Red arrow indicates the planned path.

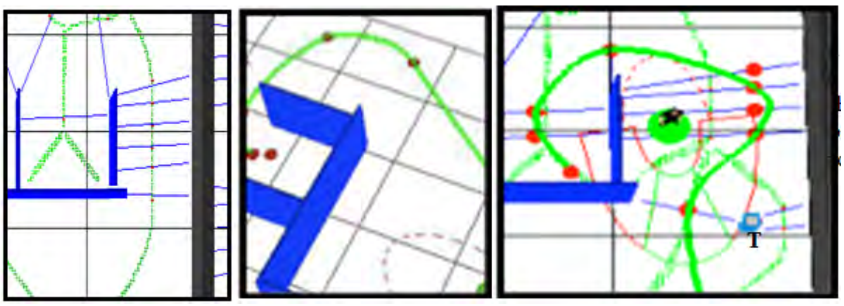


Figure 6.11: (a) Initial Voronoi Diagram and clearance (IBLs), (b) initial planned path, (c) during execution the updated clearance information and deviated path due to presence of previously unknown trash bin, marked as T.

adds new set of milestones to re-plan the smooth deviated path as shown in figure 6.11(c). Figure 6.12 shows the bigger portion of our lab having corridor. The green curve is the smooth path generated by the robot using the presented approach to reach from S to G.

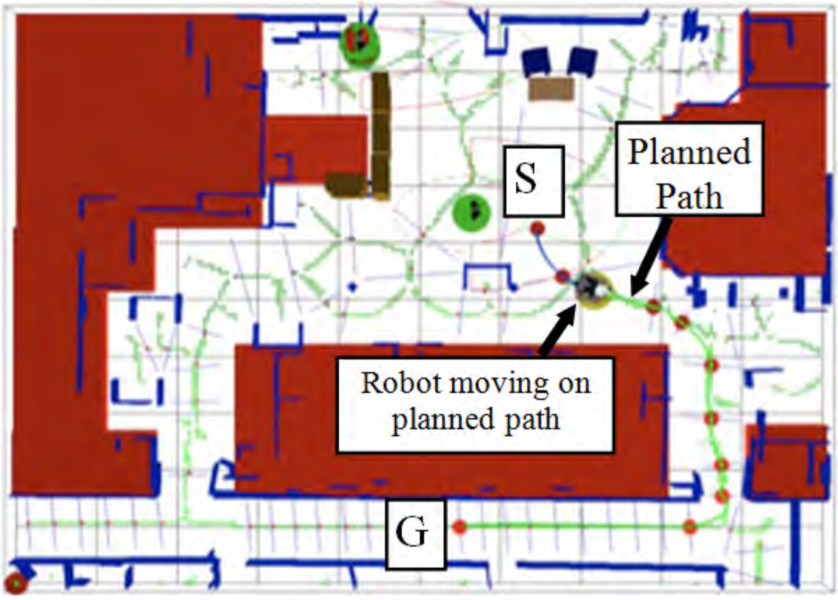


Figure 6.12: Path generated in the bigger map of our lab, from S to G using our presented framework.

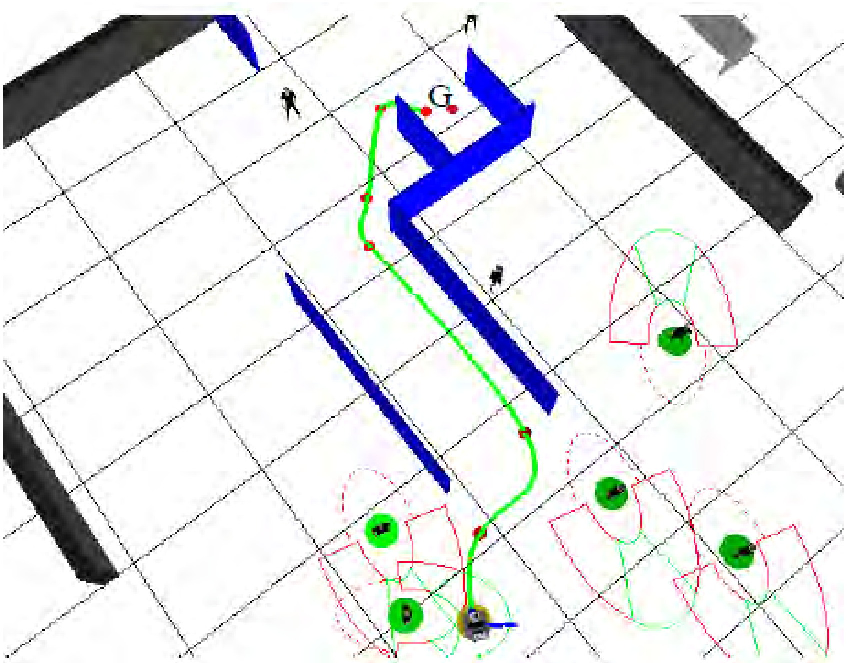


Figure 6.13: Initial socially-aware path generated by the set of social conventions, which are included at time of initial planning. Note that the robot maintains itself in the right half portion of the corridor. In addition, the entire path is smooth.

Figure 6.13 shows another initial social path generated by the robot to the goal position G. The generated green path is smooth and it maintains to be on the right

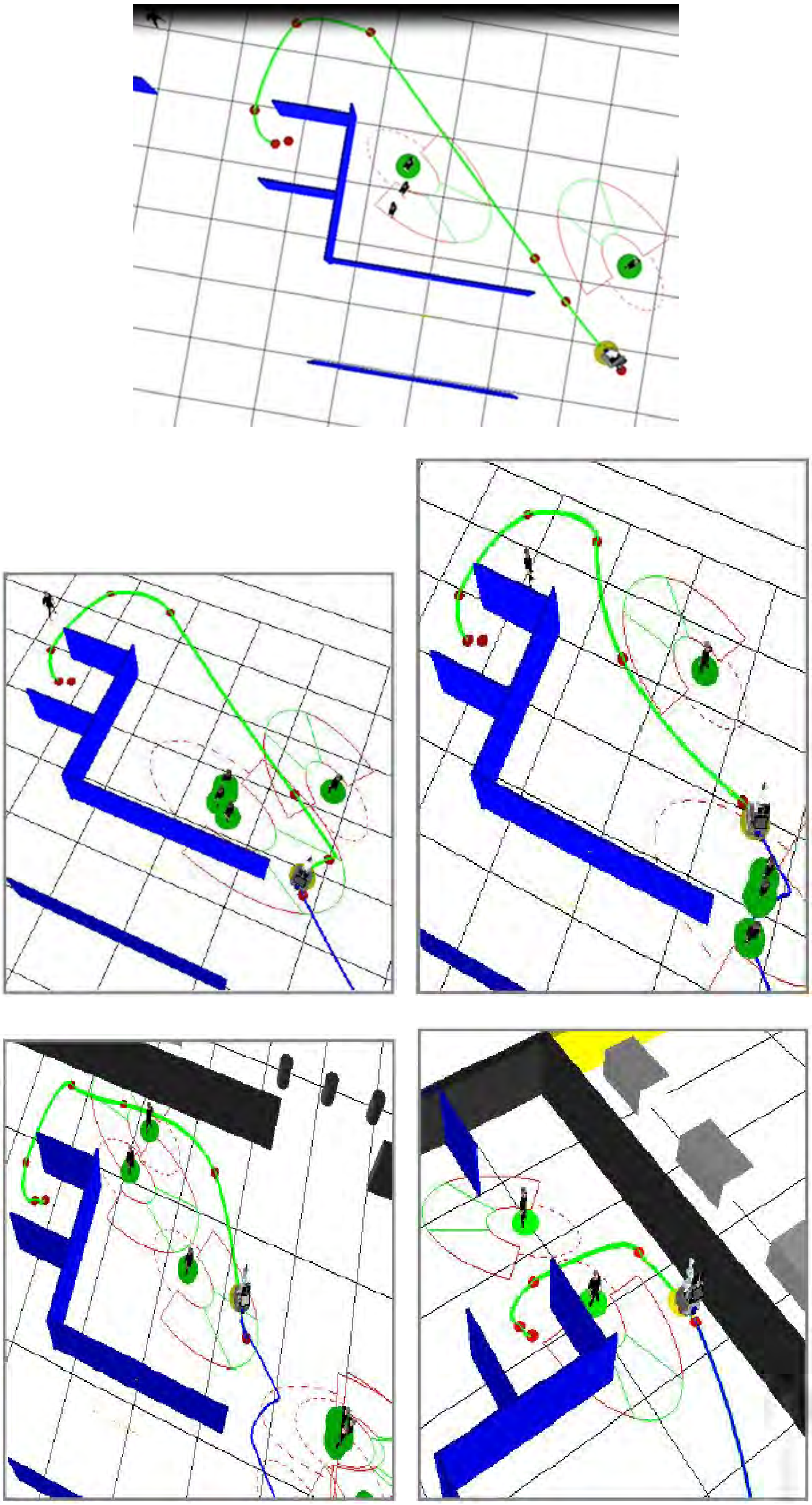


Figure 6.14: (a) Initial planned socially-aware path. (b) Group detected, smoothly passing by the group from their left. (c) Overtaking a person from his left. (d) Passing by different persons from their left sides. (e) smoothly passing by a person in a corridor. Also note the smoothness in the deviated path in all the cases and successful avoidance of unnecessary reactive behaviors or conflicting situations.

Case I	*Narrow Passage*	*Open Space*
Robot is at side of human	W1=1	W2=2
Robot is in front of human	W3=4	W4=3

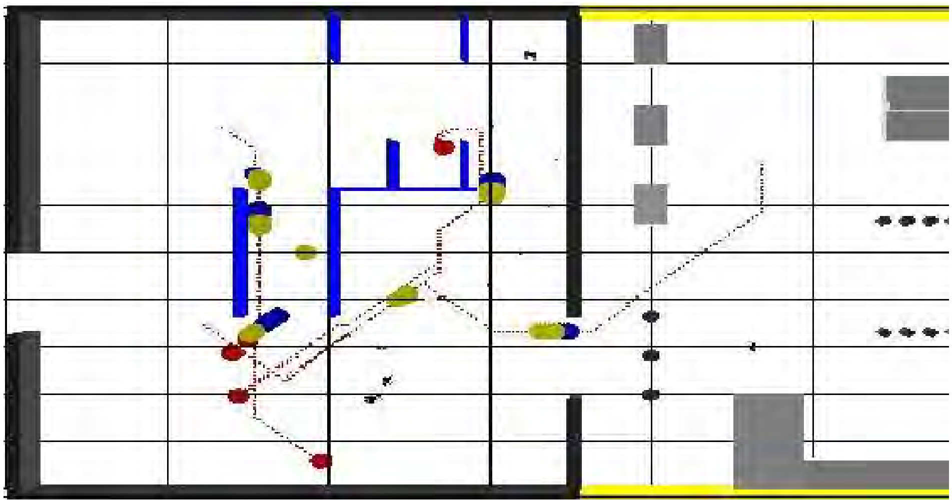
Case II	
Human Moving	W5=3
Human Standing	W6=1

Case III	*Narrow Passage*	*Open Space*
Before breaking the rule, robot was behind human	W7=4	W8=3
Before breaking the rule, robot was in front of human	W9=2	W10=**1**

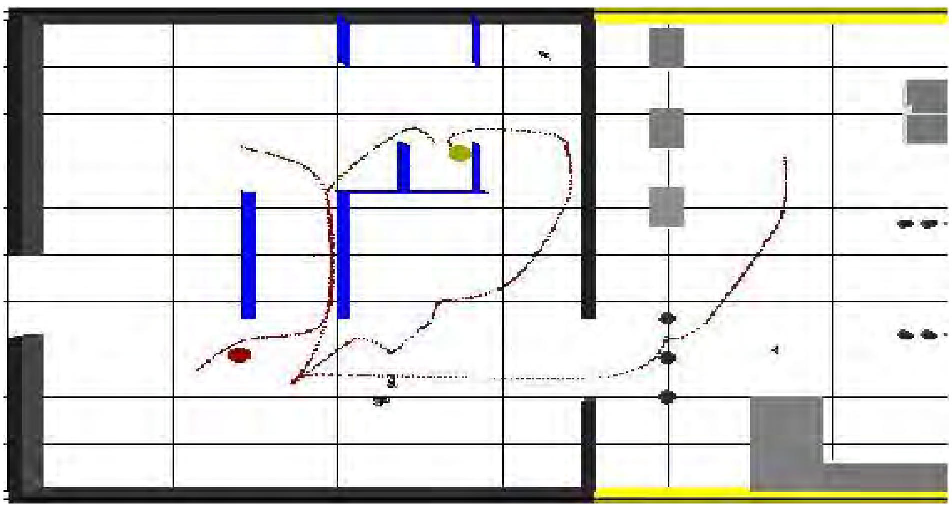
Figure 6.15: Weights for cases and sub-cases of the robot behaviors for comparing socially-aware path with purely reactive behavior based path.

half portion while inside the corridor. Figure 6.14 shows adaptation of different social rules while navigating in the human centered environment. Figure 6.14(a) shows initial path, taking into account the conflicting situations based on environment structures, and plans to moves on the right side of the narrow passage. Figure 6.14(b) shows the result of successful detection and avoidance of a group of people using social rule. Even if there was no point-to-point collision (physical collision) with the earlier path to any of the group member, the robot has generated a deviated path well in advance to signal the group that the robot is aware about them. Also note the proper passing by distance from the group while avoiding. Note the difference in shape and size of the region around the group from the regions around individual humans, as the robot has dynamically modified the parameters of the regions based on the spread of the group. Similarly, for avoiding a single person, the robot has generated deviated path with proper signaling and passing by distance. Apart from assuring gradual and smooth deviation, the robot also maintains the social conventions while passing by to avoid any conflict. As in this case, the robot's deviated path is passing by the group from the left side of the human. Figures 6.14(c) and 6.14(d) show the modified socially-aware paths in the situations of overtaking and passing by different humans. Figure 6.14(e) shows the robot passing through a narrow corridor in the presence of another human coming from opposite side, by respecting the social conventions, so there is no unnecessary reactive behavior or conflicting situation.

Our implementation is generic enough to easily switch between the right-handed and the left handed walking system.



(a)



(b)

Figure 6.16: Comparing purely reactive behavior based path with socially-aware path: (a) Different clusters of unwanted states (in overlapping blue, red and yellow circular regions along the paths) when navigated by a purely reactive robot (PRR) in the human centered environment. (b) By using our approach of socially-aware robot path (SR), different clusters of the unwanted states has been significantly reduced

6.3.3 Qualitative and Quantitative Analyses of Generated Social Navigation with Purely Reactive Navigation Behaviors

Test on the physiological or emotional response of the human is beyond the scope of this chapter. But to analyze the performance of our approach in terms of physical comfort for a human, we have formulated few criteria based on relative positions of

Case	*I. Physical Uncomfort*		*II. Unexpected*		*III. Unintuitive*		Total	
Human ID	*PRR*	*SR*	*PRR*	*SR*	*PRR*	*SR*	*PRR*	*SR*
1	2	3	1	0	0	0	*3*	*3*
2	14	0	0	5	12	2	*26*	*7*
3	24	0	6	0	10	0	*40*	*0*
4	0	0	0	0	6	0	*6*	*0*
5	28	0	12	4	8	0	*48*	*4*
6	14	4	5	2	8	6	*27*	*12*
7	3	0	0	0	5	0	*8*	*0*
8	6	0	0	0	6	0	*12*	*0*
Total	*91*	*7*	*24*	*11*	*55*	*8*	***170***	***26***
Percentage Reduction in Unwanted Behavior							***84.7%***	

PRR = Purely Reactive Robot, SR = Social Robot which uses our developed approach

Figure 6.17: Person-wise and case-wise comparison of unwanted behavior of the purely reactive robot with our developed social path planner.

the human and the robot. For comparison we use a purely reactive robot, which calculates a new path based on cost grid only if there is a point-to-point collision predicted with the human, and simply assumes the human as an obstacle. We have defined 3 terms about unwanted robot behavior:

I **Physical Uncomfort**: Whenever the robot enters into personal or intimate region of the human, without requirement of any interaction.

II **Unexpected**: Whenever the robot appears suddenly from behind a wall or from behind the human itself in his personal space. This is calculated based on the region on which the robot falls just at the instant when it gets visible to the human.

III **Unintuitive**: Whenever the robot does not meet the social expectations of the human, or cause some conflict. This is calculated by comparing the ideal social position and the actual position of the robot at the time of passing by, approaching, avoiding, taking over, etc., but only in the situations when the robot is within the social region of the human.

Figure 6.15 shows different weights assigned to the different sub-cases of these cases, based on the current and previous positions of the robot with respect to the human, environment structure and the human state. We will not provide a detailed argument for the weights but the relative order of weights could be institutively justified. For the experiments, different numbers of runs have been performed with different starting and end positions, all of them have been overlaid in the environment of figure 6.16 and summarized in figure 6.17, which compares our approach with a purely reactive robot. Two different environment types indoor and outdoor (left and right portions of both the environment of figure 6.16) have been also integrated to evaluate the performance. Different number of humans from the point of view of initial visibility, closeness to the robot and moving in a group or not have been instantiated for different runs. In addition, some humans were moving randomly, some were moving using social rules and some were not moving at all. Figure 6.17 shows the person-wise and case-wise comparison of unwanted behavior of the purely reactive robot (PRR) with our developed social robot (SR). For the same set of motion of all the humans and start and goal positions of the robot, the total weighted value of unwanted behavior for purely reactive robot was *170*, whereas with our approach it reduced to *26*. Hence, the percentage of reduction in the unwanted behavior of the robot was about *85%*. It will be evident from figures 6.16(a) and figure 6.16(b). Yellow, red and blue regions in figure 6.16(a) show the different places where the situations (I), (II) and (III) have occurred at some or the other point of time, when the robot was purely reactive. Figure 6.16(b) shows the same set of regions in the case the robot was equipped with our developed algorithm to incorporate different social conventions at different states of execution. Path planned by the robot in both the cases have been also shown in red. Presence of very few such regions in figure 6.16(b) shows the efficacy of our approach.

Until now, we have equipped the robot to navigate in the human centered environment in a socially acceptable manner. In the examples so far, there was not joint goal between the human and the robot. In next section we will incorporate the notion of joint goal from the perspective of the robot is required to guide a person from his current position to the goal location.

6.4 Social Robot Guide

As mentioned in section 2.3, monitoring the presence of person to be guided is necessary. The simple stop & wait model of co-operative task based on presence and re-appearance of the person to be guided is not socially appreciated. During the guiding process, the person can gradually switch from one side to another side of the robot, speed up or slow down, or even temporarily stop. Also at one point of time, the human may decide to follow the robot from its behind and at another point of time he could decide to accompany the robot by moving side by side. Such deviations in the human motion are categorized as non-leave-taking behaviors, in the sense the human intention is not to interrupt or suspend the guiding process. The robot should understand the human intentions, and should neither show over-reactive behavior by deviating frequently from its path, nor should it stop the guiding process, which could annoy, irritate or confuse the human.

On the other hand, there could crop up the situations, when the human deviates

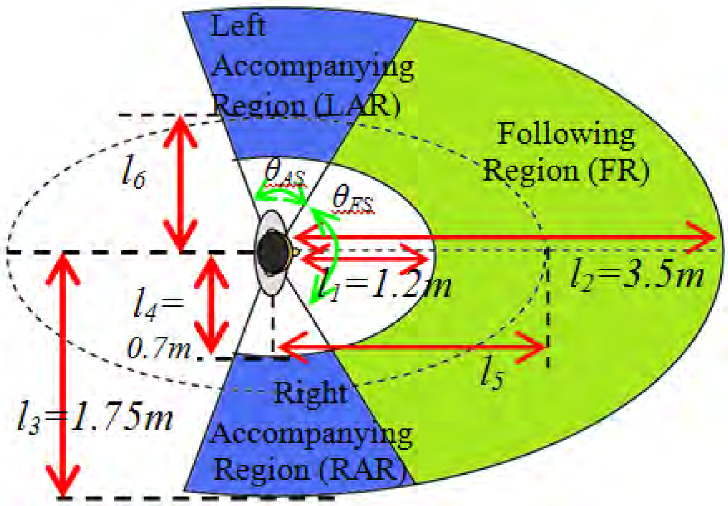


Figure 6.18: Parameters of social space around the human, and the *Following* (green) and the *Accompanying* (Blue) regions of the human.

significantly from the expected path due to some personal quest of reaching some nearby person, place or thing, due to social forces. In doing so, the human intention is not to completely break the joint commitment of guiding, but to temporarily suspend following the robot. Such deviations in the human motion are categorized as temporary leave-taking behavior. In such situation, the robot should respect the person's desire and should deviate from its original path in order to catch or approach the person as an attempt to support the human activity as well as re-engage the human in the guiding process. It will also reduce any future effort of the human for resuming the guiding process. But at the same time such deviations should be also oriented towards the goal. In this framework, the robot monitors the human behavior with respect to the guiding task and equipped with the capabilities to verify and re-initiate engagement.

Apart from assuring safety, and physical-comfort, the guiding path generated by the robot should be intuitive and socially-accepted, which could also influence the person's trajectory and fetch the person towards the goal, by exerting a kind of fetching or pushing social force. The last two characteristics will make the robot's path different from the paths generated in the cases, when the robot has to simply follow, pass, approach or accompany the person.

6.4.1 Regions around the Human

From the point of view of guiding, we have adapted the regions around the human as presented in figure 6.4 from the perspective of the task of being guided by someone, figure 6.18. Note that the angular spread of the accompanying span is slightly beyond *90 degrees* from the human axis on both sides. This is because sometimes even as an accompanying person, the human may want to move slightly ahead of the robot. As explained earlier, these regions should only serve as a reference in various decision-making processes. We will explain how the robot adjusts these parameters depending upon the situations.

6.4.2 Non-Leave-Taking Human Activities

As discussed earlier, the human can exhibit various natural deviations in his motion, on the way, even if he is supporting the guiding process. Apart from switching between following from behind to accompanying from side of the robot, he may also gradually shift from left to right side of the robot. Also, during the guiding process, the person can slightly deviate, turn left or right, speed up or slow down. Although the human is not exactly tracing the robot path, the human intention is not to break or suspend the joint commitment of guiding. So, the robot should not show any reactive behavior like deviating from its path or breaking the guiding process.

6.4.3 Belief about the Human's Joint Commitment

We model *P(JC)*, the belief of the human intention of maintaining the joint commitment of guiding process, by multi-variant Normal Distribution as follows:

$$P(JC) = \left((2\pi)^4 \left|\sum\right|\right)^{-1/2} exp\left(-\frac{1}{2}(D1 + D2)\right) \tag{6.3}$$

$$\sum = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\triangle\theta}^2 & 0 \\ 0 & 0 & 0 & \sigma_s^2 \end{bmatrix} \quad \mu = \begin{pmatrix} x_h \\ y_h \\ 0 \\ S_h \end{pmatrix} \quad X = \begin{pmatrix} x_r \\ y_r \\ \triangle\theta \\ S_r \end{pmatrix} \tag{6.4}$$

Where (x_h, y_h) and S_h are the position and speed of the human, and (x_r, y_r) and S_r are the position and speed of the robot at time t. $\triangle\theta$ is the angular position of the robot with respect to the human axis.

$$D1 = 2\left(a\,(x_r - x_h)^2 + 2b\,(x_r - x_h)\,(y_r - y_h) + c\,(y_r - y_h)^2\right) \tag{6.5}$$

D1 is exponent of the parametric form of bi-variant normal distribution in *(x,y)* plane which also takes into account the orientation θ of the distribution, which is, in fact, the orientation of the human. The parameters are:

$$a = \frac{\cos^2\theta}{2\sigma_x^2} + \frac{sin^2\theta}{2\sigma_y^2},\ b = \frac{\sin 2\theta}{4\sigma_y^2} - \frac{\sin 2\theta}{4\sigma_x^2},\ c = \frac{\sin^2\theta}{2\sigma_x^2} + \frac{cos^2\theta}{2\sigma_y^2} \tag{6.6}$$

And *D2* is the exponent of normal distribution for the rest two variables, given as :

$$D2 = (\triangle\theta)^2/\sigma_{\triangle\theta}^2 + (S_r - S_h)^2/\sigma_s^2 \tag{6.7}$$

As will be assigned in the following sections, the values of the parameters $(\sigma_x^2, \sigma_y^2, \sigma_{\triangle\theta}^2, \sigma_s^2)$ will vary according to the different states of the robot and the human.

6.4.4 Avoiding Over-Reactive Behavior

Once the joint commitment has been established and guiding process has been started, the robot is said to be in *mentor* state and the human is in *follow* state. The values of $(\sigma_x^2, \sigma_y^2, \sigma_{\triangle\theta}^2, \sigma_s^2)$ in this state will be $(3.5, 1.75, 2\pi/3, 1)$. Note that these values are inspired from figure 6.18 of our constructed regions around the human, to assign higher probability when the human maintains the robot in his accompanying or following regions. When the guiding path passes through opening or corridor which is too narrow to move for the robot and the human together side by side, the robot will relax the parameter $\sigma_{\triangle\theta}^2$ by setting it as π, hence giving the freedom to the human to move ahead of the robot to pass first, if he wants. The robot will not show any deviation from its path as long as the *P(JC)* lies within the ellipsoid that contains the top 50% of the probability distribution. For 4-dimensional normal distribution, this condition is satisfied when the square Mahalnobis distance $(D1 + D2)$ will be less than 3.36. Further, if $(D1 + D2)$ lies within the top 35% of distribution, the robot continues with its speed. This will provide the human with the freedom to decide upon the distance, position and orientation with respect to the robot, without causing the robot to react. However to adapt to the human speed, the robot will start slowing down proportionally, if $(D1 + D2)$ starts lying within the band of top 35% to top 45% of probability distribution. And the robot will completely stop and reach the *wait* state if $(D1 + D2)$ lies within the band of top 45% to 50%, which will provide the human with the freedom to halt for few moments on the way for various reasons like interacting with someone or looking at a photo frames on the wall, etc. From this *wait* state the robot will either return to the *mentor* state in which it will resume tracing the already planned path or switch to the *deviate* state. But before resuming from the *wait* state, the robot makes sure that the human is now willing to be guided. For achieving this, the robot tightens the parameters $(\sigma_{\triangle\theta}^2, \sigma_s^2)$ to $(\pi/2, 0.5)$, for assuring that the human is in higher level of harmony with the robot. And with this new values, if the square Mahalnobis distance, $(D1 + D2)$, on next few time instances starts lying within the top 45% probability distribution, the robot will return to the *mentor* state. Note that for falling into *wait* state the threshold was $>$50% but for returning to *mentor* state it is $<$45%, which is an additional attempt to ensure good harmony with the human and better confidence on the human intention of joint commitment before restarting the guiding process. The case of switching to the *deviate* state has been addressed in the following section.

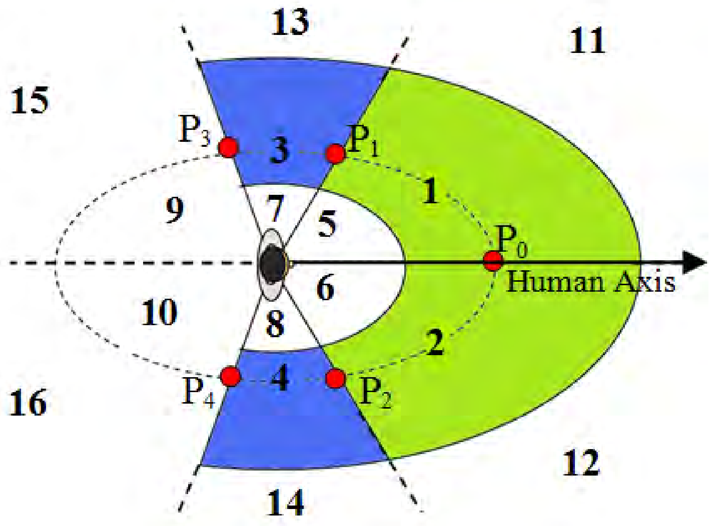


Figure 6.19: Different sub-regions, their IDs and the candidate passing by points (red circles) around the human.

6.4.5 Leave-Taking Human Activity

As discussed earlier the human may suspend the guiding process for reaching to some nearby person, place or thing, even if his intention is not to completely break the guiding process. As explained above, if the square Mahalanobis distance $(D1 - D2)$ falls outside the top 50% of the probability distribution or the waiting time has reached a maximum tolerable waiting time, the robot categorize the human activity as temporary leave taking.

6.4.6 Goal Oriented Re-engagement Effort

Once the robot will switch to the *deviate* state it will deviate from its path as an attempt to support the human activity as well as to reengage him in the guiding process. Such deviations should be goal directed and intuitive as well as try to follow or approach the human from the appropriate side and distance.

6.4.6.1 Prediction of Meeting Point

For planning the deviated path, the robot needs to first predict the future position of the human, P_{meet}, towards which the robot should start approaching. The robot will use most recent n samples of the human positions to infer about his velocity and the probable future trajectory. Then by taking into account its velocity constraints, the robot will find the nearest point on the predicted human trajectory, where if the human would reach at time t, the robot would have reached in front of the human at point P_0 shown in figure 6.19. This position of the human will serve as P_{meet}.

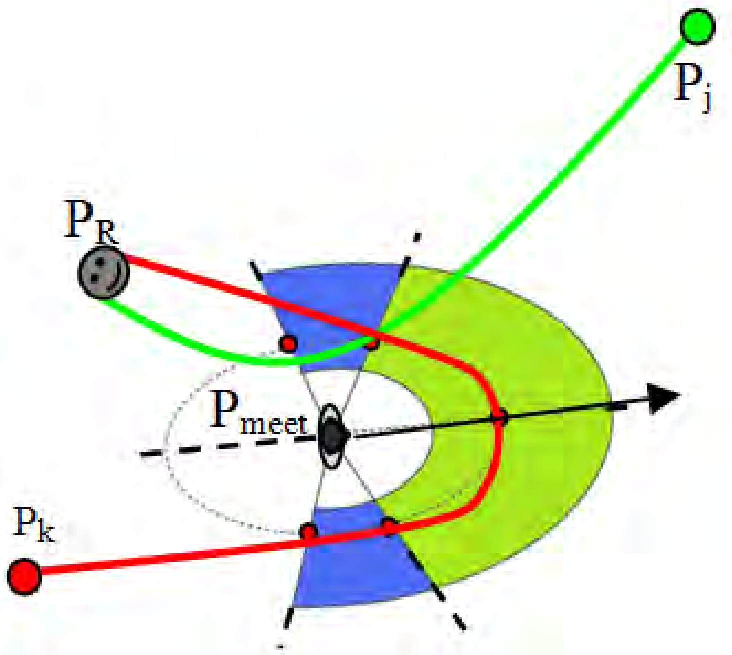


Figure 6.20: The hypothesized expected path for the robot's deviation in two different situations, when goals P_j and P_k fall in different regions, will be different.

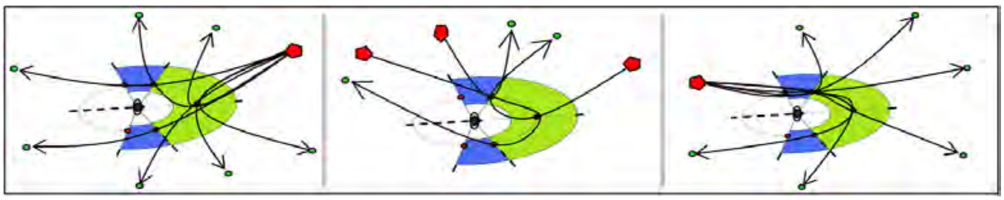


Figure 6.21: Different combination of relative position of the robot (gray circle) and the intermediate goal points (green circles), with respect to the human predicted meeting point, leads to different set of points (red circles) through which the robot should pass during deviation, to show the goal-oriented approaching behavior.

6.4.6.2 Deciding Next Point towards Goal

As the robot path should be smooth, intuitive and goal directed, the robot needs to predetermine the direction in which it should move after reaching to the human. In the simplest case, it will be the final goal point itself. If the final goal is not directly visible from the predicted point of meeting, it will be the farthest visible point on the path from meeting point to the goal. Although one can use any path planners, we will take advantage of our social planner and use node of next visible topological region as the intermediate goal, which converges towards the final goal region. Let us denote the intermediate goal point as P_{IG}.

6.4.6.3 Deciding the set of points to deviate

Let the position of the robot before deviation is P_R. First the robot will find in which region, relative to the human predicted position P_{meet}, the point P_R and intermediate goal point P_{IG} are falling. The possible different regions have been numbered in figure 6.19. Let us take an example that P_R is falling in region 15 and the point P_{IG} is in region 11. And,

to simultaneously satisfy the criteria of supporting the human activity and influencing the human towards the goal (by hinting the goal location and by exerting a kind of social pulling force), the robot should approach/catch the human in such a way, in which the human might not only be able to sense the presence of the robot but also the intention of the robot.

For satisfying the first criterion of supporting the human activity, the robot should approach and pass through any point of accompanying or following regions around the human, as shown in figure 6.18. But, to satisfy the second criterion of conveying the intended motion, the robot should give highest priority to pass through the left side of the human, as the next goal point $P_{IG} = P_j$ is in left of the human. Also since it is in region 11, it should try to pass through the point P_1 of figure 6.19. Let P_{Dev} is ordered list of milestones through which the deviated path should pass. So, in this case, $P_{Dev} = \langle P_1 \rangle$ relative to the human predicted position *Pmeet*. The green curve in figure 6.20 shows such a path passing through the point P_1.

It is not necessary that there will be only one point from where the robot should pass through during deviation. There can be situations, when both P_R and P_{ID} are predicted to fall behind the human. Let us say P_R is again in region 15, but $P_{IG} = P_k$ is in region 16. Then as shown as red curve in figure 6.20, the expected path for the robot, which also takes into account safety and comfort of the human should pass through P_1 towards P_0 and then through P_2 to P_{IG}. So, in this case, $P_{Dev} = \angle P_1, P_2\rangle$. Note that the order of deviation point is important. Similarly, for other combinations of regions of P_R and P_j with respect to the human predicted position, different choices of deviation point will be made, as partially summarized in figure 6.21. Similar set of rules are encoded in a decision tree for all other possible combinations, as done earlier for avoiding a person in a social manner. The middle scenario of figure 6.21 shows a set of important cases, where both the robot and the goal point fall in the same region. In such a situation, the relative angular positions of both with respect to the human play a decisive role in finding the order of points, through which the robot should pass.

6.4.6.4 Generating smooth path to deviate

As clear from above discussion, until reached to P_{Dev}, the robot's priority should be human reaching behavior, and beyond P_{Dev} its priority should be human fetching behavior. Also at P_{Dev} the orientation of the robot should be towards the intended

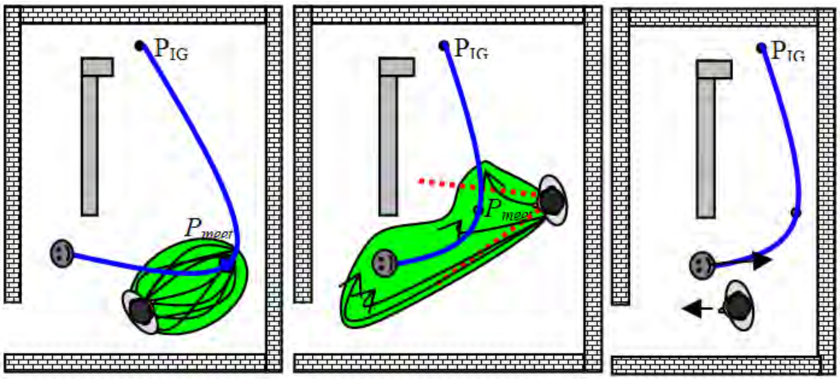


Figure 6.22: Cases of human-robot relative situations during the robot's goal directed approach to re-engage the human. (a) The human cannot see the robot. (b) The robot is in field of view of the human. (c) The human is passing by the robot in the opposite direction.

direction to goal, so that the human can better predict the robot intention and join the guiding loop in a comfortable manner. Also taking into account the human walking pattern [Arechavaleta 2008], it should be a smooth path. Such a desired path by the robot has been already shown in figure 6.20 and figure 6.21. In fact, we need a smooth path, which could join the current position of the robot to the goal position and pass through the set of deviation points P_{Dev}, by showing the above-mentioned properties. To realize such a path we use cubic B-spline curve interpolation through all these points. In fact the curve will be so smooth that even the second derivative will be continuous everywhere. Using B-splines, we will have more control flexibility, in addition, the effect of varying one control point will be local. Hence, further online smooth modification in a small segment of the path is possible, if required, which makes it a very attractive solution, in the case of highly dynamic environment. Further, to ensure smooth adaptation to the new path we use clamped B-spline to incorporate the current velocity of the robot as a constraint while generating the deviated path. However, any other interpolation approach could be used to generate a smooth path through the milestones.

6.4.7 Human Activity to be Re-engaged

If everything works fine, the robot will be able to catch the human in the desired position by following or approaching him, and hopefully the human will re-engage in the guiding process by accompanying or following the robot. The robot state will again change to *mentor* state when the robot will start falling within the desired probability distribution as explained in section 6.4.3.

But for various reasons ranging from error in prediction or change in the intention of the human, it is common that the predicted meeting point P_{meet}, will no longer be valid at some point of time during the execution of the deviated path to approach the human. Hence, the robot may need to deviate again, from its current deviated path. But, again the robot should tolerate the human deviations up to some extent, to prevent from being over-reactive. For this the robot continuously monitors the human motion and classify them as supportive or suspending to the re-engagement effort. We have identified three different cases as shown in figure 6.22:

(I). The robot is behind the human. (II). The robot is in the field of view of the human. (III). The human is passing by the robot in opposite direction from one of its sides.

In Case I, the robot will continuously predict the new trajectory of the human. A different probability distribution $P(P_{meet})$, similar to section 6.4.3 will be calculated as centered at the actual predicted meeting point P_{meet} with the three variables $(x, y, \triangle\theta)$, where x,y are co-ordinates of P_{n_meet}, which is the human's new predicted position nearest to actual P_{meet} and $\triangle\theta$ is its angular position with respect to P_{meet}. As long as the square Mahalanobis distance for $P(P_{meet})$ is decreasing and is predicted to lie within the top 30% probability distribution, the robot will not re-deviate, providing the human with the flexibility to move by choosing any path shown within green region in figure 6.22(a). As long as these criteria are being satisfied, the human is said to be in *supportive state of leave taking*, in the sense eventually he is expected to re-engage in the joint commitment of guiding without requiring the robot to re-deviate.

In Case II, the human can change his intention any time and decide to join the robot at any point within the green region of figure 6.22(b). Thus to provide the human with the flexibility to join the guiding process in the way he wants, the robot will classify the human activity as supportive as long as the square Mahalanobis distance (D1+D2) of eq. 6.3 is decreasing.

Case III could occur in two situations: the human wants to join the robot from its behind or the human is just passing by the robot for visiting some other point of interest. Thus, not to show any reactive behavior in the case, the human wants to resume the guiding process by joining the robot from its behind, the robot will classify the human activity as supporting as long as Mahalanobis distance $(D1+D2)$ of eq. 6.3 is falling within top 55% of the probability distribution. But note that in the case, when the human is intended to join the robot from behind, the human need to take few steps behind the robot, which are also opposite to the robot motion direction, so the robot will relax constraint on the human orientation by increasing the variance of relative angular position to $5\pi/6$, but maintaining or even tightening the constraints on acceptable distance by adjusting the other parameters of Σ. If the human intention was not to join the robot, eventually it will fall beyond the 55% probability distribution.

If the human activity is not classified as supportive under any of the above cases, the robot will assume that the human is trying to further suspend the re-engagement process and the human activity will be classified as suspending. Once the human activity will be in suspending state for some time period, the robot will re-deviate

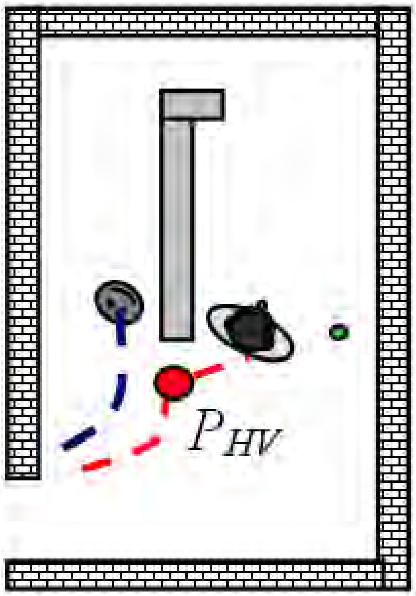


Figure 6.23: A scenario where the human is no longer visible to the robot.

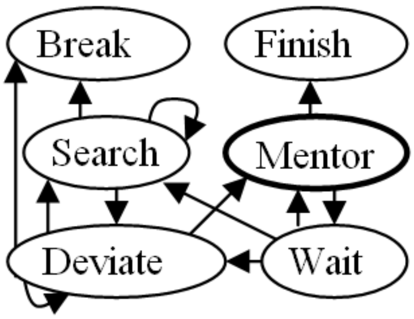


Figure 6.24: Different states of the robot and possible transitions, during the guiding process due to the human deviations. This is an attempt to guide the person in the way he/she wants to be guided.

from its planned path by using the same approach as explained in section 6.4.6.

6.4.8 Searching for the Human

During the guiding process sometimes the human might disappear from the robot's visibility. One of such situations has been shown in figure 6.23. In such cases, the robot will enter into the *search* state and will plan a path to the point P_{HV}, where the human was visible most recently. As soon as the robot again detects the human to be guided, it will re-plan a goal oriented deviated path as explained in section 6.4.6.

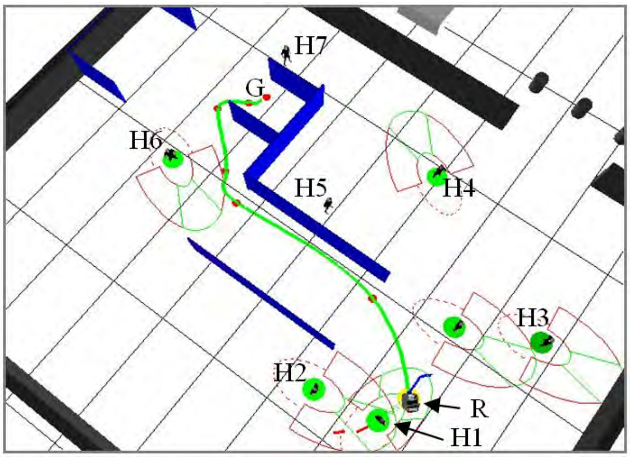


Figure 6.25: The robot JIDO approached the human H1, to start the guiding process. Green path shows the planned path to the goal after the joint commitment of guiding has been established with H1. Note that the initial planned path to guide is also generated using the social planner presented in this chapter, hence, respects the set of social rules.

6.4.9 Breaking the Guiding Process

If the time spent by the human in a particular region is beyond a threshold, the robot could decide to break the joint commitment of guiding. In such situations, the robot will either ask for confirmation or will convey the message of termination by whatever interface it has to communicate with the human. Of course there could be various other criteria for terminating the guiding process, such as critical power situation, other higher priority task, etc.

Possible transitions in the robot's state have been shown in figure 6.24. Note that as explained earlier the robot will dynamically set the values of the threshold for transition from one state to another depending upon the current state of the human and the structure of local environment.

6.5 Experimental Results and Analysis

We have used our developed Move3D software, as implementation and test environment for experimentation, by feeding the models of the real environment, experimental robot Jido and the Human. For each experiment in the simulation, the robot was fully autonomous and equipped with our developed guiding system. The behavior and motion of the human model to be guided is controlled by a real user in real time through an interface. The user was free to move the human towards any other human of the environment, to any other place, stop and wait or deviate in any direction at any moment; hence, able to mimic his own desire in different

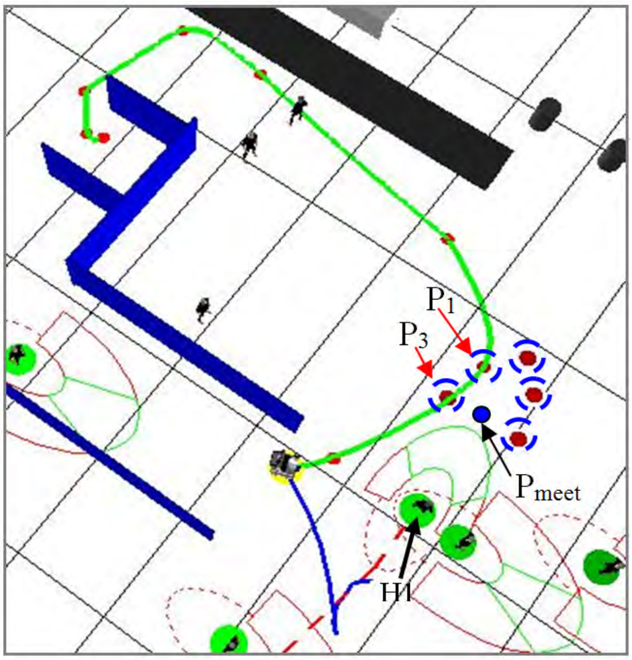


Figure 6.26: The human suspended the guiding process and the robot has planned a new path to approach and influence the human towards the goal, by exerting a kind of social fetching force with its goal directed approaching motion.

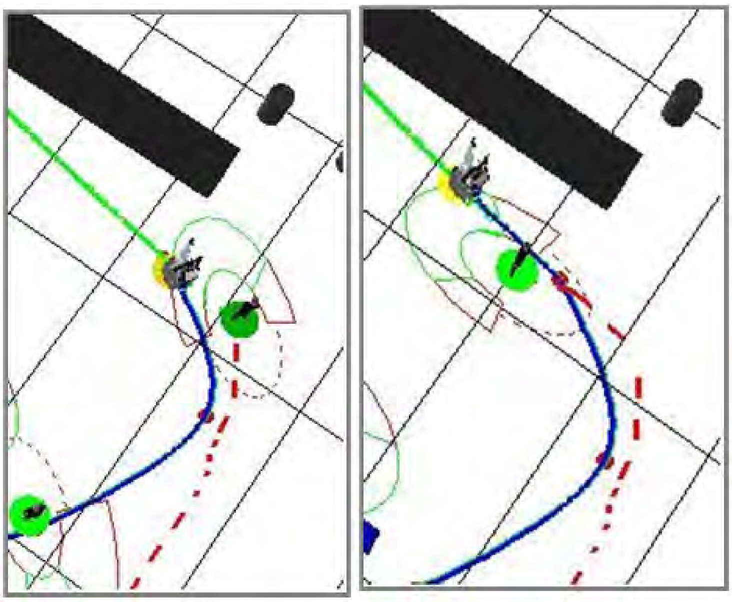


Figure 6.27: The human switches from right to left side of the robot, but the robot did not show any reactive behavior.

situations and environment.

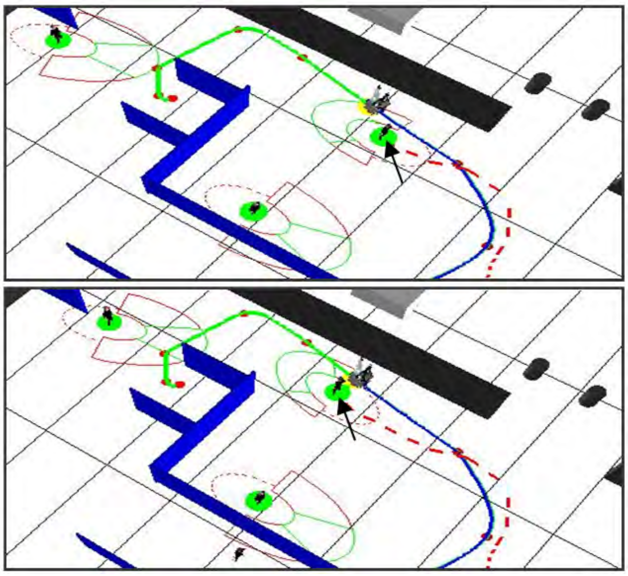


Figure 6.28: The human, indicated by black arrow, switching from behind to the left side of the robot to accompany the robot instead of following it. The robot did not show any reactive deviation in its path.

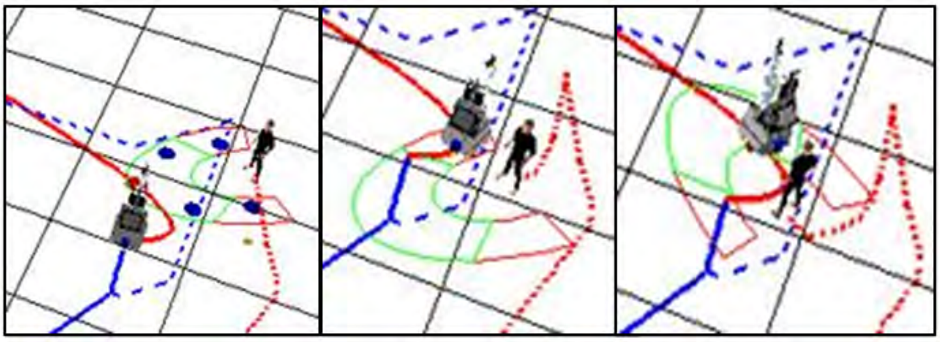


Figure 6.29: The human's re-engagement: The robot predicted that the human wants to join it from its behind so shows no reactive behavior, even if the human is moving in opposite direction of the robot.

For starting the guiding process the robot has approached to the human marked as *H1*, by following the path shown as small blue curve in figure 6.25. The green path shows the smooth trajectory, the robot has generated by using our social planner, to guide the human towards the goal region G. Note that the shape and size of the regions around the person to be guided is different from the regions around other persons in the environment, which are visible to the robot, because for other persons, the robot exhibits a socially accepted human avoidance behavior. Figure 6.26 shows the situation where instead of following to the robot, the human started moving to a new location. As the belief about the human commitment to the joint goal started decreasing below a threshold, the robot started slowing down and eventually it has

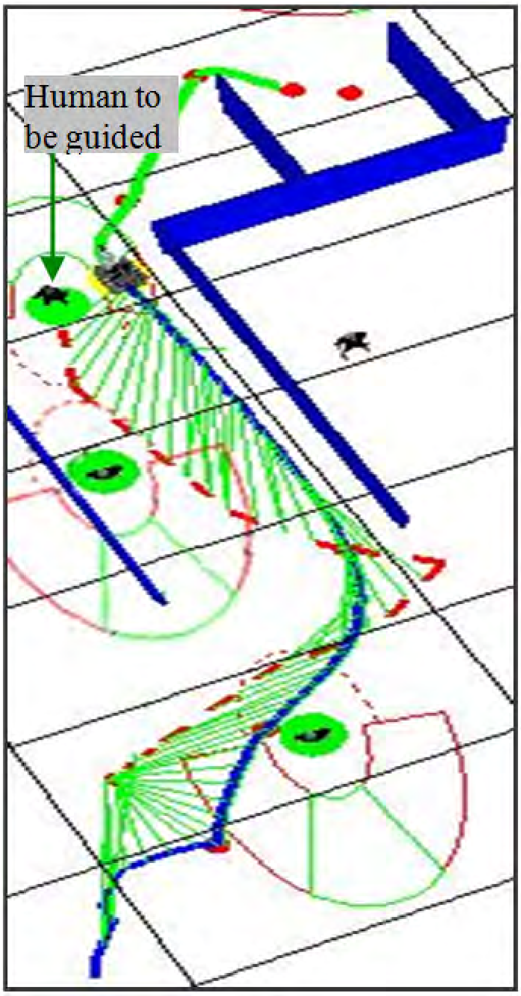


Figure 6.30: Temporal relation between the paths of the human and the guide robot. The robot did not deviate unnecessarily in the case of non-leave taking behavior of the human, hence providing the human with the flexibility to be guided in the way he wants. Note that sometimes the human followed the robot from behind, sometimes he accompanied the robot and sometimes even moved slightly ahead of the robot.

decided to deviate from its path to support such leave taking behavior of the human. Based on the predicted nearest possible meeting position and next immediate goal region for converging towards the final goal, the robot has chosen an ordered set of points, $P_{Dev} = \langle P_1, P_3 \rangle$, shown by red arrows, in figure 6.26, around the predicted position, P_{meet}, of the human, through which the robot should pass. Points P_1 and P_3 could be traced back to those of figure 6.20. Then by B-spline interpolation, it has planned the deviated path shown as green curve, the shape of which clearly depicts goal oriented human approaching behavior. Note that the robot has entirely changed its path to adapt to the human activity while maintaining the task oriented behavior to depict its intention of convergence towards the goal.

The image sequences in figures 6.27 and 6.28 are the continuation of the guiding process, which depict non-reactive behaviors of the robot in different situations, once the human has re-engaged in the joint commitment of guiding. In figure 6.27, the human has completely changed his relative position from right to the left side of the robot. In figure 6.28, the human has decided to be guided as an accompanying person instead of following the robot, so switching from behind to the left side of the robot.

In figure 6.29 the human wants to re-engage in the guiding process by joining the robot from its behind so moving in opposite direction of the robot. But in all of these cases, the robot did not show any reactive behavior, by successfully predicting different non-leave-taking and supporting behaviors of the human, hence, providing the human greater flexibility to decide upon the ways he wanted to be guided or to re-engage in the guiding process, instead of forcing him to exactly trace the path of the robot.

Hence, with our presented framework the robot exhibits neither the over-reactive behavior nor the under-reactive behavior. Finally, figure 6.30 shows the temporal relation between the points on the path taken by the robot and the human during the guiding process. This is the case, where the robot did not deviate from its path even for a single time, because it was successfully able to infer that all the deviations in the human motion are the part of non-leave taking behavior. The red dashed path is taken by the human whereas the blue path is the robots trajectory. The green line segments, joining the points on both trajectories, show the relative position of the human with respect to the robot at a particular time instant. There are few positions where multiple points on the robot trajectory are joining to the single point on the human trajectory. It indicates that the human was standing at that position for some moment. Also from the temporal relation one can easily infer the relative deviations in the human motion during the entire guiding process.

6.6 Until Now and The Next

In this chapter, we have presented frameworks for navigation planning, which is able to take into account a set of rules related to clearance in the environment, human proximity and social constraints. The framework autonomously extracts relevant local structures of the environment and dynamically selects a relevant subset of rules to be applied. Further, we have presented a framework for a robot to guide a person in a social manner. It allows various natural deviations of the person to be guided and shows reactive or re-engagement efforts only when it is necessary. Another novelty of the framework is, any re-engagement effort of the robot is goal-directed, hence, trying to exert a kind of social pulling force towards the goal. We have shown two types of comparative analyses of the presented social planner framework: comparing generated path with respect to the paths produced by typical A^* planner and the Voronoi diagram based path, analyzing the resulting behavior with respect to the purely reactive framework. To our knowledge, it is among the first works in the robot navigation in the human environment, which considers such social norms in its planning strategy, passing by and overtaking a person from appropriate side, proactively maintaining appropriate side in the corridor, avoiding passing through a group of people and carrying out appropriate goal-oriented re-engagements attempts when guiding a person.

This chapter addresses the social and human-aware aspects while planning to navigate in the human centered environment. Complementary to this is to incorporate such aspects while planning to perform human-robot interactive object manipulation task. Next chapter will explore these aspects and present framework to manipulate objects in a human-adapted manner.

Chapter 7

Planning Basic HRI Tasks

Contents

7.1 Introduction 148

7.2 How do we plan 149

7.3 Problem Statement from HRI Perspective 149

7.3.1 Components of a Placement 150

7.3.2 Synthesizing Configuration 150

7.3.3 Generating Trajectory 150

7.3.4 Grasp-Placement inter-dependency 150

7.3.5 A set of constraint classes 150

7.4 Generation of Object Property Database 151

7.4.1 Set of Possible Grasps 151

7.4.2 Set of *To Place* in space orientations 151

7.4.3 Set of *To Place* on plane orientations 152

7.5 Realization of Key Constraints 153

7.5.1 Constraint of Simultaneous Compatible Grasps 153

7.5.2 Visuo-Spatial Constraints on 'To Place' Positions 153

7.5.3 Object alignment constraints from the human's perspective . 153

7.5.4 Robot's wrist alignment constraint from the human's perspective 154

7.5.5 Collision free configuration constraint (CFC) 154

7.5.6 Constraints on quantitative visibility 155

7.6 Framework for Planning *Pick-and-Place* Tasks: Constraint Hierarchy based Approach 155

7.7 Instantiation for Basic Tasks 157

7.7.1 Show an object to the human 159

7.7.2 Make an object accessible to the human 159

7.7.3 Give an object to the human 159

7.7.4 Hide an object from the human 160

7.8 Experimental Results and Analysis 160

7.8.1 Generalized system for different robots: JIDO, PR2, HRP2 . 160

7.8.2 Effect of constraints' parameters variations 172

7.8.3 Convergence and Performance 174

7.9 Until Now and The Next 174

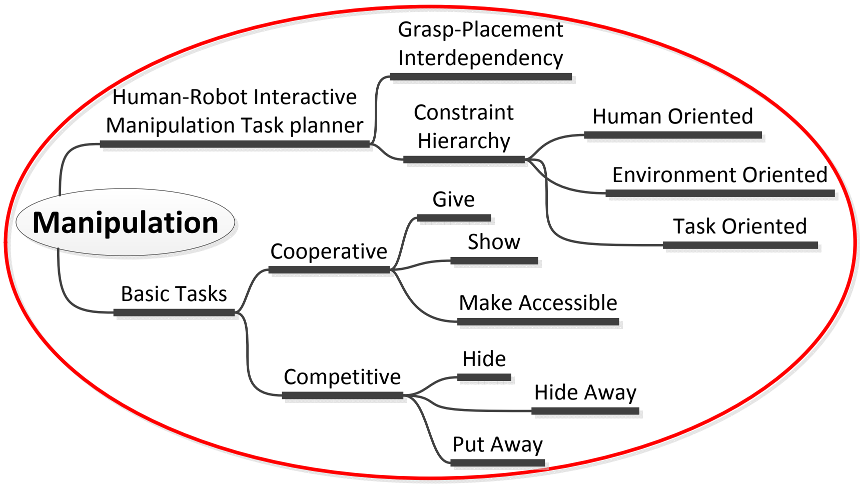


Figure 7.1: Contribution of this chapter: Taking into account *Grasp-Placement Inter-dependency* and introducing *Constraint Hierarchy* based framework for planning *basic cooperative* and *competitive* tasks. This chapter identifies and instantiates various key constraints from the perspectives of the *Human*, the *Task*, the *Environment* and the Planning. Another novelty of the framework is to introduce right constraint at right stage of planning to successively reduce the search space.

7.1 Introduction

In a typical Human-Robot Interaction (HRI) scenario, the robot needs to perform various tasks for the human, hence should take into account human oriented constraints. In this context, it is not sufficient that the robot selects grasp and placement of the object only from the stability point of view. Motivated from human behavioral psychology, in this chapter we emphasize on the mutually depended nature of grasp and placement selections, which is further constrained by the task and the human's perspective. We will further explore essential human oriented constraints on grasp and placement selection and present a framework, which incorporate such constraints to synthesize key configurations to plan basic interactive manipulation tasks.

In the context of HRI manipulation, it is assumed that either the grasp or to place position and orientation are fixed or known for a particular task, [Berenson 2008], [Xue 2008]. In addition, for human to grasp the object at the same time, robot's grasp site is just shifted [Kim 2004] or just enough space is left [Song 2010]. These approaches do not synthesize simultaneous grasps by the human and the robot for object of any shape. Further, they do not reason from the human's perspective for reachability, visibility and on effort levels. Also the set of tasks is limited: hand-

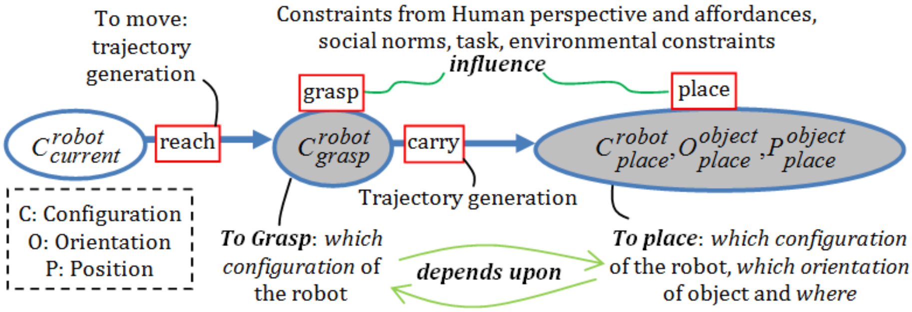


Figure 7.2: A typical *pick-and-place* task. It shows the requirement to synthesize C, O, and P components. It also shows different influencing components and inter-dependencies.

over or to place, [Cakmak 2011], [Bischoff 1999]. In this chapter, we will incorporate all these aspects to develop a generic framework for planning basic Human-Robot Interactive Manipulation tasks, which in fact could serve for complex cooperative task and shared plane generation. Figure 7.1 summarizes the contribution of this chapter.

7.2 How do we plan

As discussed in section 1.1.2 and in section 2.4, we derive following three main points: (i) A target-posture should be found before any movement. (ii) It is important to plan *pick-and-place* as one task, instead of planning and executing them separately. (iii) It is important to take into account the perspective of the human for whom the task is being performed. In this chapter, we will explore *pick-and-place* tasks for HRI manipulation along the similar guidelines by incorporating these discovered aspects.

7.3 Problem Statement from HRI Perspective

We define a task T belongs to class of *pick-and-place* task if:

$$\begin{aligned} \forall T\; T \in pick\_and\_place \; if \; A^T &= (\langle reach, grasp, carry, place\rangle \\ &\quad | \, place \in \{put\_on\_support, hold\_in\_space\}) \end{aligned} \tag{7.1}$$

Where $\langle\;\rangle$ is an ordered list (sequence) of actions. We say a task T belongs to *pick-and-place* class, if the planned action $A^T = \langle a_i \rangle$ (see eq. 3.41 and the associated section for a detailed description of actions and the planning problem) to performed the task is a sequence of *reach*, *carry*, *grasp* and *place* sub-actions. From expression 7.1, it is evident that in addition to 'putting an object on a support', we assimilate 'holding an object in space' also as a place sub-action. Figure 7.2 shows different decisional components of planning *pick-and-place* tasks. We identify the following elements for planning a *pick_and_place* type task in the context of HRI:

7.3.1 Components of a Placement

We further identify that *to-place* an object involves:

(i) P_{place}^{object} i.e. where to place and

(ii) O_{place}^{object} i.e. what should be the orientation of the object.

Together we term them as **Pose** of an object:

$$Pose = \langle orientation\, O, position\, P \rangle \tag{7.2}$$

7.3.2 Synthesizing Configuration

We also need to synthesize the configurations **C** of the robot, either to grasp or to place an object.

Planning for pose and configuration for any agent or object, we term it as **Pose & Config** planning.

7.3.3 Generating Trajectory

Once a pair of *Pose* & *Config* have been obtained or provided, then plan a trajectory between them, which we term as **Traj** planning.

Planning of the above two complementary aspects in the order, i.e. plan the *Pose* & *Config* and then the $Traj$, becomes coherent with the finding (i) discussed in section 7.2.

7.3.4 Grasp-Placement inter-dependency

As shown in figure 7.2, C_{grasp}^{robot}, i.e. *how to grasp* restricts $C_{place}^{robot}, O_{place}^{object}, P_{place}^{object}$ i.e. *how and where* the robot could place the object and vice-versa. Hence, from the perspective of robot task planning also *pick-and-place* should be planned as one task, thus coherence with finding (ii) of section 7.2.

7.3.5 A set of constraint classes

Further, as shown in figure 7.2, we have directly incorporated the finding (iii) of section 7.2 that the robot should take into account various constraints, including the restrictions from the human's perspective (object's visibility, reachability, etc.), affordances (e.g. minimizing human effort), environmental constraints (collision, etc.), task specific requirements (simultaneous grasp, placing on an object, etc.) and so on.

One of the key contributions of this chapter is:

the robot explicitly takes into account its own constraints as well as the constraints, preferences and effort of the human partner and plans for both to autonomously synthesize a feasible instance of $C_{grasp}^{robot}, C_{place}^{robot}, O_{place}^{object}, P_{place}^{object}$.

Then we can use *any* trajectory planner to plan the path between these feasible configurations, such as [Broquère 2010] to obtain a "smooth" trajectory and [Mainprice 2011] to incorporate human oriented reasoning in the complementary **Traj** aspect.

In the subsequent sections, we will first identify the key attributes of the object and the key elements of the various constraint classes identified above from the perspective of *pick-and-place* task. Then we will present the generic planning framework, which will be followed by instantiation for different tasks. **C**, **O** and **P** stand for **C**onfiguration, **O**rientation and **P**osition (see figure 7.2).

7.4 Generation of Object Property Database

The robot maintains geometric information, for each object *obj*, it encounters in its lifetime, in the form of tuple:

$$\mathbf{obj_{prop}} = \langle id, name, 3D_{model}^{mesh}, V_F, V_T, \cup_{h=1}^{n} G_h^{obj}, O_{place}^{obj} \rangle \tag{7.3}$$

V_F and V_T are manually-provided vectors associated to the symbolic front and top of the *obj*. And $O_{place}^{obj} \in \{O_{place}^{obj,plane}, O_{place}^{obj,space}\}$. These parameters are related to object's placement and grasp, which are described below.

7.4.1 Set of Possible Grasps

G_h^{obj} is the set of the possible grasps for hand type h for *obj*. Currently $h \in \{gripper_{\text{robot}} rg, hand_{\text{anthropomorphic}} ah\}$, hence $n = 2$. This set is computed as explained in section 5.2.1 of chapter 5, where we talk about agent-object affordances.

7.4.2 Set of *To Place* in space orientations

For an arbitrary point in space, the set of object's orientations are computed by rotating it around its axes. This set is denoted as $O_{place}^{obj,space}$. Figure 7.3 shows the subset of such placement orientations at an arbitrary point in space.

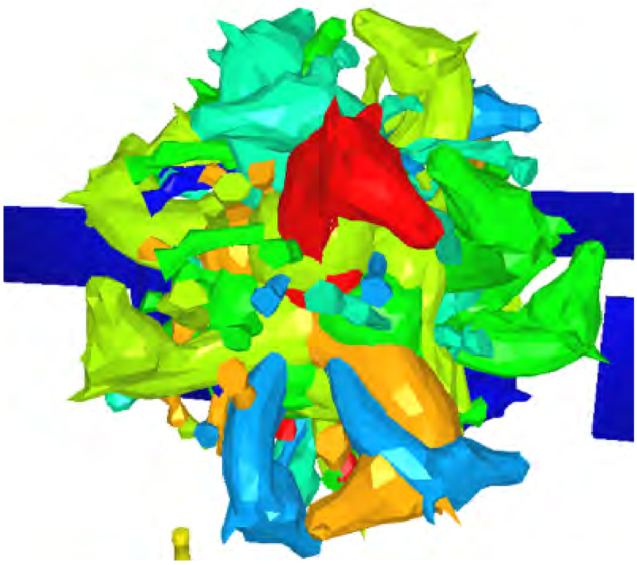


Figure 7.3: A subset of uniformly sampled different placement orientations of the toy horse at an arbitrary point in the space.

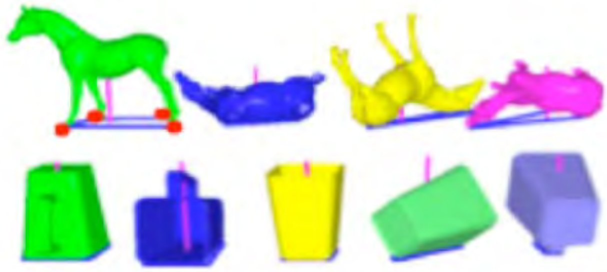


Figure 7.4: A subset of stable placements on an arbitrary horizontal support for toy horse and cup. The vertical line through the center of mass is drawn in magenta.

7.4.3 Set of *To Place* on plane orientations

As explained in section 4.2.2 the robot can autonomously extract any planer top to support an object. The robot generates and stores a set of stable orientations of the object on an imaginary support plane, which is further filtered by the shape of real support during planning. For finding a stable placement orientation on plane following approach is used: As the object's shape is modeled as a polyhedron, the stable placement is defined if the projection of object's center of mass is strictly inside the contact facet f. Contact facet f is a facet of the convex hull of the object, as drawn in blue in figure 7.4. This is 'a' placement orientation O_f based on 'a' contact facet f. Figure 7.4 shows different placement orientations with different contact facets. The robot further enriches a particular O_f by rotating the object along the vertical to get $O^{obj,plane}_{place,f}$. Finally the robot generates the set of all the stable placement orientations for all the f, denoted as:

$$O_{place}^{obj,plane} = \left\{ O_{place,f_{i:i\in[1;number\_of\_contact\_facets]}}^{obj,plane} \right\} \tag{7.4}$$

7.5 Realization of Key Constraints

In this section, we will identify the key constraints for a set of basic HRI tasks and described how those have been realized in our system.

7.5.1 Constraint of Simultaneous Compatible Grasps

To facilitate the object hand-over tasks, the robot should be able to reason on how to grasp so that the human could also grasp simultaneously. A grasp pair $\langle g_{h1} \in G_{h1}, g_{h2} \in G_{h2} \rangle$ is simultaneous compatible SC (see figure 5.3 of section 5.2.1 in chapter 5) if:

$$\begin{aligned} \mathbf{SC(g_{h1}, g_{h2}, obj)}\ if\ \ & (\text{apply}(g_{h1}, obj) \wedge \text{apply}(g_{h2}, obj) \\ & \wedge(\text{collision}(hand(h1), hand(h2)) = \emptyset)) \end{aligned} \tag{7.5}$$

7.5.2 Visuo-Spatial Constraints on 'To Place' Positions

This is to find the positions to *put* or *hold* the object. For this the planner uses the approach presented in section 5.2.4 of chapter 5, for finding candidate places based on the set of constraints $Cnts$ from the task and effort levels of the agents.

This enables the robot to find the commonly reachable and visible places for hand-over task, places to put object for hide task, etc. with particular effort levels of the agents. The set of resultant candidate places for a task is represented as $P_{place}^{obj,Cnts}$, as presented in eq. 5.5.

7.5.3 Object alignment constraints from the human's perspective

A robot should also take into account the symbolic features of the object visible from human's perspective. Hence, the set of possible orientations to place an object at a particular position p is also restricted by this. Figure 7.5 shows human-object relative situation. Blue and green frames represent human's eye and the object. Frame F_P of the object defines V_F as front direction and V_T as top vector. An object is completely aligned to the agent's view if: (i) object's front vector, V_F, points towards origin of the human's eye frame and (ii) object's top vector, V_T, is parallel to human's eye H_z-vector, as shown. Deviation in this alignment could be represented by two parameters Φ and θ, where $\pm\Phi$ is the angle to rotate the object about V_T of F_P followed by $\pm\theta$, the angle to rotate about V_F. The constraint on allowed deviations of the object's front and top from agent Ag perspective is

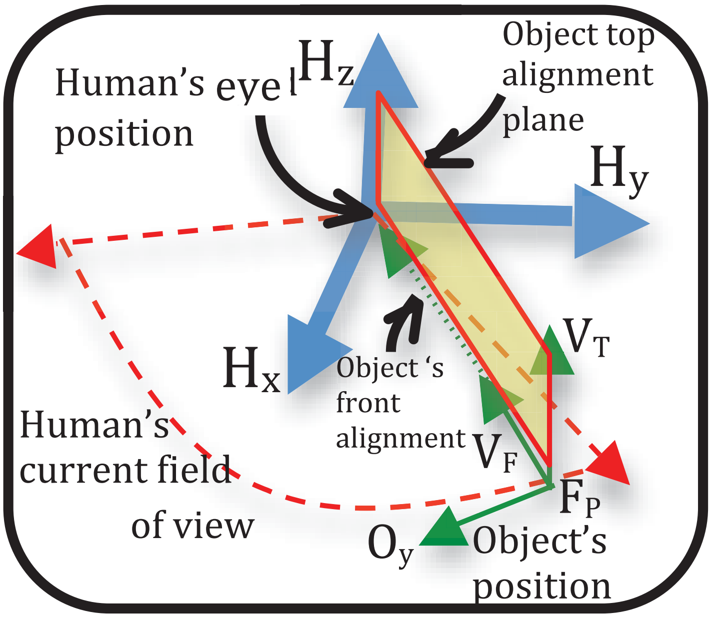


Figure 7.5: Object symbolic features' (front, top) alignments from the human's perspective.

represented as $AC_{Ag}^{obj,\Phi,\theta}$. The resultant set of orientations at a particular position p after applying alignment constraints is denoted as $O_{place}^{obj,p}$.

7.5.4 Robot's wrist alignment constraint from the human's perspective

We define a tuple T^{obj} for an object *obj* as:

$$T^{obj} = \langle grasp\ g, position\ p, orientation\ o \rangle \tag{7.6}$$

The position p to place the object, orientation o of the object at p and the selected grasp g for the object, all together define the wrist orientation of the robot. The constraints on the alignment of robot wrist from the human's perspective is used and denoted as $AC_{Ag}^{w,\Phi,\theta}$.

7.5.5 Collision free configuration constraint (CFC)

For a particular instance of T^{obj} presented above, an inverse kinematics (IK) solver is used to get the collision-free configuration to grasp or place an object, which is denoted as $Q_{grasp|place}^{robot} : (g \rightarrow obj_o^p)$ which reads as *robot's configuration after applying grasp g on object obj placed at p with orientation o.* If the IK solver fails, CFC is said to be unsatisfied in the given state of the environment.

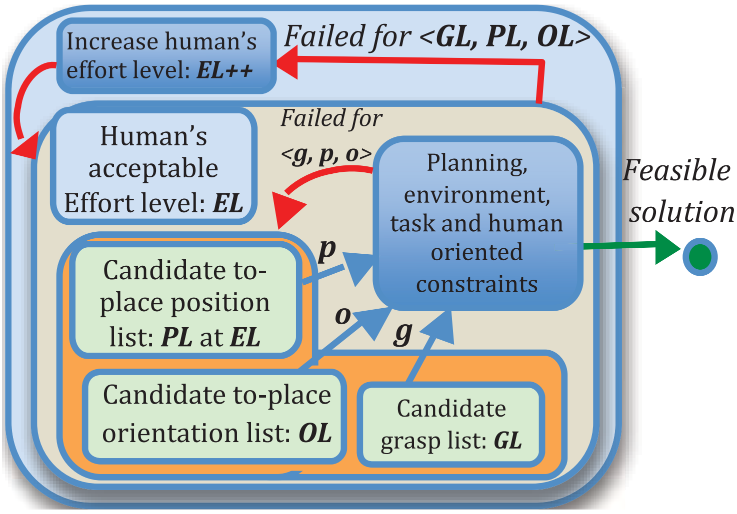


Figure 7.6: Overall planning system, it iterates on 3 candidate lists as well as on human's effort level to extract a feasible solution.

7.5.6 Constraints on quantitative visibility

A visibility score VS of an object obj from an agent Ag perspective is calculated from the eq. 4.3 as presented in section 4.3.1.2 of Mightability Analysis, chapter 4. Acceptable range of VS for a particular task is given as $[min, max]$.

7.6 Framework for Planning *Pick-and-Place* Tasks: Constraint Hierarchy based Approach

Let G be the set of all possible grasps of the object, P be the set of all possible places (3D point), where the origin of the object's frame can be placed, O be the set of all possible orientations in which the object can be placed. Then for a particular object obj, the search space for finding a solution for any task T would be $S^{obj} = G \times P \times O$. Hence, a sub-problem of finding a solution for *pick-and-place* task is to find a $s^{obj} \in S^{obj}$ where:

$$s^{obj} = \langle g, p, o \rangle | s^{obj} \ satisfies \ \forall c \in Cnts^T \quad (7.7)$$

where $Cnts^T$ is the set of constraints, expressed in terms of the key constraints discussed in previous sections.

The key feature of our planning approach is: *introduce right constraint at the right stage*. This is also supported by the posture based motion-planning model of humans [Rosenbaum 2001], which suggests that candidate postures are evaluated and eliminated by prioritized list of requirements called *constraint hierarchy*. This elimination by aspect method [Tversky 1972] has been shown to be effective in modeling

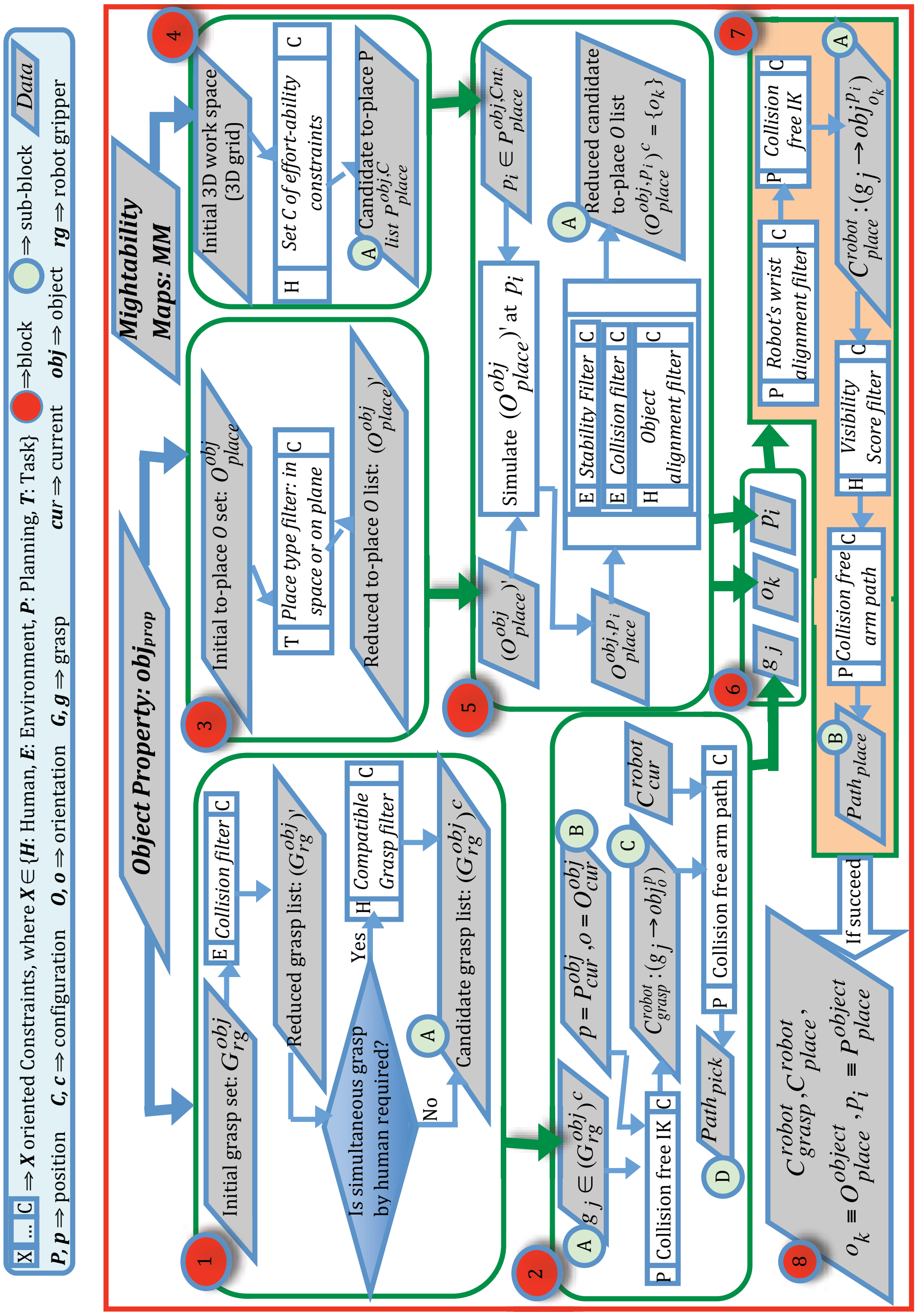


Figure 7.7: Core of the presented generic planner, showing the 4 aspects: (i) How the different candidate lists of figure 7.6 are extracted in *blocks 1-A, 4-A and 5-A*. (ii) How the candidate triplet $<$ **grasp** : **g**, **orientation** : **o**, **position** : **p** $>$ (*blocks 6*), are extracted, which in fact could lead to a feasible solution. (iii) **Constraint hierarchy**: different constraints are introduced at different stages of planning where the search spaces have been reduced significantly. (iv) All the **Pose** & **Config** components required for planning a *pick-and-place* task shown in figure 7.2 have been synthesized, as summarized in *block 8*.

flexible decision making with multiple constraints [Janis 1996]. This serves another important purpose:

Instead of introducing all the constraints at once initially, in the large search space, this approach holds the constraints to be introduced successively at appropriate stages of planning; hence significantly reducing the search spaces before introducing expensive constraints.

We have carefully chosen the *constraint hierarchy* by taking into account the importance of each constraint, their computation complexity and contribution on the reduction of the search space. Highest priority was given to the human's effort level (figure 7.6). The planner extracts candidate list of grasps GL, to-place positions PL and to-place orientations OL starting with the human's least effort. Then successively introduces various *environment*-, *planning*-, *human*- and *task*-oriented constraints at different stages (figure 7.7).

Figure 7.7 details the inner block of figure 7.6 and illustrates how different candidate lists GL (*block 1-A*), PL (*block 4-A*) and PO (*block 5-A*) are extracted. It also shows how a particular instances of T^{obj} for picking (*block 2-A, 2-B*) and for placing (*block 6*) the object are synthesized. *In each green block, if the content at the end sub-block is not* $\emptyset$, *only then the control flows to the next green block, otherwise it iterates appropriately as shown in figure 7.6.* This successive introduction of constraint significantly reduces the search spaces at each step. In *Block 7* further more expensive constraints are introduced on a particular instance of T^{obj}.

Next, the object visibility score at candidate place from the human's perspective is tested. For this, the planner virtually places the object and the robot in their current candidate final position, orientation and configuration. Next, the feasibility of the arm path between the current candidate grasp configuration obtained in *block 2-C*, and the current candidate place configuration obtained in *block 7-A* are checked. In the current implementation, the planner uses [Gharbi 2008] for finding collision-free paths, *blocks 2-D and 7-B*. If the planner succeeds to find the path, it returns with the current candidate **Pose** & **Config**, otherwise it iterates appropriately as shown in figure 7.6. Note that in *block 8*, we achieve our goal of autonomously synthesizing all the **Pose** & **Config** components required for a *pick-and-place* task as shown in figure 7.2. The presented planner is generic in the sense it can find solution for basic human robot interactive manipulation tasks of different natures, when represented in terms of various constraints. Next we will explore such tasks, which are building blocks for complex HRI task planning.

7.7 Instantiation for Basic Tasks

Most of the constraints related to IK, collision, human least effort, etc. are common for the HRI tasks. We discuss below some task specific constraints, provided to the presented planner to get a feasible solution.

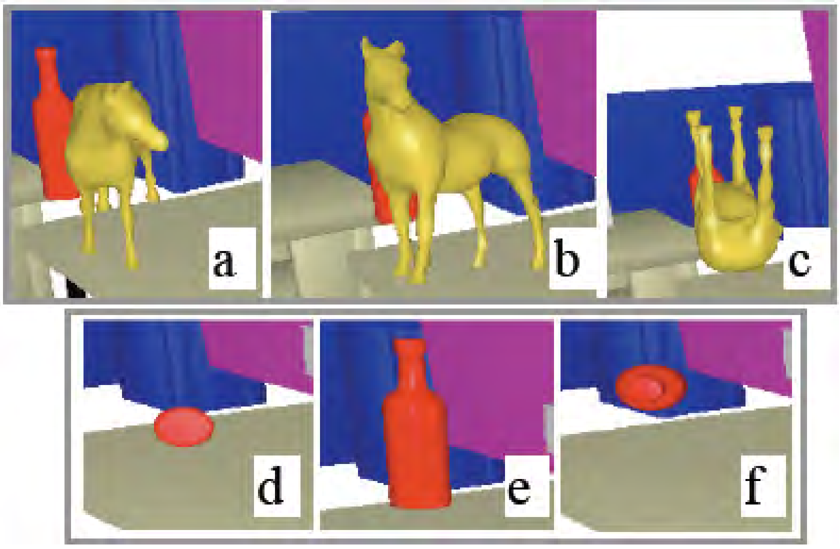


Figure 7.8: Different placement orientations of a toy horse and a red bottle from the human's perspective. The placements (b) and (e) make the objects better recognizable from the human's perspective as they maintain the symbolic features (front, top) and maximal parts of the object visible to the human.

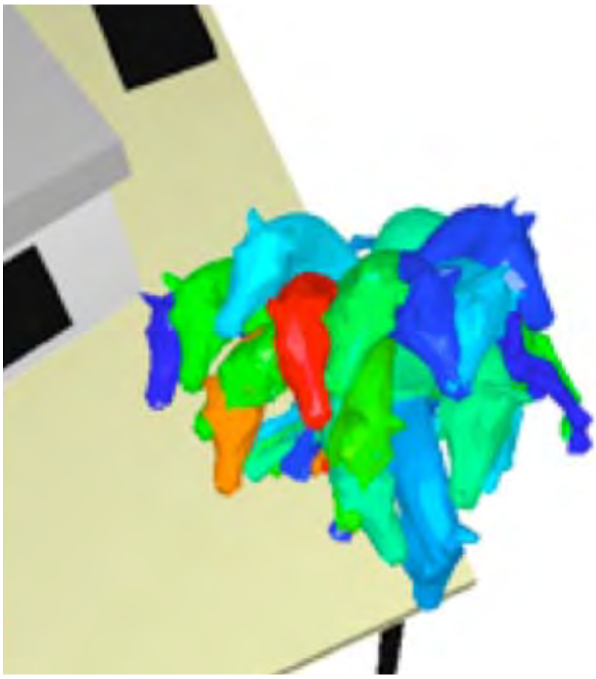


Figure 7.9: Acceptable placement orientations of the toy horse from the human's perspective. Note that in all these orientations, the front is visible and the top is maintained upward from the human's perspective. Blue to Red: Highest to lowest ranking, based on how much part of the horse will be visible.

7.7.1 Show an object to the human

The task requires grasping an object and holding it in a way so that the human can see it with least feasible effort. But it is not sufficient to hold the object in any orientation. As illustrated in figure 7.8, showing the toy horse by placing it in the ways shown in (a) and (c) do not reveal much information about the symbolic feature of the object from human's perspective to identify it correctly, as compared to the one shown in (b). Similarly, for the red bottle, for the same reason the placement in (e) is better than (d) and (f) from the human's perspective. In fact, if there exists a symbolic *top* or *front* of the object, we prefer to maintain that from the perspective of the human, to whom the robot will try to show that object. So, for the task of showing, the constraints on placement are: (i) Front should be visible to the human. (ii) Object should maintain its top upward from human's perspective. (iii) Maximal parts of the objects should be visible.

These constraints could be imposed to the system by providing appropriate parameters of the object's alignment constraint $AC_{Ag}^{obj,\Phi,\theta}$ by allowing a deviation by setting Φ and θ to be $60°$ and then ranking the orientations based on their visibility scores. This value has been chosen arbitrarily to avoid the system to be over-constrained as well as to satisfy the requirements. Figure 7.9 shows the accepted range of object's orientations $O_{place}^{obj,p}$, from human's perspective by using these thresholds, if placed at a particular position p. Note that in all these orientations, the front is visible and the top is maintained upward from the human's perspective. Further, based on the visibility score these orientations are ranked. Blue to red show decreasing order of rankings. The orientations similar to the one shown in figure 7.8(b) automatically get higher ranking because of visibility of relatively larger part of the object to the human.

We also introduce an intuitive constraint to maintain the wrist orientation towards the human. For the same reason of avoiding the system to be over-constrained, we allow a deviation of $\pm 75°$ for the wrist frame.

7.7.2 Make an object accessible to the human

The goal is to place an object, which is currently hidden and/or unreachable to the human, on some support plane so that the human can see and reach it with least feasible effort. Additional constraint on object orientation to maintain the top upright from the human's perspective is imposed for this task.

7.7.3 Give an object to the human

In addition to the constraints of show an object task, the hand-over task imposes the constraint of the simultaneous compatible grasps and reachability by the human with least feasible effort.

Figure 7.10: **Show Object Task**: **Maximally visible orientation, maintaining object's front and top**, PR2 robot is showing an object, in an orientation to ensure its maximal part is visible, while maintaining the front and top of the object from the human's perspective.

7.7.4 Hide an object from the human

The task is to place the object somewhere on a support plane, so that the human cannot see it, with a particular effort level. Unlike the task of making an object accessible to the human, there will be no constraint about maintaining the object upright or reachability by the human.

7.8 Experimental Results and Analysis

The system has been tested using our integrated 3D representation and planning software Move3D [Simeon 2001]. Objects are identified and localized by stereovision-based tag identification system. The human is tracked by Kinect motion sensor. The human's gaze is simplified to head orientation obtained through markers-based motion-capture system. In the figures' captions, the *directly observable* key components are in bold.

7.8.1 Generalized system for different robots: JIDO, PR2, HRP2

The presented framework is general and the robot type is just a parameter to the planner presented. To show this we will illustrate the experimental results on 3 real robots of different structures: JIDO (Single arm mobile manipulator), PR2 (Dual arm semi humanoid mobile robot) and HRP2 (Humanoid robot).

(a)

(b)

Figure 7.11: **Show Object Task**: **Maximally visible orientation, least human effort**, PR2 robot is showing an object, in an orientation to ensure its maximal part is visible and at a place which requires least effort to see the object. Note the difference in final object orientation for different relative human-robot positions.

7.8.1.1 Show Task

In figure 7.10 PR2 shows an initially hidden object to the human. The selected grasp and orientation show the inclusion of the constraints of visibility of object's front while ensuring maximal visibility of the object.

Figure 7.11 illustrates two different scenarios to show an object to the human by PR2 robot. Except the relative position of the human with respect to the robot, rest of the initial world state was almost same. Note that the robot has autonomously decided different global orientation of the object so that maximum part of it is visible from the human's perspective. Figure 7.12 illustrates the show object task

(a)

(b)

Figure 7.12: **Show Object Task**: **Ensuring least human effort**: The HRP2 robot is showing an object at a place which requires the human to put least effort to see the object.

by another robot HRP2.

7.8.1.2 Give Task

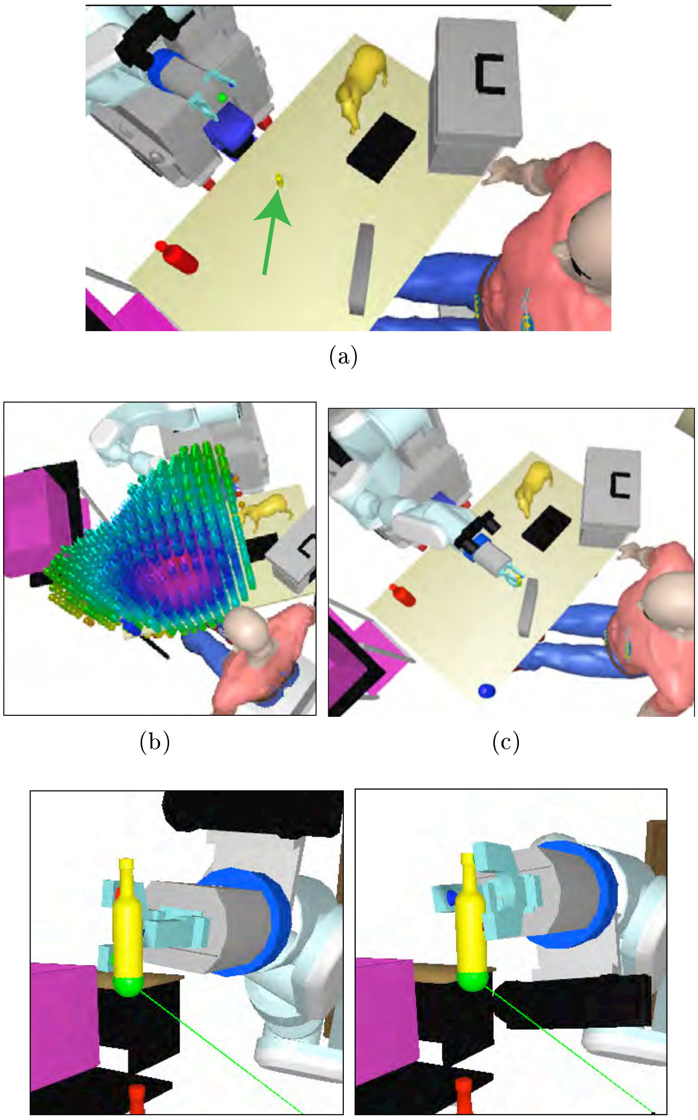
Figure 7.13 shows PR2 is giving an object to the human by maintaining the front of the object and the wrist towards the human. Figure 7.14(a) shows a different scenario in which the robot JIDO is required to give the small yellow bottle to

Figure 7.13: **Give Task: Maintaining symbolic features**, PR2 is giving an object maintaining the object's front towards the human.

the human. Figure Figure 7.14(b) shows the weighted candidate points extracted using the Mightability Maps of both the agents. 7.14(c) shows the robot's final configuration to hand over the object to the human. The interesting fact is shown in figure 7.14(d) and (e), from the human's perspective. For figure 7.14(d), the constraint of simultaneous grasp by the human hand was relaxed. In this case the robot has selected the most stable grasp, at the center of the bottle. But with the simultaneous grasp constraint in figure 7.14(e), the planner selected a different grasp by analyzing the feasibility of simultaneous grasps by the human hand, ensuring space for the human to grasp and take the bottle.

7.8.1.3 Make-Accessible Task

Figure 7.15 shows the case where the planner found a stable placement at the top of an object other than the table plane, because that was the least effort reachable place by the human. Note in a different scenario figure 7.16(a) where initially the toy horse was in a more constrained place and lying by its side, the robot autonomously selected the grasp 7.16(b), which facilitated the synthesized final placement of figure 7.16(c). This final placement is having different orientation than the initial one because of maintaining object's upright constraint. Figure 7.17 shows the sequential *make accessible* task by PR2 for three objects. The robot is able to take into account



(a) (b) (c) (d) (e)

Figure 7.14: **Give Task: Maintaining symbolic features**, **Simultaneous dual grasp**, Jido robot gives small yellow bottle (pointed by the arrow). We deliberately chose the small sized bottled to clearly show the effect of dual grasp computation. (a) Initial scenario. (b) Weighted candidate placement positions of the object. Green least preferred and red most preferred.(c) Final hand-over configuration of the robot. Views from the human's perspective: (d) Without introducing the constraint of dual grasp, Jido grasps the bottle at middle. (e) With introducing the constraint of dual grasp, planner selects to grasp the upper part of bottle by geometrically analyzing the possibility of simultaneous graspability of the human to take it. Note that both placements also satisfy the constraint of maintaining the bottle upright from the human's perspective.

(a)

(b)

Figure 7.15: **Make Accessible Task**: **Reasoning on the human's effort levels, stable placement on non-table plane**, (a) JIDO is picking and (b) making the object accessible by placing it on the white box, so the human can take it with least feasible effort.

the changes in the environment due to its previous actions and synthesizes different feasible placements while maintaining various constraints: stability, visibility, reachability, least feasible human effort, etc.

Figure 7.18 shows another scenario of making two objects accessible to the human sitting in a different relative position than figure 7.17. The robot is able to take into account the changes in the environment due to its previous actions and synthesizes a different feasible placement for the second object. It found a stable placement on the top of the box, as that was the feasible position to ensure least possible effort of the human. As shown in figure 7.18(e) and (f) the human is now able to see the

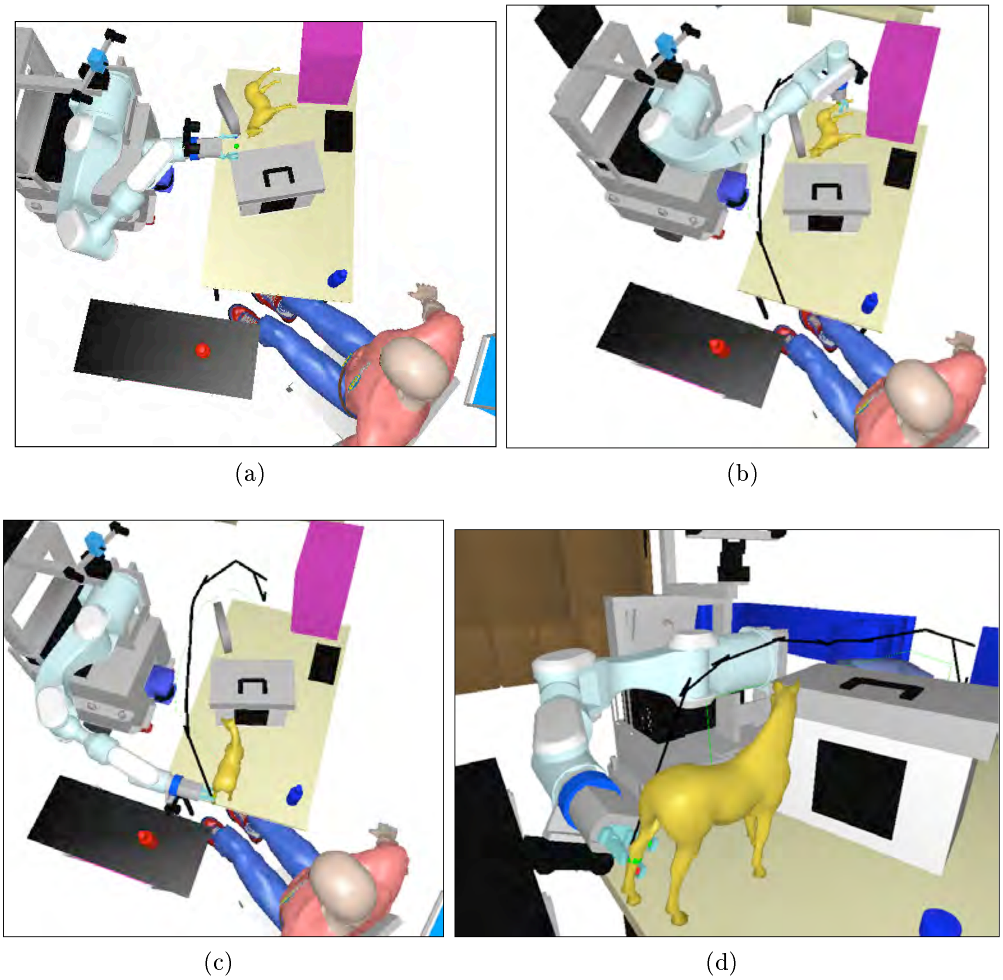
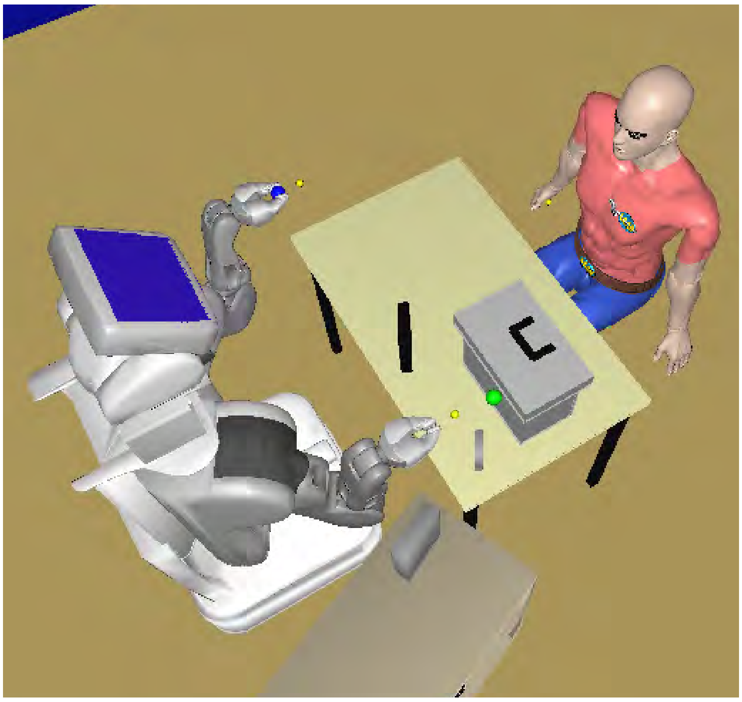


Figure 7.16: **Make Accessible Task**: **maintaining upright, grasp-placement interdependency**, JIDO is making accessible the toy horse. (a) Initial scenario. (b) Autonomously selected feasible grasp in the constrained environment, which facilitates the final placement with various task and planning oriented constraints. (c) Autonomously synthesized final placement by maintaining the constraint of stability and placing upright. Therefore, even if the toy horse was initially lying by its side, the robot has finally placed it in standing position. (d) Final placement from the human's perspective.

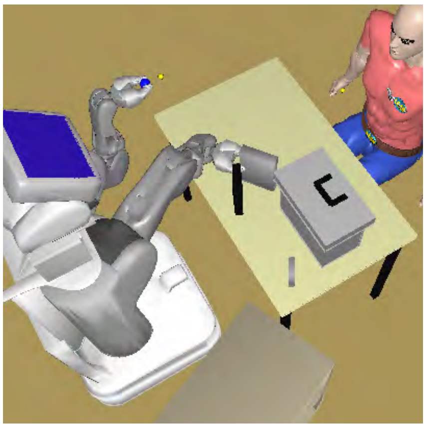
object without any effort and reach the object just with *Arm_ Torso_ Effort*.

7.8.1.4 Hide Task

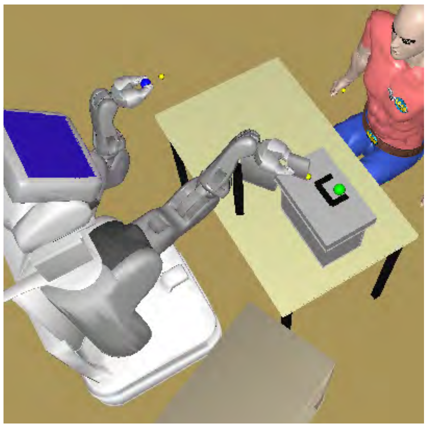
Figure 7.19 and figure 7.20 show the results of hiding two different objects in different situations. It is interesting to note that, due to the non-visibility constraint from the human's perspective, the planner discovered that no orientation is allowing the



(a)



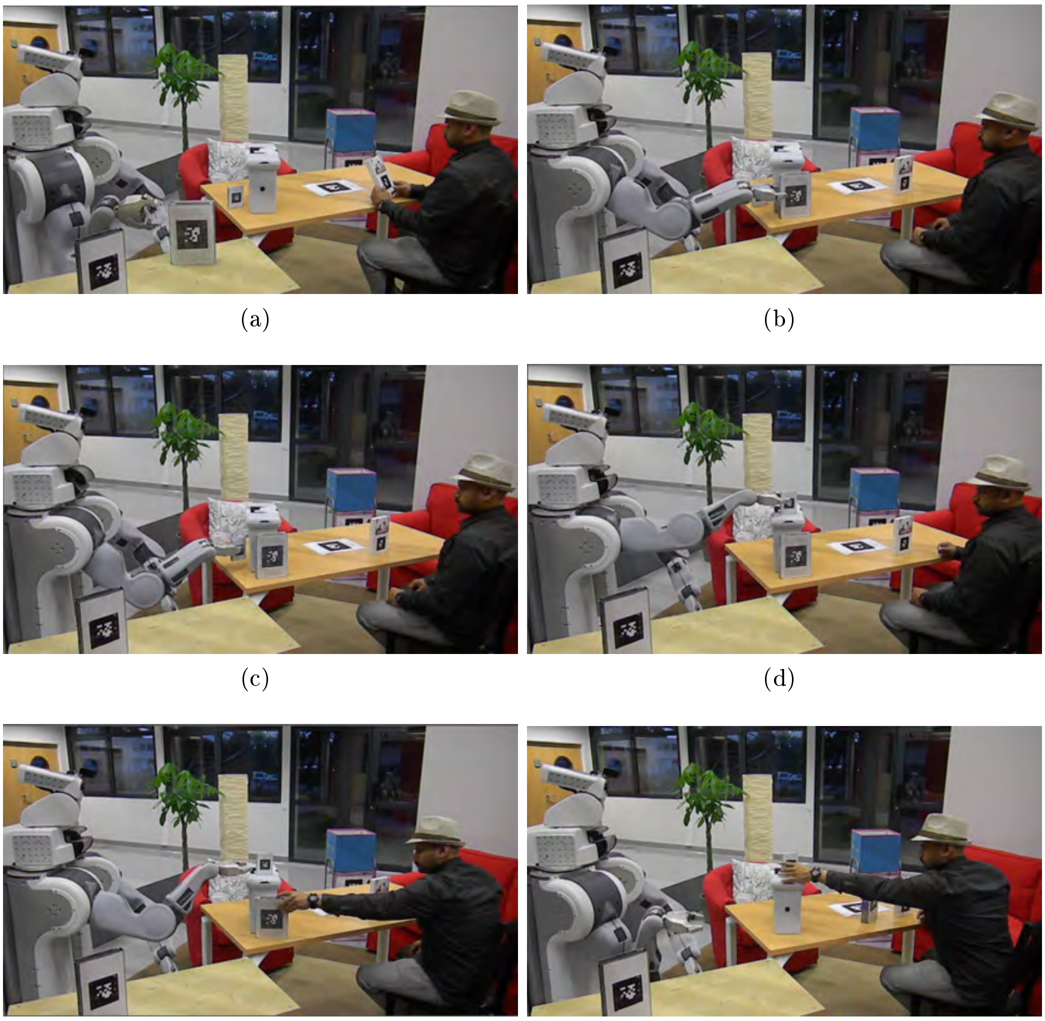
(b)



(c)

(d)

Figure 7.17: **Make Accessible Task**: **Different placements for same task, taking into account changes in the environment by previous actions, stable Placement on the top of other object**, PR2 is sequentially making accessible three different objects. (a): Initial positions. (b) Making first object accessible at the feasible place. (c) Making second object accessible by synthesizing a new feasible placement on the top of the box by taking into account the changes made by its previous action. (d) View from the human's perspective. The robot has just made the third object accessible by synthesizing a placement next to second object on the top of the box.

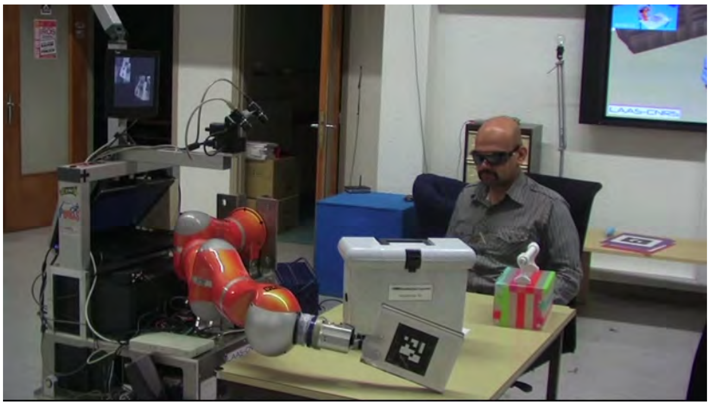


(a) (b) (c) (d) (e) (f)

Figure 7.18: **Make Accessible Task**: **Stable placement on the top of other object, ensuring least feasible human effort**,*(a)-(f)*: PR2 is sequentially making accessible two different objects to the human, who is sitting in relatively different position than the figure 7.17(a). (a) The first object, which is on the right of the robot is not reachable to the human from his current position and the second object, which is behind the box from the human's perspective is not visible to the human from his current position. (b) The planner autonomously finds a feasible placement for first object. (c)-(d) By taking into account the changes made by its previous action, which leaves no space to place the second object on the table top while ensuring least feasible human effort, the presented framework autonomously finds a stable placement for the second object on the top of the box. (e)-(f) Now the human can easily see both the objects and also take just with *Arm_ Torso_ Effort*.

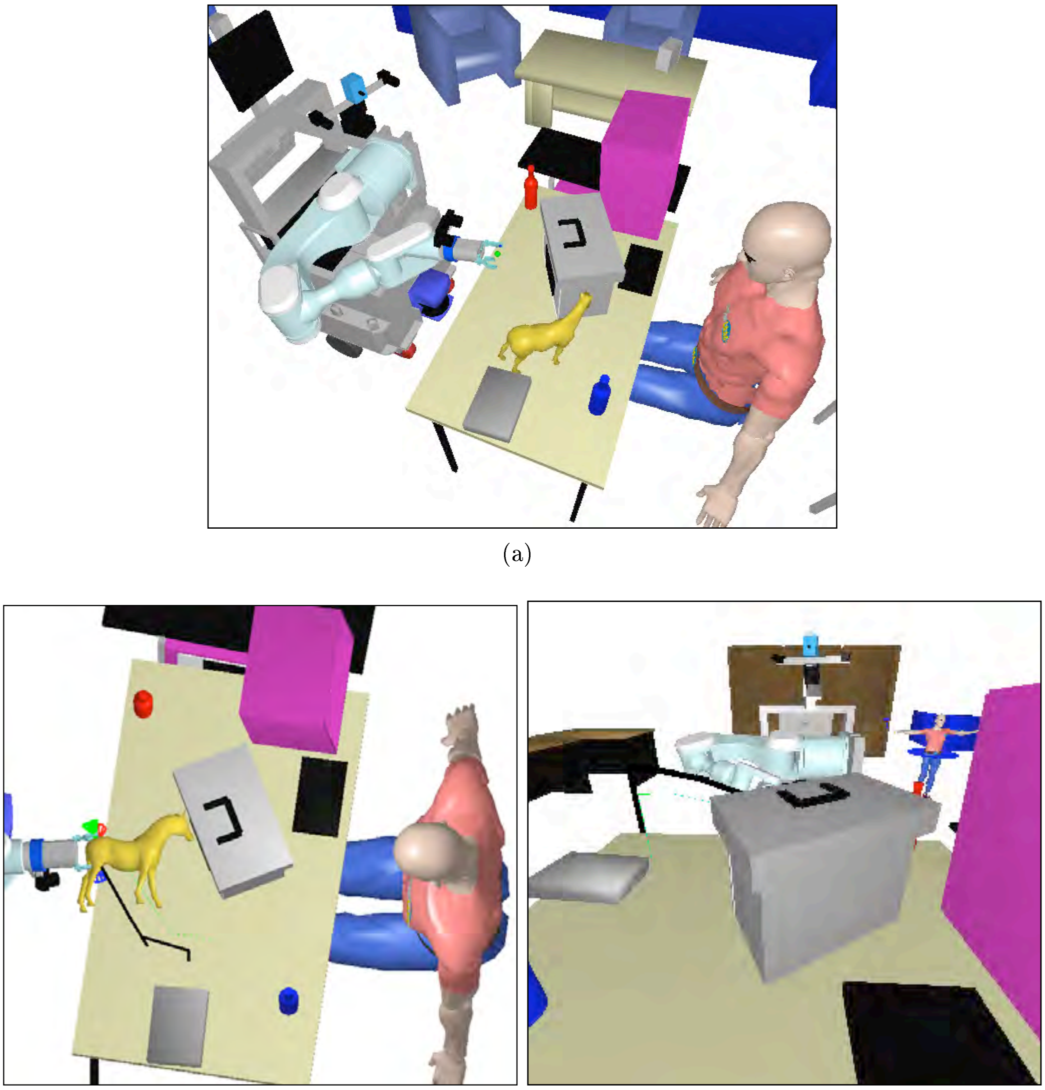
(a)

(b)



(c)

Figure 7.19: **Hide Object Task**: **Grasp-Placement Interdependency**, The planner found feasible object orientation different than its initial one, (a) Initial Scenario. (b)The selected grasp, which facilitates the final placement. (b) The final placement which places the object by a different contact facet to make it completely hidden from the human's perspective.

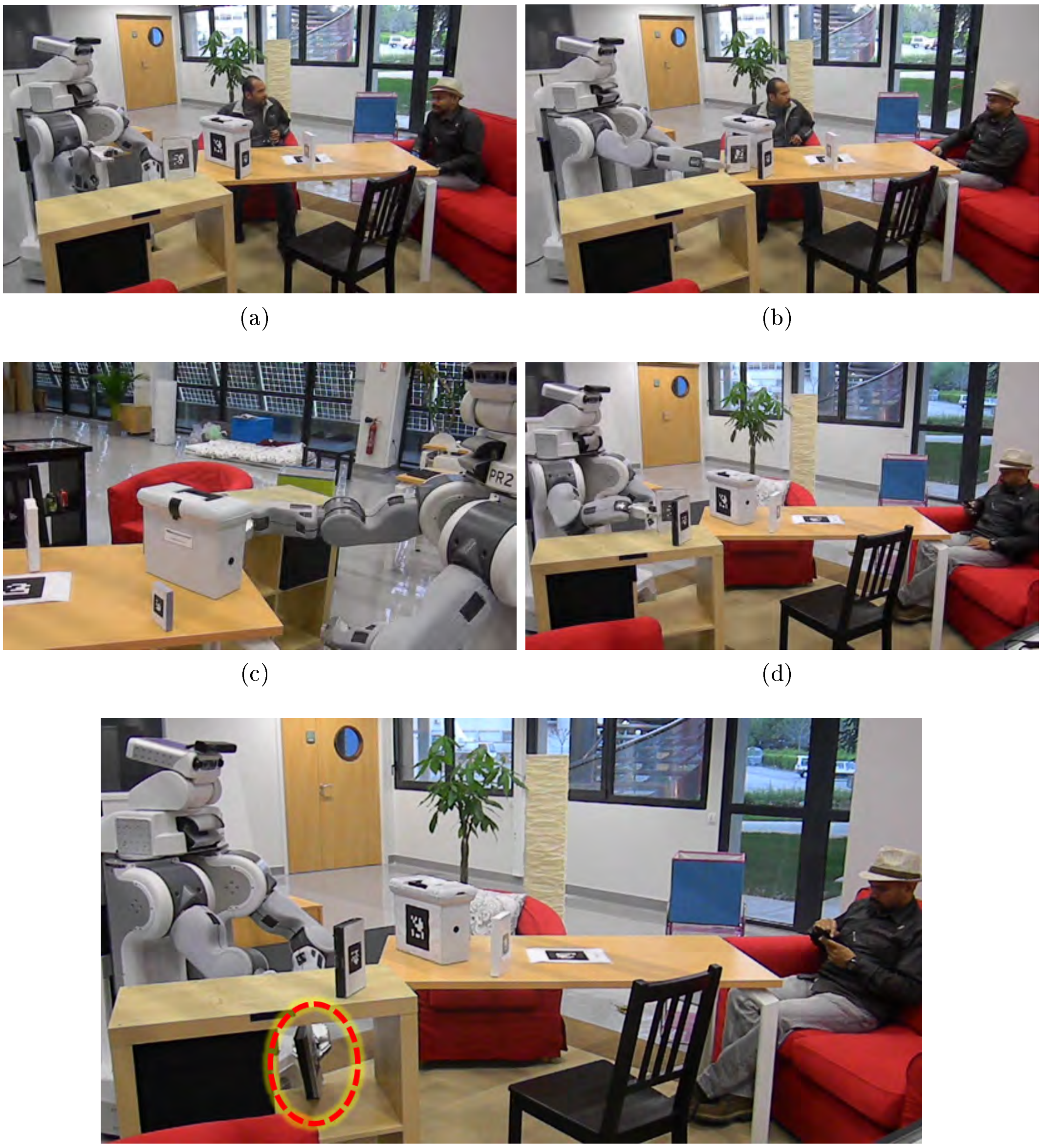


(a)

(b) (c)

Figure 7.20: **Hide Object Task**: **Grasp-Placement Interdependency**, The planner found feasible object orientation different than its initial one, so that the object will be completely hidden from the human. (a) Initial Scenario, the toy horse is standing upright and visible to the human. (b) The robot puts the toy horse such that it lying by its side. (c) The toy horse is completely hidden from the human's perspective.

object to put upright to hide and finds a different final to-place orientation. Further, it selects the grasps, which facilitate to put the objects lying by its side on the table different from their initial supporting facets; hence clearly shows the grasp-placement inter-dependency. Figure 7.21 shows PR2 hiding objects in two different situations from two different humans. Note that in figures 7.21(d) and (e), interestingly the robot has autonomously found a stable placement to hide inside the shelf so that the



(a) (b) (c) (d) (e)

Figure 7.21: **Hide Object**: **Different Placements to hide, ensuring more human effort to see**, *(a)-(e) top-down*, (a)-(c) PR2 is hiding an object from the human sitting in the middle. Note that as shown in (c) the object is completely hidden from the human's perspective. (d)-(e) Shows the case of hiding the object from the human on the right. The robot has autonomously found a stable placement to hide inside the shelf so that the human will be required to put maximum effort to see it.

human will be required to put maximum effort to see it. Figure 7.22 shows HRP2 hiding an object from the human.

(a)

(b)

Figure 7.22: **Hide Object**: HRP2 robot hiding an object from the human.

7.8.2 Effect of constraints' parameters variations

Figure 7.23 shows effect of parameter variation in a different scenario with JIDO robot. It shows interesting variations of the grasp and the final placement based on the changes in the parameter's value for the task of showing a toy horse by the Jido robot to the human. Hence, also illustrating the inter-dependency nature of grasp and placement. See the caption for the description.

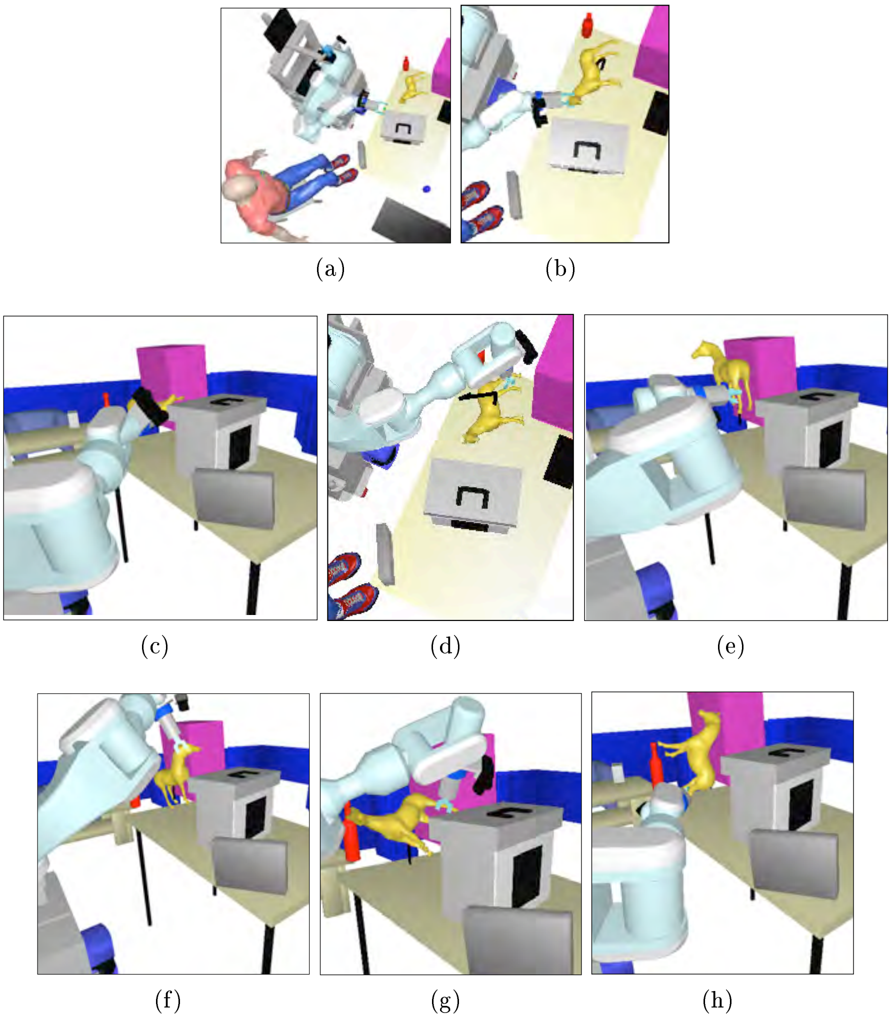


Figure 7.23: **Show Object task**: **Effect of variations on parameters' values**, JIDO shows the toy horse. (a) Initial scenario. (b) Selected grasp of higher stability for the case: the constraint on the visibility score of the object at final placement was relaxed. (c) View from the human's perspective, the object is placed just based on the visible position in the space. The final configuration of the robot itself hides the object from the human. (d) With the constraint on visibility score and to maintain the top upright from the human's perspective. The planner selected a different feasible grasp. (e)-(h): Views from the human's perspective. (e) Final placement for case (d). (f) Final placement with the additional constraint of maintaining the object's front towards the human. (g) Final placement when the constraint to maintain the top was relaxed up to a greater extent. (h) Final placement when the constraints to maintain the top as well as the front were relaxed up to greater extent. Hence, we can clearly observe that ***the constraints and their values restrict the final placements, which effects the initial grasp.***

Also observe that final placements are avoiding exact alignment of the front or back of the toy horse towards the human, as due to the constraint of maximal visibility such orientations are ranked lower. See the figure caption for detail.

7.8.3 Convergence and Performance

As it is based on iterative search, the planner will always converge to a solution if there exists one in the discrete search space. The computation of sets of grasps and placement orientations are one-time process, and do not contribute to the runtime complexity. In the presented results, the computation time varies from 0.5s to 1min, depending on the complexity of the environment. But it could be reduced by further optimizations, such as by adding a 'memory' to the system to quickly converge if a similar situation had been encountered earlier. Future, the iterations due to failure of collision-free IK or path tests are time-consuming. For quicker convergence, an approximate path could be planned during the feasibility test by delaying the planning of complete collision-free path until execution.

7.9 Until Now and The Next

This chapter is an attempt to fuse HRI and Manipulation planning. We have identified a set of constraints, which must be taken into account for planning basic Human-Robot Interactive object manipulation tasks. Further, we have presented a framework, which takes into account the important notion of grasp-placement inter-dependency and which is able to incorporate a set of constraints from the perspective of human, environment and the task. To our knowledge it is the first planner to consider this type of rich human-oriented constraints and grasp-placement inter-dependency for planning object manipulation tasks for HRI context. Another novelty of our presented approach of constraint hierarchy based search space pruning is it introduces only the relevant constraint at appropriate state of planning. Hence, sequentially reducing the search space before introducing computationally more expensive constraints. We have shown that the parameters of the constraints can be adapted depending upon the requirements and could result into different solutions. We have demonstrated that the framework is generic, by planning for a set of tasks: show, hide, give, make-accessible, on three different robots: HRP2, PR2 and Jido.

Until now, we have demonstrated the basic capabilities of planning for interactive object manipulation tasks by one agent for another agent. This makes the robot ready to move to next level and develop the pro-social cognitive and behavioral capabilities as shown in figure 1.1 of our pyramid. Next chapter will use the contributions until now to develop frameworks to equip the robot with such grounding and cooperation capabilities.

Chapter 8

Affordance Graph: an Effort-based Framework to Ground Interaction and Changes, to Generate Shared Cooperative Plan

Contents

8.1 Introduction . . . 176

8.2 Incorporating Effort in Grounding and Planning Cooperative Tasks . . . 178

8.3 Decision on Effort Levels . . . 179

8.4 Taskability Graph . . . 180

8.5 Manipulability Graph . . . 183

8.6 Affordance Graph . . . 185

8.7 Computation Time . . . 188

8.8 Potential Applications . . . 189

8.8.1 Grounding Interaction, Agent, Action and Object . . . 190

8.8.2 Generation of Shared Cooperative Plan . . . 190

8.8.3 A remark on planning complexity . . . 196

8.8.4 Grounding Changes, Analyzing Effects and Guessing Potential Action and Effort . . . 198

8.8.5 Supporting High-Level Symbolic Task Planners . . . 202

8.9 Two Way Hand Shaking of Geometric-Symbolic Planners . 202

8.9.1 The Geometric Task Planner . . . 202

8.9.2 The Symbolic Planner . . . 205

8.9.3 The Hybrid Planning Scheme . . . 205

8.10 Until Now and The Next . . . 209

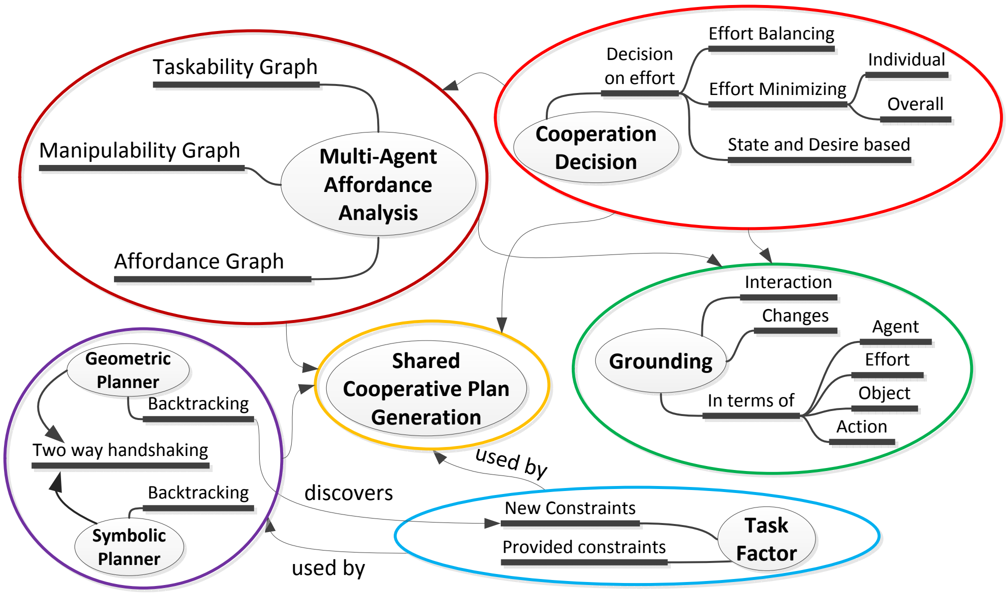


Figure 8.1: Contributions of this chapter: in terms of introducing the concept of *Taskability Graph*, *Manipulability Graph* and *Affordance Graph* to encode different possible affordances in the environment with the provision to incorporate different decisions on efforts; in terms of grounding environmental changes; in terms of providing a framework to plan for complex tasks through human-robot cooperation, by communicating between symbolic and geometric planner and provision to backtrack at different levels.

8.1 Introduction

Previous chapter deals with performing basic tasks between two agents. Very often, the agents have to solve more complex tasks by planning a series of such basic tasks. Further, the robot should be able to ground the interaction, in terms of the object referred, to ground the changes, which might occur in the absence of the robot, in terms what, who, how aspects.

For all these, one common mechanism is to reason about the affordances of all the agents in order to solve a task planning problem by involving more than two agents. Whether it is to give something to some agent by the help of a third agent, or to cooperatively clean a table by putting all the objects in the trashbin, or to ground what the agent is referring, or to guess what has changed and who might have done those changes, the robot should be able to reason about different possibilities by generating a set of actions not only by planning for itself but also for all the agents in the environment.

At symbolic level, this is solved through Hierarchical Task Network based planners such as [Alili 2009] but their main focus is to decompose a complex task in terms of symbolic sub tasks and actions to achieve the desired effect. However, a rich sub-symbolic and geometric counterpart is absolutely necessary to reason on task's feasibility and that too from multiple effort levels of the agents', instead of just relying on the agents' current states. This chapter will complement the existing symbolic task planning approaches by providing a rich geometric and sub-symbolic counterpart and elevating the geometric task planner to a level where it can be beneficial to high-level symbolic planner.

As discussed in the section 2.5 of chapter 2, we are interested in elevating such grounding and shared task planning capabilities by incorporating a rich set of affordances, by incorporating the notion of effort and by enlarging the domain to multi-agent context. By doing so, a subset of grounding problems becomes the planning problem among different agents with different efforts. In this chapter, we are interested in the problem of grounding the interaction and changes, and shared planning through a rich perspective taking mechanism.

This chapter will first illustrate the necessity of considering efforts while planning for cooperative manipulation tasks, grounding interaction and changes. Then we will discuss how individual and context based preferences, social norms, social status, agent's state could be incorporated in terms of effort levels to influence the decisions of grounding interactions, grounding changes in the environment and generating shared plans.

Then we present a novel framework of *Affordance Graph*, which will convert such decision making, grounding and planning the cooperative object manipulation task problems into graph search problem and facilitate the realization of alternative feasible plans with the possibility to incorporate agent oriented and global constraints. *Multi-Agent Affordance Analysis*, *Cooperation Decision* and *Grounding* blocks of figure 8.1 show this contribution of the thesis.

Second half of this chapter will focus on the another contribution of this chapter, which facilitates the robot to solve complex tasks (planned as a series of sub-tasks) by communication between symbolic and geometric planners, which we call as *two-way handshaking*. The novelty of this approach is, each of the planners maintains a local solution planned for the task and backtracks at its level to find alternative plans. Such backtracking might be needed in various cases e.g. the current plan turns out to be not feasible because of change in the environment or because a new action/sub-task could not be validated. The rest three block of figure 8.1 shows this contribution of the thesis. This contribution of the thesis also elevates the geometric trajectory planner component of the traditional systems with a geometric task planner developed earlier in this thesis. Hence, allows to search for alternative solutions of a basic action at geometric level, not just a trajectory. We will show through example that such enhancements facilitate to solve a task without unnecessary flooding of fail reports to the symbolic planner as well as avoid the symbolic

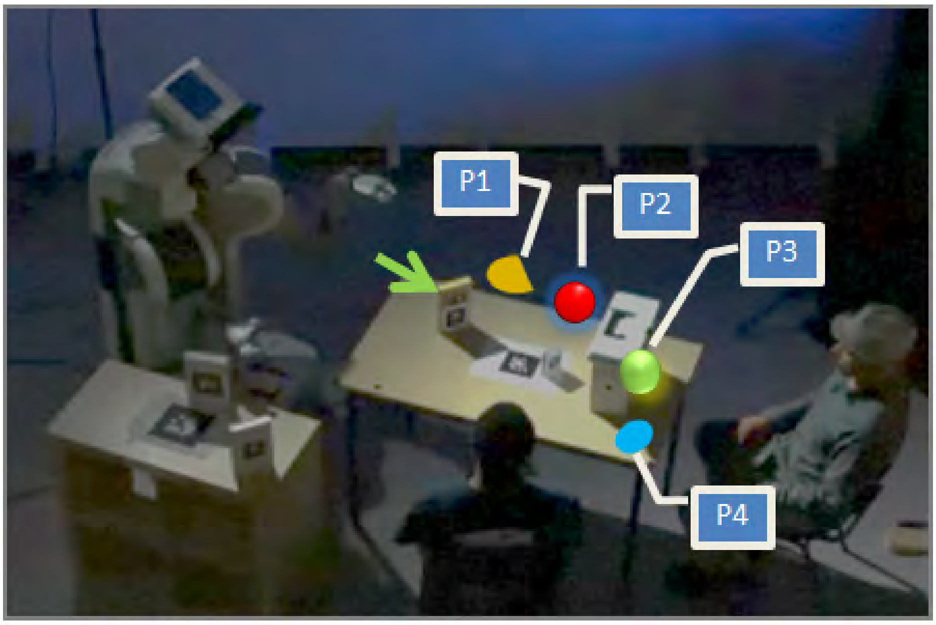


Figure 8.2: The robot's rich reasoning capability about the possibilities of the Human on the right to 'have' the object (marked by arrow), based on Mightability Analysis and Affordance Analysis, the possibilities could be: (i) The human can move to *P1* and take it. (ii) The robot can hold it at *P2* and the human can take it by leaning forward. (iii) The human in the middle can give it to the human on the right at *P3*. (iv) The human in the middle could make accessible to the human on the right by putting at *P4*.

planner to bother about the details related to geometric constraints and parameters of the task.

8.2 Incorporating Effort in Grounding and Planning Cooperative Tasks

As long as the robot reasons on the current states of the agents the complexity as well as the flexibility for reasoning about grounding and generating cooperative plans are bounded in the sense if the agent cannot reach an object from current state, it means that agent cannot manipulate that object, similarly if the agent cannot give an object to another agent it means he/she/it will not do so. But thanks to the Mightability Analysis (see chapter 4), our robot is equipped with rich reasoning of agents' abilities from multiple states. This introduces another dimension: *effort in the grounding and cooperative task planning*, as theoretically every agent would be able to perform a task, only the effort to do so will vary.

As shown in figure 8.2 the robot not only knows that the human on the right (*Human*

1) cannot take the object (marked by arrow) from his *current position*, but also is intelligent enough to know that he can 'have' the object (i) if he will *stand up* and *move to* position *P1*, or (ii) if the robot will *give* him the object at position *P2*, or (iii) if human in the middle (*Human 2*) will *give* the object at position *P3* or (iv) if *Human 2* will *make the object accessible* to *Human 1* by putting it at a place *P4*. Thanks to *Mightability Analysis* (chapter 4) and *Affordance Analysis* (chapter 5), the robot could not only know all these possibilities, but also the associated effort level of the agents as well as the candidate places and positions. Hence, the question of incorporating *efforts* in generating any shared/cooperative object manipulation task, in reasoning about *what*, *who* and *how* aspects behind some change or in grounding *what* one agent is referring, is pertinent.

8.3 Decision on Effort Levels

In this chapter, we will confine the scope of a task involving cooperative object manipulation. As being a social agent the robot will be expected to help, cooperate and collaborate and generate shared plans to distribute the workload and balance/reduce the efforts. One such criteria of deciding about effort could be social status of each agent. In most of the examples, we assume each agent including the robot is having same social status, and hence they will be expected to be *mutually responsive* and *mutually responsible*. Hence, any task has to be performed as an attempt to *effort balance*. This means the *performing agent* will not behave like a slave, it will expect the *target agent* (for whom the task is being performed) also to be involving in the joint goal achievement by putting some effort. However, to be polite and prosocial and to be informative, the *performing agent* could put a little more effort than the *target agent*.

The effort level could further be restricted based on the current context, for example, if all the agents are sitting around a dining table then they will not prefer to stand up and move to perform some task, if there exists some better alternatives. In this scenario, the maximum desired effort level of the agent could be further restricted to *Whole_Body_Effort*, avoiding any displacement. Also depending upon individual's current state and desire the effort level of that agent could further be restricted.

However, if the robot is playing the role of a caregiver for a person in need or as a servant, the effort level decision will be different. Because, in such cases the robot should try to put maximal effort while trying to reduce the effort of the human. Similar is the case if the target-agent is tired.

Similarly, various global and individual factors could decide the maximum allowed effort levels of the agents: *social status* (collaborator, servant), *current context* (dining, reception, living room), *individual desire* (not willing to move), state (tired, reduced mobility, back problem so cannot lean or turn, neck problem so cannot turn head around, old person, agent already holding something), *role* (caregiver, boss, friend), etc.

8.4 Taskability Graph

Taskability Graph encodes what each agent in the environment might be able to do for every other agent, with which effort levels, and where. It basically encodes *agent-agent affordance* between each pair of agents.

We represent a taskability graph for a task as a directed graph TG_{task}:

$$TG_{task} = (V(TG), E(TG)) \tag{8.1}$$

$V(TG)$ is set of vertices representing agents in the environment:

$$V(TG) = \{v(TG) | v(TG) \in AG\} \tag{8.2}$$

where AG is the set of all the agents defined in chapter 3. $E(TG)$ is set of edges between an ordered pair of agents:

$$E(TG) = \{e(TG) | e(TG) = \langle v_i(TG), v_j(TG), e_{prop}\rangle \wedge v_i(TG) \neq v_j(TG)\} \tag{8.3}$$

e_{prop} is property of an edge:

$$e_{prop}^{TG} = \left(CSS, EC = \langle EC_{ab}^{ag} | \forall ag \in \{source(e), target(e)\} \wedge \forall ab \in RelAb_{ag}\rangle\right) \tag{8.4}$$

where CSS is candidate solution space, in which the solution of the task would lie. Depending upon the set of constraints and their types, which will be used to analyze the feasibility of the task, CSS could even be a single solution of the task. EC is a list of enabling condition. Each enabling condition EC_{ab}^{ag} corresponds to a particular basic ability type $ab \in TypeAb$, as defined in eq. 3.29, for a particular agent ag. This enabling condition as defined in eq. 3.28, can be a sequence of actions, an effort to apply, a state of the agent, etc. $RelAb_{ag} \subseteq TypeAb$ is set of ability types, which are relevant in finding the feasibility of the current task for a particular agent.

In the current implementation, for the taskability graph we restrict candidate space as the places to perform the task. We also restrict $RelAb \in \{see, reach\}$. Further, we assume that the enabling condition is expressed in terms of effort levels as presented in section 4.4.1 of chapter 4. In a more general implementation, depending upon the task, the CSS could be the Cartesian product of multiple parameters of the task, such as $position \times grasp \times orientation$.

Hence, with the assumptions in the current implementation, the problem of finding a taskability graph for a particular task $T \in BT$ (BT is set of basic HRI tasks, as shown in eq. 5.2), is a problem of finding *Agent-Agent affordances* (see section 5.2.4 of affordance analysis chapter 5) for every pair of agents in a given state of the

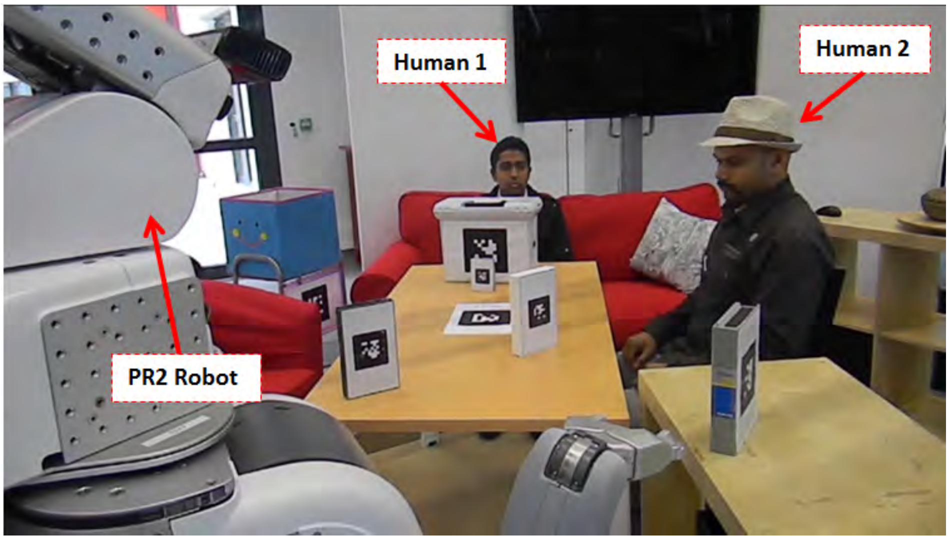


Figure 8.3: Scenario of a typical human-robot interaction around a table.

environment. Therefore, each edge of the taskability graph will encode the following information:

$$e^{TG}_{prop} = \left(point\_cloud, \left\langle effort^{performing\_agent}_{see}, effort^{performing\_agent}_{reach}, effort^{target\_agent}_{see}, effort^{target\_agent}_{reach}\right\rangle\right) \tag{8.5}$$

Figure 8.4 shows the taskability graphs for 4 different tasks: Make Accessible, Show, Give and Hide for scenario of figure 8.3 among all the agents in the environment.

Figure 8.5 shows an edge of a taskability graph and figure 8.6 explains what does an edge revel. It is a directed edge from performing agent to target agent. The spheres show the effort levels of both the agents and the point cloud *PtCloud* shows the candidate places where the task could potentially be performed with these effort levels. The instance of an edge of taskability graph as denoted in eq. 8.3 for this particular edge is:

$$e\,(TG) = \langle PR2\_Robot, Human2, (point\_cloud = PtCloud, \left\langle effort^{PR2}_{see} = NO\_Effort, effort^{PR2}_{reach} = Arm\_Effort, effort^{Human2}_{see} = NO\_Effort, effort^{Human2}_{reach} = Arm\_Effort\right\rangle)\rangle \tag{8.6}$$

In fact, there exists an edge only if the task is feasible with desired effort level. In the current implementation, the feasibility criterion is the existence of candidate places to perform the task corresponding to an edge. Assuming equal social status the effort balancing criteria was used for each task in this example. Further, the current context of sitting around a table has been also used to restrict the individual maximum desired effort as *Arm_ Torso_ Effort*. That is why between the human on the right and the robot there is no possibility of *give* and *make accessible* tasks as reflected from the missing edges between these two agents in the taskability graphs.

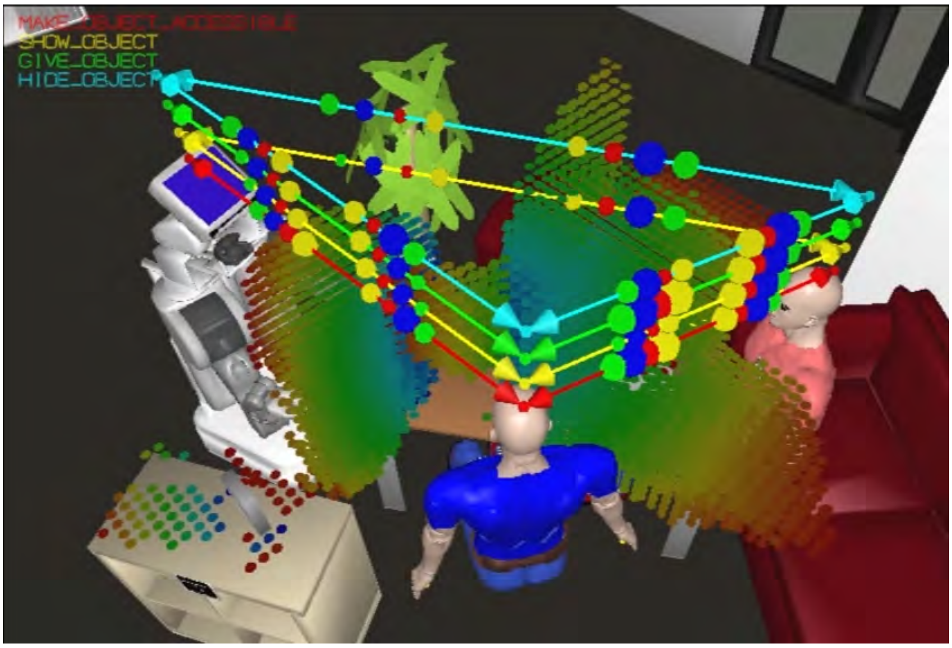


Figure 8.4: Taskability Graph for different tasks based on effort balancing, assuming equal social status and allowed maximum effort levels as *Arm_ Torso_ Effort* for each agent. See figures 8.5 and 8.6 for description about an edge of the graph.

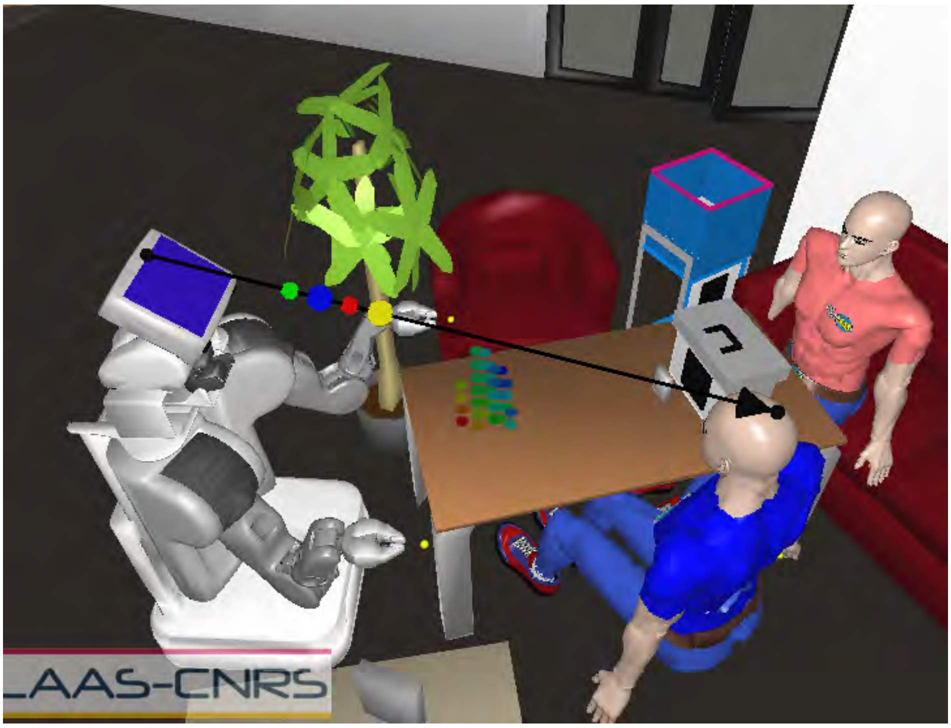
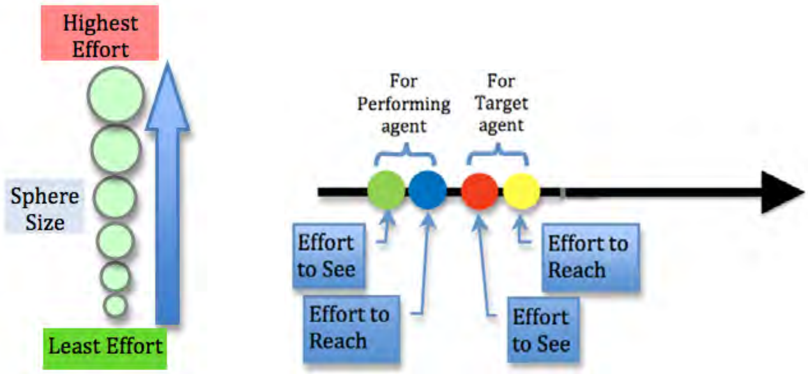


Figure 8.5: Example of an edge of a taskability graph



(a) Effort Sphere (b) Edge Description

Figure 8.6: Explanation of edge

For finding the edge with criteria of effort-balancing, the approach is as follows: Set the initial effort levels for both agents as least (*Arm_Effort* to reach and *Head_Effort* to see in current implementation). Then the planner uses the eq. 5.5 presented in Agent-Agent Affordance section 5.2.4 of chapter 5 to find the candidate place based on the requirement of the task. If the resultant candidate set is NULL then the planner increases the efforts of both the agents to next level incrementally until a NOT NULL candidate space is obtained or maximum effort level of each agent is reached.

Similarly, different criteria of mutual efforts are incorporated. For example, if it is to minimize the target agent's effort, in the iteration only the effort of performing agent is increased while maintaining the target agent's effort as lowest. Opposite is done if the performing agent effort has to be minimized.

8.5 Manipulability Graph

Manipulability Graph encodes what an agent might be able to do with an object, with which effort level, and where (if applicable).

Complementary to Taskability Graph, which encodes agent-agent affordances, Manipulability graph represents agent-object affordances. Currently there are four such affordances: *Touch*, *Pick*, *PutOnto* and *PutInto*. *Pick* is ability to $See \wedge Reach \wedge Grasp$ as explained in section 5.2.1, whereas *Touch* is ability to just $See \wedge Reach$, *PutOnto* is ability to $See \wedge Reach$ the places, which belongs to horizontal planes. *PutInto* is ability to put something into some container object, as explained in section 5.2.2 of affordance analysis in chapter 5.

Similar to taskability graph we represent a manipulability graph for a task as a directed graph MG_{task}:

$$MG_{task} = (V\,(MG)\,, E\,(MG)) \tag{8.7}$$

$V(MG)$ is set of vertices representing entities $ET = AG \cup OBJ$ (OBJ is set of objects in the environment):

$$V(MG) = \{v(MG) | v(MG) \in AG \vee v(MG) \in OBJ\} \tag{8.8}$$

$E(MG)$ is set of edges between an ordered pair of agent and object:

$$E(MG) = \big\{e(MG) | e(MG) = \langle v_i(MG), v_j(MG), e^{MG}_{prop}\rangle \wedge v_i(MG) \in AG \wedge v_j(MG) \in OBJ\} \tag{8.9}$$

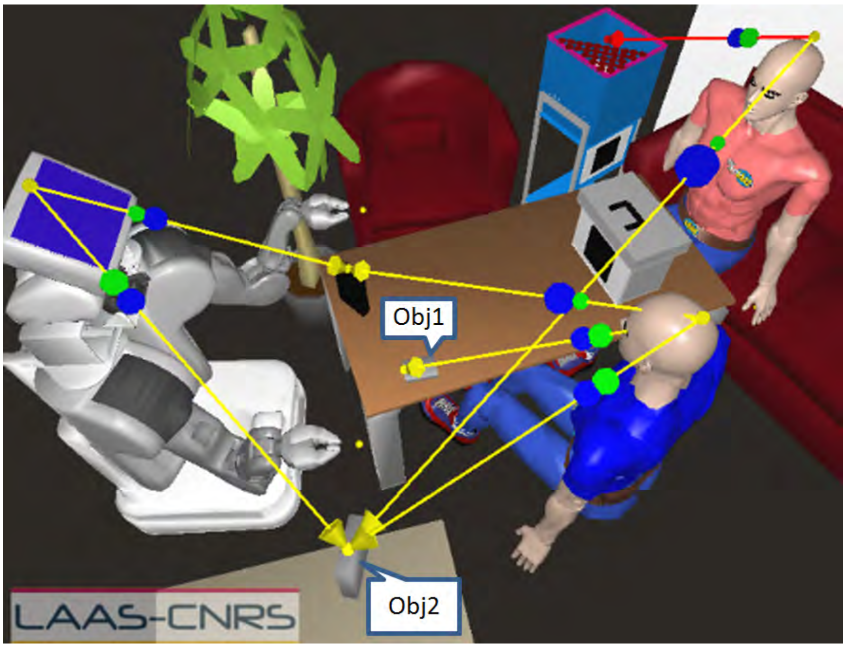
e^{MG}_{prop} is property of an edge of the manipulability graph:

$$e^{MG}_{prop} = \big(CSS, EC = \langle EC^{ag}_{ab} | ag = source(e) \wedge \forall ab \in RelAb_{ag}\rangle\big) \tag{8.10}$$

One of the main differences from the edge property of a taskability graph is, each edge in manipulability graph contains the list of enabling condition for one agent, which is the source vertex of the edge, because the target vertex will belong to an object. CSS, the candidate search space in which the feasible solution will lie will depend upon the type of the task. It could be the set of collision free grasp configurations for $Pick$ or $Take$, places on the plane for $PutOnto$ and so on.

Figure 8.7(a) shows the partial *Manipulability Graph*, demonstrating the *Pick* and *PutInto* affordances. Different maximum desired effort levels can be assigned for different affordances and agents, while constructing the graph. To show this, the maximum allowed effort for *Human 1* has been provided as *Displacement_Effort*, whereas for PR2 and *Human 2*, it has been given as $Torso\_Effort$. Hence, the resulted graph shows that *Human 1* can pick *Obj2*, as there exist collision free placements around the object, as shown in figure 4.16, from where he can reach, see and grasp the object. It also shows that *Human 1* can put something into the trashbin on his right, whereas *Human 2* cannot, because of his more restricted maximum allowed effort level. Each edge of the *Manipulability Graph* shows the agent's least feasible effort to see and reach the objects. Note the difference among the effort levels of all the agents to pick the objects, e.g. one on the right of the robot, successfully encoded in the *Manipulability Graph*.

For the sake of clarity, we do not superimpose the *Touch* and *PutOnto* affordances. Note that there is no edge from PR2 to *Obj1*, this is because in fact PR2 can reach and see, i.e. touch *Obj1*, there exists no collision free grasp to pick it, because of the object placement and PR2 gripper's size. Figure 8.8 shows effect of the environmental changes on the *Manipulability Graph*. We have displaced the *Obj1* of figure 8.7(a) behind the box, as shown in figure 8.8(a). The robot finds and updates the *Pick* affordance of *Human 1*. Earlier there was no edge in the *Manipulability Graph* because of the non-existence of collision free placement of *Human 1* around *Obj1*, even with *Displacement_Effort*. In the changed situation, the robot finds a feasible *Human 1 - Obj1 Pick* affordance with $Whole\_Body\_Effort$. Hence, there



(a) Manipulability Graph corresponding to *Pick* and *PutInto* affordances.

(b) Edge Description

Figure 8.7: Manipulability Graph and an edge description.

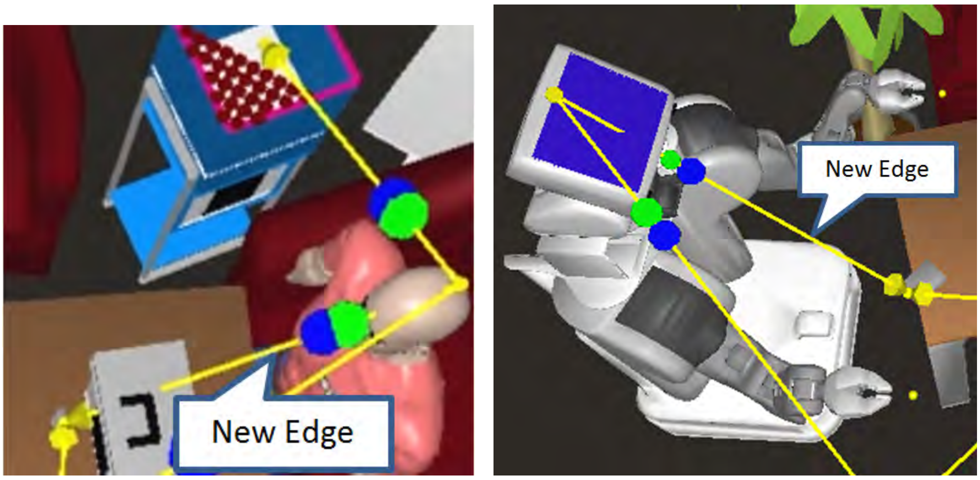
exists a new edge in the corresponding *Manipulability Graph*, as shown in figure 8.8(a).

In another experiment, we placed *Obj1* at the edge of the table, which facilitated collision free grasp by the PR2 gripper. In this case a new edge has been inserted for PR2 - *Obj1* *Pick* affordance, figure 8.8(b).

8.6 Affordance Graph

By combining a set of *Taskability Graphs* and a set of *Manipulability Graphs*, we have developed the concept of *Affordance Graph (AG)*. Hence, the *Affordance Graph* will tell the action-possibilities of manipulating the objects among the agents and across the places, along with the information about the required level of efforts and the potential spaces. Affordance graph (AG) is given as:

$$AG = \biguplus_{\forall tg \in TGr} TG_{tg} \uplus \biguplus_{\forall mg \in MGr} MG_{mg} \tag{8.11}$$



(a) Human's new affordance edge (b) PR2's new affordance edge

Figure 8.8: Environment change and its effect on the *Manipulability Graph*.

$\uplus$ is the operator, which depending upon the type of task, appropriately merges a takability or manipulability graph in affordance graph. As will be explained below it will create some virtual edges to ensure one of the desired properties to be maintained, that is between any pair of vertices of the affordance graph, there should be at most one edge. Further, it also assigns proper labels, weights and directions to the edges, which will become evident from the discussion below.

Figure 8.9 shows the *Affordance graph* of the current scenario. The $\uplus$ operator uses following set of rules for constructing *Affordance Graph*:

(i) Create unique vertices for each agent and each object in the environment.

(ii) For each edge *Et* of *Taskability Graph* from the performing agent *PA* to the target agent *TA*, introduce an intermediate virtual vertex *Vt* and split *Et* into two edges, *E1*, connecting *PA* and *Vt*; and *E2*, connecting *Vt* and *TA*.

(iii) The direction of *E1* and *E2* depends upon the task:

  (a) If the task is to *Give* or *Make-Accessible*, *E1* will be directed inward to *Vt* and *E2* will be directed outward from *Vt* towards *TA*.

  (b) If the task is to *Hide* or *Show* the object, *E2* will also be directed towards *Vt* from *TA*. This is to incorporate the intention behind such tasks, i.e. the object is not expected to be transported to the *TA*, and *E2* is for the purpose of grounding *Vt* to corresponding *TA*.

(iv) Assign meaningful symbolic labels to each of the new edges *E1* and *E2*. For example, if *Et* belongs to *Give* task, then label *E1* as *To_ Give* and label *E2* as *To_ Take*; if *Et* belongs to *Make-Accessible* task, then label *E1* as *To_ Place* and label *E2* as *To_ Pick* and so on.

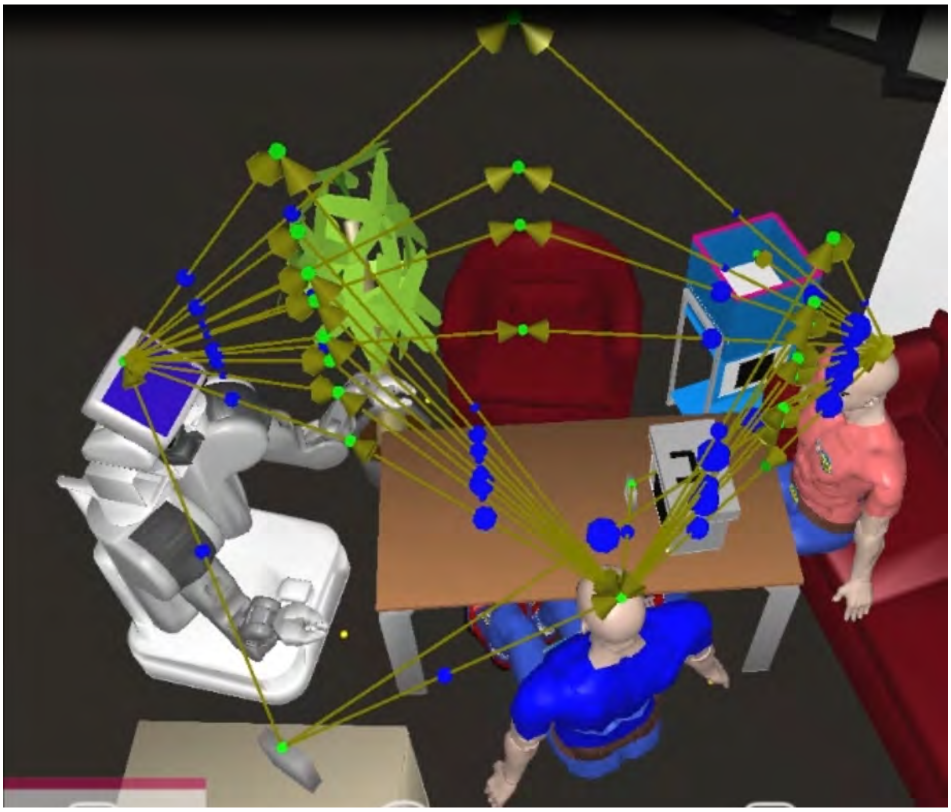


Figure 8.9: Affordance Graph constructed for the real scenario of figure 8.3. It is obtained by merging the corresponding Taskability and Manipulability Graphs. Because of the rules applied in constructing the affordance graph (such as, the edges are directed from the objects to the agents for pick task, and from the agents to the places for the place task), in one sense it encodes various possibilities about how the objects could *flow* across places and among agents. Hence, sometime we also refer this as *Object Flow Graph*. An Affordance Graph converts a subset of *grounding* and *shared planning* problems into a graph search problem.

(v) For each edge *Emt* of *Manipulability Graph* to *Pick* an object, an edge is introduced in the *Affordance Graph* directing from the object to the *PA*.

(vi) For each edge *Emp* of *Manipulability Graph* for *PutInto* and *PutOnto* affordance, an edge is introduced in the *Affordance Graph* from the *PA* to the container and the supporting objects.

Rule (iii) encodes potential flow of an object between two agents whereas rules (v) and (vi) encode the possible flow of objects corresponding to pick and put tasks, hence we sometimes refer *Affordance Graph* also as *Object Flow Graph*.

Each edge will have a weight depending upon the efforts encoded in the edges of the parent Taskability and Manipulability graphs. There could be various criteria to assign weights to the edges. Next, we will discuss about one possible choice of

Table 8.1: Computation Time in s

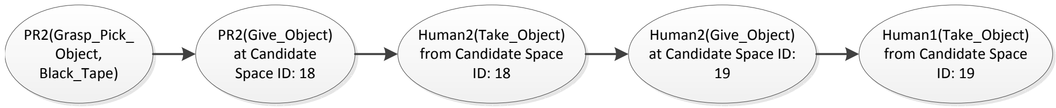
3D grid size: $60 \times 60 \times 60$ cells, each of dimension $5cm \times 5cm \times 5cm$	
3D grid creation and Initialization (one time process)	*1.6*
Computation of Mightability Analysis (provided 3D grid initialization is done)	*0.446*
Computation of Taskability Graph (provided Mightability Analysis is done)	*1.06*
Computation of Manipulability Graph (provided Mightability Analysis is done)	*0.14*
Computation of Affordance Graph (provided Taskability and Manipulabity graphs are calculated)	*0.002*
To obtain the shared plan to clean the table (see figure 8.11(a), provided affordance graph has been created)	*0.01*

such weight assignments.

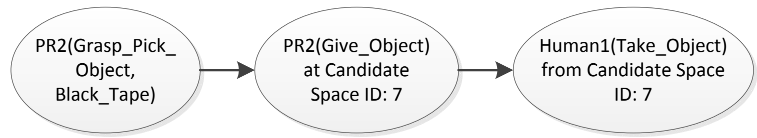
The weights shown in the affordance graph of figure 8.9 have been selected based on the maximum effort of the relevant abilities to see and/or reach. For example, if the edge corresponds to the *Give* task, the maximum effort between reach and see, encoded in the *Taskability Graph* will be assigned for both the agents: performing agent, *PA* and target agent, *TA*. But, if the task is to *Show*, then for the edge *PA-Vt* (*Vt* is the virtual vertex as discussed above) the highest between the reach and see effort will be assigned, but for the edge *Vt-TA*, the weight is assigned as effort to see. This is because, *TA* is not required to reach the object. In fact, the relevant abilities for a task are provided to the system a priori, which could also be learnt for basic HRI tasks as we will show in chapter 10.

8.7 Computation Time

Table 8.1 shows the computation time for different components to obtain the *Affordance Graph* of figure 8.3. Note that it is for the first time creation of the graphs, which is acceptable for a typical human-robot interaction scenario. As during the course of interaction, only a part of the environment changes, hence the selective updation of these graphs will be even faster. However, the exponential complexity of the system will be more evident, when more types of affordances and more number of agents will be present. Hence, the future work is to interface the system with our supervisor system [Clodic 2009], which will decide which part, and how much of these graphs will be updated depending upon the changes, situation and requirement.



(a)



(b)

Figure 8.10: Generated shared plan for the object (referred as *Black_ Tape*), which is on the right side of the robot, to give it to the human on the right (referred as *Human 1*) (see figure 8.9). The generated plan based on graph search in the Affordance Graph: (a) involves the middle human, (*Human 2*) to pass the object, as the criterion was to restrict the effort level of each agent as the *Arm_ Torso_ Effort*; (b) directly gives to the *Human 1*, in the case his maximum allowed effort level has been increased. In this case the plan does not involve *Human 2* as the path through *Human 2* will be more costly, as encoded in the Affordance Graph.

The novelty of affordance graph includes:

(i) It provides a graph based framework to query about affordances using any standard graph search algorithm.

(ii) By playing with edges, vertices, weights, the graph search can be guided. This facilitates to incorporate various desires, preferences, social constraints, effort criteria, in finding desirable/suitable affordance potentialities.

(iii) It provides capability to reason on human/agents effort levels.

(iv) It transforms a subset of human-robot shared task planning problem into a graph search problem, as will be demonstrated in section 8.8.2.

8.8 Potential Applications

Once the robot is equipped with the capabilities to analyze the potential flow of the object, i.e. having the Affordance Graph, it could be used for variety of purposes: to enhance human robot interaction, to ground interaction, to generate shared cooperative plan for tasks, to ground the changes to the agent and the actions, to facilitate action recognition and proactive behavior, to supporting high-level symbolic task planners, etc. Below we will discuss some of the aspects of these applications and the results.

8.8.1 Grounding Interaction, Agent, Action and Object

Based on Affordance Graph and Mightability Analysis the robot can disambiguate and ground the object. For example, again consider the scenario of figure 8.2, where two human and PR2 robot are sitting around a table. There are objects that are reachable and visible by different agents with different effort levels. If human on the right asks the robot: "where is the tape?" Then the robot can ground the object, which requires maximum effort to be seen by the human. Note that this grounding could further be enriched by taking into account the interaction history, which is beyond the scope of the thesis. Similarly, if the human asks the robot: "give me the tape", then Affordance graph facilitates to ground the object based on different possible reasoning. If robot assumes that if human wants a tape, which is easily reachable by the second human, it would have asked to the second human instead of the robot. Then based on analyzing which tape requires least effort to be taken by the robot than by the second human, it could plan to give that particular tape.

8.8.2 Generation of Shared Cooperative Plan

Using Affordance graph, a cooperative shared plan to perform low level task such as to give some object as well as high-level task such as cooperatively clean the table could be found.

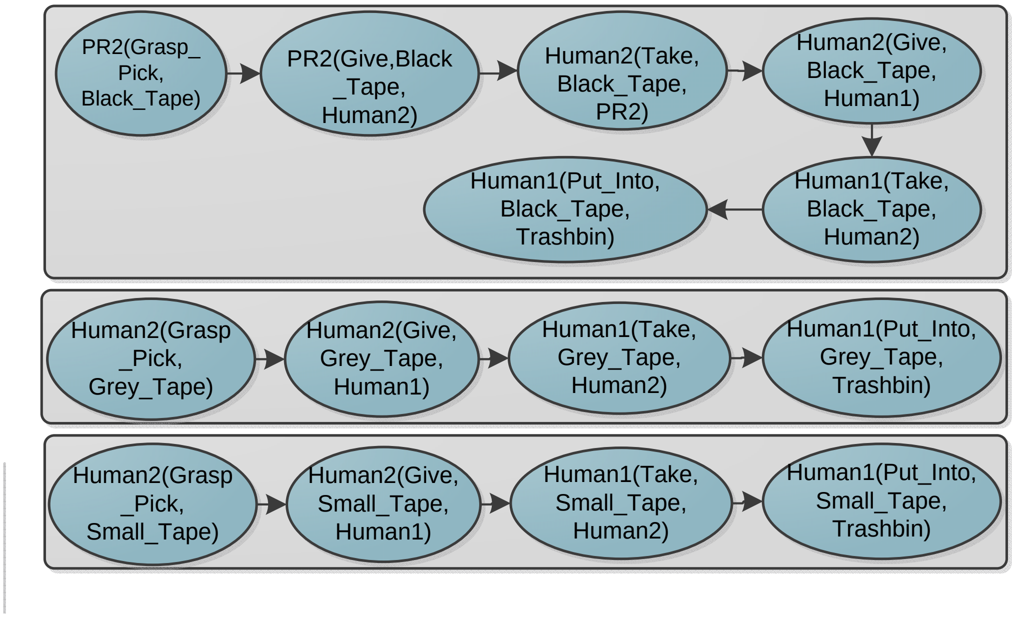
For example, consider the task is to give the tape (named as *Black_ Tape*), to the human on the right (referred as *Human 1*) (see figure 8.9). In the Affordance graph, the planner first finds the node *N1* corresponding to the object black tape and the node *N2* corresponds to the *Human 1*. Then uses a standard shortest path routine such as Dijkstra shortest path to find a path between nodes *N1* and *N2*. As the weight of edges reflects the efforts, the found path will be minimizing overall effort.

Figure 8.10(a) shows the found plan in the current scenario. Note that it has automatically planned a way, which requires the help of the human in the middle, as due to the situation based preference no agent was allowed to move and *Human 1* was willing to put maximum of *Arm_ Torso_ Effort*.

But in the case *Human 1* desired maximum effort level was *Whole_ Body_ Effort*, which allowed him to stand up and lean, the automatically generated refined plan is shown in figure 8.10(b). Note that the new plan does not involve *Human 2* as the path through *Human 2* will be more costly, as encoded in edges of Affordance Graph.

Similarly, if the task is to clean the table, the same framework could be used to find the shortest path between each object's node and the *PutInto* node corresponding to the *Trashbin*.

The framework autonomously finds the object-wise shared cooperative plan for cleaning the table as shown in figure 8.11(a).



(a)

(b)

Figure 8.11: Generated shared plan for clean the table. (a) With maximum effort level of all the agents as *Arm_ Torso_ Effort*. (b) Partially modified plan for small tape in case of *Human 1* (human on the right) effort level has been increased to *Whole_ Body_ Effort*.

But in the case *Human 1*'s willingness to put his effort was increased to *Whole_ Body_ Effort* then the sub-plan to trash the small tape was changed as shown in figure 8.11(b). This reflects the robot's capability to estimate that if *Human 1* will stand up and lean forward he can grasp the small tape behind the box and put it into the trashbin.

Figure 8.12 shows the execution of first cooperative action of the generated shared plan to clean the table by putting the objects in the trashbin. The generated plan is similar to the top block of 8.11(a). The maximum desired effort levels of each agent was set as *Arm_ Torso_ Effort*, hence the generated plan is autonomously involving each agent to achieve the task but each of them is putting least feasible effort.

The Affordance Graph encodes different ways (edges) to move the object from one node to another. To take the full advantage of this, we have defined "*agent busy*"

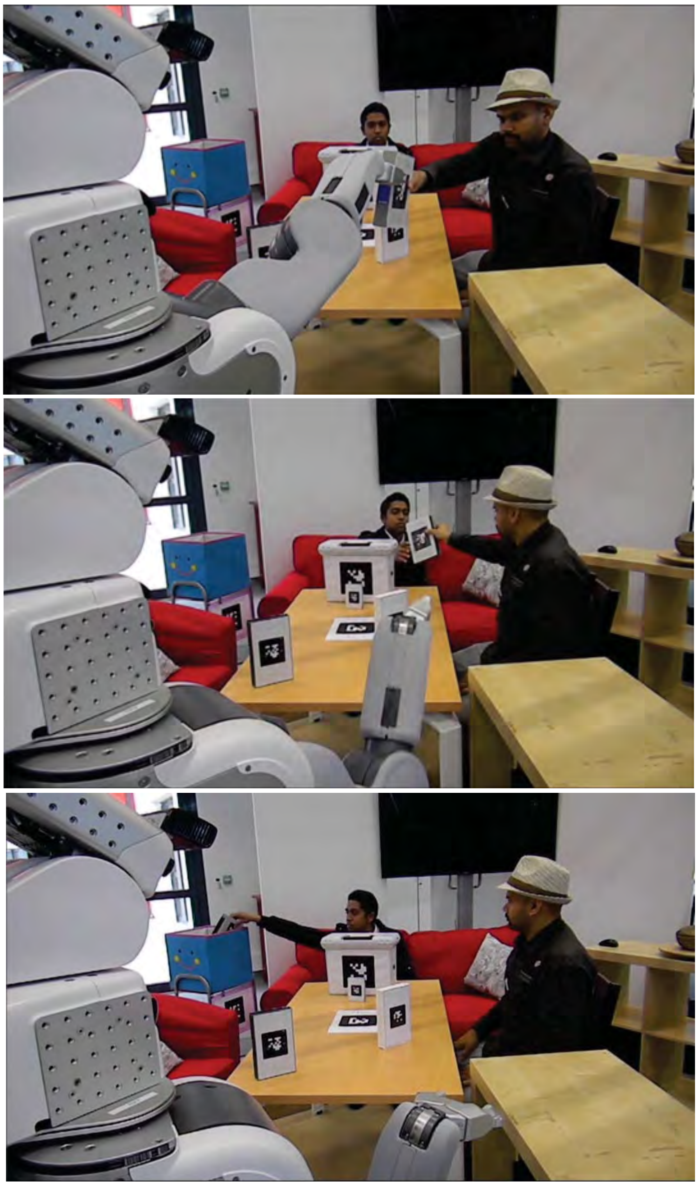
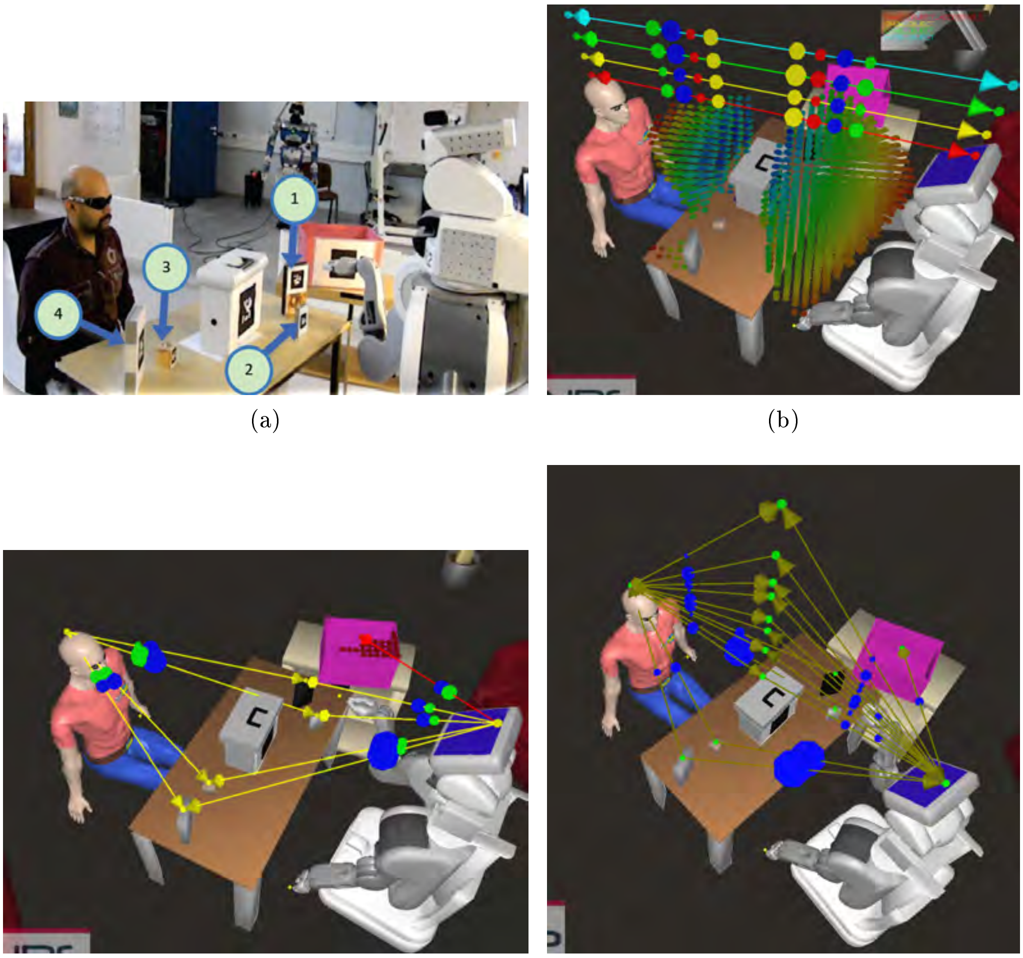


Figure 8.12: Execution of first cooperative action of the generated shared plan to clean the table by putting the objects in the trashbin. The maximum desired effort levels of each agent was set as *Arm_ Torso_ Effort*, hence the generated plan is involving each agent to achieve the task but each of them is putting least feasible effort.

flag, which if set as *True* for a particular agent, all the *To_ Take* edges corresponding to that agent is set as *inactive*. Hence, avoiding the possibility of *Give* task to the agent. Hence, any part of the plan that requires to pass the object through or to that agent, will autonomously take the path that will be corresponding to the *make-accessible* task (because the *To_ Pick* edges are still active).



(a) (b)

(c) (d)

Figure 8.13: (a) Initial scenario for which the planner has to find cooperative shared plan to clean the table. (b) Generated taskability graphs, encoding *agent-agent affordances* for different tasks. (c) Generated manipulability graphs, encoding *agent-object affordances*. (d) Generated affordance graph by merging the set of both the graphs.

Such flags could help in easily incorporating various criteria for generating the shared plan. For example, if there is a requirement to minimize the number of times the robot should try to engage the human in the task and once the human is engaged it should get maximum out of him/her so that he/she can be released. Consider the task of "Clean the Table" task in another scenario as shown in figure 8.13(a) by putting the small objects (marked as 1-4) on the table into the pink trashbin. The corresponding taskability graphs, manipulability graphs and the resulting affordance graph by merging them have been shown in figure 8.13. During the planning, this time, after finding the first cooperative action and assuming that it will make the robot busy during real execution of the complete plan, the planner sets "agent busy"

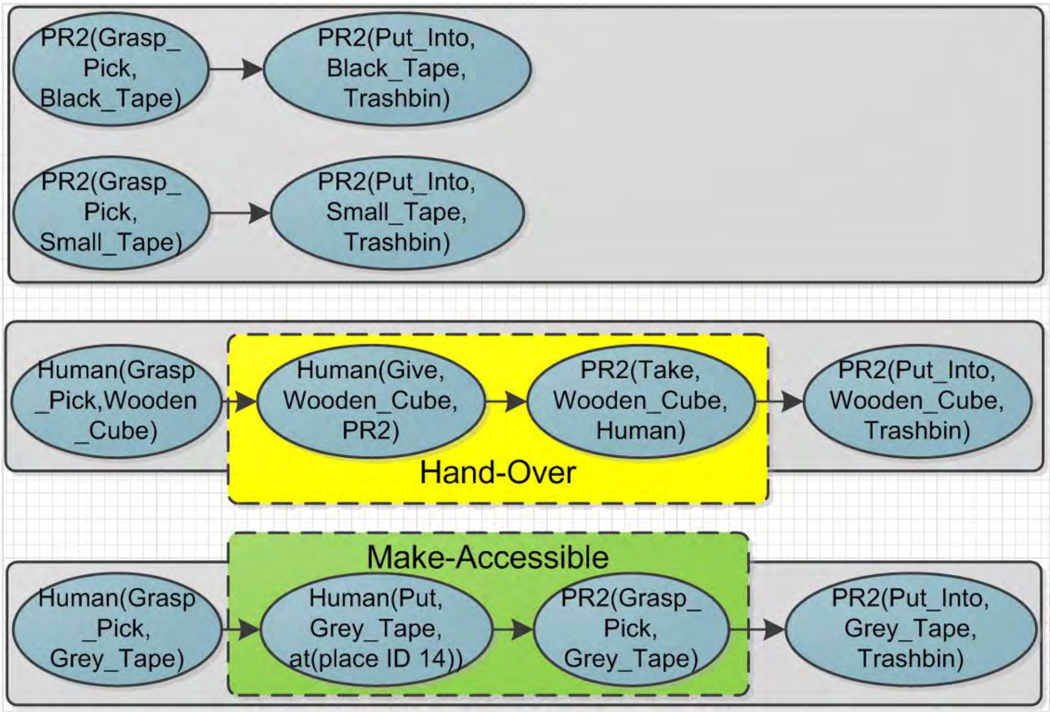


Figure 8.14: Generated shared plan to clean the table for the scenario of figure 8.13. During the planning, after obtaining the first "hand-over" plan, the *agent busy* flag was set as true for PR2 robot. Therefore, the first cooperative task corresponds to object hand-over whereas the next cooperative plan is make object accessible by the human. This facilitates the human to finish all his part in the shared task and avoid him to wait for the robot to finish its previous action so that he will give another object in its hand.

flag as true for the robot. This is with the intention to avoid human waiting or to avoid the robot to reengage the human again in a different cooperative action after it has executed the first one. This will automatically find the plan for rest of the task assuming the robot will be busy; hence any cooperative plan will be produced without simultaneously involving the robot. In the current situation, the generated shared plan to clean the table has been shown in figure 8.14. Again the maximum effort for each agent was set as *Arm_ Torso_ Effort*.

It is interesting to note that after the first cooperative hand-over action, next cooperative action is related to make-accessible (instead of another hand-over action), which the robot can communicate to the human by verbal request, while taking the object 3 from the human. The complete execution of the task has been shown in figure 8.15. Note that in figure 8.15(f) the human is directly giving the object to the robot, while in figure 8.15(h), he is making 4th object accessible, while the robot is putting the 3rd object taken from human into the trashbin. Thus releasing the human from the task as early as possible. And later on when robot finished putting

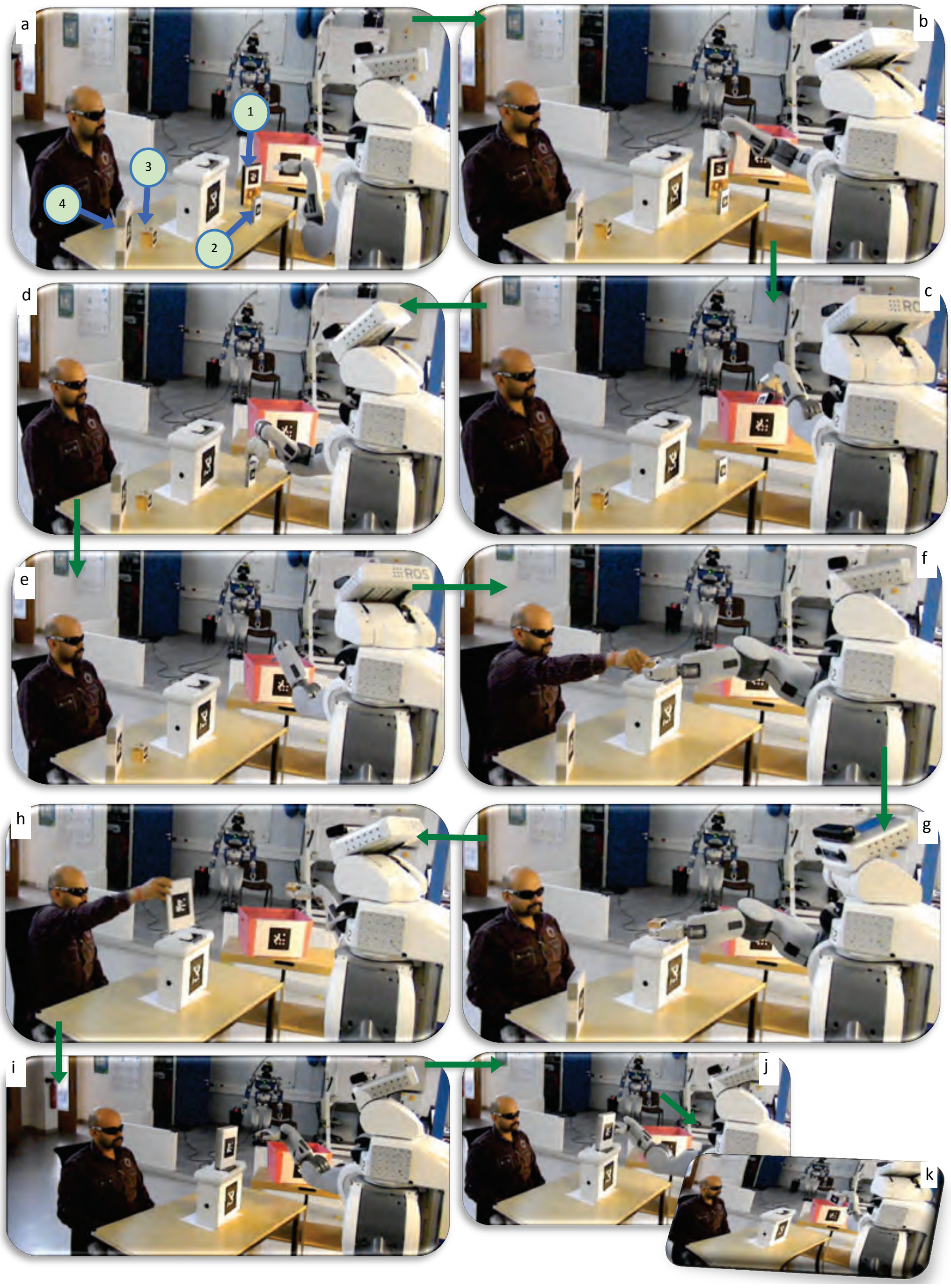


Figure 8.15: Execution of the Generated Shared Plan to clean the table (shown in figure 8.14) by putting the objects in the trashbin. Minimize the *number of times to engage human* as well as *least human effort desired* has been given as criteria to plan the cooperative actions. So the first cooperative task corresponds to *give object* (see (f)), whereas the next one corresponds to *make object accessible* (see (h)), while the robot will be continuing to trash the previous object given by the human.

the 3rd object it reaches to take the 4th object to put it into trashbin, figures 8.15(i)-8.15(k). In this task two types of proactive behaviors of the robot have also been integrated: proactively reaching out to take the object from the human and proactively suggesting the place where the human can make the object accessible to the robot. This is to reduce the confusion and effort of human. Detail explanation and instantiation of such proactive behaviors have been presented in the next chapter (chapter 9).

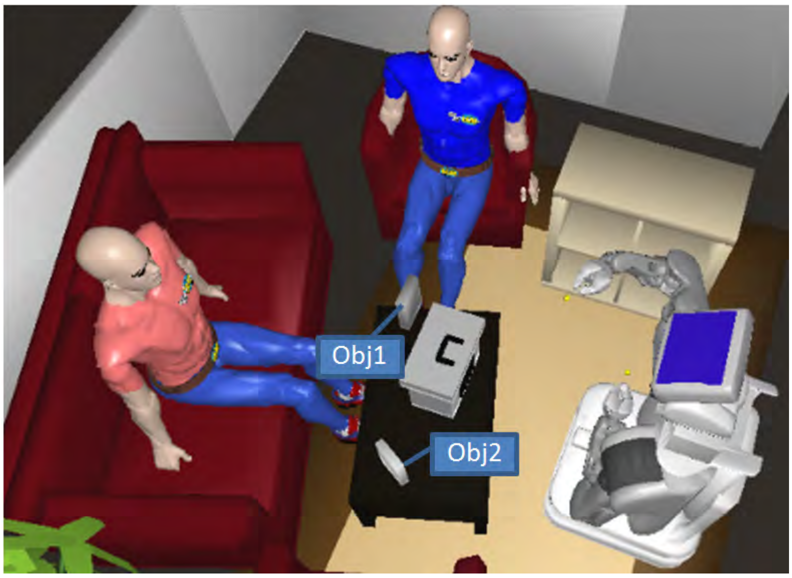
Similarly, different constraints could be introduced either by manipulating the edge weight or restricting/introducing the nodes. For example, least effort by each agent, least effort by a particular agent, overall least effort, least engagement time, maximum engagement time, temporarily suspending the involvement of a particular agent and so on. Above example shows introducing the constraint of reducing number of times to engage the human or to set him free as soon as possible. It can be opposite in the case the robot is required to keep the human engaged from time to time to avoid him/her being bored. For this instead of directly planning end-to-end from source to target node, the human node could be used as a *via node* to plan for each sub-task.

Note that this is just an example to easily incorporate such preferences in finding plan through Affordance Graph. In principle this will be a higher-level robot supervisor system such as SHARY [Clodic 2009], which will take such decisions on preferences based on the requirements and adjust such parameters of the presented Affordance Graph based shared plan generation.

8.8.3 A remark on planning complexity

Here it is important to discuss about actual complexity of such task planning. As already mentioned in this chapters, all such graphs are for a given environment state $s_0 \in S$, with the aim to provide rich information about various types of affordances in the given situation. Hence, while finding cooperative shared plan, transition from one vertex to another will make some changes in the environment state and result into a new state $s_1 \in S$. Hence, the graphs computed in s_0 might no longer be representing the actual affordances of s_1, at least in terms of the effort levels. Hence, the graphs might require to be recomputed partially or fully before searching for the next segment of the plan, which in fact will lead to exponential complexity. In our example scenarios, we assume that as the agents' relative positions are not changing significantly, and the objects' are not large enough to significantly affect the taskability graphs from one state to another. Hence, we relaxed the need of updating these graphs and relying on the plan entirely produced by searching in the initial graphs. However, it is another interesting research challenge to design and develop algorithm, which intelligently updates such graphs (completely or partially) during the process of finding a plan, to avoid combinatorial explosion as well as to drive the search towards the convergence.

(a)



(b)

Figure 8.16: (a) Initial state of the environment, s_0. (b) A changed state of the environment, s_1. *Obj3* is lost from the PR2 robots perspective, whereas the positions of *Obj2* and *Obj1* have been changed. During the course of changes the robot is blindfolded. Now the robot has to ground the changes by comparing s_0 and s_1.

Regardless of the planning approach or the assumptions, such graphs provide the high-level task planners with the flexibility to choose from different feasible sub-actions and the associated cost, at any stage of planning.

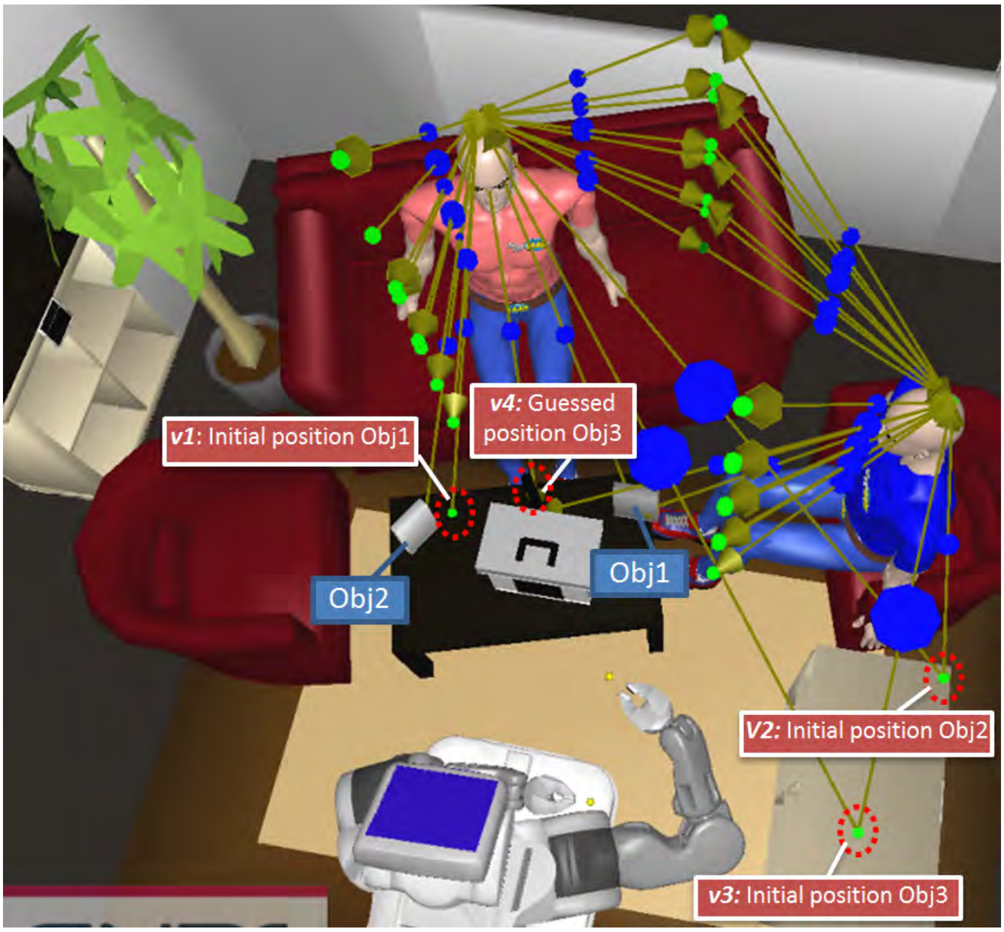


Figure 8.17: Modified affordance graph to ground the changes. Constructed by first introducing virtual object nodes, corresponding to the old positions of the displaced or lost objects in state s_0 in the new state s_1 of the environment and then all the graphs are calculated. For the lost objects, the places where the objects could be hidden are found (by using the agent-agent affordances for hide task) and then one feasible placement orientation of the object is found, which should be making the object invisible to the robot.

8.8.4 Grounding Changes, Analyzing Effects and Guessing Potential Action and Effort

Based on Affordance graph and Mightability Analysis a set of hypothesis could be generated about potential agents and actions, which might be responsible for some changes in the environment, cause of which was oblivious to the robot. For example, consider that at a particular instance of time the state of the environment is $s0$ as shown in figure 8.16(a). Now, assume the robot moves away for some purpose and meanwhile the humans have made some changes in the environment. Now the robot is back and observes the new state of the environment as s_1. In s_1, the robots find that the position of the two objects, *Obj1* and *Obj2* have been changed and the position of *Obj3* is *UNKNOWN*, as the robot could not see them from its current position. Now the problem of grounding the changes is:

given $\langle s_0, s_1 \rangle$***, find*** $\langle C, E, A \rangle$

where C is the changes, E is the effect of those changes on abilities and affordances and A is the potential sequence of actions behind such changes.

To analyze the effect of the change on the agents abilities, the planner compares the Manipulability Graphs computed corresponding to the states s_0 and s_1, and generates a set of comparative and qualifying facts. (In chapter 10 we will explain how such facts have been constructed by comparing two instances of the environment.)

To ground this change to the agent and action, the planner adds state s_0 position of those objects, which are now displaced in state s_1 in the list of dummy object vertices, DV. For the lost objects, using the agent-agent affordance analysis for the hide task, the planner finds the places where the object could be made hidden from the robot. Then it finds a placement of the object with an orientation, such that the lost object would be completely hidden from the robot. Currently to the planner it was explicitly provided that the object could be lying on the planes belonging to the wooden furniture (table, shelf, in our current scenario). Hence, for the current example, it guessed one possible placement of the lost object $Obj3$ behind the white box, as indicated in figure 8.17. Then this guessed position is also inserted in the list of dummy object vertices DV.

Then Manipulability graph and Taskability graphs are constructed and Affordance graph is found, using the set of vertices $V(TG) \cup V(MG) \cup DV$. As the robot was not the active agent during the changes, it removes all the outgoing edges from the vertex belonging to the robot. Now, to guess the potential action behind the change in position of each displaced object, first a vertex pair $\langle vs, vg \rangle$ is extracted. $vs \in DV$, which belongs to the position of the object at s_0. $vg \in (DV \bigcup \cup \{v(MG) | v(MG) \in OBJ\})$. Now simply finding a shortest path in the affordance graph for the vertex pair $\langle vs, vg \rangle$ the robot can reason about the agent, his/her/its action as well as the effort behind the change in that particular object.

For our current example, the set of dummy vertices found by the planner is $DV = \{v1, v2, v3, v4\}$ as shown in figure 8.17, encircled in red. The set of vertex pairs to find the path for finding the possible explanation behind the change of each object are $\{\langle v1, Obj1 \rangle, \langle v2, Obj2 \rangle, \langle v3, v4 \rangle\}$.

Below is the overall output of the planner by comparing the two environment states as shown in figure 8.16. Following is the mapping of names of agents and objects of figure 8.16(a) and the unique names of the entities used by the system for internal representation of the environment:

$Obj1 \rightarrow GREY\_TAPE$

$Obj2 \rightarrow WALLE\_TAPE$

$Obj3 \rightarrow LOTR\_TAPE$

$Human1 \rightarrow HERAKLES\_HUMAN1$

$Human2 \rightarrow HERAKLES\_HUMAN2$

Explanation and interpretation is provided within ⌊⌊ ... ⌋⌋.

=** Result of comparing world states with ids 0 and 1 =**

=========== PHYSICAL CHANGES =======

- Object LOTR_TAPE *MOVED.*
- Object WALLE_TAPE *MOVED.*
- Object GREY_TAPE *MOVED.*

====== EFFECT on ABILITIES and AFFORDANCES ========

- For the pair [LOTR_TAPE , PR2_ROBOT] to SEE the ability : *LOST* ⌊⌊ *as robot cannot see it from its perspective* ⌋⌋
- For the pair [LOTR_TAPE , PR2_ROBOT] to REACH the effort : *LOST* ⌊⌊ *as robot don't know the object's position* ⌋⌋
- For the pair [WALLE_TAPE , HERAKLES_HUMAN1] to REACH the effort : *DECREASED* ⌊⌊ *as in state* s_0 *human1 was required to put Displacement_Effort to reach the object* ⌋⌋
- For the pair [WALLE_TAPE , HERAKLES_HUMAN2] to SEE the effort : DECREASED ⌊⌊ *as in state* s_0 *human2 was required to turn head to see the object* ⌋⌋
- For the pair [WALLE_TAPE , HERAKLES_HUMAN2] to REACH the ability : LOST ⌊⌊ *as in state* s_1 *the planner did not find any collision free placement of human2 for him to reach the object, which was reachable in* s_0 ⌋⌋
- For the pair [WALLE_TAPE , PR2_ROBOT] to SEE the effort : DECREASED ⌊⌊ *as in state* s_0 *the robot was required to turn head to see the object, which is now in the current field of view of the robot.* ⌋⌋
- For the pair [WALLE_TAPE , PR2_ROBOT] to REACH the ability : GAINED ⌊⌊ *as in state* s_0 *the planner did not find any collision free placement of robot to reach the object, which is now reachable in* s_1 ⌋⌋
- For the pair [GREY_TAPE , HERAKLES_HUMAN2] to REACH the ability : GAINED

- For the pair [GREY_TAPE , PR2_ROBOT] to REACH the ability : LOST

============ POSSIBLE EXPLANATIONS ===============

- For LOTR_TAPE:

  LOTR_TAPE GRASP_PICK by HERAKLES_HUMAN2 GIVE_OBJECT at_a_place TAKE_OBJECT HERAKLES_HUMAN1 PUT_ONTO at_a_place ⌊⌊ *human2 has picked the object and gave it to human1 and then human1 has placed it at its new guessed position* ⌋⌋

  =====

- For WALLE_TAPE:

  WALLE_TAPE GRASP_PICK by HERAKLES_HUMAN2 GIVE_OBJECT at_a_place TAKE_OBJECT HERAKLES_HUMAN1 PUT_ONTO at_a_place ⌊⌊ *human2 has picked the object and gave it to human1 and then human1 has placed it at its new position* ⌋⌋

  =====

- For GREY_TAPE:

  GREY_TAPE GRASP_PICK by HERAKLES_HUMAN1 PUT_ONTO at_a_place ⌊⌊ *human1 picked it and placed it at its new position* ⌋⌋

The above result shows the capability of the system to infer the potential cause of changes with one possible explanation. This is based on different assumptions about the agents and their willingness to put efforts and cooperate, hence not necessarily be guessing the actual course of actions. Depending upon various factors as discussed earlier, such as the criteria of mutual effort used for computing taskability and manipulability graph, the criteria for assigning weight to the edges of affordance graph, and the criteria used to find the path in the affordance graph, the resultant path of the graph could be different and could imply different assumption for guessing the actions. In the current example, as the criteria was effort balancing and the resultant path was minimizing overall effort, hence the explanation in some sense assumes that whenever possible and feasible, agents will try to cooperate to achieve the changes. But that is how we also guess, based on some assumptions about the agents and their behaviors. In appendix B, we will demonstrate this aspect where we will place a human competitor to play the game with the robot to ground the changes in terms of abilities and actions.

In fact the output generated in the format presented above is further parsed and verbalized by the supervisor and speech synthesis system to play the game in an interactive manner, as we will show in appendix B, where we will compare with human's responses.

8.8.5 Supporting High-Level Symbolic Task Planners

The *Affordance Graph* could be utilized in its full potential by a high-level task planner such as ours [Alili 2009], for which the tasks encoded in the graph serve as the atomic level tasks. This can be done for various purposes, for finding the feasibility of a task, to get suggestion for geometrically grounded feasible alternative plans, and so on. Linked to this aspect, in the next section we will discuss another contribution of this chapter, establishing a link between geometric and symbolic task planner. We have elevated the typical notion of the geometric counterpart of a task planer from *geometric path planner* to *geometric task planner*. Further, we have introduced the concept of *geometric backtracking*. The framework, instead of directly returning a *fail* to the symbolic planner about the feasibility of a requested task, the geometric planner first tries to find a solution by backtracking at its level in the tasks planned so far to achieve the goal. Next section will illustrate the framework.

8.9 Two Way Hand Shaking of Geometric-Symbolic Planners

In this section, we extend our approach of solving intricate motion, manipulation and task planning problems based on the link between geometric and symbolic planners [Cambon 2004], [Cambon 2009] by applying it to the challenging context of human-robot cooperative manipulation. We propose a scheme that is based on the coalition of a symbolic task planner and a geometric task planner and provides a more elaborate interaction between the two planning environments.

The overall planning system starts from a goal or a situation to achieve, and builds a so-called cooperative plan, which is based on planned actions for the robot and estimation of feasibility of actions for the human. We describe below the two components, the geometric and the symbolic planners, how they are invoked in a hybrid planning scheme and finally we illustrate their interleaved cooperation through an example.

8.9.1 The Geometric Task Planner

In this section, we will describe different layers of geometric task planner from the perspective of facilitating the discussion of link with symbolic planner as well as the concept of geometric backtracking.

8.9.1.1 Layers of Geometric Planner

As illustrated in figure 8.18, the geometric planner consists of 3 main layers:

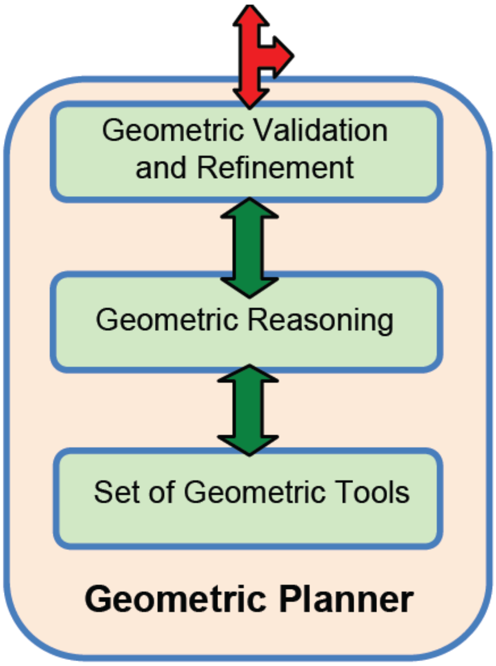


Figure 8.18: Layers of Geometric Planner. Top layer provides the interface for symbolic planner.

Geometric Tools Layer: This layer consists of a set of geometrical tools, which provides the robot with the Mightability and Situations Assessment capabilities.

Geometric Reasoning Layer: At the top of the basic layer, we have the geometrical reasoning layer, which finds out the candidate spaces for performing various basic actions. The affordance analysis presented in chapter 5 and the basic human-robot interactive task planner presented in chapter 7 are the key components of this layer.

Validation and Refinement Layer: This top layer of geometric planner communicates with external modules (symbolic planner in our case) and handles external requests, about feasibility of a particular basic task. It maintains and updates the plan, as well as contains logics for backtracking at geometric level. As shown in figure 8.19, in our current implementation, the symbolic planner sends a request to the geometric planner with following information:

- Name of the basic actions take, Make accessible, give, show, etc.
- The parameters of the action: Object Name, Agent Name, etc.
- The set of additional constraints, for example, a particular object should always be visible, or should be put on a particular table, etc.

Types of constraints We have three types of constraints at geometric level:

- Internal constraints known to the geometric reasoning system for performing

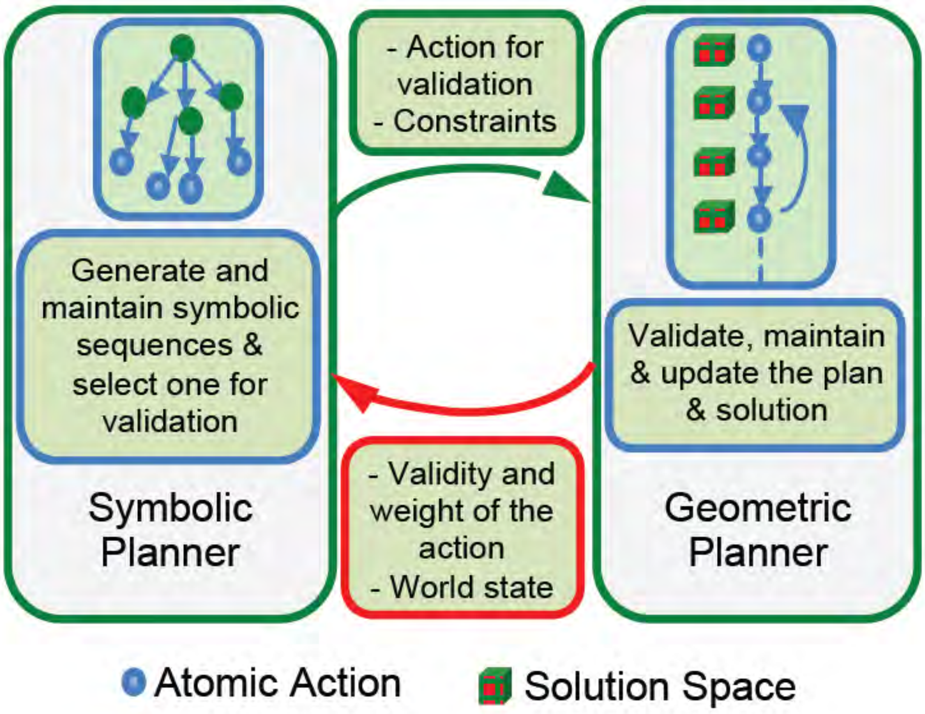


Figure 8.19: Communication between geometrical and symbolic planner and their roles in the overall system.

a particular basic task.

- External/additional constraints provided by the symbolic planner for the same basic task.
- Discovered constraints by geometric planner due to failure at later stages of validating another task in the sequence.

In fact, the beauty of the system is, the geometric planner will already have a basic set of constraints for reasoning to perform a particular action without need of any external constraint as illustrated in the framework presented in chapter 7. Hence, additional constraints, if any, from symbolic planner will refine the solution space for better converging towards the final task to achieve.

Another novelty of our system is that, in case of non-feasibility of a solution for a particular action in the sequence, instead of directly sending the fail message to the symbolic planner, it will first try to backtrack at geometric level. This backtracking is to find the possible cause of failure and refine the solution space of a particular action. In that sense robot will have a third set of geometric constraints, which it has discovered due to failure during validating the sequence of actions. This third set of discovered geometric constraints together with the set of constraints already known to the robot at geometric planner level and the set of additional constraints provided by the symbolic planner, will serve as the new set of constraints for avoiding failure

Algorithm .1: Symbolic planer procedure

```
   Input: Ws, Pro
1  Prj ← ∅;
2  repeat
3      Refinement(Tree, Ws);
4      if Refinement reach an action a then
5          if Precondition(a, Ws)= True then
6              if Geometric_Refinement then
7                  Update Prj;
8                  ApplyEffects(a, Ws);
9              else
10                 Symbolic backtrack ;
11         else
12             Symbolic back track;
13 until the Tree is explored;
```

Figure 8.20: Backtracking at Symbolic Planner level.

at future steps while reiterating for validation of the plan after backtracking.

8.9.2 The Symbolic Planner

For planning at symbolic level, we have interfaced with two such planners: Affordance Graph based planner presented earlier in this chapter and HATP (Human-Aware Task Planner) a hierarchical task planner [Alili 2009] designed to synthesize desire-effect based plans to achieve goals. The robot will have to produce a plan by taking into account its capacities and the current state of the environment.

The geometric planner enhances the symbolic reasoning by allowing it to use facts that depend exclusively on the geometry like "visibility" and "accessibility", also it allows to reject plans that are feasible at symbolic level, but do not have a valid geometric refinement.

8.9.3 The Hybrid Planning Scheme

The symbolic planner has the world state s and the planning problem *Tree* (in case of HATP) and graph (in case of Affordance Graph) as input. The algorithm as shown in figure 8.20 starts by initializing actions' temporal projection *Prj* to an empty set (line 1). The main loop of the algorithm (line 2 to 13) runs until all the tree/graph is explored. The refinement function at line 3 is responsible for the tree/graph refinement. It decomposes all high-level tasks present in the tree/graph until the reach of a sequence of action. When an action appears in the tree/graph, the algorithm checks its precondition (line 5). If the precondition does not hold, the algorithm makes a backtrack to the refinement step (line 3) to continue the exploration of other branches. If the precondition holds in the current world

Algorithm .2: Geometric refinement procedure

```
   Input: Ws, a
1  if Geometric_validation(a, Ws) then
2      Update Prj cost;
3      Update Spatial_facts(Ws);
4      return true;
5  else
6      Geometric backtrack;
7      if New Solution then
8          Update Prj cost;
9          Update Spatial_facts(Ws);
10         return true;
11     else
12         return false;
```

Figure 8.21: Backtracking at Geometric Planner level.

state, the algorithm calls the geometric refinement procedure to query the geometric validation of the action (line 6). If geometric refinement procedure validates the action, the algorithm updates the plan and applies the effects of the action on the current world state. On the other hand, if the geometric planner fails, the algorithm needs to backtrack to explore other tree branches.

Figure 8.21 describes the geometric refinement procedure. With a given world state and an action, this procedure is in charge of planning a motion to achieve the action while maintaining a continuous geometric motion plan. The algorithm first tests if there is a solution to achieve the given action with the given symbolic world state and its "internal geometric state". For validating actions, it mainly uses the constraint based basic task planner presented in the chapter 7. In case of a successful validation, the cost and facts are sent to the symbolic planner and the geometric state is saved. In case of a failure in this step, the geometric planner looks for alternatives in its previously planned nodes, which might have an effect to the achievement of the current action, by backtracking (line 6). If the current action can be achieved by modifying properties of the previously planned motions, the algorithm returns true and updates costs and facts. Otherwise it fails. But instead of just returning 'fail' it also checks for the feasibility of the current request with increasing the desired effort levels and/or relaxing some of the constraints. If it finds a solution with such modifications, it returns the 'fail' along with the suggestions to have the feasible solution. For example, if the symbolic planner requests to make an object accessible to the human by putting it on a specific table and the geometric planner fails to validate it even with backtracking, it will relax the criteria of putting on table and find all the support in the environment (such as on the top of another object, as the robot is capable of finding placement on the top of any surface see section 4.2.2 of chapter 4) and could return with suggestion that the task could be achieved if the object will be put at the box top. Other types of suggestions includes: if no path found to grasp and hold an object to show, it can suggest to try removing a

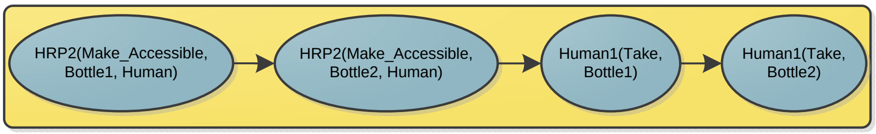


Figure 8.22: Generated symbolic plan for helping the human in making the drink.

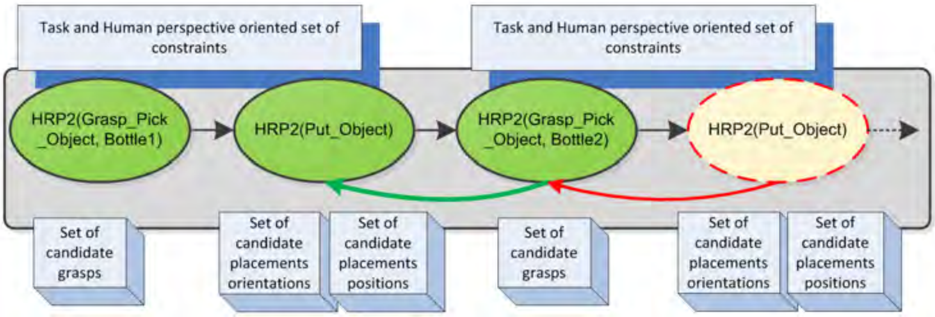


Figure 8.23: Internal representation at geometric level while validating the symbolic plan of figure 8.22. In case of failure to validate 4th node, instead of returning with fail message, the planner backtracks to explore with the alternative to perform the previous actions, so that the current action might become feasible.

particular object, as the robot keeps tracks of obstructing and occluding objects from an agent's perspective, see section 4.3 of chapter 4. This suggestion of alternatives to the symbolic planner, based on rich understanding of agent's abilities, affordances and situation, makes the two-way handshaking more prominent.

8.9.3.1 System Demonstration

Let us consider a scenario where the robot has to help the human in making a drink. As shown in figure 8.24(a), the human will need to have the two bottles of drink (Red and Blue bottles) to prepare the drink. Assuming the *tired* human does not want to stand up and lean to take the bottles, which in fact are also hidden from his current perspective, the desired maximum effort of the human is set as *Arm_ Torso_ Effort*. On the other hand, the HRP2 robot cannot stand up by itself from the chair, so the maximum allowed effort of HRP2 is also restricted to *Arm_ Torso_ Effort*. The plan generated at symbolic level has been shown in figure 8.22. Then the task planner starts validating the nodes one by one with its geometric counterpart. Note that we have chosen to first plan the entire task at the symbolic level, then validate. However, one can also decide to validate the partial symbolic plan before converging to a big full plan.

Backtracking at Geometric Level The geometric planner successfully validates the first node of the symbolic plan of figure 8.22 and stored an internal representation

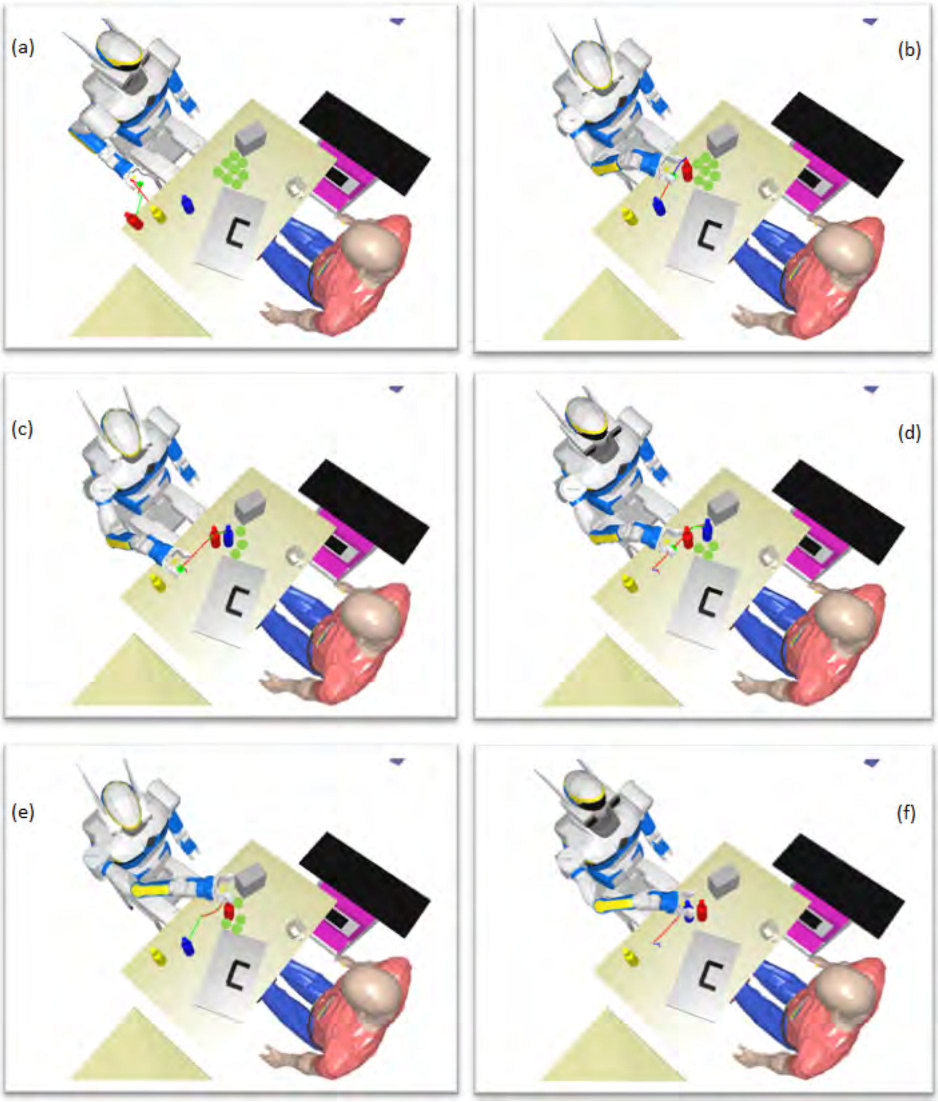


Figure 8.24: Demonstration of backtracking at geometric level to find a feasible solution for making two bottles (red and blue) accessible to the human. Maximum allowed effort level for both the agents is set as *Arm_ Torso_ Effort*. (a) The initial scenario, and the green circles show the autonomously found current candidate placement space. (b) Solution found for validating the first node of symbolic plan (see figure 8.22). (c)-(d) Failed to find a feasible path to make accessible second bottle, corresponding to second node of symbolic plan. Instead of returning with a fail message, the geometric layer backtracked to find alternate solution for previous actions, which might make the current action feasible. (e) By backtracking in candidate space of putting the first bottle, it found another placement for first bottle, by respecting the constraints. (f) With this new solution of making first bottle accessible, now the planner found a feasible placement for making second bottle also accessible to the human. Hence, returned a success for the entire plan to the symbolic layer.

of the task along with the candidate search spaces (detail about such candidate search spaces could be found in chapter 7 of planning basic human-robot interactive manipulation tasks using a hierarchy of constraints). The first two nodes of figure 8.23 shows the internal representation of the successfully validated make-accessible task for the first bottle, see figure 8.24(b). While validating the second node of the symbolic plan of figure 8.22, it did not find the solution in fourth node of geometric plan as shown in figure 8.23. Because the placement of first bottle is not allowing to place the second bottle with a collision free path, while maintaining the task oriented constraints of reachability and visibility, with desired effort level and it fails as shown in figures 8.24(c) and (d). At this stage, instead of communicating the non-feasibility of the solution to the symbolic level, the planner will first try other possible solutions at geometric level itself by backtracking. Hence, it backtracks to previous node, and explored with alternative grasps. But no other pickup configuration, facilitated the action. It fails again and backtracks further to the second node. There it finds another solution for the first task, i.e. alternative placement of the first bottle, which still respects the constraints, see figure 8.24(e). Then it again starts validating from that second node onward. The process continues until search space for all the nodes is exhausted or a maximum number of backtracking has been performed or threshold on reply time is reached. For the current situation, the planner is able to find the solution for entire task by backtracking and placing the first bottle at a different place than the initial one, which made placement of second bottle also feasible. Figure 8.24(f) shows the final solution, note that the placement of first bottle has been changed from the initial planned placement of figure 8.24(b), hence the second bottle has been also placed with a collision free path.

Backtracking at Symbolic level It may be the case that even after backtracking and exploring all the possible solution at geometric level, the feasibility of a particular node of symbolic plan could not be validated. In that case, the geometric planner would inform the symbolic planner about non-availability of solution along with the suggestion to achieve the task, as explained earlier. Then the symbolic planner could decide to explore another (sub)branch of its plan or to explore the suggestion if it does not contrast with other global constraints.

In our example, if the geometric planner would have failed to validate the second request of make accessible, because of the absence of a stable position to put the bottle down, the symbolic planner may switch to an alternative solution where the robot waits and "gives" the object directly to the human or could follow the suggestions of geometric planner and try allowing the object to be placed at the top of another object or increase the effort level of the agents.

8.10 Until Now and The Next

In this chapter, we have presented the concepts of *Manipulability Graph*, which encodes agent-object affordances, *Taskability Graph*, which encodes agent-agent affordances and merged them to construct *Affordance Graph*, which tells about variety of action possibilities to make some physical changes in the environment. All these graphs have a component of effort associated with each edge, hence facilitates to incorporate various desire and constraints on the plan. The novelty of this framework is that it converts a variety of HRI problems, such as Grounding interaction, grounding environmental changes to agents, actions and objects, generating shared cooperative plans, etc., into a graph search problem. We have shown the results of these aspects, where the robot is guessing what all might have happened in its absence, and also generating shared cooperative plans to clean a table.

Further, we have presented the concept of backtracking at geometric task level to solve a high-level goal in cooperation with the symbolic counterpart. We have argued and illustrated that elevating the geometric planner from a simple trajectory planner to a task planner, converges to a plan without unnecessarily flooding the symbolic planner with fail reports, thus avoid to force unnecessary backtracking at symbolic level. The rich geometric task planner also avoids the symbolic planner to bother about the details related to geometric constraints and parameters of the task.

Until now, we have equipped the robot to reason about the cooperation. In the next chapter, we will equip the robot to reason about the proactivity. The robot will instantiate the plans, which involve an element of proactivity to cooperate as an attempt to support the human partner and to reduce his/her effort and confusion.

Chapter 9

Prosocial Proactive Behavior

Contents

9.1 Introduction 211

9.2 Generalized Theory of Proactivity for HRI 213

9.2.1 Proactive Action 213

9.2.2 Proactive Action Planning Problem 213

9.2.3 Spaces for Proactivity 213

9.2.4 Proposed Levels of Proactive Behaviors 215

9.3 Instantiation 220

9.3.1 Objective of the hypothesized proactive behavior 221

9.3.2 Hypothesized Proactive Behavior for Evaluation 224

9.3.3 Hypotheses about the effects of the human-adapted proactive behaviors in the joint task 224

9.3.4 Framework to Instantiate 'where' based Proactive Action . . 225

9.4 Illustration of the framework for different tasks 227

9.4.1 For "Give" task by the human: Proactively reaching out . . . 227

9.4.2 For "Make Accessible" task by human: Suggesting 'where' to place 230

9.4.3 Remark on convergence time 230

9.5 Experimental results 231

9.5.1 Demonstration of the proactive planner and analysis of human effort reduction in different scenarios 232

9.5.2 Validation of Hypotheses and Discoveries through User Studies 236

9.6 Discussion on some complementary aspects and measure of proactivity 244

9.7 Until Now and The Next 246

9.1 Introduction

As discussed in introduction chapter, proactive behaviors are essential building blocks for supporting socio-cognitive expectation and adaptation. Robot, equipped with such proactive behavior could have enriched multi-modal interaction and co-operation capabilities as well as could develop more complex social behaviors in

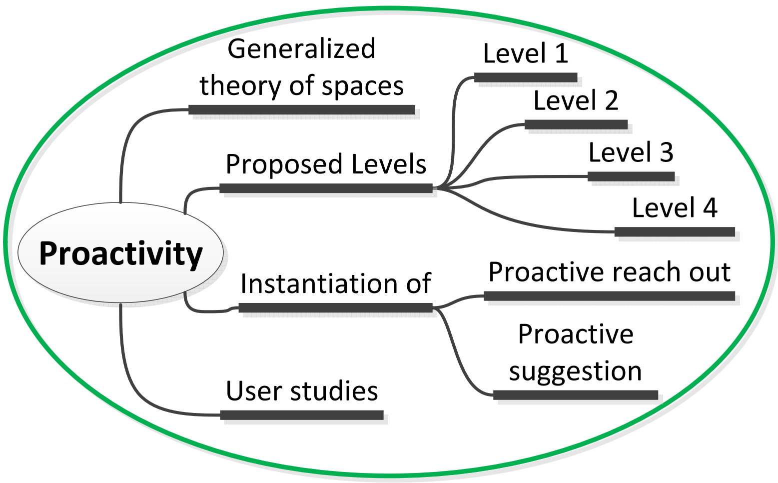


Figure 9.1: Contributions of this chapter: in terms of proposing a generalized theory for identifying spaces for synthesizing proactive behaviors and a way to regulate the "allowed proactivity" of an autonomous agent. Another contribution is a framework for instantiation of two proactive behaviors: reaching out proactively to take something in the case when human is expected to give something, and proactively suggesting the place where the human can put something so that robot will be able to take it later on. We have performed preliminary user studies to validate these proactive behaviors, which suggest that such behaviors reduce the human partner's *effort* and *confusion*, and the robots appear to be more *aware* to the users.

human centered environment. Such proactive behaviors are wide ranging and pose different challenges of synthesis and execution. However, we still lack a comprehensive analytical framework as a basis for their autonomous synthesis. In this chapter, we will first propose a generalized theory for synthesizing proactive behaviors by categorizing environment and action spaces and present 4 different levels of proactive behaviors. These categorizations could be used: (i) to reason upon the potential spaces to produce proactive behaviors, (ii) to formally regulate the "allowed proactivity" of an autonomous agent, (iii) to provide a mean to measure how much influential/severe the proactive behavior will be. Further we will show practical instantiations of few of their examples in human-robot cooperative manipulation scenario and validation through preliminary user studies, which will suggest that such behaviors reduce the human partner's effort and confusion. Figure 9.1 summarizes the contribution of the thesis.

9.2 Generalized Theory of Proactivity for HRI

9.2.1 Proactive Action

Dictionary definition of "proactive" is: "Acting in anticipation of future problems, needs and changes." [merriam webster.com b].

Hence, any action defined in section 3.3.3 of chapter 3 of general HRI theory, is proactive if it satisfies the additional characteristics mentioned above. Proactive actions by an autonomous intelligent agent could be synthesized in different spaces depending upon "how much" and "which parts" of the currently planned or being executed actions/roles of all the agents, and the outcomes will be altered. Next, we will define the proactive planning problem followed by identification of different spaces for reasoning on proactive actions.

9.2.2 Proactive Action Planning Problem

We define a proactive action planning $\mathcal{P}r$ problem as tuple:

$$\mathcal{P}r = \langle \mathcal{P}, A \rangle \tag{9.1}$$

where $\mathcal{P}$ is an instance of planning problem defined in eq. 3.39 as $\mathcal{P} = (\Sigma, s_0, g, F\_in, A\_in, F\_av, A\_av,)$, for the planning domain $\Sigma = (\mathcal{S}, \mathcal{A}, \mathcal{E}, \gamma)$ and A is the plan produced/provided for $\mathcal{P}$. To recall, s_0 is initial state of the environment (see eq. 3.33), g is the set of expressions must be satisfied in the goal state. F_in and F_av are set of expressions, which should be maintained and avoided during each step of the plan. A_av and A_in are the sets of actions, which should be avoided or incorporated in the plan.

$\mathcal{P}$ and A can be used to identify different spaces in environmental state and in action state to synthesize different proactive behaviors. It is important to note that A is already known to the proactive planner, which is either planned based on $\mathcal{P}$ or provided/proposed by agents in the environment, does not include proactive actions A^{pro} to be planned. Note that as mentioned in the section 3.4.4.8 of chapter 3, this provided plan A could even be a partial plan π, see eq. 3.42. The result of the proactive planning problem is the proactive plan as a sequence of actions $A^{pro} = \langle a_1^{pro}, a_2^{pro}, ...a_k^{pro} \rangle$.

One of the important aspects of proactive planning problem is to identify spaces for proactivity, which could be in the environmental state space $\mathcal{S}$ as well as in the action space $\mathcal{A}$: $S^{pro} \subseteq \mathcal{S} \times \mathcal{A}$. In the next section, we will address this issue.

9.2.3 Spaces for Proactivity

As mentioned in chapter 3, specifications about goal state and constraints result into a space $S_f \subseteq V_f$ for fact variable $f \in F$ in which its value can lie. V_f is set of all possible values of fact f and F is set of all facts, if grounded will determine the state of the environment (see chapter 3 for detail). A fact f is said to be *grounded* if $S_f = v_f$ such that $v_f \in V_f$ and $|S_f| = 1$, otherwise f is said to be *ungrounded*. If a fact f is ungrounded, we sub-classified it either as *constrained*, if $S_f \subset V_f$ and $|S_f| > 1$ or as *unconstrained*, if $S_f = V_f$ and $|S_f| > 1$. Conceptually, unconstrained fact variables are those whose values are not specified or restricted in the planning problem or in terms of constraints. Theoretically, such fact could take any value in its domain if the fact is independent of any other fact. The *undesired space* of a fact f is $S_f^{ud} = V_f - S_f$. Note that this definition assumes that if a fact is grounded or constrained, then all the values except the grounded value or outside the constrain region is undesired. Hence, this assumption also requires that if the values of a fact is constrained only in terms of undesired values then S_f must have been deduced as $S_f = V_f - S_f^{ud}$. Further, if nothing is specified about the value of f, then in that case $S_f = V_f$ and the undesired space will be $NULL$. Let us denote F_{gr} as the set of all the grounded fact variables and S_{gr} as the value space corresponding to all the grounded facts; F_{cn} as the set of all the constrained fact variables and S_{cn} is the values space corresponding to all the constrained facts; F_{uc} as the set of all the fact variables, which can take any value in their domain and S_{uc} as the value space corresponding to all such unconstrained facts; and F_{ud} as the set of all the fact variables, which has NOT_NULL undesired space and S_{ud} as the value space corresponding to all the undesired spaces of the facts. Hence, at a given instance of the planning problem, any fact variable f will either be $f \in F_{gr}$ or $f \in F_{cn}$ or $f \in F_{uc}$. However, if $f \notin F_{uc}$ then it will also be $f \in F_{ud}$, as there will always be some values, which will not be desirable. A state space S^g of the goal environment will be:

$$S^g = \overbrace{\prod_{f \in F_{gr}} v_f}^{S_{gr}} \times \overbrace{\prod_{f \in F_{cn}} S_f}^{S_{cn}} \times \overbrace{\prod_{f \in F_{uc}} V_f}^{S_{un}} \tag{9.2}$$

And a state s^g of the goal environment, provided S^g, will be $s^g \in S^g$ and given as:

$$s^g = \overbrace{\bigcup_{f \in F_{gr}} v_f}^{s^g_{gr}} \cup \overbrace{\bigcup_{f \in F_{cn}} v_f | v_f \in S_f}^{s^g_{cn}} \cup \overbrace{\bigcup_{f \in F_{uc}} v_f | v_f \in V_f}^{s^g_{uc}} \tag{9.3}$$

Let us assume that the provided plan A leads to an inferred goal environment s^g. This state was obtained by grounding the facts belonging to F_{cn} and F_{uc} of eq. 9.3. We say that any proactive behavior could try to choose a different value for a fact variable in eq. 9.3 while maintaining the state space or even could try to change various state spaces of eq. 9.2.

Further, at any instant of planning or execution there will be a set of restricted actions A_{res}, to facilitate avoiding unwanted behaviors. We use subscript *pa* for the agent intended to behave proactively and *oa* for all other agents in the environment. At a particular time instant, without synthesizing the proactive action, the action of all agents A as discussed above is $A = \{A_{pa} \cup A_{oa}\}$. If agent *pa* synthesizes/takes proactive action A_{pa}^{pro} then theoretically it can modify the existing sequence of actions of all the agents and the restricted actions space.

Hence, a proactive planner could decide to choose a particular value of the fact variables based on predicting the future needs and plan appropriate action to verbally or non-verbally communicate it, or could decide to plan some new action to facilitate A or avoid some future problems or ease some future requirements. Here the *future* could even be the very next sub-action and/or environmental state. Let us denote that proactive planner results into state space $S^{g\prime}$, state $s^{g\prime}$, action $A' = \{A_{pa}{}' \cup A_{oa}{}'\}$ and restricted action $A_{res}{}'$.

In the next section, we will present different levels of proactive behavior depending upon which spaces of the environmental state and action are getting modified and its impact on the beliefs of all the agents about the current environment, influence on the other agents' action, and the effect on the expected final environment. The focus of the chapter is not to provide an autonomous reasoning capability to synthesize a particular type of proactive action; however, we will device HRI based examples during the discussion of each level.

9.2.4 Proposed Levels of Proactive Behaviors

9.2.4.1 Level-1 Proactive Behavior

In this type of proactive behavior, the proactive agent *pa* takes initiatives to facilitate smooth, problem free achievement of the task but does not try to directly impose any additional desire or restriction on ongoing plan A as well on the goal state space, S^g of the environment. However, it can alter the set $s^g_{cn} \cup s^g_{uc}$ for some of the ungrounded fact variables by choosing different or better values than the current inferred one in goal environment state s^g. The proactive action A_{pa}^{pro} gets inserted at appropriate place in the sequence of already planned action of *pa*, i.e. $A_{pa}{}' = \{A_{pa}, A_{pa}^{pro}\}$ and is not intended to alter the actions of other agents i.e. $A_{oa} = A_{oa}{}'$. In this type of proactive behavior, $(S_{gr} = S_{gr}{}')$, $(S_{cn} = S_{cn}{}')$, $(S_{uc} = S_{uc}{}')$, $\left(s^g_{gr} = s^g_{gr}{}'\right)$, $\left((s^g_{cn} \cup s^g_{uc}) \neq \left(s^g_{cn}{}' \cup s^g_{uc}{}'\right)\right)$.

Further, the restricted action space will also be unaltered i.e. $A_{res} = A_{res}{}'$. This type of proactive actions will be executed stand-alone or in parallel to other agents actions.

Examples:

(i) Assume that the robot anticipates during the course of human's action that he can obliviously hit object O, and categorizes it as a state to be avoided. Then to avoid the event $hit\,(human, O)$, the proactive planner could find a new goal state, which prevents the *hit* event. If this new goal state is found by searching in the already provided ungrounded state space and the proactive plan to achieve that does not alter the plan of other agents, this can fall into level-1 proactive behavior. One possible new goal state could contain a different position P of O than its current position. And the synthesized proactive action could be $A\_pa^{pro} = Put\,(O, P)$, which can conceptually be viewed as putting the object away from the human's predicted trajectory.

(ii) Another example of level-1 proactive behavior could be, the robot cleans the table proactively by anticipating its future use by the human, $A_{pa}^{pro} = Clean\_Table\,(Robot, tab)$.

(iii) By anticipating some problem, the robot could synthesize a verbal proactive action $A_{pa}^{pro} = Say\,(\text{"}Be\ Careful...\text{"})$ to result into an 'aware' mental attentional state of the other agent.

Note that the same examples could fall into a different level of proactive behavior, if there were constraints, which would result into a more restricted state space, such as *maintain the current position of O* in the example (i), or *don't change the mental attention of human* in the example (iii). In such cases, the solutions found would not be lying within the ungrounded state space, $(S_{cn} \cup S'_{uc})$. However, they could still qualify for a proactive behavior of different levels as presented below.

9.2.4.2 Level-2 Proactive Behavior

The intention behind level-2 proactive behavior is to suggest some desire or constraint for 'better' achievement of the task, but not to contradict already 'specified' desire/constraints or the action types. This is basically done by reasoning on the values of those fact variables, which could put burden on other agents or could potentially create confusion, or achieved by better specifying the parameters of already planned actions.

In this case, space $S_{cn} \cup S_{uc}$ corresponding to ungrounded part is made more constrained or even grounded. And based on this new ungrounded state space, a goal state is chosen by altering $s^g_{cn} \cup s^g_{uc}$ and/or adding more elements in s^g_{gr}. Then the proactive action is planned to fully or partially achieve that goal state. Hence, $((S_{cn}{}' \cup S_{uc}{}') \subset (S_{cn} \cup S_{uc}))$, $s^g_{gr} \subseteq s^g_{gr}{}'$.

Here, the main difference lies in the fact that in level-1 those facts of the environment are changed, which are not directly affecting the action of other agents, whereas in level-2 other agents are expected to adapt by modifying the parameters of his/her/its already planned action. In this level $type\,(A_{oa}) = type\,(A_{oa}{}')$, but the parameters of the actions could be changed, $parm\,(A_{oa}) \neq parm\,(A_{oa}{}')$ which will become clear in the examples.

Generally A_{pa}^{pro} results into an intermediate environment, which in fact leads/influences other agents to adapt. Similar to level-1, $A_{pa}{}' = \{A_{pa}, A_{pa}^{pro}\}$. Further, $A_{res} \subseteq A_{res}{}'$, i.e. the restricted action space could also be appended but not reduced.

Apart from the intentions behind level-1 proactive behavior, level-2 proactive behavior is further intended to achieve some social constraints, to reduce the other agents' effort or confusion. Elevating the nature of level-1 proactive behavior, which is more inclined towards reactive side, level-2 proactive behavior tries to be suggestive.

Examples:

(i) Consider a *give object* task, in which the human has to give object O to the robot. Let us assume that the goal state specifications include the fact that the object O should be in the hand of the robot. And the already planned action includes that the human will pick and carry the object and the robot will grasp the object, i.e. $A = \langle pick(human, O), carry(human, O), grasp(robot, O)\rangle$. If there is no other constraint on the position of the robot's hand in the goal state, the proactive planner could find a "better" hand position instead of the current hand position, by considering social norms, human's effort, visibility, perspective taking, etc. Then it could plan some proactive action to achieve that "better" hand position for the task. One of such possible proactive actions could be $A_{pa}^{pro} = move\_hand\,(Robot, right, P)$ and insert at appropriate place in the already planned sequence of actions. This in-fact will be proactive reaching out to a position P as an attempt to 'ease' the object hand-over for the human. Note that the types of other actions have not been changed. Depending upon, at which stage of execution A_{pa}^{pro} is executed, this reaching out to take will generate an intermediate environment, and the parameter of human's action, i.e. the point, where to *carry*, is influenced.

(ii) Another example could be, the human is required to make an object accessible to the robot by putting it somewhere. The robot, instead of entirely leaving the human to decide about where to put, could proactively restrict the places where the human could put the object based on various human-adapted criteria. Then a proactive action could be planned to suggest this choice: $A_{pa}^{pro} = say\,(\text{"}you\ could\ place\ it\ at\ << description >> \text{"})$.

As will be shown in next section, such proactive behaviors also reduce the effort and confusion of the human partner.

9.2.4.3 Level-3 Proactive Behavior

Level-3 proactive behaviors are intended to provide a 'better alternative' for performing the task even by removing elements from already grounded part of the goal state of the environment or modifying the agent's roles/actions, i.e. $s_{gr}^{g} - {s_{gr}^{g}}' \neq NULL$, $A \neq A'$, $A_{res} \subseteq {A_{res}}'$. Hence, in this type of proactive behavior, ${s_{gr}^{g}}'$ will not necessarily contain all the elements of the grounded facts of the initially planned goal state of the environment. Such behaviors are generally to avoid the future problems or to incorporate the future requirements, left unnoticed during the initial planning or germinated by the plan itself or during the course of execution.

Examples:

(i) Assume that during the course of interaction, the existing plan consists of a task $show\left(object\left(O1\right), by\left(robot\right), to\left(human\right)\right)$. The constraint $visible\left(O1, human\right)$ is used to decide the space in which the goal environment will lie. And the planned action sequence is $A = \langle grasp\left(O1\right), carry\left(O1\right), hold\left(O1, at\left(P\right)\right)\rangle$. Now assume that the robot anticipated future need that the human might need another object *O2*, which is currently unreachable to the human. Further, assume that *O2* is occluding *O1* from the human's perspective. So, to incorporate this future need, the robot could proactively introduce another constraint $reachable\left(O2, human\right)$ and refine the space in which the goal state will lie. And proactive planner could result into a new plan $A_{pa}^{pro} = \langle grasp\left(O2\right), carry\left(O2\right), place\left(O2, at\left(P2\right)\right)\rangle$ to displace *O2* in a way so that *O1* becomes visible to the human and also the human can easily take *O2* in the future. In this case $A \neq A'$.

(ii) Let us consider a plan for the task of cleaning a table in cooperation with the human partner. Let us assume that the grounded part of goal state consists of $position\left(glass1, tray2\right) \in s_{gr}^{g}$ with the provided partial plan which consists of $A = \langle a1, a2, a3\rangle$ where $a1 = make\_accessible\left(robot, human, tray2\right)$, $a2 = put\_into\left(human, glass1, tray2\right)$ and $a3 = take\_away\left(robot, tray2\right)\rangle$, i.e. the robot has to make the *tray2* accessible to the human and the human is required to put *glass1* into *tray2* so that the robot will take it away. Now assume that during the course of execution at a particular moment, the human is currently holding the *glass1*, however, *tray2* is with the robot and the robot is away from the human. Assume that the robot has anticipated some significant delay in performing its part of action. Then being a social robot and to avoid the human holding the glass and unnecessary waiting for the robot, it can proactively generate a different goal state by altering already grounded fact about the $glass1$ position to $position\left(glass1, tray1\right) \in {s_{gr}^{g}}'$ and generate a proactive action $A_{pa}^{pro} = say\left(\text{"}you\ could\ place\ it\ on\ tray1\text{"}\right)$, and the robot will take it away later on. Hence, $s_{gr}^{g} - {s_{gr}^{g}}' = position\left(glass1, tray2\right)$ i.e. $s_{gr}^{g} - {s_{gr}^{g}}' \neq NULL$.

Synthesizing such proactive behavior requires a stronger and more reliable reasoning mechanism, as both the desired grounded environment state and the way to achieve are altered. The robot has to make sure that there is no "problem", "conflict" and "harm" in making such alteration.

9.2.4.4 Level-4 Proactive Behavior

This type of proactive actions influence the restricted spaces of the goal environment and of the action, by removing some of the restrictions, i.e. $A_{res} - {A_{res}}' \neq NULL$ and $S_{ud} - {S_{ud}}' \neq NULL$. This is mainly desirable in the cases where: (a) the *pa* anticipates that removing them will cause no problem/damage and will facilitate

Level of Proactive Action A^{pro}	Allowed changes (s: set of grounded facts, S: space in which the fact variables should be grounded, A: action)				Intentions from task's perspective	Intention/ nature of A^{pro}	Influence on *environment* sub-spaces	Desirable level of reliability of the proactive agent	Severity of A^{pro}
	can alter the goal state part	can alter the current plan part	can expand the goal state space and action space parts	can reduce the goal state space and action space parts					
Level-1	s^g_{cn} , s^g_{uc}	A_{pa}			smooth, problem free achievement	responsive, alterning	better value for un-grounded part	Average	Lowest
Level-2		$parm(A_{oa})$	S_{ud} , A_{res}	S_{uc} , S_{cn}	better achievement	suggestive, influencing, adjusting, fine-tuning	better specify ungrounded space		
Level-3	s^g_{gr}	A_{pa} , A_{oa}	A_{res}		better alternative	imposing	alter desired part		
Level-4				S_{ud} , A_{res}	avoid severer problem	obligating, risking	alter some undesired parts	Highly reliable	Highest

Figure 9.2: Summary of different levels of proactive actions proposed in this chapter. Only the key distinguishing aspects of allowed changes have been shown from the second to the fourth columns.

better achievement of the task, (b) the pa anticipates some more severe problems and to proactively avoid them require violating some of the restrictions.

A further strong and reliable reasoning mechanism is required for an artificial agent to take such decisions proactively, because of possible underlying safety concerns and undesirable consequences.

Examples:

(i) The robot proactively fetches the attention of the human to warn about some anticipated problem where initially the restricted state space was determined by the constraint: $change\,(attention\_mental \wedge attention\_physical, Human)$ should be avoided.

(ii) The robot moves its hand away or turns its wrist with jerk while carrying a glass of water by anticipating collision due to the movement of other agents, where initially $\{jerk, turn\_wrist\} \in A_{res}$ for the task of carrying a filled glass.

Designing an artificial agent capable of autonomously synthesizing level-4 proactive behaviors requires that the artificial agent should satisfy highest level of reliability and safety criteria. Because, it will be capable of deciding to perform some action or exhibit some behavior which might be restricted for the current context. Also, it could change the state of the environment into a state, which is restricted in the given context of the planning problem.

Figure 9.2 summarizes the proposed 4 levels of proactive behaviors from various perspectives. Note that it shows only the key distinguishing aspects of allowed changes, not what all could be changed at different levels. Further, note that severity is highest in level-4 as it removes restricted actions or states and lowest in level-1 as it just modifies the state within the acceptable ranges of unconstrained or unrestricted facts and that too by assuring no conflicts with existing specifications. The range of qualifying the reliability requirements begins with *Average*, because as it is proactive planning problem $\mathcal{P}r$. And as shown in eq. 9.1, it already includes a plan A by a task planner. Hence, we find it logical to place reliability requirement of a proactive planner above the task planners, which reasons on $\mathcal{P}$.

9.3 Instantiation

As a step towards validating different proactive behaviors, we are essentially interested here in one particular non-trivial aspect: to determine *where* (an important aspect of joint task [Sebanz 2009]) a joint action should preferably take place and what the robot can do to *propose/communicate* the choice it has made. Hence, we chose to instantiate the two example scenarios of level-2 proactive behavior mentioned earlier: (i) give object and (ii) make object accessible tasks by the human to the robot. We made this choice because firstly we could explore the two different modes of proactive action and secondly it is based on physical cooperative tasks, so the user's effort and confusion could be visually analyzed from the point of proactivity of the robot.

We have already developed a system, which enables the robot to perform a set of basic *human-robot interactive manipulation* tasks such as give, show, hide, make an object accessible to the human in chapter 7. In this chapter, we will focus on a complementary aspect of such interactive manipulation in which the human will be required to perform the task or to contribute to a joint task in order to achieve some joint goal. We will consider two such tasks: (i) *Give*: The human has to give some object to the robot by holding it somewhere. (ii) *Make Accessible*: The human has to make some object accessible to the robot by placing it somewhere.

As shown in figure 9.3(a) if the robot asks, "*Please give me the toy dog.*" and remains in the rest position, this might create confusion for human about how and where to give: "Should I move and reach near to the robot?" or "should I stand and put the object in the hand of the robot?" or "should I put it at a place somewhere on the table for the robot to take it?" etc. But if the robot, along with the request to give, shows proactive reach out behavior by moving its hand towards a feasible and convenient place for the human, to take the object, as shown in figure 9.3(b), it will significantly guide the human about how to perform the task and where to give. Further, the robot might better communicate its abilities and its understanding about the abilities of the human partner.

Similarly, if the human has to make some object accessible to the robot and if the

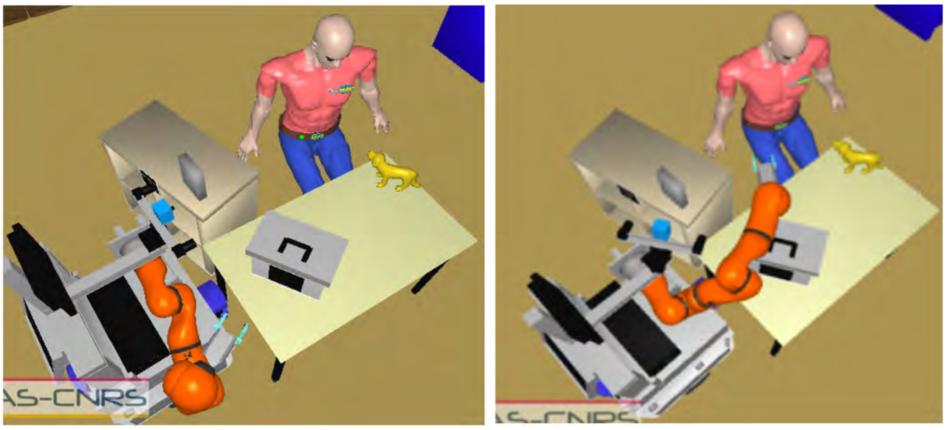


Figure 9.3: Robot asks to the human, "*please give me the toy dog.*" (a) by maintaining its rest position, (b) by proactively moving its hand to appropriate place to guide human's action.

robot proactively advises the human about where to put the object so that the robot could take it, it will significantly guide the human about how and where to perform the task. Again the robot might better communicate its abilities and its understanding about the abilities of the human partner.

9.3.1 Objective of the hypothesized proactive behavior

Actually, in both the tasks discussed above, in the type of proactive behavior we are talking about, the robot partially or fully synthesizes a solution for the task and proactively communicates it through different means. In our examples, it is computing 'where' the human can give or put the object and communicating it by reaching out to take or suggesting the place to put. The intention behind such proactive behavior is to guide and facilitate the human partner to better perform the task or the joint goal by reducing task related confusion and effort of the human.

For illustration, see figure 9.4, where an ellipse shows one state of the environment, and an edge shows an action. In the absence of any proactive behavior, one of the states of the expected final environment will be s^g. In the case of proactive behavior, an intermediate state of the environment s_{intr} is created by instantiating or better specifying the facts related to the ungrounded space of the environment and achieved by the proactive action A_r^{pro}. Hence, the proactive behavior leads to a change in the physical and/or mental state, which also incorporates the belief, of the human. And the human adapts to it to achieve the task by action A_h^{adp}, by modifying the parameters of his/her already planned action. This might result into a partial different final state of the environment, $s^{g'}$. In this section, we assume that the proactive behaviors of level-2, which do not remove the set of already planned action A as well as preserve the already grounded part of the desired goal state. Hence, there is a common part in the goal states of the environment, which

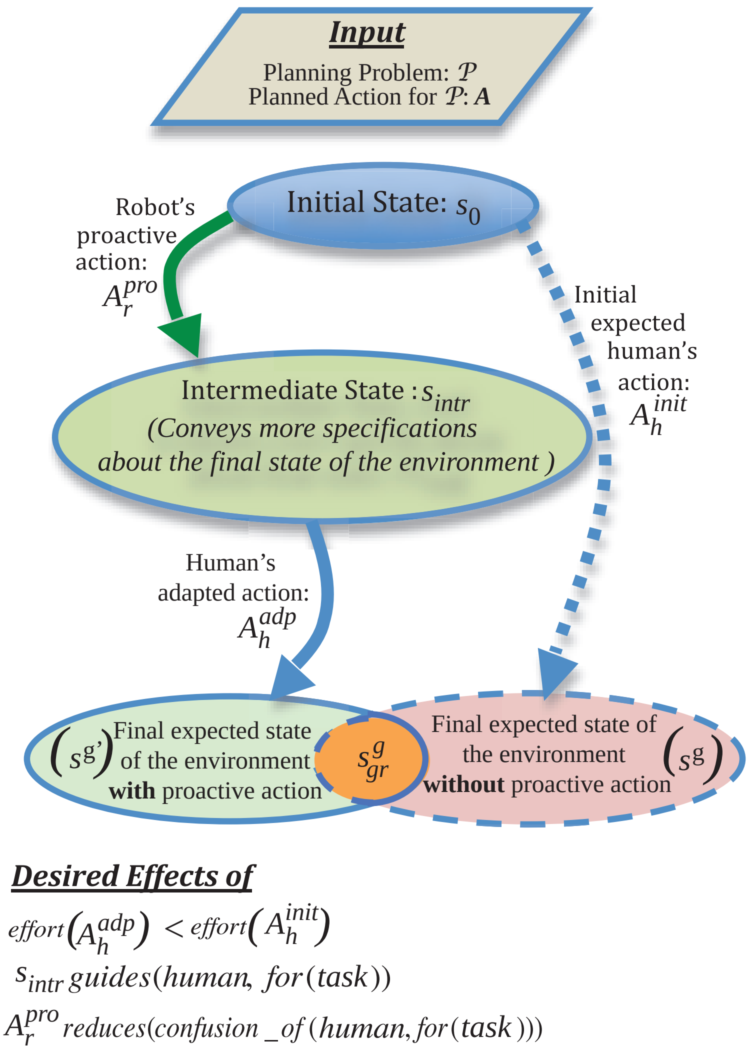


Figure 9.4: A type of proactive behavior, which results into an intermediate physical state of the world and/or changed mental state of the human, mainly to reduce the confusion and effort of the human for smooth execution of the joint task.

is denoted as s_{gr}^{g} in figure 9.4.

Before formally hypothesizing different proactive behaviors for basic human-robot interactive tasks, first we will briefly describe the essential ingredients to be incorporated in such behaviors (for comprehensive features, see the survey of socially interactive robots [Kopp 2004]).

Our interest is to device proactive behaviors based on 'where' the task could be performed. Let us derive the interest behind such proactive behaviors and at the same time identify some of their key ingredients. In [Holthaus 2011], it has been shown that in general the participants appreciate the robot's initiatives of showing some movements as an engagement attempt. The experiment was in a receptionist-visitor scenario, and the type of initiative was gaze shifting, which is different from our HRI object manipulation scenario and the associated proactive behaviors' types. However, their finding that the participants rated the robot higher when it showed some movements in its gaze than just being still, points that proactive initiatives of the robot could better engage the human and serve the purpose of interaction opening. Further, in [Li 2011], it has been shown that simple arm-head gesture increases the expressive power of the social robots. Although, such movements have not been directly studied in relation to proactivity, we will hypothesize proactive behaviors, which will incorporate arm and head movements, as an attempt to be expressive.

[Kozima 2001] suggests that to be intentional the robot should exhibit goal-directed actions. In [Holthaus 2010], it has been shown that simple changes in robot's gaze could show robot's attention and intentions to the human partner. Further, in [Imai 2003], it has been found that robot's eye contact and hand movement with situated dialog help in achieving joint attention with human partner. Moreover, a robot moving its arm to a location could induce human goal anticipatory response as demonstrated in children [Gredeback 2010]. Further, regarding the object manipulation, which is our focus, we find that gazing plays an important role in pointing-based object-reference conversation [Iio 2011]. Therefore, in our hypothesized proactive behaviors, we incorporate to look at the object and the place of interest.

All these suggest that goal-directed gaze and hand movement are basic blocks for engaging the human and expressing intention. Hence, our hypothesized proactive behavior associated with a particular task incorporates goal directed behaviors at two levels: (a) at task level: by reaching out to take the object from the human or looking at the suggested location at which the human can place the object, (b) at object level: by reaching out towards the desired object or looking at the desired object and the places of interest, thus also incorporates the gaze and the hand movements.

Expressive behavior coupled with perspective taking has been shown to be important for socio-cognitive aspect of Human-Robot Interaction, [Breazeal 2009]. Hence, we elevate the proactive behavior by incorporating the reasoning on the effort from the human partner's perspective. This aspect of *human-adapted proactive behavior*, makes the robot to find a solution, which tries to maintain the least feasible effort from the visuo-spatial perspective of the human partner.

Estimating 'where' the human can perform a task is helpful for sharing attention with others [Tomasello 2005] and to predict spatial characteristics of the others actions [Jordan 2008]. These are essential for building the robot's theory of mind [Scassellati 2002] and consequently can help to guide human's behavior [Zwickel 2009], [Kockler 2010] towards robot.

So, our focus will be this aspect of estimating 'where' a task could be performed by the human and we will hypothesize different proactive behavior based on this 'where' information.

9.3.2 Hypothesized Proactive Behavior for Evaluation

For interactive human robot joint tasks, we hypothesize the following:

9.3.2.1 Proactive Reach Out to *Take* from the Human

We postulate that along with informing verbally, the robot should proactively reach out to take, in the case the human has to give something to it.

9.3.2.2 Proactively Suggesting 'where' to Place

We postulate that the robot should proactively suggest (verbally and by gaze shifting) about 'where' to place an object, in the case the human has to make the object accessible to the robot.

9.3.3 Hypotheses about the effects of the human-adapted proactive behaviors in the joint task

9.3.3.1 Reduction in human partner's *confusion*

We hypothesize that such proactive behaviors will be preferred over non-proactive behavior and will reduce the 'confusion' of the human partner.

9.3.3.2 Reduction in human partner's *effort*

We will present a generic framework to find a solution for different proactive behaviors. The framework estimates 'where' the human can perform the task by respecting environmental and postural constraints and where the robot can support the task. Our framework will provide a *human adaptive proactive solution*, by taking into account human perspective and effort.

We further hypothesize that such proactive behaviors, by incorporating the *human-adapted* aspect will also reduce the 'effort' of the human.

9.3.3.3 Effect on *perceived awareness* of the robot

We further hypothesize that with such human-adapted proactive behaviors, the robot should be perceived as being more 'aware' about the human's capabilities,

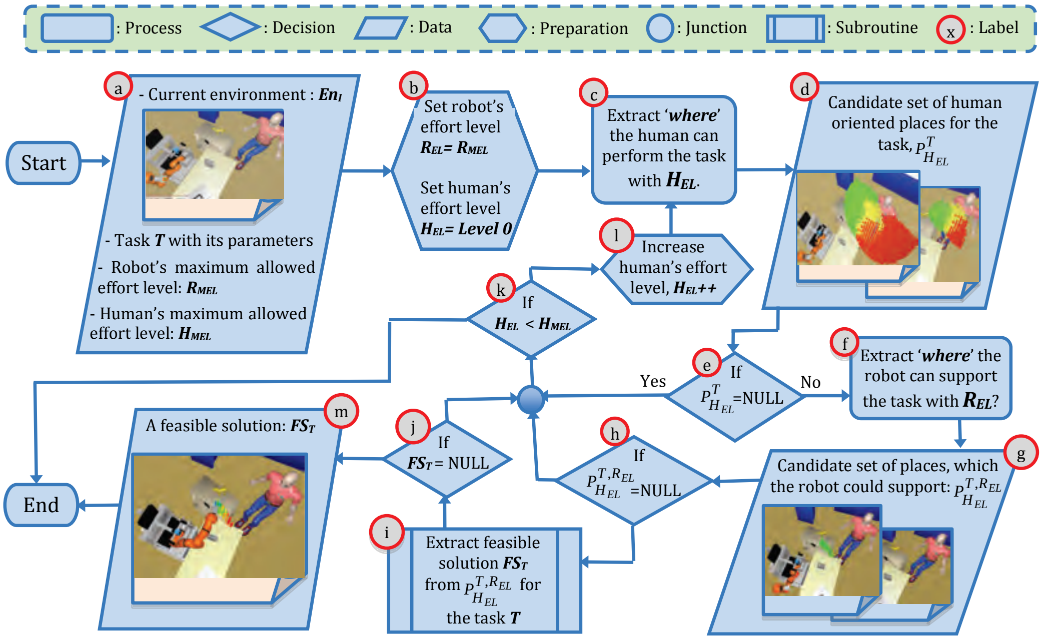


Figure 9.5: Proactive planner: Reasoning to find a solution for proactive behavior while ensuring least feasible effort of the human partner for performing the joint task.

more 'supportive' to the task and to the human and more 'communicative' about its own capabilities.

Next, we will discuss the framework to instantiate the hypothesized proactive behaviors. Further, we will discuss the results of the user studies as an attempt to validate the above hypotheses.

9.3.4 Framework to Instantiate 'where' based Proactive Action

The task is assumed to be known to the robot and the type of the proactive behavior associated to the task is also known to the robot. In the context of behaving proactive, it is important for the robot to be aware about human's capabilities and show proactive behaviors, which reduce the effort of the human as well. We have adapted our framework presented in chapter 7 to find a solution 'where' the human can perform the task by respecting environmental and postural constraints and where the robot can support the task, providing a partial but proactive solution to the joint task. It will further maintain the least feasible effort of the human partner while finding the feasible solution.

Figure 9.5 shows the reasoning process for extracting a solution to behave proactively. As shown in block *'a'*, input consists of the current environment state

$En_I = s_0$, the task T with its parameters: performing agent, target agent (for whom the task would be performed), the name of the object. In addition, the current maximum allowed effort level of the robot, R_{MEL} and of the human, H_{MEL} are provided. The planner reasons about 'where' the human can perform the task. As an attempt to ensure minimum feasible effort by the human, initially the maximum allowed effort level of the human, H_{EL} is set as *level 0*, i.e. *No_Effort_Required* to *see* and *reach* the places to perform the task (block '*b*' of figure 9.5). In block '*c*' of figure 9.5, the planner will find the candidate places where the human can perform the task T with his current maximum effort level and in block 'f' the planner finds the set of candidate points where the robot can support the task performed by the human. Both these can be found in one step by solving the expression 5.5 presented in affordance analysis section 5.2.4 of chapter 5. Depending upon the task T, for a particular agent Ag, the planner is already provided with a set of constraints to find the set of candidate places as done in chapter 7 for different tasks.

Block *'g'* of figure 9.5 shows the set of places where the human might be able to perform the task T for the given effort level, H_{EL} and the robot will be able to support it. For effort level 0, in which the human is not expected to even move the arm, this will be *NOT NULL* only if the object is already in the human's hand. The human has to just maintain his/her posture and the robot will be expected to take the object from his/her hand. In this case, $P_{H_{EL}}^{T,R_{EL}}$ will be a set of points corresponding to the object's current position.

The planner further ranks the candidate places obtained in block *'g'* and rank them. Currently we assign weights based on the closeness to the target-object position. This is based on the assumption that the human needs to put less effort in placing or holding the object if he/she has to carry the object for a shorter distance. Another motivation behind such weight assignment is to exhibit goal-directed behavior. In [Kozima 2001], it has been suggested that to be intentional the robot should exhibit goal-directed actions. Hence, this weight assignment also drives the solution to be directed towards the object, which might also inherit the notion of *pointing* to the desired object.

Assignment of such weights for candidate places for proactive behavior and the studies and frameworks for preferable hand-over positions such as [Cakmak 2011], [Huber 2009], [Dehais 2011] could mutually benefit. For the latter case the possibility of proactive behavior by the human, who will be the receiver, could be incorporated. Whereas, for the former case in which the robot will be the receiver could take into account further aspects related to preferable hand-over positions.

If candidate places to support the task is NOT NULL then the planner performs various feasibility tests on these candidate places to extract a feasible solution. And for this we feed the candidate places to the generalized HRI object manipulation task planner presented in section 7.6 and illustrated in figure 7.7 of chapter 7. Note that at any stage of planning if the planner fails to find a candidate place or a feasible solution, and if there is still a possibility of increasing the effort level (block

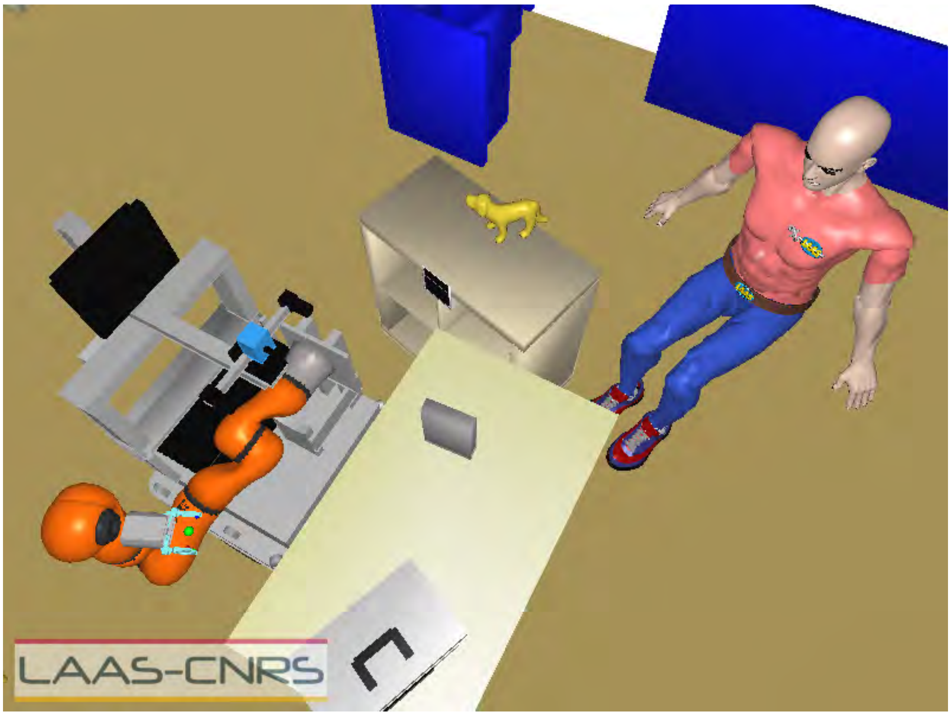


Figure 9.6: Initial scenario in which the human has to give the toy horse to the robot. The robot will plan to proactively reach out to take.

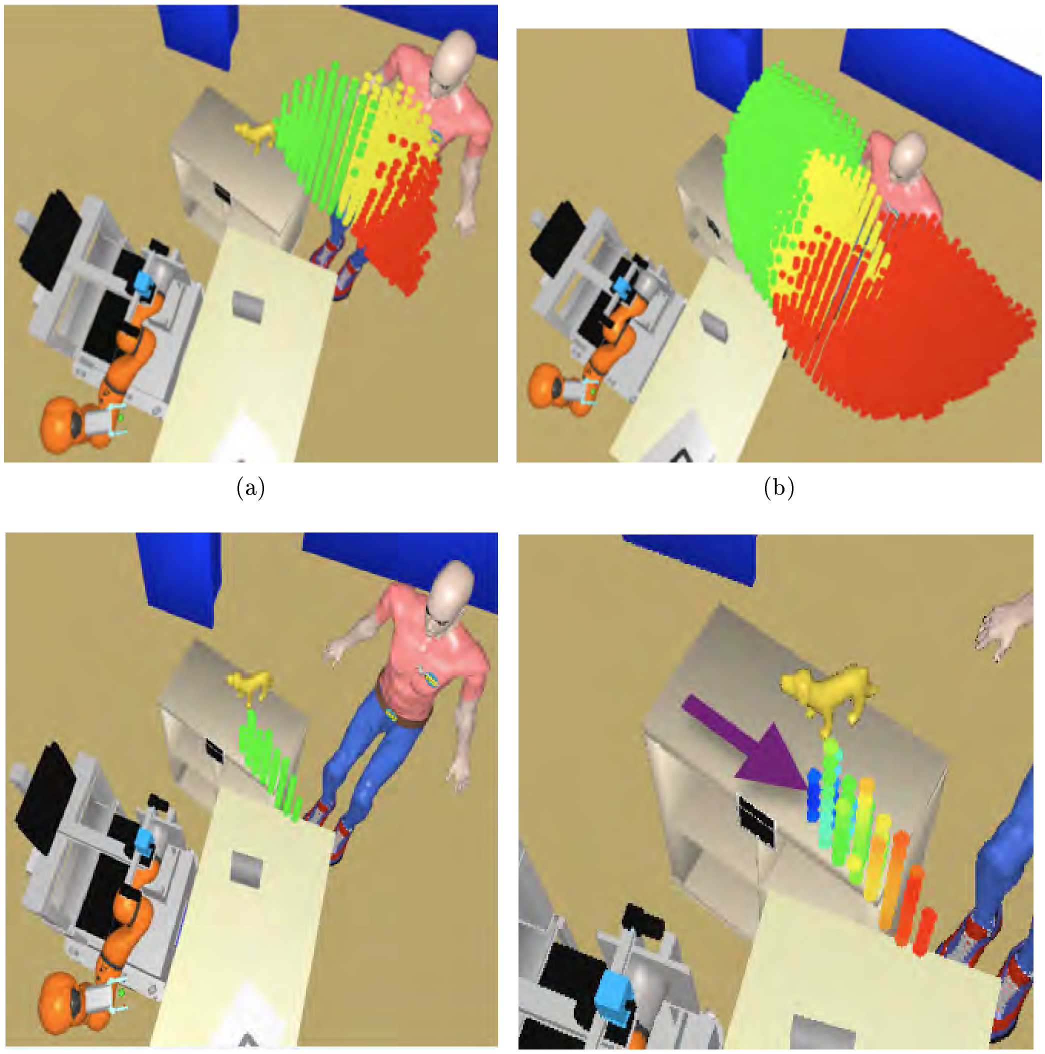
'k'), it increases the acceptable effort level of the human in block *'l'* and begins the next iteration.

9.4 Illustration of the framework for different tasks

In this section we will illustrate the presented proactive planner for finding a feasible solution to behave proactively. We will illustrate for two different tasks performed by the human: giving an object to the robot and making an object accessible to the robot.

9.4.1 For "Give" task by the human: Proactively reaching out

Figure 9.6 shows the initial scenario in which the human has to give the toy dog, placed on his right, to the robot. The robot will plan to proactively reach out to take. For the current example, as it is a tabletop cooperative manipulation scenario, to avoid more expensive motions of the robot, its maximum allowed effort is set as *Arm_Effort*. This restricts the robot from planning to turn or move its base, it can only move its arm for achieving the current sub-task of getting the object from the human. And the human maximum effort level is provided as *Whole_Body_Effort*.



(a) (b)

(c) (d)

Figure 9.7: Candidate points for giving an object by human (green: by right hand, red: by left hand, yellow: by both hands) (a) from his current position, *No_Effort_Required*, (b) if human will make effort to move his torso (lean forward or turn) while remain seated, *Arm_Torso_Effort*. (c) Candidate points from where robot can take the object for the effort (b) of human, (d) weight assignment on the candidate points based on the nearness to the target-object, the toy dog.

However, these could be modified online by higher-level decision-making or supervisor systems, such as ours [Alili 2009], [Clodic 2009].

As already explained in section 9.3.4, for the human, initially the least effort level, *No_Effort_Required*, will be tested for extracting the candidate places to give the object. The planner will get the candidate set of places in block *'d'* of figure 9.5 as *NULL*, as the object is not already in the human's hand. Hence, the control reaches

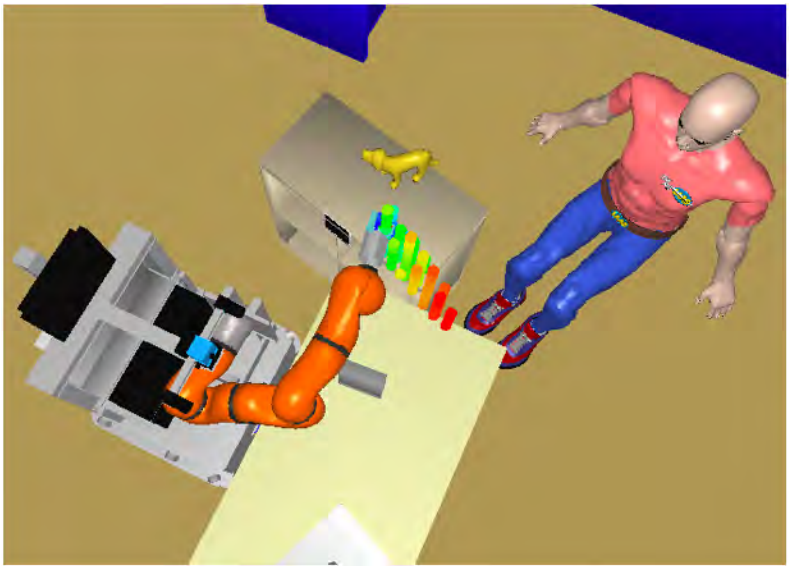


Figure 9.8: Estimated final robot configuration for proactive reach.

to the block *'k'* and increases the human's reach effort level to *Arm_Effort* in block *'l'*, (we chose to maintain the corresponding effort level to see as *No_Effort*). This means that the planner estimates the places, which are in the current field of view of the human and where the human could give the object to someone, if he will only stretch out his arm. Figure 9.7(a) shows the candidate places for giving the object by the human for this level of effort, obtained in block *'d'* of figure 9.5 in the next iteration. Green, red and yellow points show giving possibilities by right, left and both hands respectively. In block *'g'* of figure 9.5 with the maximum allowed effort level of the robot, the planner extracts the subset of the places where it can support the human. For our current example, this turned out to be *NULL*, as there was no point reachable by the robot with its current effort level among the points of figure 9.7(a). Hence, the planner again reaches to the block *'l'* of figure 9.5 to test for the next effort level of the human, by setting *Arm_Torso_Effort* to reach and *Head_Torso_Effort* to see. Figure 9.7(b) shows the candidate points for giving the object by the human, who is now expected to lean forward and/or turn around while being seated. In this iteration, in block *'g'* of figure 9.5 the planner finds a set of candidate places, from where the robot could take an object from the human. Figure 9.7(c) shows these candidate places as green point cloud. The resultant candidate places after the weight assignment as explained in section 9.3.4, have been shown in figure 9.7(d). Blue points have the highest weight in the sense they will be preferred over the red points having lowest weights. The feasible solution corresponds to the first highest weight candidate point, which passes the rest of the grasp, placement, object visibility and trajectory oriented feasibility tests. This feasible solution obtained in block *'m'* of figure 9.5, has been indicated in figure 9.7(d). At the end, depending upon the task, the planner returns appropriate data for exhibiting proactive behavior. For the current task, it returns the winner feasible

Figure 9.9: : Task of making an object accessible by the human to the robot by putting it at appropriate place for the robot to take. (a) Places on the support planes where the human can put the object with least effort. (b) Weighted points where the robot can support the human by taking the object. (c) Robot found the predicted possible placement of the object on the box from where it is feasible for the robot to take.

place obtained in figure 9.7(d), the corresponding levels of efforts for the human and for the robot, the trajectory to reach the place, and the estimated end configuration of the robot, as shown in figure 9.8.

9.4.2 For "Make Accessible" task by human: Suggesting 'where' to place

The robot finds 'where' the human can put the object for the robot to take and proactively suggests the human about that place. As the robot is able to find the horizontal surfaces in the environment as the candidate points to place some object. Hence, in block '*d*' of figure 9.5, the planner finds the places at the top of the box as well, where the human can put the object as shown figure 9.9(a). Figure 9.9(b) shows the weighted candidate points to perform the task. Figure 9.9(c) shows the feasible estimated placement of the object obtained in block '*m*' from where the robot could take it. Apart from the similar information for proactive reach out task, the proactive planner also provides the symbolic information that the placement is '*on the box*' based on the reasoning on the inter-object spatial relations. Incorporating other predicates such as left, right, next to, etc. could further enrich the location description while suggesting the place to put.

9.4.3 Remark on convergence time

As the first step, the main focus of this thesis is to incorporate the key elements of grasp, visibility, placement, feasibility of trajectory, etc. from the perspective of both agents: the robot and the human. One of our future works is to further optimize the iterative approach presented in figure 9.5. So, we will provide an approximate idea about the convergence time.

As the candidate search space based on Mightability Maps could be updated online,

Figure 9.10: (a) Initial scenario for giving the object grey tape marked by red arrow. (b) The human is trying to give by standing up, *Whole_ Body_ Effort*, in the absence of proactive reach behavior by the robot. (c) The human is giving just by leaning forward, *Arm_ Torso_ Effort* in the case of proactive reach by the robot.

Figure 9.11: (a) Another Scenario for the task of giving an object to the robot. (b) In the absence of any proactive behavior human is standing up and reaching to the robot (*Whole_ Body_ Effort*) to give the object (c) With proactive reach behavior of robot user is giving the object by only *Arm_ Effort*.

and the initial lists of grasp and placement are stored by calculating only once for each new object. Hence, the convergence time for the algorithm mainly depends upon the number of times it has to backtrack due to failure of any of the tests in figure 9.5 and the time taken by the path planner, which is presently a *RRT* based planner [Gharbi 2008]. The algorithm finds a feasible solution for the typical scenario, shown in figure 9.6, in *1.6* seconds. The convergence times for other scenarios and tasks presented throughout the chapter varies between *0.5* seconds to *30* seconds.

9.5 Experimental results

We have tested our system on two different robots: JIDO a home-built mobile manipulator equipped with a LWR Kuka arm and PR2 from Willow Garage.

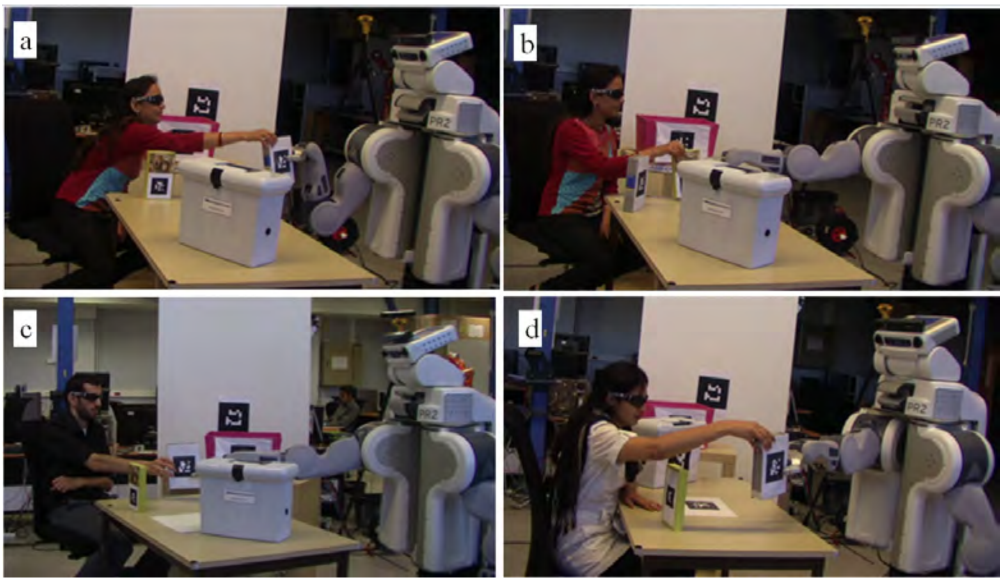


Figure 9.12: : Experiments with another robot PR2 for give task by the users. (a) The user is putting more effort (*Arm_ Torso_ Effort*) in the absence of any proactive reach towards object behavior by robot. (b) The user is giving with less effort (*Arm_ Effort*) when robot is reaching out proactively. (c) and (d) The robot is successfully able to find a solution for proactive reach out in different scenarios and the user is putting only the *Arm_ Effort* to give the object in both scenarios.

9.5.1 Demonstration of the proactive planner and analysis of human effort reduction in different scenarios

This sub-section will be confined to showing two aspects: (i) The planner is generic and independent of the scenario and the robot. (ii) The resultant solution visibly reduces the human efforts in different situations, when seen through the perspective of effort levels presented in table 4.5. In the next sub-section, we will show and analyze the results of the preliminary user studies to demonstrate the supportive and encouraging evidences of the proactive behaviors hypothesized in this chapter.

9.5.1.1 For proactive reach out for 'give' task by the human in different scenarios

Figure 9.10(a) shows an initial scenario in which the robot requests the human to give the object indicated by the red arrow. Figure 9.10(b) shows the final scenario, where the human is giving an object to the robot for the case when the robot did not move its hand proactively. Human is standing and trying to give the object, hence putting *Whole_ Body_ Effort* (see table 4.5). But in the case when the robot was allowed to behave proactively, the proactive planner successfully finds a feasible place to take the object from the human, while ensuring minimum feasible effort by the human. Figure 9.10(c) shows the case in which the robot is proactively reaching to the feasible place to take the object. This proactive behavior has reduced the human's effort for the task as the human is just leaning forward from the seated position to give the object. Hence, the effort is *Arm_ Torso_ Effort* instead of *Whole_ Body_ Effort* of standing up and leaning. Figure 9.11 shows another scenario where the human and the robot are sitting in a different spatial arrangement than scenario of figure 9.10(a). Figure 9.11(a) shows initial scenario and the position of the object to be given by the human. Figure 9.11(b) shows the situation of non-proactive behavior, the human is standing and giving the object to the robot. But as shown in figure 9.11(c), the proactive planner is able to find a different human adapted reach out place, than that of figure 9.10(c). And the human can hand over the object to the robot from his current position itself. This time the human effort level has been reduced from *Whole_ Body_ Effort* to *Arm_ Effort*.

We have further tested our system on another robot PR2, to illustrate the portability of the system and the ability of the proactive planner to take into account different robots of different kinematic structures.

Figure 9.12(a) shows the user giving the object without robot's proactive reach behavior, whereas figure 9.12(b) shows the user is giving the object with less effort when the robot has proactively moved its arm. In this case, the human effort has been reduced from *Arm_ Torso_ Effort* to *Arm_ Effort*. Figure 9.12(c) and (d) show two different scenarios and the planner is able to find a feasible reach out solution for PR2 robot. Both the users are giving the object with *Arm_ Effort* in the case of proactive reach out by the robot.

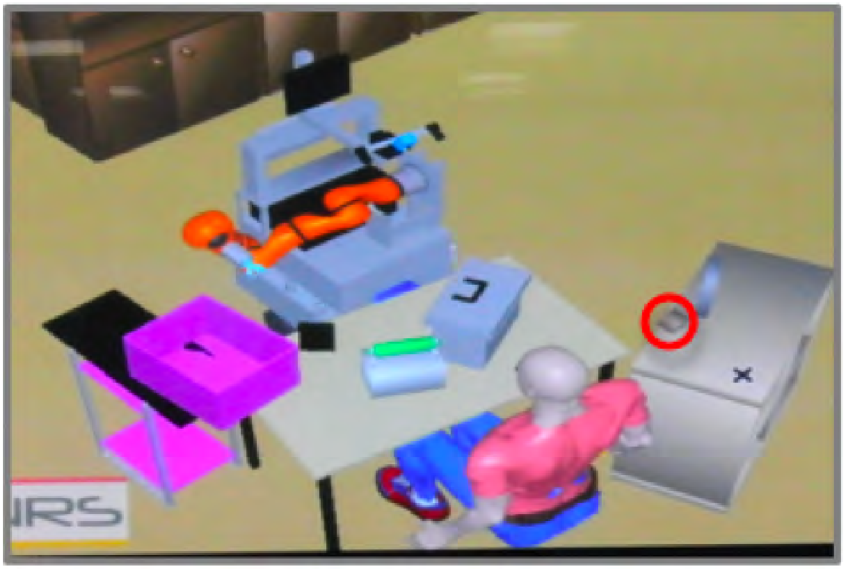
9.5.1.2 Finding solution to proactively suggest the place for make accessible task in different scenarios

In this sub-section we will show the results for finding the solution for proactively suggesting the human about where to put the object to make it accessible to the robot, while ensuring the least feasible effort of the human.

Figure 9.13 shows the initial scenario and its real time 3D representation. Figure 9.14(a) shows the human is putting the object close to the robot on the table to make it accessible to the robot. This required *Arm_ Torso_ Effort* of the human. In the second case of showing proactive behavior, the planner finds a feasible place for the human to put from where the robot could take the object, while ensuring least feasible effort by the human. The robot proactively suggests the human to put the object on the box and as shown in figure 9.14(b) the human is placing the object at the white box with less effort, *Arm_ Effort* only.

In figure 9.15, the chair has been placed away from the table to make the human sitting relatively away from the table compared to the scenario of figure 9.14. In this scenario, in the absence of proactive suggestion from the robot, the human is standing up and leaning forward to make the object accessible to the robot, i.e. with

(a)



(b)

Figure 9.13: Task of making an object accessible by the human to the robot by putting it at appropriate place, so that the robot might be able to see and take it. (a) initial scenario (b) Real time 3D representation of initial scenario, by the robot through various sensors. The object, which the robot will request to make accessible is encircled in red in (a) and (b).

Whole_ Body_ Effort, figure 9.15(a). But when the robot uses the proactive planner, it finds a feasible place with reduced human effort and suggests it to the human. As shown in figure 9.15(b) with such proactive suggestion, now the human is putting the object on the box by leaning forward, which is *Arm_ Torso_ Effort*. Note that

Figure 9.14: The human is making an object accessible to the robot for the initial scenario of figure 9.13. (a) Without proactive suggestion about where to place, the human is putting it close to the robot with *Arm_Torso_Effort*. (b) With the human adapted proactive suggestion by the robot, the human is now putting it on the white box as suggested by the robot. This has reduced the human's effort to *Arm_Effort*.

Figure 9.15: Make accessible task: (a) Without proactive suggestion about where to place, the human is putting it close to the robot on table by standing and leaning forward with *Whole_Body_Effort*. (b) With the human adapted proactive suggestion by the robot to put it on the white box, the human is now required to put *Arm_Torso_Effort* only. Note that the planner could not find a feasible solution for *Arm_Effort* of the human, as was the case for figure 9.14. This is because the human was sitting relatively away from the table and the robot was not able to support the task for *Arm_Effort* of the human with its maximum allowed effort level, which was also set as *Arm_Effort*.

in this case, the robot with its current allowed maximum effort level, which is set as *Arm_Effort*, was not able to support the human for his *Arm_Effort* level, as was the case for scenario of figure 9.14.

Hence, the presented planner is not only able to find a feasible solution for different scenarios for both the tasks, but also in most of the cases it is successfully able to reduce the effort of the human partner.

In the next section, we will present the results and interesting facts revealed through preliminary user studies to further analyze the effect of such proactive behaviors and to validate our hypotheses.

9.5.2 Validation of Hypotheses and Discoveries through User Studies

In all the experiments the speech of the robot was scripted, only some of the parameters were synthesized, such as the name of the object and the name of the support (object or piece of furniture) returned by the proactive planner.

The experiments are controlled in the sense when the user sits comfortably on the chair in the scenario, then the remote operator starts the script, which begins by the robot speaking "I need your help..." and if the human looks at the robot (detected by the robot through visual perspective taking of the human) it assumes that the joint attention has been established otherwise it continues to repeat the initial sentence. Once the joint attention has been established, it shows non-proactive or proactive behavior by finding the solution. The names of the task and the target object are provided to the script.

We have performed a series of preliminary user studies to validate the hypotheses and discover the effects of the proactive behaviors of the robot on the users compared to the non-proactive behaviors. In fact, the figures shown in previous sections are from that user study. The two main aspects we want to validate are:

(i) Whether the users are experiencing the *reduction in confusion* about the task because of the hypothesized expressive proactive behaviors or not.

(ii) As the presented framework takes into account the human partner's visuo-spatial perspective and effort to find a solution not only to behave proactively but also to reduce the human effort. Therefore, we further want to validate whether the users are experiencing the *reduction in effort* or not. Also we want to know that in the case of such human-adapted proactive behaviors, whether the users find the robot to be *'aware'* and *'supportive'* to their capabilities or not.

There were a total of *30* users divided into three groups of *10* users, two groups for the *give* task and one group for the *make accessible* task. Each user group was a mix of different users based on their exposure to the real robots: *no exposure*, *little exposure*, and *rich exposure*. This was to compensate any bias from the experienced and non-experienced users of robots in general. At the beginning of the experiment, each user was informed that the robot will interact but not about the behaviors and the task. Further, no particular instruction was given to the users about 'how' they should behave.

9.5.2.1 For "give" task by the user

We setup different scenarios having different relative positions of the robot, the human and the objects. Broadly, the scenarios could be divided into two categories:

(i) The human is sitting away from the robot and there is some furniture between them, similar to figure 9.10(a).

Table 9.1: Type of users' confusions for the *give* task

Type of confusion	Where to give	When to give	Overall % of users having at least one confusion
In Non-proactive behavior	55%	50%	85%
In Proactive reach behavior	10%	15%	25%

(ii) The human is sitting relatively closer to the robot with different relative position and there is no furniture between them, similar to figure 9.11(a). The users were randomly selected to sit in one or the other scenario.

There were two user groups for the *give* task: *group I* and *group II* consisting of 10 users in each group. The main difference between the both groups was that they have been exposed to the robots of different appearances: *JIDO* and *PR2*. This was to compensate any bias due to the robot's appearance or kinematic structure while validating our hypotheses.

Each user has been exposed to two different behavior of the robot: *NPB* and *PB*. *NPB (Non-Proactive Behavior)*: The robot just asks to the user "Please give me the << *object_name* >>" and waits in its current state. *PB (Proactive Behavior)*: The robot asks the same but also starts moving its arm along the trajectory obtained through the presented proactive planner. In the *PB* case, it also starts turning its head to look at the object as an attempt to incorporate goal-object-directed gaze movement (head movement in our case) as discussed earlier in this chapter.

During the entire experiment, the decision whether *PB* or *NPB* should be exhibited first to a particular user was random. After being demonstrated to both behaviors, each user was requested to fill a questionnaire with first behavior referred as *B1* and the second behavior as *B2*. Note that for some of the users *B1* was *NPB* and for some it was *PB*.

Below we will first analyze the common part of the questionnaire of *group I* and *group II*, to show that independent of the appearance of the robots, the proactive reach behavior is preferable over the non-proactive behavior. Then we will present the analyses of the part of the questionnaire, which is exclusive to *group I* and explore the nature of the confusion and the effect on the effort. (We excluded these questions for *group II* users to maintain the compactness of the questionnaire, as they were required to answer about a few additional questions).

Table 9.1 shows that in the case of proactive reach out behavior of the robot, the total number of the users having at least one type of confusion has been significantly reduced. This supports the hypothesis that the proactive reaching out to take something reduces the confusion of the user.

Note that the sum total (%) of the data of these tables and of the tables following may not be *100* as the users were allowed to mark multiple options or none.

Figure 9.16: Task of giving an object to the robot. (a) In the absence of any proactive behavior the user is holding the object and waiting for the robot to take. (b) With proactive reach behavior from the robot, the user is also putting some effort to give the object to the robot.

Table 9.2: Users' responses about the confusion on 'how' to perform the *give* task in the **NPB** of the robot

Confusions in NPB were: should the user...	...go and give it to the robot?	...stand up and give it to the robot?	...put it somewhere for the robot to take?	...hold it somewhere and wait for the robot to move and take?	...wait for the robot to show some activity?
When first **NPB** has been shown	28%	42%	42%	42%	42%
When first **PB** has been shown	33%	0%	33%	0%	66%

Table 9.2 shows the users' confusions, reported by *group I* users, about how to perform the task. It shows the data for two different cases: (i) *NPB-PB:* When the non-proactive behavior (*NPB*) has been shown first followed by the proactive behavior (*PB*). (ii) *PB-NPB:* When *PB* has been exhibited first followed by the *NPB*. The percentage (%) is calculated based on the total number of the users belonging to a particular case (i) or (ii). Note that for the case (ii) in which *PB* has been demonstrated first, users have been found to be biased towards expecting similar behavior for the next demonstration, which was going to be *NPB*. Last column of table 9.2 reflects this as more users are expecting the robot to show some activity when *PB* has been exhibited first. In such cases user responses were, *"I thought that the experiment has failed, since the robot didn't move"*, *"I was waiting for the robot to take it from me."*

Table 9.3 shows *group I* users' responses about the change in their perceived efforts. It shows that *71%* users of the *NPB-PB* case explicitly mentioned that the second behavior, i.e. the *PB* has reduced their effort to give the object compared to the

Table 9.3: Users' experience on change in effort for the *give* task

Change in the human's effort in the behavior shown second, B2, compared to the behavior shown first, B1.	Reducing human's effort	Demanding more effort
When **B1** was **NPB** and **B2** was **PB**	71%	0%
When **B1** was **PB** and **B2** was **NPB**	0%	66%
% users reported **PB** reduces human effort compared to **NPB** = 70%		

Table 9.4: Users' experience about awareness, supportiveness and the guiding nature of PB for the *give* task

Compared to the NPB, the % users explicitly indicated that in the **PB** the robot was...	
...more aware about the user's abilities and possible confusions	70%
...more supportive and helping to the task and to the user	85%
Total % of users explicitly reported that proactive reach guided them about where to perform the task	80%

first behavior, i.e. the *NPB*. Further, *66%* users of the *PB-NPB* case explicitly mentioned that the second behavior, i.e. the *NPB* has demanded more effort to give the object compared to the first behavior, i.e. the *PB*. On combining both, a majority of the users, *70%* of the total users of *group I*, reported that the proactive reach out behavior of the robot reduces their efforts compared to non-proactive behavior. Hence, it supports our hypothesis that the *human adapted reach out* will also make the users to feel a reduction in their efforts in the joint tasks. It also validates that the presented framework is indeed able to find a solution while maintaining least feasible effort of the human partner.

Table 9.4 (combines *group I* and *group II* responses) shows that a majority of the users reported the robots to be more 'aware' and 'supportive' to them and to the task in the cases it behaved proactively. Table 9.4 also shows that *80%* of users of *group I* explicitly mentioned that proactive reach behavior guides them about where to perform the task. Hence, validating the perspective taking capability of the robot.

A Few Interesting Observations: Apart from the direct responses from the users, we observed following interesting situations:

(i) Without any proactive reaching behavior the user in figure 9.16(a) is holding the object and waiting for the robot to take. Whereas, as shown in figure 9.16(b), in the presence of proactive reaching behavior of the robot, the human is also putting some effort to lean and give the object to the robot. This suggests to be validating the studies of human-behavioral psychology that goal anticipation during action observation is influenced by synonymous action capabilities [Gredeback 2010].

(ii) For the cases where non-proactive behaviors have been shown first, few users have been found to spend some time 'searching' for the object to give, if the table top environment was somewhat cluttered, even if the robot has asked to give the object by name. This suggests that such goal-directed proactive reach behaviors also help in fetching the human's attention to the object of interest. Which further suggests that such goal-directed proactive reach behaviors (should) directly/indirectly incorporate the component of pointing, which in our experiments have been partially achieved by assigning higher weights to the places close to the object. This seems to be supporting the findings in [Louwerse 2005] and [Clark 2003] that directing-to gesture help drawing user focus of attention towards the object.

Further user studies are required to properly validate and establish these observations as facts.

9.5.2.2 For "make accessible" task by the user

The robot requests the human partner to make an object accessible, so that the robot could take it sometime later. As explained earlier, the robot is able to find a feasible place where the human can put the object with least possible effort and the robot could take it from there. We have deliberately built the scenario in which the least possible effort for making an object accessible to the robot by the human is to put it on the top of a white box.

There were 10 users forming the *group III*. For this task, instead of exposing the two behaviors randomly to a user, we decided to first show the non-proactive behavior *(NPB)* followed by the proactive behavior *(PB)*. This is because if the user will be first exposed to the *PB*, he/she might be biased towards putting the object at the same place in the case of *NPB* also, as the scenario would be the same.

For the non-proactive behavior, *(NPB)*, the robot looks at the human and utters the scripted sentence:

"Hey, I need your help. Can you please make the $<<$ object_name $>>$ accessible to me."

For the proactive behavior, *(PB)*, the robot says:

"Hey can you make the $<<$ object_name $>>$ accessible to me, you can put it on the $<<$ support_name $>>$".

As an attempt to incorporate the goal-directed gaze movement (head movement in this case) of the robot, it looks at the object while uttering the first part and then it starts turning its head towards the place where it would suggest the human to put the object.

Table 9.5: Nature of the users' confusions for the *make-accessible* task

The user was confused about:	Meaning of the task: How to perform (give in hand, put somewhere)	Where to make accessible	Overall % of users having at least one confusion
In non-proactive behavior	30%	60%	80%
In proactive suggesting behavior	10%	30%	30%

Table 9.6: Users' suspicions about the robot's capabilities for the *make accessible* task

The users were suspicious about the robot's capabilities ...	From where the robot will be able to take	At which places the robot will be able to see	Overall % of users having at least one suspicion
In non-proactive behavior	70%	20%	70%
In proactive suggesting behavior	20%	10%	30%

As shown in table 9.5, about *80%* of users have reported confusion about *how* and *where* to make the object accessible in the case of *NPB*. This has been significantly reduced to *30%* in the case of *PB*.

Table 9.6 shows the percentage of users who were suspicious about the robot's ability about from 'where' it could take or see the object. Note that in the case of proactive behavior, as the robot was explicitly suggesting, "$\cdots$ *you could put it on the white box*", hence restricting the search space for the user to perform the task, such suspicions have reduced significantly.

These findings seem to be also supporting the result of [Louwerse 2005], which shows that the use of location description increases accuracy in finding the target. In the current experiment, the location description was not for localizing the object, but instead for the place to put the object; hence guiding the user for efficient task realization.

As shown in table 9.7, a majority of the users found the proactive suggestion by the robot more compelling. Table 9.8 shows that *60%* of the users found that the *human adapted* proactive behavior reduced their efforts.

A few Interesting Observations:

Table 9.7: Users' responses about the robot's awareness through the *PB* for the *make accessible* task

% of users explicitly mention that in PB compared to NPB	
The robot seems to be more aware about user's capabilities and possible confusion	70%
The robot has better communicated its capabilities	80%

Table 9.8: Users' responses about their relative efforts in the make accessible task

Users' efforts in PB compared to NPB			
Human effort reducing	Mutual effort Balancing	Demanding more human effort	Can't say
60%	20%	10%	10%

Figure 9.17: Task of making an object (marked as red arrow) accessible to the robot. In the absence of proactive behavior this user has taken away the white box as an attempt to clear the obstruction for the robot, so that the robot would be able to take the object by itself.

Figure 9.18: Task of making an object accessible to the robot. In the absence of proactive behavior the user is holding the object and waiting for the robot to take.

(i) One of the interesting observations was related to the human's interpretation about how to perform the task of making an object accessible. As shown in figure 9.17(a), in the case of non-proactive behavior, the user took the white box away

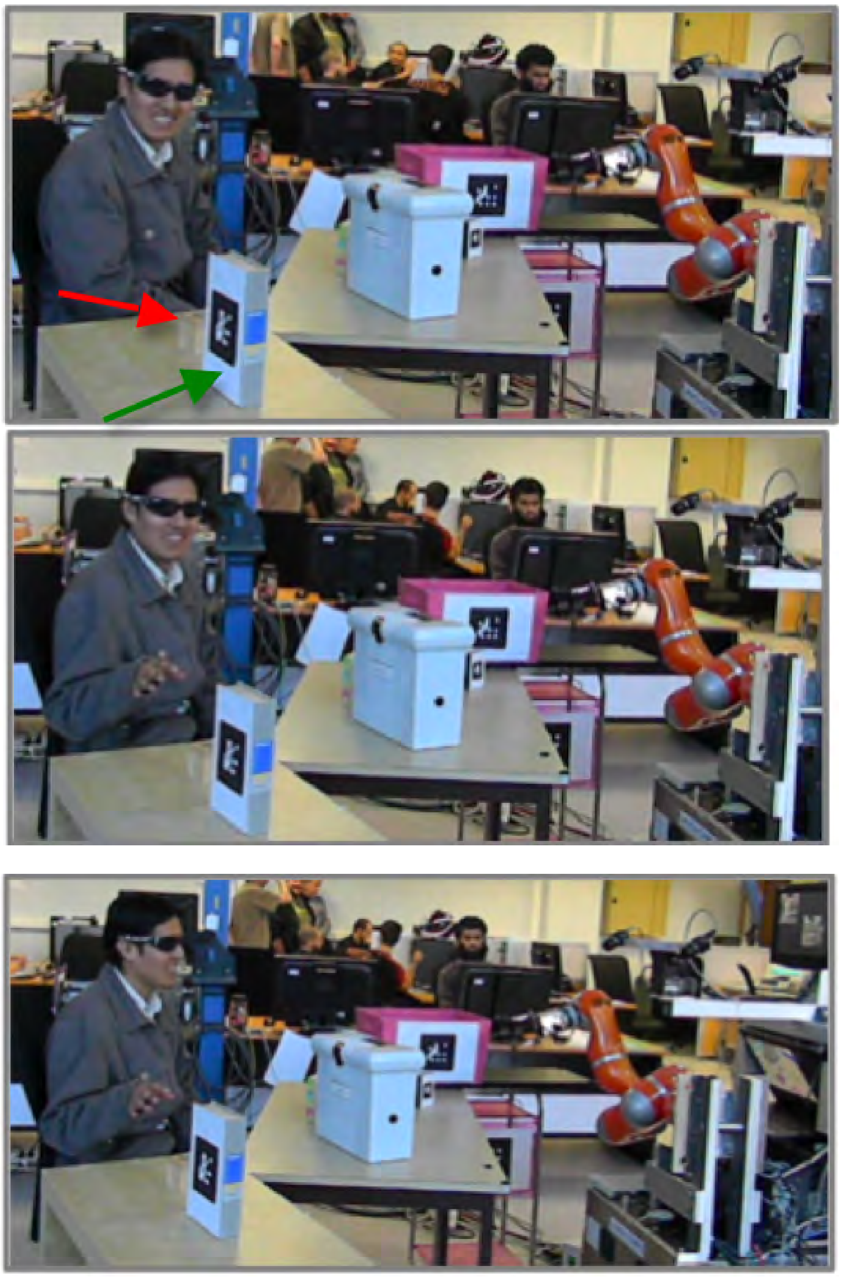


Figure 9.19: Task of making an object (marked as red arrow) accessible to the robot. In the absence of further feedback from the robot, the human is confused about which object to make accessible, as he failed to ground the object referred by the robot.

for making the object (marked by red arrow) accessible to the robot. Although he overestimated the reach of the robot but his interesting explanation was that he thought if he would move the box away, which was obstruction from the robot's perspective to reach the object, robot would be free to take the object in the way it wants.

Figure 9.18 shows another scenario in which the user is holding the object close to the robot for the robot to take it. Such observations suggest the need of proactive suggestions about *'how'* to perform the task, whenever necessary.

(ii) As shown in figure 9.19, this user is confused about which object the robot has requested to make accessible. Such confusion has been reported by at least 3 users, because of various factors, such as background noise, difficulty to ground the object by name, novice to the computer-synthesized sound, etc. Moreover, such confusion has been reported in both the cases: non-proactive and proactive. In this particular case, the user is trying to reach towards the objects on his left side based on predicting the robot's attention, figure 9.19(b), but looking at robot to get some additional information, figure 9.19(c).

This suggests that the element of pointing should be also included in robot's behaviors whenever is required. Another component suggested by figure 9.19(c) is to have a feedback mechanism from the robot also. It suggests that not only the robot requires a feedback from the human but the robot should also provide feedback to the human in natural human-robot interaction scenario. Works on such complementary issues of grounding references through interaction, such as ours [Ros 2010], [Lemaignan 2012], could be adapted for this purpose of proactive behavior with feedback.

As mentioned earlier, this is preliminary user study, which seems to be in agreement with our hypotheses and the existing works in human behavioral psychology and encourages for further analyses with bigger group of people to establish such observations as facts from Human-Robot Interaction point of view.

9.5.2.3 Overall inter-task observations

In this section, we will combine the results of both the tasks to draw some global conclusions. Table 9.9 (by combining table 9.1 and table 9.5) shows an overall *66%* reduction in confusion in the case of proactive behavior. Table 9.10 shows that a majority of the users, *65%*, experienced that the *human adapted* proactive behavior reduced their efforts. Table 9.11 shows that a majority of the users, *85%*, reported that the proactive behavior has better communicated the robot's capabilities and was more supportive to the task and to them.

9.6 Discussion on some complementary aspects and measure of proactivity

In Human-Human interaction, the notion of proactive eye movement have been identified [Flanagan 2003], and further in [Sciutti 2012] such proactive gaze have been suggested as an important aspect to be incorporated in developing methods

Table 9.9: Overall reduction in the users' confusion because of the robot's proactive behavior

For give task by the human	70%
For make accessible task by the human	62%
Overall by combining both the tasks	66%

Table 9.10: Overall reduction in users' effort because of the robot's proactive behavior

For give task by the human	70%
For make accessible task by the human	60%
Overall by combining both the tasks	65%

to measure HRI through motor resonance. However, their notion of proactive gaze corresponds to predicting the goal of the action, and then proactively shifting the gaze directly towards the goal. This notion of proactivity is complementary to the proactive behaviors within the scope of the thesis, in the sense instead of shifting its gaze proactively based on the human's action, the robot proactively finds a solution for the human action and suggests it through its proactive actions. However, such proactive actions might include proactive gaze as a component or might induce the human partner's proactive gaze.

However, we feel the need of further user studies from the perspective of long-term human-robot interaction in the context of high-level tasks. Regarding this, the proactive gaze model as discussed above could be adapted to develop the measure of proactivity in HRI, based on how much the proactive action of the robot induces proactive gaze of the human partner, indicating the predictiveness in the proactive behavior. Developing such measures with other metrics as identified in [Olsen 2003], [Steinfeld 2006] will also help in identifying the necessary enhancements at different levels of planning and execution of such proactive behaviors and in HRI in general.

Table 9.11: Overall responses about supportiveness and communicativeness of the proactive behavior

Total % of users explicitly reported that the robot has better communicated its capabilities and was more supportive to the task and to the user in the proactive behaviors	85%

9.7 Until Now and The Next

In this chapter, we have identified various spaces of action and environmental states, in which reasoning about proactive behavior could be done. Based on *which part* and *how much* of these spaces will be altered by the proactive behavior, we have presented a theoretical basis for synthesizing and regulating the proactivity. Using this we have identified 4 levels of proactivity, based on its effect on the ongoing interaction, and on already planned actions and desired state. Further, we have instantiated a couple of such proactive behaviors and shown through user studies that the human-adapted proactive behaviors reduce the effort and confusion of the human partner as well as enhances the user's experience with the robot. The users find the robot to be more aware and supportive in the cases the robot behaves proactively for different types of tasks.

Until now, we have assumed that the desired effect of a task is already known to the planner, whether it is to plan for basic HRI tasks, to plan for cooperatively sharing the task or to plan to behave proactively. However, it would be nice if the robot would be able to understand the desired effects of a task autonomously through demonstrations. That will greatly support the existence of the robot in our day-to-day life, as the robot will be able to understand various tasks and even perform them differently in different situations. In the next chapter, we will address this issue of emulation aspect of social learning for a subset of basic HRI tasks and present a framework to understand the task semantics at appropriate level of abstraction.

Chapter 10

Task Understanding from Demonstration

Contents

10.1 Introduction 248

10.2 Predicates as Hierarchical Knowledge Building 249

10.2.1 Quantitative facts: agent's least efforts 249

10.2.2 Comparative fact: relative effort class 250

10.2.3 Qualitative facts: nature of relative effort class 251

10.2.4 Visibility score based hierarchy of facts 251

10.2.5 Symbolic postures of agent and relative class 252

10.2.6 Symbolic status of objects 252

10.2.7 Object status relative class and nature 253

10.2.8 Human's hand status 253

10.2.9 Hand status relative class and nature 254

10.2.10 Object motion status and relative motion status class 254

10.3 Explanation based Task Understanding 255

10.3.1 General Target Goal Concept To Learn 256

10.3.2 Provided Domain Theory 256

10.3.3 m-estimate based refinement 257

10.3.4 Consistency Factor 258

10.4 Experimental Results and Analysis 260

10.4.1 Show an object 262

10.4.2 Hide an object 265

10.4.3 Make an object accessible 267

10.4.4 Give an Object 268

10.4.5 Put-away an object 269

10.4.6 Hide-away an object 270

10.5 Performance Analysis 271

10.5.1 Processing Time 271

10.5.2 Analyzing Intuitive and Learnt Understanding 272

10.6 Practical Limitations 274

10.7 Potential Applications and Benefits 274

10.7.1 Reproducing Learnt Task 274

10.7.2 Generalization to novel scenario 275
10.7.3 Greater flexibility to high-level task planners 276
10.7.4 Transfer of understanding among heterogeneous agents 277
10.7.5 Understanding by observing heterogeneous agents 277
10.7.6 Generalization for multiple target-agents 277
10.7.7 Facilitate task/action recognition and proactive behavior . . 277
10.7.8 Enriching Human-Robot interaction 278
10.7.9 Understanding other types of tasks 278
10.8 Until Now and The Next 278

10.1 Introduction

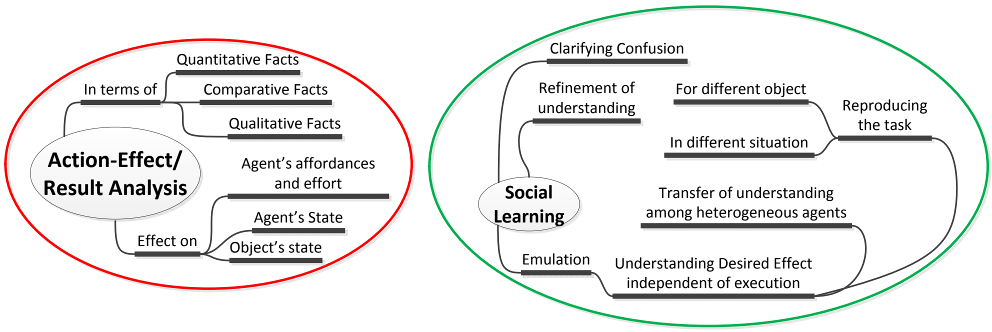


Figure 10.1: Contribution of the chapter in terms of analyzing effect of an action based on effect-based hierarchical knowledge building and understanding tasks' semantics independent to how it has been demonstrated, which could facilitate planning and executing a task differently in different situations.

Until now, we assumed that the semantics of a task is known to the robot whether it has to perform a task for the human or to behave in a proactive way. Now, we will present a framework, which learns the tasks' semantics in terms of the effects to be achieved from the human demonstrations. This is an important aspect of autonomous robot with the capabilities of lifelong learning from day-to-day demonstrations and reproducing the task in different situations. As mentioned in section 1.1.1, from the perspective of social learning, which in loose sense is "*A observes B and then 'acts' like B*", *Emulation*, is regarded as a powerful social learning skill. This is related to understanding the effect or changes of the task, which in fact facilitates to perform a task in a different way. For successful *Emulation* (i.e. bringing the same result, which might be with different means/actions than the demonstrated one), understanding the "effect" of the task is an important aspect. We have developed a framework, which enables the robot to autonomously understand different tasks at appropriate levels of abstraction, by comparing environmental state before and after the task. This facilitates task understanding in a 'meaningful' term as well as provides flexibility of planning alternatively for a task depending upon the situation. Figure 10.1 summarizes the contribution of the chapter as well as the benefits.

10.2 Predicates as Hierarchical Knowledge Building

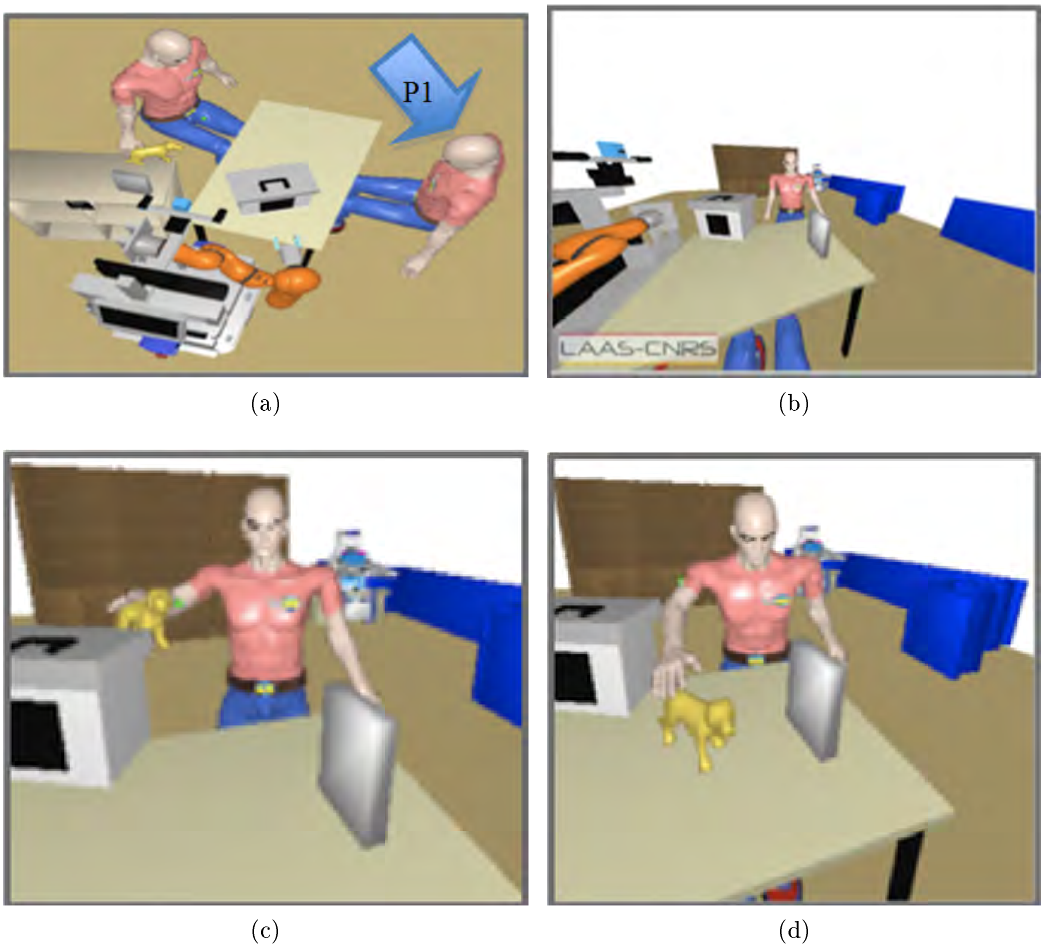
As demonstrated through the example in section 2.7 of chapter 2, same task of making an object accessible could be performed in different ways based on the situation, preferences, posture, etc. So, it is important to be able to reason about the capabilities and constraints of the agents involved at a level of proper abstraction, to capture the 'meaning' of the task. Hence, below we will present the first part of the contribution of this chapter: hierarchical knowledge building, by enabling the robot to infer the facts at a level of abstractions, which are not directly observable, such as comparative facts like *easier*, *difficult*, *reduced*, etc.; qualitative facts like *supportive*, *non-supportive*, etc. The robot's knowledge has been further enriched with hierarchy of facts related to the object's state.

10.2.1 Quantitative facts: agent's least efforts

As already mentioned in chapter 4, the robot infers abilities of the agent: Ability to Reach (Re) and See (Se). Further, the Ability to Grasp (Gr) is perceived. For this, if there exists at least one collision free grasp for the reachable object, the object is assumed to be graspable for that agent.

Visibility Score (ViS) for an object from an agent's perspective presented in section 4.3.1.2 of chapter 4 is also used as a predicate for task understanding. Figure 10.2 shows different visibility scores for toy horse from human *P1*'s perspective from his current state.

As explained in section 4.4 of Mightability Analysis chapter (chapter 4), we have a human-aware measure of *effort types* as summarized in figure 4.5 of that chapter. Further, as explained in section 4.7 of the chapter, the robot is able to find the least effort associated with an object for an ability Ab (reach, see, Grasp) for an object Obj from an agent's perspective. We denote the type of the least effort as T_E.



(a) (b) (c) (d)

Figure 10.2: (a) The robot is observing a human-human interaction. (b) Person *P1*'s current state visual perspective. Visibility scores of the toy dog for person *P1* are *0.0* for the currently hidden toy dog as in (b), *0.001* when the toy dog is partially occluded and relatively far as in (c) and *0.003* when it is non-occluded and relatively closer as in (d).

10.2.2 Comparative fact: relative effort class

The robot should be able to relatively analyze two efforts. For this, we define operator C_{et}, which compares two effort levels and assigns a class C_{RE}, as:

$$C_{RE}\left(T_E^1, T_E^2\right) = \begin{cases} Remains\_Same & if\ T_E^1 = T_E^2 \\ Becomes\_Easier & if\ T_E^1 < T_E^2 \\ Becomes\_Difficult & if\ T_E^1 > T_E^2 \end{cases} \quad (10.1)$$

Note that $C_{RE}\left(T_E^1, T_E^2\right) \neq C_{RE}\left(T_E^2, T_E^1\right)$.

Although not used in current implementation of learning, we further have a measure of amount of effort for a particular effort level in terms of how much the agent

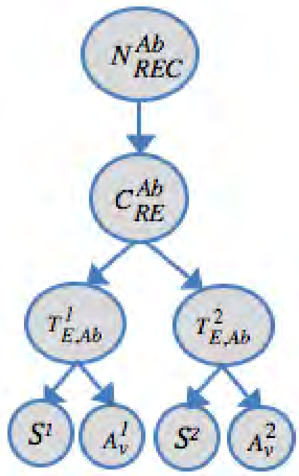


Figure 10.3: Effort based hierarchy of facts.

has to turn/lean, etc, as explained in chapter 4. Hence, the robot could further compare two efforts of same effort level. This could be further enhanced based on the studies of musculoskeletal kinematics and dynamics models, [Khatib 2009], [Sapio 2006]. Whether the input is as effort level or as amount of effort the robot finds the comparative facts of expression 10.1.

10.2.3 Qualitative facts: nature of relative effort class

We have further enhanced the robot's knowledge-base with another layer of abstraction by qualifying the Relative Effort Classes (C_{RE}) as *supportive* and *not supportive*. Based on the intuitive reasoning that if an object becomes difficult to be reached by a person, the intention/nature behind it is not to support the person's ability to reach the object. Hence, we qualify the intention behind the change in effort level by assigning a nature, N^{Ab}_{REC} as:

$$N^{Ab}_{REC}\left(C^{Ab}_{RE}\right) = \begin{cases} S : Supportive & if\ C^{Ab}_{RE} \in \{Remains\_Same, Becomes\_Easier\} \\ NS : Not\_Supportive & if\ C^{Ab}_{RE} \in \{Becomes\_Difficult\} \end{cases} \tag{10.2}$$

where Ab is a particular ability of the agent. Figure 10.3 shows the hierarchy of facts based on efforts.

10.2.4 Visibility score based hierarchy of facts

The robot performs hierarchical analysis by comparing two Visibility Scores, ViS^1 and ViS^2 to have relative visibility score classes as:

$$C_{RViS}\left(Vis^1, Vis^2\right) = \begin{cases} Almost\_Same & if\ (Vis^1 - Vis^2 \approx 0) \\ Increased & if\ Vis^1 << Vis^2 \\ Decreased & if\ Vis^1 >> Vis^2 \end{cases} \tag{10.3}$$

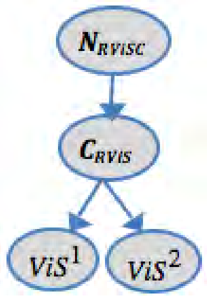


Figure 10.4: Visibility scores based hierarchy of facts.

Similarly, we qualify the nature N_{RViSC} to the relative class based on whether the quantitative visibility of the object is supported or not:

$$N_{RViSC}\left(C_{RViS}\right) = \begin{cases} S : Supportive & if\ C_{RViS} \in \{Almost\_Same, Increased\} \\ NS : Not\_Supportive & if\ C_{RViS} \in \{Decreased\} \end{cases} \tag{10.4}$$

Figure 10.4 shows the hierarchy of facts by analyzing the visibility scores.

10.2.5 Symbolic postures of agent and relative class

As mentioned in section 5.4.1 in situation assessment part of chapter 5, the robot tracks the human's body parts and distinguishes between standing and sitting postures of the human online. We use agent's posture as predicate *Post*:

$$Post \ \in \{Standing, Sitting\} \tag{10.5}$$

Further, by comparing two postures a class is assigned as:

$$C_{RPost}\left(Post^1, Post^2\right) = \begin{cases} M : Maintained \ \ if\ Post^1 = Post^2 \\ C : Changed \ \ otherwise \end{cases} \tag{10.6}$$

10.2.6 Symbolic status of objects

Based on relative positions of an object with human's hand and with other objects, as explained in situation assessment part of chapter 5, a symbolic status to the object is assigned. The object status predicate is:

$$O_s \in \{Inside\_Container, On\_Support, In\_Hand, In\_Air\} \tag{10.7}$$

Ambiguity in object status is resolved based on simple case based rules. Such as if the object is on a support and hand is also in contact with the object, it returns object *On_ Support*.

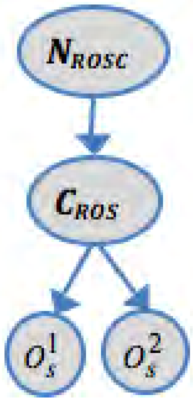


Figure 10.5: Object state based hierarchy of facts.

10.2.7 Object status relative class and nature

By comparing two ordered instances of O_s, a class is assigned as:

$$C_{ROS}\left(O_s^1 \to O_s^2\right) = \begin{cases} M : Maintaining\left(O_s^1\right) & if\ O_S^1 = O_S^2 \\ G : Gaining\left(O_S^2\right) \wedge L : Losing\left(O_S^1\right) & otherwise \end{cases} \tag{10.8}$$

Note the second case results into two simultaneous facts to encode the transition: gaining and losing states by the object. For example, for the *lift object* task, if initially the object was on support and now it is in hand, then the expression 10.8 will result into two facts: *Losing On_Support state* and *Gaining In_Hand state*, to encode the transition.

Further, we qualify the nature of the changes $c = C_{ROS}\left(O_s^1 \to O_s^2\right)$ as *supportive* to the final state if the transition maintains or gains that state, as (see expression 10.8 for abbreviations):

$$N_{ROS}\left(c\right) = \begin{cases} S : Supportive\left(O_S^2\right) & if\ c \in \left\{M\left(O_S^2\right), G\left(O_S^2\right)\right\} \\ NS : Not\_Supportive & if\ c = L\left(O_S^2\right) \end{cases} \tag{10.9}$$

Hence, a hierarchy of facts based on object's states is built as shown in figure 10.5.

10.2.8 Human's hand status

As explained in situation assessment part of chapter 5, a symbolic status to human's hand could be obtained. From the human's perspective we use the human hand status predicate:

$$H_S \in \{Holding\_Object : OH, Free\_of\_object : OF, Resting\_on\_Support : RS\} \tag{10.10}$$

10.2.9 Hand status relative class and nature

The robot further compares two instances of status of the human's hand from the point of view of manipulability of the object. Based on the reasoning that if the object is in either of the hands, then human can directly manipulate it, a comparative class is assigned as follows (*Manip* stands for Manipulability, see expression 10.10 for other abbreviations):

$$C_{RHS}\left(H_S^1 \to H_S^2\right) = \begin{cases} M : Manip\_Maintained & if\ H_S^1 = H_S^2 \wedge H_S^2 = OH \\ G : Manip\_Gained & if\ H_S^1 \neq H_S^2 \wedge H_S^2 = OH \\ L : Manip\_Lost & if\ H_S^1 \neq H_S^2 \wedge H_S^1 = OH \\ V : Manip\_Avoided & if\ H_S^1 \neq OH \wedge H_S^2 \neq OH \end{cases} \tag{10.11}$$

Further, a qualifying nature for relative hand status class $c = C_{RHS}\left(H_S^1 \to H_S^2\right)$ from the agent's perspective is assigned as (see expression 10.11 for abbreviations):

$$N_{RHSC}\left(c\right) = \begin{cases} MD : Manip\_Desired & if\ c\ \in \{M, G\} \\ MND : Manip\_Not\_Desired & if\ c\ \in \{L, V\} \end{cases} \tag{10.12}$$

This again results into hierarchy of facts based on human's hand status. Note that in the current implementation, if the state of either of the hand changes, it is treated as change in manipulability.

10.2.10 Object motion status and relative motion status class

As already mentioned in chapter 3 and illustrated in figure 3.1, the environment observation and inference is continuous in time. Hence, based on the temporal reasoning on the object's position, at any point of time the motion status of the object is knows as:

$$O_{ms} \in \{Moving : Mv, Static : St\} \tag{10.13}$$

Further, by comparing two instances of motion status, a relative status class for the object's motion state transition is assigned as follows (see expression 10.13 for abbreviations):

$$C_{ROMS}\left(O_{ms}^1 \to O_{ms}^2\right) = \begin{cases} motion\_gained & if\ O_{ms}^1 = St \wedge O_{ms}^2 = Mv \\ motion\_lost & if\ O_{ms}^1 = Mv \wedge O_{ms}^2 = St \\ motion\_maintained & if\ O_{ms}^1 = Mv \wedge O_{ms}^2 = O_{ms}^1 \\ motion\_avoided & if\ O_{ms}^1 = St \wedge O_{ms}^2 = O_{ms}^1 \end{cases} \tag{10.14}$$

In this section, we have enriched the robot's knowledgebase with a set of hierarchy of facts related to the human and the object. Next section will describe our generalized task understanding framework based on *explanation-based learning* and *m-estimate based refinement*. The framework takes into account such hierarchies of facts and autonomously learns the tasks' semantics at appropriate level of abstractions.

10.3 Explanation based Task Understanding

Apart from understanding the task independent of how to execute it, another motivation behind current work is to enable the robot to begin learning the task even from a single positive demonstration. So we have adapted the framework of Explanation Based Learning (EBL) (see the survey [Wusteman 1992]), which has been shown to possess the desired characteristics and could be used for concept refinement (i.e. specialization) as well as concept generalization, [Dejong 1986]. For continuity, below we mention the components of a typical *EBL* system (see [Dejong 1986] for detail):

- *Goal Concept*: A definition of the concept to be learnt. Given in terms of high-level properties, which are not directly available in the representation of an example.
- *Training Example*: A lower level representation of the examples.
- *Domain Theory*: A set of inference rules and facts sufficient for providing that a training example meets the high-level definition of the concept.
- *Operationality Criterion*: Defines the form in which the learnt concept definition must be expressed.

Generally domain theory and operationality criterion are devised to restrict the allowable learnt vocabulary and initial hypothesis space, to ensure that the new concept is 'meaningful' to the problem solver (the task planner).

Our approach will be similar to *EBL* in the following manner [Wusteman 1992], [Flann 1989]:

(i) It constructs an explanation tree for each example of a task.

(ii) Compares these trees to find largest sub tree.

(iii) Forms the horn clause using the leaf nodes of the largest sub tree to find the general rule.

Our approach will differ from EBL in the sense, instead of providing a proper domain theory and operationality criterion for the target-concept to precise the hypothesis space; we will provide a general goal concept in terms of the effect of the task. This will initialize the hypothesis space with highest-level abstract knowledge of the robot. This will ensure to learn any task, which could possibly incorporate any of the effect related predicates known to the robot. Then based on the demonstrations, the robot has to autonomously refine/prune the hypothesis space. This will prevent providing separate domain theory for each and every task the robot will encounter in its lifetime, as well as will enable the robot to autonomously extract relevant features for a particular task.

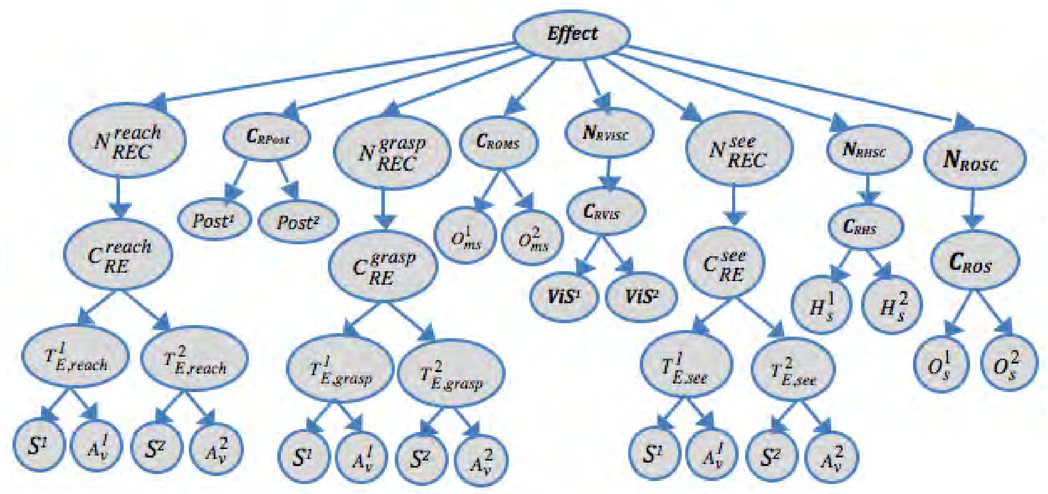


Figure 10.6: Initial generalized hypothesis space for effect-based understanding of tasks' semantics.

10.3.1 General Target Goal Concept To Learn

We provide for any task T, performed by a *performing-agent* P_{ag} for a *target-agent* T_{ag} on a *target-object* T_{obj}, the generalized goal concept to learn as:

$$Task\,(name(T)) \leftarrow \mathit{effect}\,(WI, WF, T_{ag}, T_{obj}) \tag{10.15}$$

As illustrated in figure 3.1 of chapter 3, *WI* and *WF* are snapshots of the continuously inferred facts and continuously observed world states at the time stamps t_i and t_f marking the start and the end of a demonstration.

10.3.2 Provided Domain Theory

The following *domain theory* is provided:

$$\begin{aligned} \mathit{effect}\,(WI, WF, T_{ag}, T_{obj}) \leftarrow N^{reach}_{REC}\,(T_{ag}, T_{obj}) \wedge N^{grasp}_{REC}\,(T_{ag}, T_{obj}) \wedge \\ N^{see}_{REC}\,(T_{ag}, T_{obj}) \wedge N_{RViS}\,(T_{obj}, T_{ag}) \wedge C_{RPost}\,(T_{ag}) \wedge N_{RHSC}\,(T_{ag}) \wedge \\ N_{ROSC}\,(T_{obj}) \wedge C_{ROMS}\,(T_{obj}) \end{aligned} \tag{10.16}$$

The task is learnt in the form of *desired effects* from any *target-agent's* perspective for any *target-object*.

Above expression when mapped into the definitions of inferred facts discussed earlier in this chapter, results into following representation:

$$\begin{aligned} \mathit{effect}\left(WI, WF, T_{ag}, T_{obj}\right) \leftarrow\ & \mathit{Nature\_Effect\_Class\_to\_Reach}\left(T_{ag}, T_{obj}\right) \wedge \\ & \mathit{Nature\_Effect\_Class\_to\_Grasp}\left(T_{ag}, T_{obj}\right) \wedge \\ & \mathit{Nature\_Effect\_Class\_to\_See}\left(T_{ag}, T_{obj}\right) \wedge \\ & \mathit{Nature\_Visibility\_Score}\left(T_{obj}, T_{ag}\right) \wedge \\ & \mathit{Effect\_Relative\_Posture}\left(T_{ag}\right) \wedge \\ & \mathit{Nature\_Effect\_Hand\_Status}\left(T_{ag}\right) \wedge \\ & \mathit{Nature\_Effect\_Object\_Status}\left(T_{obj}\right) \wedge \\ & \mathit{Effect\_Object\_Motion\_Status}\left(T_{obj}\right) \end{aligned} \tag{10.17}$$

And rest of the definitions of the domain theory is presented in expressions of section 10.2.

Above domain theory when unfolded results into a *general initial hypothesis space* as shown in figure 10.6.

The *training examples* are provided as the lowest level, i.e. in 3D world model consisting of the positions and configurations of the objects and the agents. As the robot continuously observes and infers the environment, based on the time stamps of start and end of a demonstration, the robot autonomously instantiates the hierarchies of the facts of the domain theory. Further, to be generalized enough to learn different tasks; we do not strictly provide the form of the learnt concept as *operationality criterion*. It could be composed of any of the nodes of the initial hypothesis space as shown in figure 10.6.

10.3.3 m-estimate based refinement

Each node of initial hypothesis space of figure 10.6 serves as a predicate. For refining the learnt concept based on multiple demonstrations, instead of directly pruning the explanation sub-tree based on getting two different values for a node, we use *m-estimate* based reasoning. *m-estimate* has been shown to be useful for rule evaluation, [Furnkranz 2003] and to avoid premature conclusions [Agostini 2011], in the cases where only a few examples have been demonstrated. This is because the generalized definition of *m-estimate* incorporates the notion of *experience*, as described below.

Let us say a value v for a particular predicate p for a particular task T has been observed in n number of demonstrations, out of total N demonstrations. The possibility of observing the same value v for the next demonstration within the *m-estimate* framework will be given as:

$$Q_p^{v,T}\left(n, N\right) = \frac{n + a}{N + a + b} \tag{10.18}$$

where $a > 0, b > 0, a + b = m$ *and* $a = m \times P_v$. m is domain dependent and could also be used to include noise, [Cestnik 1990]. From the above eq. 10.18, following properties could be deduced:

$$Q_p^{v,T}(0, 0) = P_v > 0 \tag{10.19}$$

$$Q_p^{v,T}(0, N) = \frac{a}{N + a + b} > 0 \tag{10.20}$$

$$Q_p^{v,T}(N, N) = \frac{N + a}{N + a + b} < 1 \tag{10.21}$$

The robot will not assume a close world in the sense if it did not observe v for predicate p, it does not mean that possibility of the existence of v is NULL. In fact Pv is prior probability of v. Also if it always observed the same value, that too will not be accepted as universal rule that p will always have the value v for the task T. Hence, it takes into account the possibility of unseen demonstrations. These properties allow lifelong refinement of the learnt concept.

$$Q_p^{v,T}(N + 1, N + 1) > Q_p^{v,T}(N, N) \tag{10.22}$$

Above property ensures that even if the value v has been observed for all the examples, the possibility to observe same value will be more if more number of examples have been demonstrated, thus incorporating the notion of *experience*.

$$Q_p^{v,T}(0, N) < Q_p^{v,T}(0, N + 1) \tag{10.23}$$

This property ensures that even if the value v has never been observed, the possibility that v will not be observed in the future will be less if less number of examples have been demonstrated, thus again incorporating the notion of experience.

One acceptable instantiation of *m-estimate* is using *Laplace's law of succession*. This states that if in the sample of N trials, there were n successes, the probability of the next trial being successful is *(n+1)/(N+2)*, assuming that the initial distribution of success and failure is uniform. With the similar initial assumption, we also use *a=1* and *a+b=2* for *m-estimate* of eq. 10.18.

10.3.4 Consistency Factor

As the robot is required to autonomously find out whether a predicate p is relevant or not, it analyzes the consistency in the observed value of the predicate. If the values are not always the same, it means the predicate might not be relevant for that task and the values are just the side effects, not the desired effect. We further assume that v_h is the value for p having the highest *m-estimate* obtained from eq. 10.18. If this value is consistent over demonstrations, then the predicate p is relevant and its desired value will be v_h. Let, for a particular predicate p, over N demonstrations,

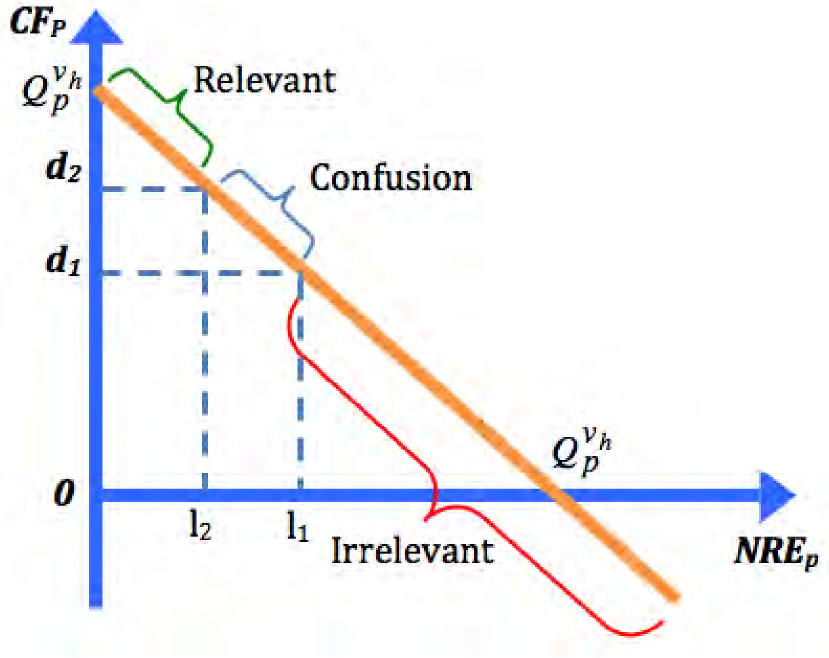


Figure 10.7: Deciding relevance and irrelevance of a predicate, as well as potential confusion.

N_p different values $\{v_1, v_2, v_3, ...v_{Np}\}$ have been observed. We define a *consistency factor (CF)* of p for task T to decide about the relevance of p as:

$$CF_p^T = \overbrace{Q_p^{v_h,T}}^{\text{relevance evidence}} - \underbrace{\sum_{i=1 \wedge i \neq h}^{Np} Q_p^{v_i,T}}_{\text{non-relevance evidence}} \tag{10.24}$$

The first part on the right side of the equation shows the evidence of p being relevant for the task. Higher this value, more will be the possibility that the most observed single value, v_h, for p is the part of the desired effect for task T. The second part gives the possibility of obtaining any of the observed value other than v_h. This in fact represents *non-relevant evidence* of p, NRE_p, because, higher this value, lower the possibility of p having a consistent value. Hence, based on the value of the consistency factor after any demonstration, we define following 3 situations for a particular predicate p for a particular task T (see figure 10.7):

(i) ***Contradiction**, irrelevant predicate p*: A predicate p will be assumed to be non-relevant based on contradiction in its value, (a) if $CF < 0$; non-relevant evidences are collectively higher than the relevant evidence, or (b) If $0 <= CF <= d_1$; non-relevant evidences are significant to contradicting the possibility of v_h being the expected consistent value of p.

(ii) ***Consistency***, *relevant predicate* p: if $CF > d_2$; as the non-relevant evidences are significantly lower and could be ignored.

(iii) ***Confusion***, *confusing predicate*: if $d_1 < CF < d_2$; as the non-relevant evidences are not sufficient to contradict the current understanding but also not small enough to be ignored directly. In this case, the robot has to ask the human partner for clarification about the significance of the predicate p and its desired value by framing a sentence including the values causing the confusions.

As the demonstrations are assumed to be positive, which means we will not try to teach a child with wrong examples, little evidence of a predicate assuming different values should be sufficient to prune that node from the explanation tree, resulting into almost coinciding d_1, d_2 and $Q_p^{v_h}$. However, we prefer to maintain the separate boundaries to allow to tune d_1 and d_2 based on various practical factors such as the reliability of the demonstration, the accuracy of the inferred fact, noise at different levels of the system, nature, sensitivity and criticality of the predicate, preferences on inconsistency tolerances, etc. We set d_2 based on the *10%* tolerance of the inconsistency in relevant predicate, hence $l_2 = 0.1 \times Q_p^{v_h}$ and $d_2 = Q_p^{v_h} - 0.1 \times Q_p^{v_h}$. We set d_1 by giving autonomy to the robot to decide a predicate to be irrelevant if there exists non-relevant evidence as low as *30%* of the relevant evidence, i.e. $l_1 = 0.3 \times Q_p^{v_h}$ which results into $d_1 = Q_p^{v_h} - 0.3 \times Q_p^{v_h}$. We set d_1. Hence, only in the case where the non-relevant evidence is between *10%* and *30%* of the relevant evidence, the robot will ask the human for the clarifications.

One example situation in which case (iii) may arise is, if the robot had consistently observed a particular value *v1* for a predicate p in many past demonstrations but recently it started observing another value *v2* in successive demonstrations. Initially the robot will ignore *v2* for next few demonstrations but as *v2*'s m-estimate will be becoming significant compared to its experience based expectation of obtaining *v1*, it should ask the human partner for clarification.

Here it is important to note that the robot keeps track of *m-estimate* of all the observed values for all the predicates to maintain the notion of inter-value experiences, even if currently a particular value has been found irrelevant. This facilitates the robot to incorporate experience and allow modifying its understandings lifelong.

10.4 Experimental Results and Analysis

We have tested our system on two different robots: *JIDO* a home-built mobile manipulator equipped with a LWR Kuka arm and *PR2* robot from Willow Garage. As shown in figures 10.8(a) and (b), Jido and PR2 robots are observing the environment. Figures 10.8(c) and (d) show the 3D world representation of the environment built and updated online by the robots. The robots use Move3D, [Simeon 2001], an integrated planning and visualization platform. The robots, through their various sensors, maintain and update the 3D world state in real time. For object identifi-

Figure 10.8: Mobile robot JIDO, (a), and PR2, (b), are observing Human-Human interaction scenario. (c) and (d) 3D representation of the world built and updated online by the robots.

cation and localization, they use tags based stereo-vision system. For localizing the human, they use data from Kinect (Microsoft) sensor mounted on it. The human's gaze is simplified to his/her head orientation, estimated through markers tracked by a motion capture system in real time.

After every demonstration, the *task name*, time stamps for starting and finishing of the task, information about the *performing-agent*, *target-agent* and *target-object* are provided to the robot.

In the current approach, the name of the *target-object* is explicitly provided to the robot. However, works on autonomous learning on task-relevant objects such as [Lee 2009] could be adapted for this purpose.

In all the demonstrations, the explanation tree has been constructed by inferring the facts from the *target-agent*'s perspective. This is to find the desired effect for the person for whom the task has to be performed. However, the similar tree could be constructed from the perspective of the performing-agent, to find how the agent prefers to perform the task.

As explained in section 10.3, the robot constructs an explanation tree for each new demonstration of the task, by instantiating the hypothesis tree of figure 10.6. For instantiating the leaf nodes, the predicates with superscript *1* is provided with the data from *WI*, initial world state, whereas for superscript *2* the data is provided from WF, final world state.

Greater the diversity among the demonstrations for the same task, faster the non-

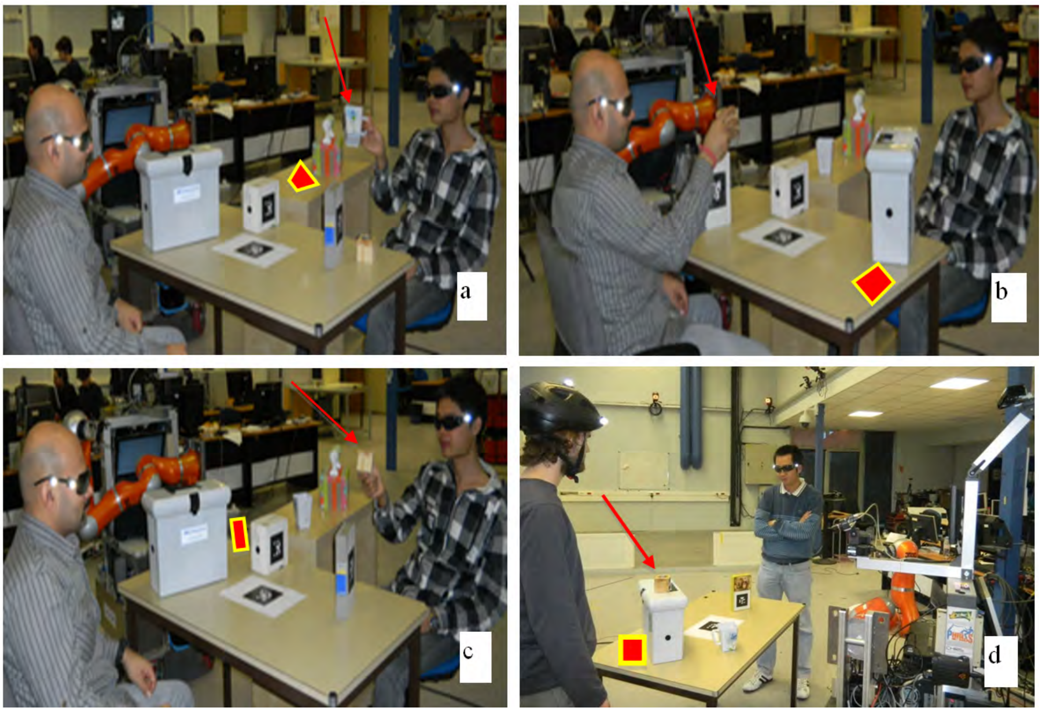


Figure 10.9: Human-Human performing the task to 'show' an object. Initial positions of the *target-object* are shown by red quadrilaterals. (a) Right human is showing the cup by holding it. (b) Left human is showing the wooden cube by holding it. (c) Right human is showing the wooden cube by holding it. (d) Left human is showing the wooden cube by making it visible by putting it on the top of the white box.

relevant predicates will be pruned out from the task's understanding. Therefore, to achieve diversity we have changed the initial scenarios by changing the relative arrangements of the performing- and target- agents, the initial position of the objects, etc.

10.4.1 Show an object

The first task demonstrated to the robot was to *show* an object. Figures 10.9(a)-(d) show final scenarios of four different demonstrations of the task. The red quadrilaterals show initial positions of the *target-object* (which is the cup in figure 10.9(a) and the wooden cube in figures 10.9(b) and (c)), the red arrows mark the final position of the *target-object* at the end of the task. In situations of figures 10.9(a) and (c), the *target-agent* was the person on the left whereas for figures 10.9(b), he was the *performing-agent*. The largest consistent sub-tree after first two demonstrations, figures 10.9(a) and (b), has been shown in figure 10.10. Below each node of the tree, the corresponding inferred values of the predicates have been shown in parenthesis {}. The learnt target concept for the task is obtained in terms of horn clause from

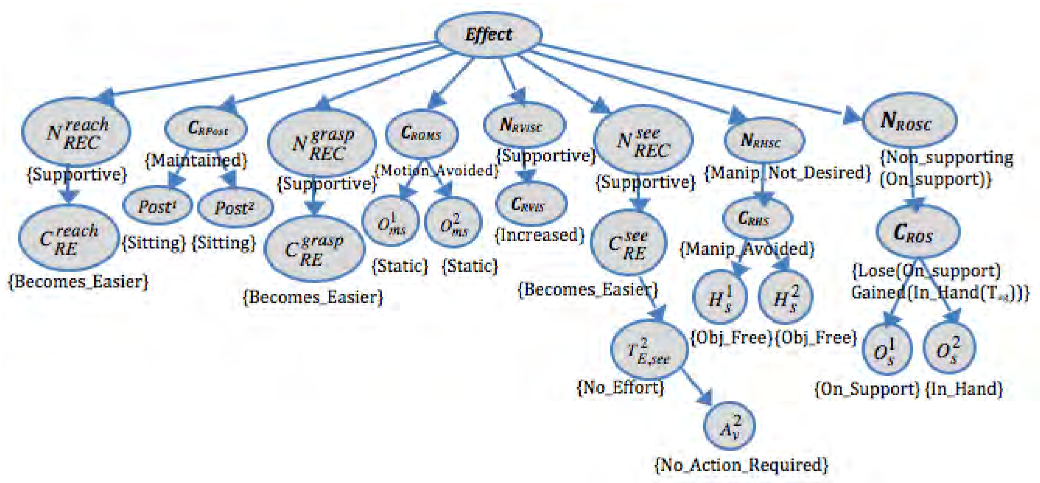


Figure 10.10: Explanation tree for the show task after 2 demonstrations (a) and (b) of figure 10.9.

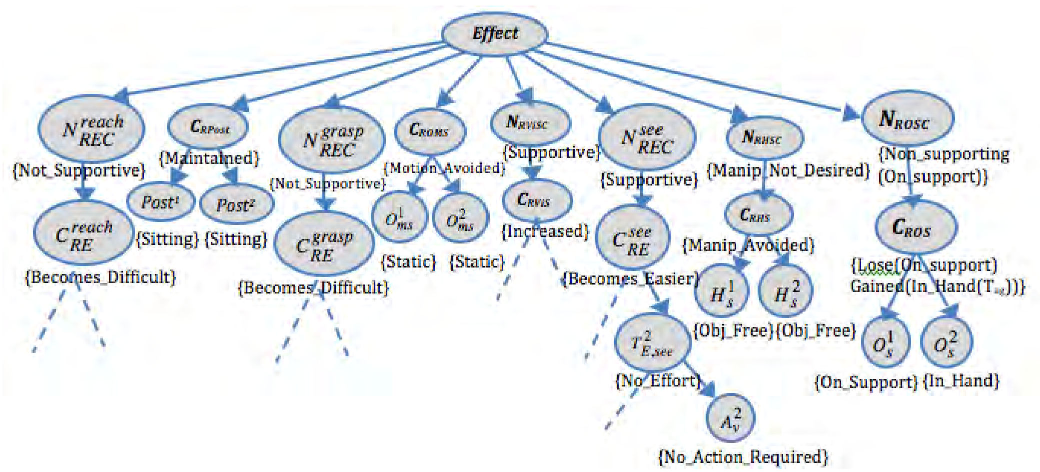


Figure 10.11: Partial instantiation of the hypothesis space for explaining the show task of the demonstration (c) of figure 10.9

the leaves nodes of this sub-tree.

Figure 10.11 shows partial instantiation of the hypothesis space for the individual demonstration (c) of figure 10.9. And figure 10.12 shows the refined explanation, based on the largest common, *m-estimate* based consistent, sub-tree for all the three demonstrations. In the fourth demonstration for the same task, a different pair of *performing-* and *target- agents* demonstrated the task in standing postures. The *performing-agent* has put the *target-object*, the wooden cube, on another object, white box, to make it visible, as shown in figure 10.9(d). Figure 10.13 shows the refined explanation tree. The refined understanding after these four demonstrations,

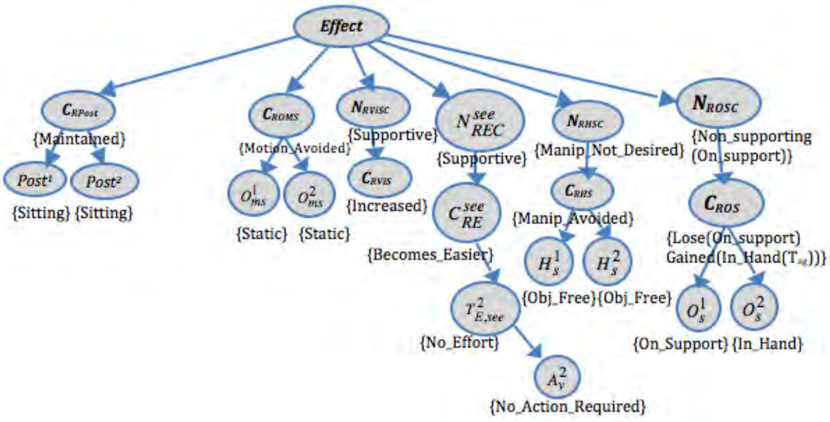


Figure 10.12: Refined consistent explanation tree after three demonstrations (a), (b) and (c) of figure 10.9 for the show object task.

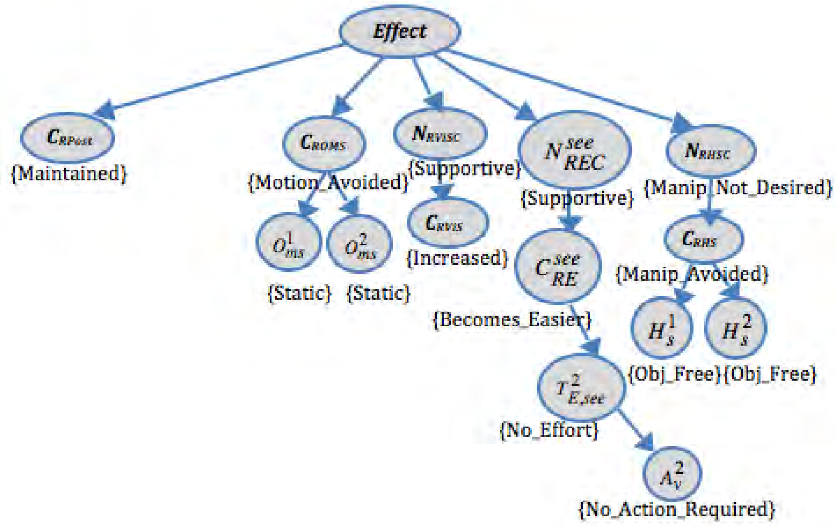


Figure 10.13: Refined consistent explanation tree after fourth demonstrations (d) of figure 10.9 for the show object task.

formed by the horn clause of the leaves node is:

$$\begin{aligned} Task\,(Show\_Object) \leftarrow (C_{RPost} = Maintained) \wedge (O^1_{ms} = Static) \wedge \\ (O^2_{ms} = Static) \wedge (C_{RViS} = Increased) \wedge (A^2_v(see) = No\_Action\_Required) \wedge \\ (H^1_s = Object\_Free) \wedge (H^2_s = Object\_Free) \end{aligned} \tag{10.25}$$

By replacing the abbreviations with the symbolic terms, presented in section 10.2, the above understanding comes out to be:

$$\begin{aligned} Task\,(Show\_Object) \leftarrow (Relative\_Posture = Maintained)\wedge \\ (Object\_Initial\_Motion\_Status = Static)\wedge \\ (Object\_Final\_Motion\_Status = Static)\wedge \\ (Object\_Relative\_Visibility\_Score = Increased)\wedge \\ (Action\_to\_See = No\_Action\_Required)\wedge \\ (Initial\_Hand\_Status = Object\_Free)\wedge \\ (Final\_Hand\_Status = Object\_Free) \end{aligned} \tag{10.26}$$

Note that the above understanding is from the *target-agent's* perspective. This means the *target-agent* should put no effort to see the *target-object*, the visibility score of the *target-object* should be increased from the *target-agent's* perspective, the hand of the *target-agent* should be free of object, etc. As mentioned earlier, such analysis from the *performing-agent*'s perspective could be used to learn the preferences of the *performing-agents* in different situations.

Note that the irrelevant predicates such as reachability and graspability of the *target-agent* as well as the object's status have been autonomously pruned out completely from the learnt desired effect of the task.

Explicitly learning the preconditions for a task is not the scope of this chapter, however, when a leaf node corresponding to *WI* appears in the learnt concept, we let it there to be treated at the task planning level to enrich the list of preconditions based desired effects, while planning.

10.4.2 Hide an object

The next task demonstrated to the robot was to hide an object. Figure 10.14 shows three demonstrations for the task of hiding an object with different initial scenarios. The understanding of the task after these three demonstrations, formed by the horn clause of the leaf nodes is:

$$\begin{aligned} Task\,(Hide\_Object) \leftarrow (Post^1 = Sitting) \wedge (Post^2 = Sitting)\wedge \\ (O^1_{ms} = Static) \wedge (O^2_{ms} = Static) \wedge (ViS^2 \approx 0) \wedge (C^{see}_{RE} = Becomes\_Difficult)\wedge \\ (H^1_s = Object\_Free) \wedge (H^2_s = Object\_Free)\wedge \\ (O^1_s = On\_Support) \wedge (O^2_s = On\_Support) \end{aligned} \tag{10.27}$$

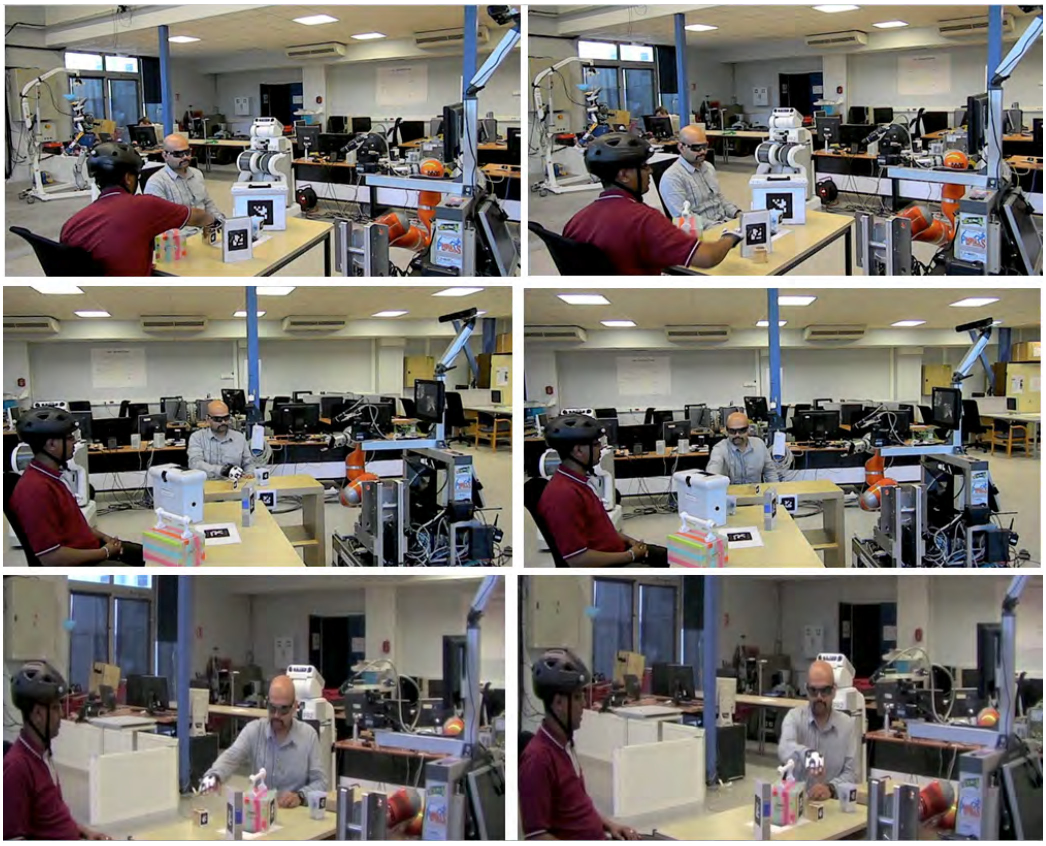


Figure 10.14: Three demonstrations for the task of hiding an object, observed by JIDO robot. First column shows the initial scenarios and the second column shows the final scenarios after performing the task.

This results into following representation of the hide task's effect:

$$\begin{aligned}
Task\,(Hide_Object) \leftarrow (Human_Initial_Posture = Sitting)\wedge \\
(Human_Final_Posture = Sitting)\wedge \\
(Object_Initial_Motion_Status = Static)\wedge \\
(Object_Final_Motion_Status = Static)\wedge \\
(Object_Final_Visibility_Score \approx 0)\wedge \\
(Relative_Effort_Class_to_See = Becomes_Difficult)\wedge \\
(Initial_Hand_Status = Object_Free)\wedge \\
(Final_Hand_Status = Object_Free)\wedge \\
(Object_Initial_Status = On_Support)\wedge \\
(Object_Final_Status = On_Support)
\end{aligned} \tag{10.28}$$

Similar to show task, further demonstrations in which the *target-agent* would be standing, would result into refined and more abstract level understanding about the *target-agent*'s posture. However, note that the main differences between the understanding of the show and the hide tasks have been captured. In hide task, the effort hierarchy corresponding to see the object is pruned at relative effort class level, instead of maintaining the lowest level node, of required action, as was the case of show task. This results into the understanding of the hide task that the relative effort to see the object should become difficult for the *target-agent*. Also, the visibility score hierarchy has been pruned at lowest level of absolute value of visibility score. For the show task, the actual values of visibility score from the *target-agent*'s perspective were not consistent but were always greater than the initial values. Hence, the framework autonomously results into a higher level of abstraction, which is: increased relative visibility score. Whereas, in the case of the hide task the absolute value of visibility score itself is always negligible, hence making the visibility score node as consistent and relevant.

Note that again the effect on the abilities to reach and grasp the object by the *target-agent* have been autonomously found to be irrelevant and pruned out from the explanation tree, as was the case for show task.

10.4.3 Make an object accessible

Next, the task of make an object accessible has been demonstrated to the robot. There were total 5 demonstrations, 3 of them were similar to the figures 2.2(b), (c) and (d) as illustrated in chapter 2. The rest two demonstrations were in different relative arrangements of the object and the humans, and the *target-agent* was in standing posture.

Note that as the robot does not autonomously find out the end of a task, we explicitly provide the time stamp of the end. In this case, the end time stamps were not the instants shown in figures 2.2(b), (c) and (d), where the *target-agent* has already begun to reach the object. The intention behind the make accessible task is to make the object easier to be reached and seen by the *target-agent*; and depending upon the requirement and the situation, the *target-agent* will take it sometime in future. Hence, the end of the make-accessible task provided to the robot is the instant when the *performing-agent* has finished putting the object on the table to make it accessible. This is different from the give task where the task is said to be finished only when the object is in the hand of *target-agent*.

The understanding of the robot about the make accessible task after these five demonstrations was:

$$\begin{aligned}
Task\,(Make_Accessible) \leftarrow (C_{REC}^{reach} = Becomes_Easier)\wedge \\
(C_{RPost} = Maintained) \wedge (O_{ms}^{1} = Static) \wedge (O_{ms}^{2} = Static)\wedge \\
(C_{REC}^{grasp} = Becomes_Easier) \wedge (C_{RViS} = Increased)\wedge \\
(N_{REC}^{see} = Supportive) \wedge (H_{s}^{1} = Object_Free)\wedge \\
(H_{s}^{2} = Object_Free) \wedge (O_{s}^{1} = On_Support)\wedge \\
(O_{s}^{2} = On_Support)
\end{aligned} \tag{10.29}$$

Which results into following symbolic representation:

$$\begin{aligned}
Task\,(Make_Accessible) \leftarrow (Relative_Effort_to_Reach = Becomes_Easier)\wedge \\
(Relative_Posture = Maintained)\wedge \\
(Object_Initial_Motion_Status = Static)\wedge \\
(Object_Final_Motion_Status = Static)\wedge \\
(Relative_Effort_to_Grasp = Becomes_Easier)\wedge \\
(Object_Relative_Visibility_Score = Increased)\wedge \\
(Nature_Effort_Class_to_See = Supportive)\wedge \\
(Initial_Hand_Status = Object_Free)\wedge \\
(Final_Hand_Status = Object_Free)\wedge \\
(Object_Initial_Status = On_Support)\wedge \\
(Object_Final_Status = On_Support)
\end{aligned} \tag{10.30}$$

Note that the predicate related to the posture of agent is from the point of view of desired effect on the posture due to the task. This posture does not indicate the change of posture, which might occur due to the actions required by the agent to see, reach or grasp the object. This is captured in another fact, which is encoded in effort-based hierarchy. For example, the *Relative_Posture* predicate obtained in above understanding indicates that the task does not change the posture of the agent, but it does not say that to see, take or reach the object the agent would not be required to change his/her posture. The virtual action estimated by the robot for the agent to take an object might itself include $A_v^{posture}$, i.e. the agent has to change his/her posture.

The interesting observation for the make accessible task understanding is that, it did not filter out reachability and graspability as irrelevant predicates, as were the cases for the show and the hide tasks. It found that the reachability and graspability of the *target-object* by the *target-agent* should become easier.

10.4.4 Give an Object

The next task demonstrated to the robot was of giving an object. The scenarios were similar to the make accessible task, the only difference was that the *performing-*

agent was holding the *target-object* at appropriate place in the space and waiting for the *target-agent* to take it, instead of putting the object on the support. For this task the end time stamps, indicated to the robot, were the moments when the *target-agent* takes the object from the *performing-agent*. There were total three demonstrations and the task understanding based on the leaf nodes of the *m-estimate* based consistent explanation sub-tree is:

$$\begin{aligned} Task\,(Give) \leftarrow (A_v^2(reach) = No\_Action\_Required) \wedge \\ (C_{RPost} = Maintained) \wedge (A_v^2(grasp) = No\_Action\_Required) \wedge \\ (O_{ms}^1 = Static) \wedge (O_{ms}^2 = Static) \wedge (C_{RViS} = Increased) \wedge \\ (N_{REC}^{see} = Supportive) \wedge (C_{RHS} = Manipulability\_Gained) \wedge \\ (O_s^1 = On\_Support) \wedge (O_s^2 = In\_Hand) \end{aligned} \tag{10.31}$$

Which results into following symbolic representation:

$$\begin{aligned} Task\,(Give) \leftarrow (Action\_to\_Reach = No\_Action\_Required) \wedge \\ (Relative\_Posture = Maintained) \wedge \\ (Action\_to\_Grasp = No\_Action\_Required) \wedge \\ (Object\_Initial\_Motion\_Status = Static) \wedge \\ (Object\_Final\_Motion\_Status = Static) \wedge \\ (Object\_Relative\_Visibility\_Score = Increased) \wedge \\ (Nature\_Effort\_Class\_to\_See = Supportive) \wedge \\ (Relative\_Hand\_Status = Manipulability\_Gained) \wedge \\ (Object\_Initial\_Status = On\_Support) \wedge \\ (Object\_Final\_Status = In\_Hand) \end{aligned} \tag{10.32}$$

Compared to the make accessible task, the three main differences, which in-fact are interrelated, in the understanding of the give task are: the *target-agent* should apply no action to reach and to grasp the object, the object should be in the hand of *target-agent* and the manipulability of the *target-object* is gained by the *target-agent*. It encodes that the give task will not be finished until the object is in *target-agent*'s hand, whereas for make accessible task it is sufficient to make the *target-object* easier to be seen and reached by the *target-agent*.

Note that, in the future demonstrations, initially sitting *target-agent* might be required to standup to take the *target-object* from the *performing-agent*. In that case the currently learnt desired effect about the *Relative_posture* predicate with its value maintained will be autonomously pruned out to refine the understanding of the task.

10.4.5 Put-away an object

Next task to demonstrate was to put-away an object. There were four demonstrations in different situations. Following is the robot's understanding about the put-away task:

$$\begin{aligned}
Task(Put_Object_Away) \leftarrow (C_{RE}^{reach} = Becomes_Difficult)\wedge \\
(C_{RPost} = Maintained) \wedge (O_{ms}^{1} = Static) \wedge (O_{ms}^{2} = Static)\wedge \\
(C_{RE}^{grasp} = Becomes_Difficult) \wedge (C_{RViS} = Decreased)\wedge \\
(C_{RE}^{see} = Maintained) \wedge (H_{S}^{1} = Object_Free)\wedge \\
(H_{S}^{2} = Object_Free) \wedge (O_{S}^{1} = On_Support)\wedge \\
(O_{S}^{2} = On_Support)
\end{aligned} \tag{10.33}$$

Which results into following representation:

$$\begin{aligned}
Task(Put_Object_Away) \leftarrow \\
(Relative_Effort_to_Reach = Becomes_Difficult)\wedge \\
(Relative_Posture = Maintained)\wedge \\
(Object_Initial_Motion_Status = Static)\wedge \\
(Object_Final_Motion_Status = Static)\wedge \\
(Relative_Effort_to_Grasp = Becomes_Difficult)\wedge \\
(Relative_Visibility_Score = Decreased)\wedge \\
(Relative_Effort_to_See = Maintained)\wedge \\
(Initial_Hand_Status = Object_Free)\wedge \\
(Final_Hand_Status = Object_Free)\wedge \\
(Object_Initial_Status = On_Support)\wedge \\
(Object_Final_Status = On_Support)
\end{aligned} \tag{10.34}$$

10.4.6 Hide-away an object

Next task to demonstrate was to hide-away an object. There were four demonstrations in different situations. Following is the robot's understanding about this task:

$$\begin{aligned}
Task\,(Hide_Object_Away) \leftarrow (C_{RE}^{reach} = Becomes_Difficult)\wedge \\
(C_{RPost} = Maintained) \wedge (O_{ms}^{1} = Static) \wedge (O_{ms}^{2} = Static)\wedge \\
(N_{RE}^{grasp} = Becomes_Difficult) \wedge (ViS^{2} \approx 0.0)\wedge \\
(N_{RE}^{see} = Becomes_Difficult) \wedge (H_{s}^{1} = Object_Free)\wedge \\
(H_{s}^{2} = Object_Free) \wedge (O_{s}^{1} = On_Support) \wedge (O_{s}^{2} = On_Support)
\end{aligned} \tag{10.35}$$

Which results into following representation:

$$\begin{aligned} Task\,(Hide\_Object\_Away) \leftarrow \\ (Relative\_Effort\_to\_Reach = Becomes\_Difficult)\wedge \\ (Relative\_posture = Maintained)\wedge \\ (Object\_Initial\_Motion\_Status = Static)\wedge \\ (Object\_Final\_Motion\_Status = Static)\wedge \\ (Relative\_Effort\_to\_Grasp = Becomes\_Difficult)\wedge \\ (Object\_Final\_Visibility\_Score \approx 0.0)\wedge \\ (Relative\_Effort\_to\_See = Becomes\_Difficult)\wedge \\ (Initial\_Hand\_Status = Object\_Free)\wedge \\ (Final\_Hand\_Status = Object\_Free)\wedge \\ (Object\_Initial\_Status = On\_Support)\wedge \\ (Object\_Final\_Status = On\_Support) \end{aligned} \tag{10.36}$$

Note that the above understanding of the hide-away task tries to inherit the properties of the hide and put-away tasks. For the hide task the ability to reach and grasp were found to be irrelevant, whereas in the hide-away task, similar to the put-away task, these have been found relevant with the corresponding values, which make these abilities difficult for the *target-object* from *target-agent*'s perspective. For the put-away task, the robot found relative visibility score to be decreasing because of the relatively away position of the object, whereas for hide-away task the absolute visibility score itself has been found to be negligible from the *target-agent*'s perspective, as was the case for the hide task.

Note that the effects related to the reach and grasp, wherever appeared in the learnt concepts were similar. This is because of the type of task demonstrated. However, if the robot will be demonstrated with the tasks such as, put the *target-object* to enable the *target-agent* to touch it, the learnt concept would successfully capture the effect of reach independent of the grasp.

10.5 Performance Analysis

10.5.1 Processing Time

Figure 10.15 shows number of demonstrations per task, N, and the average processing time per demonstration, T, after finishing each demonstration. It is interesting to observe that T is more for the tasks, which require the robot to apply more number of virtual actions on the *target-agent* for finding the reach feasible effort for an ability. For example, in the case of hide task, the *target-object* is placed to be invisible by the *performing-agent* from the perspective of *target-agent*. In such

Task	Total Number of Demonstrations	Average Time for learning after each demonstration in *s*
Make Accessible	5	0.56
Hide	3	0.81
Show	4	0.63
Give	3	0.59
Put Away	4	0.61
Hide Away	4	1.2

Figure 10.15: Number of demonstration per task and average learning/refinement time (in s) after each demonstration for each task.

cases, most of the time the robot needs to apply *Whole_ Body_ Effort* or even *Displacement_ Effort* to find the least feasible effort to see or reach the *target-object* by the *target-agent* but before that it has to test for lower effort levels. Whereas, for the tasks where the least effort of the *target-agent* is found by applying the virtual actions corresponding to lower-level efforts such as *Head_ Effort* or *Arm_ Effort*, the computation time is less. For example, for the give task, from *target-agent*'s perspective, the least feasible efforts to see and to reach, both are lower, hence lower processing time.

In fact, the convergence after each demonstration is *O(n)* where n is the total number of predicates in the domain theory, as even the learnt concept appears to be pruned significantly, the robot maintains the *m-estimate* of all the predicates to incorporate the possibility of lifelong learning and confusion based refinement. This is a choice we have made. However, one could chose to refine only the tree learnt so far and batch-process the remaining data offline. This will make the system learn faster but may need offline processing.

10.5.2 Analyzing Intuitive and Learnt Understanding

The first question is how we can define a fully accurate model of "what" does a task mean in terms of effect. We do not assume a close world assumption and in fact we should not, as a task could have effect on mental and emotional states of the *target-agent*, his desire and so on. Hence, it is practically not possible to define a domain theory, which will be 'complete' or 'accurate'. Further, to figure out a ground truth exact model of the task semantics is also not possible.

In this chapter, we have tried to incorporate a subset of predicates to better understand a task, but we can't claim it to be a complete domain theory for such task understanding. One of the contribution of this thesis is to identify the key predicates (which we think are more relevant, from the perspective of human-robot day-to-day interactive manipulation tasks), which could lead to a symbolic understanding of the tasks. We enabled the robot to infer their values online. Hence, we could not

Task	Intuitive understanding of task			Number of demonstration to reach at intuitive understanding			Number of trial to get the intuitive sub-tree of hypothesis
	For target agent's ability		For target object state	For target agent's ability		For target object state	
	To see	To Reach		To see	To reach		
Make Accessible	Supportive	Easier	Static on support	4	2	1	4
Hide	Difficult	(Not Relevant)	Static on support	2	2	1	2
Show	Directly	(Not Relevant)	(Not Relevant)	1	3	4	4
Give	Supportive	No Action*	In target-agent's hand	3	1	1	3
Put Away	Maintain	Difficult	Static on support	2	2	1	2
Hide Away	Difficult	Difficult	Static on support	3	3	1	3

*assuming task ends when object is grasped by target agent

Figure 10.16: Analyzing the key attributes understanding by the robot with our intuitive understanding for a task.

provide any analysis based on accurate ground truth models. Further, it should be noted that the 'accuracy' is a term from our understanding of what does the task means, which itself might not be accurate.

The question of "correct tree" will probably be unanswered until we will really have a "complete" domain knowledge. We can intuitively say something about a "partially correct" explanation of a task. Since the focus of the chapter is on understanding human robot interactive manipulation tasks, we have chosen the predicates related to the effect on *target-agent*'s ability and the effect on object's state to analyze the "partial correctness". For this, purely based on our intuitive understanding of task we have compared the robot's understanding. This is just to demonstrate the strength of the presented framework that it can learn such understanding even from two successive demonstrations, if demonstrated as differently as feasible, while maintaining the semantics of the task. Figure 10.16 summarizes this comparison. Note that for the values belonging to higher levels in hypothesis tree, such as *supportive*, it takes more number of trials than the lower levels, such as *directly*. This is obvious, as the pruning of sub-tree is bottom up, for the sake of understanding a task at appropriate level of abstraction. So number of demonstrations to conclude non-relevance accumulates as the level of abstraction goes up.

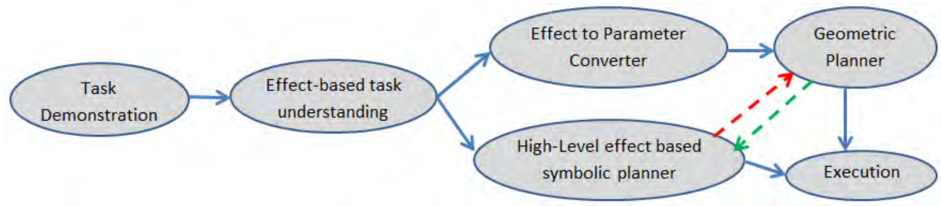


Figure 10.17: Different ways to reproduce a learnt task, based on desired effect. Either the effect could be converted in terms of constraints for geometric planner or could be used by high-level symbolic planner.

10.6 Practical Limitations

The practical issues related to inference of various facts limit the output of the presented framework. For example, to infer *object_ in_ hand* fact, the robot attaches the object to the human hand if the object is not on a support and the hand is close to the object. This limits estimating how the object is being grasped by the agent, i.e. what is the relative position and orientation of the hand with respect to the grasped object. This further limits the inference of dual grasp, i.e. is there sufficient space available to grasp the object simultaneously by another human or not? So, for the give task we have deliberately chosen the object of bigger dimensions. This always leaves sufficient space, so that the robot can positively infer the possibility of the dual grasp by the t*arget-agent*. This enables time stamping the end of the task as soon as the *target-agent* grasps the *target-object*, and could also be used for finding for hand-over action. Another limitation arises from the localization of the object. Since it is based on the tag on the object, the *performing-agent* was instructed to manipulate the object such that the tag always faces towards the camera of the robot, with special care at the end of the task.

10.7 Potential Applications and Benefits

The symbolic understanding of a task along with its geometric counterpart makes the robot more 'aware' about its behavior. Below we discuss few of the potential applications.

10.7.1 Reproducing Learnt Task

As shown in figure 10.17 a task learnt in terms of desired effect could be reproduced in different ways: (i) *Effect-to-Parameter Converter*: By using a converter, which will interpret the effect and convert it into the parameters of geometric planner, such as one presented in HRI task planner in chapter 7. (ii) *High-Level task Planner*: However, not all effects could be necessarily represented/converted in terms of parameters of a geometric planner. For example, if the desired effect is: the person should be "Happier", a High-Level task planner, such as HATP [Alili 2009] will be required to find the actions, which could make desired change in the emotional state of the person. Such high-level symbolic planner could also be used instead of the converter, to plan to achieve an effect in different ways. For example, to achieve the effect of "show" an object, such high-level planner could plan to displace an occluding object from the target person's perspective, instead of directly manipulating the *target-object*. Of course, as we have already discussed in chapter 8, a two-way handshaking between the low-level planner and the high-level planner will be required to better converge to a feasible plan.

In the current demonstrations, since the domain of the demonstrated tasks is Human-Human Interactive Manipulation Tasks, we have implemented a simple converter to ground the effects in terms of the parameters of our HRI Task Planner presented in chapter 7. It has different sub-modules to convert different predicates. For example, to convert the effect "Easier to be reached by the target person", it finds the target person's current least effort, C_{LE} to reach the target object, by using the approach presented in section 4.7 of Mightability Analysis chapter 4. Then it finds the places, which are reachable by effort levels $< C_{LE}$ by the target person, which serves as the input candidate places in the presented framework of chapter 7. Similarly, different efforts are converted to the parameters of the framework. As the effect of task understanding is at a level of abstraction, which makes it independent from the kinematics of the agent. We have simply used the converter to feed the constraints for planning the hide task by PR2 robot, using the same framework of our HRI Task Planner presented in chapter 7. Figure 10.18 shows PR2 robot reproduces the task of hiding a *target-object* (grey-tape) form a *target-agent* (human) in a different scenario.

Note that the framework presented in chapter 7 is just an example, which could be used in geometric planner block of figure 10.17. For a different planner an appropriate effect to parameter converter is required to be designed.

10.7.2 Generalization to novel scenario

The understanding of a task is independent of the shape and size of the object, the trajectory as well as of the absolute/relative distances among the agents and objects. This facilitates the robot to perform the task in an entirely different scenario, when integrated with our constraint based task planners, presented in chapter 7. For example, the robot will be able to perform the task of making an object accessible by putting it at the top of the box, even if it would have been never demonstrated to the robot.

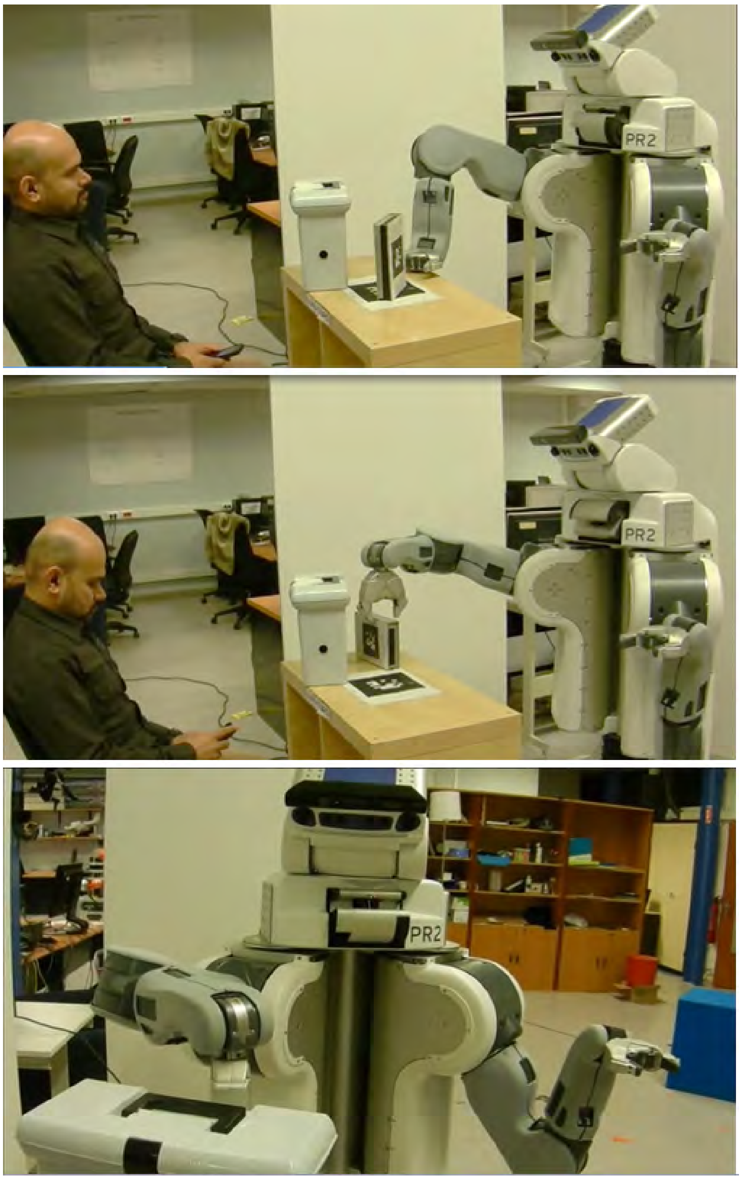


Figure 10.18: Using the effect-to-parameter converter, learnt effects of hide task has been converted into constraints comparable with HRI task planner presented in chapter 7. By using this, the PR2 robot reproduces the task of hiding a *target-object* (Grey tape) from a *target-agent* (Human) in a different scenario.

10.7.3 Greater flexibility to high-level task planners

Once the planner at symbolic level knows the semantics of a task independent of the execution; it could plan to achieve the task in a variety of ways. Such as if it 'understands' that hiding means object should be difficult to be seen by *target-agent*, depending upon the situation, it could decide to cover the *target-object* with some other container type object to make it invisible. Similarly, for showing or making an object accessible, again instead of directly manipulating the *target-object*, it could plan to displace the occluding or obstructing object from the human's perspective to achieve the same desired effects. The task-planner could even involve third agent to achieve the task. The high-level effect based task planners [Cambon 2009], [Alili 2009], could take into account such learnt desired effect for a task in the planning. These planners are capable of generating co-operative plan for achieving a task, based on the reasoning: which agent can cause which part of the desired effects of the task with which level of effort.

10.7.4 Transfer of understanding among heterogeneous agents

Since the robot understands the task independent of the trajectory planning and control level execution it can easily transfer the task semantics to another robot of entirely different kinematics structure and shape. And the other robot equipped with similar capabilities of visuo-spatial perspective taking of the agents could then interpret the understanding and perform it by respecting its own constraint of whole body planning.

10.7.5 Understanding by observing heterogeneous agents

The robots are equipped with visuo-spatial perspective takings for different set of states for a variety of other agents (JIDO, HRP2, PR2, etc.) apart from the human. Hence, the robot could understand demonstrations by different types of agents. Such as if the human will perform the task for the robot itself, it could understand the task with the same framework by putting itself as the *target-agent* and inferring the facts from its perspectives. Also, if human will perform a task for HRP2 robot, PR2 robot could understand the task by using the same framework.

10.7.6 Generalization for multiple target-agents

Another interesting research work in future is to generalize the understood task, to perform for multiple target humans. Such as hide an object from two humans at the same time, show an object to a group of people, etc. The symbolic level understanding of tasks will facilitate such generalizations.

10.7.7 Facilitate task/action recognition and proactive behavior

By partially observing the human's action and its effect the robot could probabilistically classify the task or the desired changes of the action, by the human. Even if the complete task is known to the robot, again based on the desired effects of the task, the robot could show proactive behaviors to partially/fully facilitate the task while reducing the human's effort. For example, if the robot infers/knows that the human is trying to reach an object, it could proactively offer help by making the object accessible. Similarly, if the robot knows at symbolic level that the human wants to show or give an object to it, knowing the 'meaning' of task robot could proactively turn its head or move its arm to facilitate achieving the desired effect as an attempt to guide as well as support the task.

10.7.8 Enriching Human-Robot interaction

Such symbolic awareness about the task's semantics could also enrich the verbalize interaction with the human partner, as the robot will be able to communicate the task at the level of abstraction understandable by the human.

10.7.9 Understanding other types of tasks

The focus of the thesis was on basic human-human interactive tasks where one human is performing for the other human. However, the symbolic facts and the learning framework presented in this thesis will help in effect-based understanding of various other types of tasks, such as tap, lift, drop, dump an object in trash bin, throw an object, etc. For example, after tapping a ball initially laying on a support, the final state of ball is moving on support, hence the effect contains object symbolic status: *on_support*, motion status: *moving*. Similarly, for the *throw* task the effect contains object symbolic status: *in_air*, motion status: *moving*. Similarly, for the *lift* task, the object *on_support* will be lost and object *in_hand* will be gained, etc. Certainly more predicates and reasoning on the dynamics of action will further be required to understand the complete effect of such tasks. The framework will autonomously prune out irrelevant facts, associated either with the human or with the object.

10.8 Until Now and The Next

In this chapter, we have elevated the capability of the robot to autonomously understand the semantics of task in terms of desired effect to be achieved for the success of the task. We have identified the hierarchy of essential facts, without which a subset of basic HRI tasks could not be understood. Then we have adapted explanation based learning framework to construct and refine the hypothesis of a task. The robot begins to learn the task semantics from the very first demonstrations and refines it in successive demonstration. We have shown that the framework is able to find the distinguishing aspects of different tasks, of opposite natures, such as show and hide.

We have argued that such level of task understanding facilitates smooth interaction, transfer of understanding among different agents, reproduction of the learnt task in different scenario without providing the learning data for that scenario. This also facilitates the planning for the same task in different ways by a high-level effect based task planner.

This chapter wraps the scientific contribution of the thesis with a step towards the emulation aspects of social learning. This in fact is very important aspect for existence of a social robot in our day-to-day lives, but a lot more to be explored and done for making such socially intelligent robots, which can learn wide ranging tasks by observing us and 'grow' lifelong.

Chapter 11

Conclusion

Contents

11.1 Main Contributions **281**
11.2 Prospects **285**
11.2.1 Immediate Potential Applications 285
11.2.2 Future Work 286
11.2.3 Future Technology Transfer Activities 287
11.3 Two Lines **288**
11.4 One Line **288**

11.1 Main Contributions

The focus of the thesis is bottom up embodiment of social and human aware factors and abilities for the robot's 'development'.

The main scientific contributions of the thesis are:

- **Social intelligence embodiment pyramid**: We begin by identifying the basic cognitive and behavioral capabilities for the robots to co-exist in human centered environment in socially intelligent i.e. socially acceptable and expected manner. This is motivated from psychology, child development and human behavioral research. Based on this we have conceived a *social intelligence embodiment pyramid.*
- **Generalized theories**: We have presented a generalized theory of HRI, derived different HRI research challenges within this unified framework, and further presented a generalized theoretical framework for regulating the proactivity of the robot.
- **New concepts**: At different levels of social embodiment pyramid sketched in chapter 1, from the Human-Robot Interaction point of view, we have discussed state of the art and identified research challenges. Then at each level, from HRI perspective, we have introduced new concepts such as *Mightability Analysis*, *Agent-Agent Affordances*, *Affordance Graphs*, *Geometric task space backtracking*, *Symbolic and Geometric task planner handshaking*, and shown

these to be important and useful for elevating robots' capabilities for efficient human robot interaction.

- **Generic frameworks and algorithms**: Further, at each level, we have presented generic frameworks for socially-aware path planning, planning human-robot interactive object manipulation tasks, instantiating proactive robot behavior, learning effect-based task semantics. All the presented frameworks are generic in the sense:
  - it can plan and adapt for different situations and tasks,
  - type of the robot (PR2, HRP2, Jido) is a parameter to the frameworks,
  - facilitate to incorporate human, human oriented constraints, preferences and social norms and expectations in key decisional and planning aspects.

This is a step towards harmonizing the human-robot co-existence.

In fact, the robot *Manava*, dreamed in the introduction of the thesis is motivated from the contribution of the thesis and closely resembles the socio-cognitive capabilities and behaviors developed in this thesis.

Below we will summarize main contribution of each of the chapters.

Chapter 1: We tried to sketch a social intelligence embodiment pyramid, by identifying key components from psychology, child development and human behavioral research. This serves another purpose to 'locate' the scientific contribution of the thesis at different levels of social, cognitive and behavioral aspects.

Chapter 2: The contribution of this chapter in addition to present the state of the art is that we used this opportunity to identify and present various sub-categories of research challenges and aspects from the HRI perspective, complementing the discussion of introduction chapter (chapter 1).

Chapter 3: Before moving to the practical contribution of the thesis, we have developed a generalized framework for HRI, based on the causal nature of changes in the environment's state.

We presented a generalized definition of Environment, its attribute and action from the perspective and requirements of HRI.

We discussed that it serves as an unified framework for perceiving various aspects of Human Robot Interaction based on how much of the world state and action are known: to do perspective taking, affordance analysis; to plan basic HRI tasks based on constraints; to behave proactively; to learn (emulate, imitate); to ground action, agent, object, changes; to predict effect; and so on.

Rest of the chapters instantiate some of the important attributes of this framework by introducing new concepts as well as present algorithms and frameworks to address a rich set of key research aspects to elevate socio-cognitive capabilities of the robots.

Chapter 4: We equipped the robot with a rich visuo-spatial perspective taking ability, which not only analyzes what is visible and reachable, but also what is not and why.

Further, we have presented a new level of abstraction for analyzing effort, and developed an effort-hierarchy, based on the type of body parts involved. This facilitates the robot to communicate and understand human effort in a 'meaningful' way.

Further, we have developed the concept of *Mightability Analysis*, derived by fusing visuo-spatial perspective taking and effort analysis. *Mightability* stands for "*Might be Able to* ..." and facilitates the robot to reason about various abilities of the agent not only from his/her/its current state but also from a set of different states attainable by the agent. Further, we have shown that the Mightability information required for ongoing interaction can be updated online.

We have shown that using Mightability Analysis, the robot can find the least-feasible efforts to see and reach different objects and places by the agents. For the object-oriented part of such analyses, we have encoded such information in a graph, which we termed as *visuo-spatial ability graph*.

This chapter builds a base and serves throughout the thesis to incorporate the important aspect of multi-effort ability analysis of the agents in various decision-making and planning problems.

Chapter 5: Introduced the concept of *Agent-Agent Affordances*. We have presented a framework to infer such *agent-agent affordances* for a set of basic tasks. Further, we have equipped the robot to infer various types of *agent-object affordances* and agent's physical states, to facilitate the robot supervision system. We have also identified different sub-categories of *object-object relative spatial relation* to facilitate a rich situation assessment based task planning. We have shown the results of different affordances and situation assessment analyses on real robots, which build and use a real time 3D representation of the environment.

Chapter 6: Presented a framework for planning a socially acceptable path. It facilitates the robot to selectively adapt a rule depending upon the dynamics and local structure of the environment.

We have also shown that the planned path tries to inherits the natures of Voronoi Diagram based path as well as A* based path, in the situations where they perform better. We have also shown enhancement of the performance by quantitative and qualitative analyses with respect to purely reactive behavior of the robot.

Further, we have presented a framework to guide a person in a way he/she wants to be guided. This framework allows natural deviations of the person to be guided. Another novelty of the framework lies in the fact that the robot carries out appropriate re-engagement efforts, but they are goal-directed instead of just reaching or following. Hence, trying to exert a kind of social force towards the goal, which has been shown to be existing in human-human interaction and relative navigation scenarios.

Chapter 7: In this chapter, we have exploited the notion of grasp-placement inter-dependency, fuse it with a set of HRI aspects, and presented a generic framework to plan a set of basic Human-Robot Interactive Manipulation tasks.

The presented framework is based on constraint-hierarchy based and the novelty is it incorporates novels constraints from the perspectives of the task, the human and the environment, which have not been incorporated before.

Another novelty of the framework is, it introduces right constraints at right stages of planning to subsequently reduce the search space and holds the expensive constraints for very later stages of planning.

We have shown its application for planning basic cooperative tasks: *show*, *give*, *make-accessible* as well as competitive tasks *hide*, *put away*.

We have demonstrated on three different robotics platforms: JIDO, PR2 and HRP2.

Chapter 8: Introduced the concepts of *Manipulability Graph*, which encodes Agent-Object Affordances, *Taskability Graph*, which encodes Agent-Agent Affordances and fuses them to introduce the *Affordance Graph*. The novelty of the concept is, it contains a rich information about the action possibilities along with the associated efforts and converts a variety of decision-making, grounding and planning problem into a graph search problem.

We have shown its applications in grounding interaction, grounding changes in the environments and shared task planning.

We have also discussed that 'playing' with the parameters of the edge, the vertices and the criteria of the graph construction, various social and individual preferences & constraints could be incorporated.

Further, we have introduced the notion of *two-way handshaking of geometric and symbolic planner* in the context of human-robot interaction. The novelty is to elevating the typical geometric trajectory planner with the geometric task planner, which reduces the burden of the symbolic planner to bother about the geometric parameters and actions of the task.

Further, we have introduced the concept of *backtracking at geometric level* for HRI task planning to find a feasible solution for a series of symbolic actions. The novelty of the concept is it reduces the overhead of unnecessary fail messages to flood the symbolic planner, in the case the task could be solved by searching for alternative solutions within the geometric task plan obtained so far.

Chapter 9: Presented a generalized theory of proactivity by identifying and introducing different spaces, in which proactive behaviors could be synthesized and presented different levels of proactivity. this is an attempt to drive the future research in devising proactive behaviors and a way to regulate the "allowed proactivity" of an autonomous social agent.

Further, we have instantiated a couple of human-adapted proactive behaviors and validated through preliminary user studies that such behaviors indeed reduce the human partner's effort and confusion in the joint tasks.

Chapter 10: Identified the necessity to learn a task independent from its execution.

Equipped the robot to the find comparative and qualifying predicates to understand the effect of the task at the appropriate level of abstraction, so that it will be 'meaningful' as well as could be performed in different scenarios and among heterogeneous agents.

Presented an explanation based learning framework and its application to autonomously learn the effect of a variety of basic tasks at appropriate level of abstraction.

We have shown that the approach is able to find the distinguishing characteristics of tasks of opposite natures such as *show* and *hide*, as well as the unique features of the tasks having minor difference such as *make-accessible* and *give*, *hide* and *hide-away*.

11.2 Prospects

11.2.1 Immediate Potential Applications

The developed system will be used for two main immediate future developments: (i) To be used as a platform for various user studies related to interactive manipulation, social navigation, proactivity, cooperative task solving with naive users. This will serve for two main purposes: (a) to identify various other factors related to human preferences, socially expected behaviors, etc. (b) for the self-improvement of the system by fine-tuning the parameters of the system. (ii) To build more complex socio-cognitive abilities and behaviors, by using the basic blocks developed in the thesis.

In this thesis, we have presented a generalized framework for HRI and instantiated many of the environmental attributes. However, many attributes such as agent's emotional state and divergent believes have not been explicitly exploited enough in the thesis. In fact, based on the components developed in this thesis, such aspects are being developed by other contributors of the group, see figure A.1 of system architecture in appendix A. The system architecture also shows the work in progress towards a robot having a rich *theory of mind (TOM)* capabilities, as it involves reasoning from the human's perspective in various levels of decision and planning aspects.

11.2.2 Future Work

We have provided a theoretical framework to reason on the spaces to synthesize various levels of proactive behavior. We have instantiated some of the examples and shown supporting results of the user studies. However, one of the challenging and interesting future works is to develop intelligence for autonomous decision and synthesis of a particular proactive behavior depending upon the situation.

As an initiative towards making the boundary between task primitives and execution primitives visible, we developed framework to enables the robot to understand task's semantics independent of the means to achieve it. However, to understand more complex tasks, it is necessary to incorporate a wider domain theory with richer set of predicates. Exploring this from HRI perspective, by incorporating rich set of attributes of the generalized HRI domain theory as presented in chapter 3, will be an interesting future work. Further, developing autonomous learning capabilities of the execution preferences will be also complementing the contribution of the thesis. Another complementary research challenge is to develop frameworks to understand undesirable effects from demonstration. Integrating such effect-based understanding of the task with high-level task planner will really help towards life-long social-learning and task-reproducing capabilities of the autonomous domestic robot.

We have identified a subset of block in our sketch of socially intelligent pyramid presented in figure 1.1, which we thought to be important. There remain many unexplored blocks and even non-addressed aspects, such as related to emotion, expression, speech synthesis, etc. It would be interesting to identify and develop them and integrate with the contribution of the thesis. Some of the aspects, such as the dialogue module, which is being developed in our group, is being integrated and used for different purposes. Such as for grounding natural language based interaction, where the facts developed in this thesis are used, and for the interpretation of human's desire and the execution of verbal task requests. The basic HRI task planner developed in this thesis elevates the reasoning capabilities of the robot for a smooth and natural human robot interaction. For example, the human will not be required to say verbally "pick the bottle, put is on this table at this place", instead he can communicate in a more natural manner "make the bottle accessible to me" and the robot will autonomously decide where and how to perform the task by reasoning from the human's perspective and effort. All this will help to converge towards a 'better' socially intelligent robot.

From the perspective of navigating in human presence, we have addressed the aspects where the goal location and the task (such as to reach, to guide) are known to the robot. However, there remains a range of complementary aspects where neither the final goal nor the complete task is known or provided to the robot, e.g. to accompany a person or follow a group of people in different types of environments. It would be interesting to explore the additional challenges associated with such aspects and develop a coherent framework, which allow the robot to move and behave in a socially expected manner.

We have shown that the lower layers of the pyramid help in achieving the socio-cognitive capabilities of the layers above. However, the thesis contribution concludes at the layer of pro-social behavioral aspect. It would be interesting to use the components of these layers to develop frameworks to achieve more complex behavioral and cognitive abilities, such as intention understanding, collaborate to compete, helping proactively instead of on demand, and even to develop real negotiation capabilities. The perspective taking, affordance and situation analysis capabilities developed in this thesis will serve a basis for such complex decision-making and planning capabilities.

We are already working on most of the aspects discussed above within the framework of various national and European Union (EU) projects as well as in collaboration with our industrial partners, as outlined next. In this context, the contribution of the thesis is playing a significant role by providing basic tools for further research and development.

11.2.3 Future Technology Transfer Activities

Following are some of the technology transfer activities planned based on the scientific contribution of the thesis:

- As a partner of the *ROMEO2 FUI* project proposal, coordinated by the industrial partner *Aldebaran Robotics*, based on the scientific contribution of the thesis, a set of services have been planned to be developed and transferred to Aldeberan Robotics robot Romeo.
- In EU FP7 *SAPHARI* project (http://www.saphari.eu/), the large part of the system developed in this thesis is planned to be provided as a tool to serve to other partners and to facilitate collaborative research work.
- As a partner of proposed EU FP7 project *SPENCER* (Social situation-aware perception and action for cognitive robots), the scientific contribution of the thesis related to the social navigation and reasoning about the human will be provided as a basic tool to facilitate further research.

11.3 Two Lines

The ***core*** of this thesis is based on bottom up social embodiment, the ***motive*** is to provide basis for developing more complex socio-cognitive behaviors, with the ***vision*** of *Manava* like robot (hypothesized in introduction chapter) to become a reality. The bottom up approach of this thesis helps in achieving this vision by providing open nodes to explore and to 'grow' the robot as a socially intelligent agent.

11.4 One Line

"...PR2 showed good interaction with its environment and intelligence in its responses and behavior."

- a visiting Ph.D. researcher at LAAS-CNRS, Toulouse.

Appendix A

System Architecture

Contents

A.1 System Components 290
A.2 Perception of the World 290

Throughout the thesis period, the overall system has been continuously evolving to incorporate new components and as a team effort to separate robot specific modules from the platform independent modules. Therefore, through the figure we will illustrate the different components of the overall perception, planning, decision and supervision process and the corresponding serving modules of the system architecture.

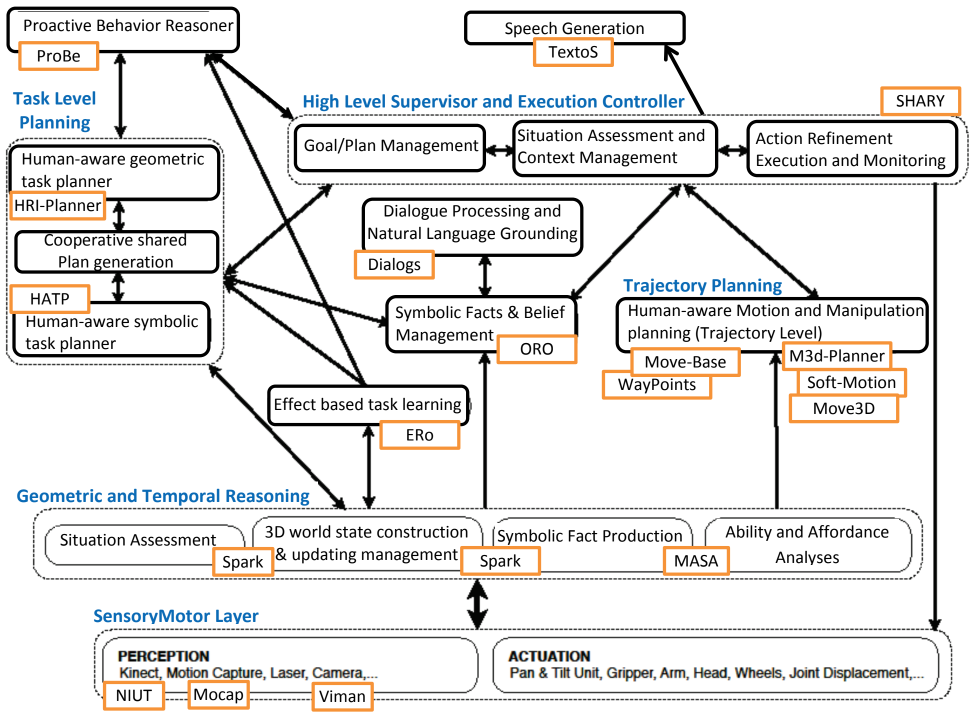


Figure A.1: Main components of the overall HRI system of LAAS and the modules (in orange boxes), which contribute to those components. (Drawing courtesy to Lavindra de Silva, LAAS-CNRS)

A.1 System Components

As shown in figure A.1, the different components and the sub-components have been separated in our overall HRI system and in the orange rectangles, different modules contributing in a particular component have been named. Below is the list of some important modules:

- *MASA*: Mightability and Agent State Analysis
- *SHARY*: Supervision for Human-Robot Interaction ([Clodic 2009])
- *ERo*: Effect Reasoner
- *TextoS*: Text to Speech
- *SPARK*: SPAtial Reasoning and Knowledge
- *ProBe*: Proactive Behavior Reasoner
- *HATP*: Human Aware Task Planner ([Alili 2009])
- *ORO*: Open Robot Ontology ([Lemaignan 2010])
- *Dialogs*: Natural language parsing and interpretation module ([Lemaignan 2011a])
- *MoCap*: Motion Capture Module

Some of the modules are C/C++ libraries, such as *move3d*, *HRI task planner*, some are *GenoM* modules (GenoM is developed at LAAS for distributed robot architecture [Fleury 1997]), such as *Spark*, some are *python* and *tcl* scripts based components, such as *Dialogs*, some are *Open PRS (OPRS)* based systems ([Ingrand], an open source version of the Procedural Reasoning Systems), such as *SHARY*, some are robot specific components, such as various *ROS nodes* ([WillowGarage]) in the case of PR2 robot for sensory-motor controls, *HRP2-Genom* for sensory motor control of HRP2 robot and many more. Arrows show the exchange of data between two components. The form of the data exchange depends upon the type of communicating modules. For example, if it is a *Genom* module communicating with *OPRS* or other *Genom* module it will use posters, the shared memory. The type of data flow is wide ranging and depends upon the purpose. So, we will avoid explicitly mentioning them all, however below we will mention about a couple of the basic components, perceiving the world and executing the trajectory.

A.2 Perception of the World

One of the important component of HRI system, which is the perception of the environment is achieved through a variety of sensors, see figure A.2. Human position and whole body tracking is achieved by the Kinect sensor mounted on different robots (encircled in blue in figure A.2(a)) and in the environment. Precise orientation of

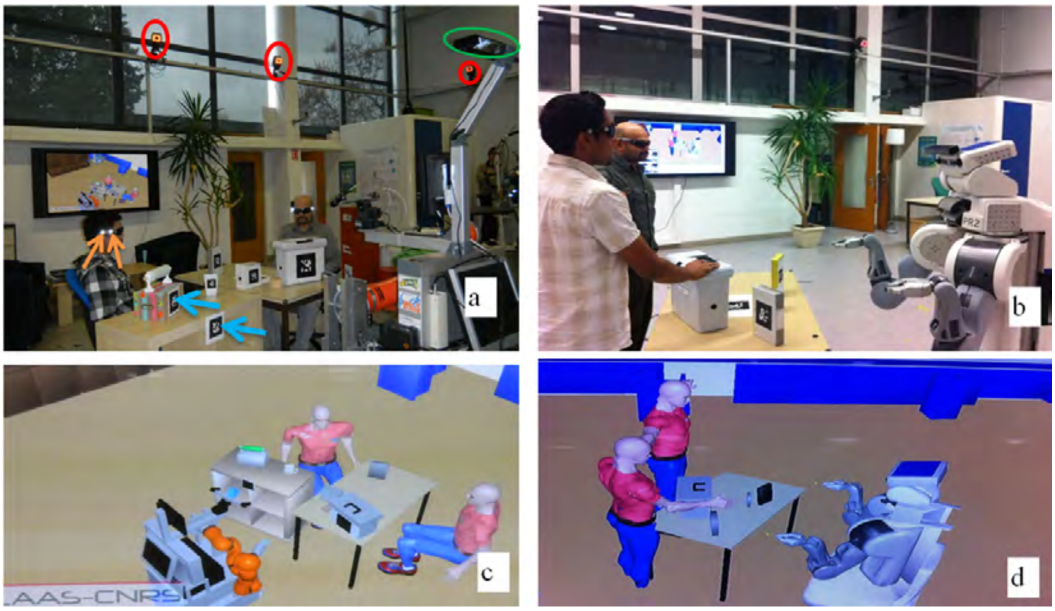


Figure A.2: Information about various sensors and real time construction of 3D world model.

the human head, if required, is obtained by motion capture system, encircled in red. For this human is required to wear calibrated goggles with markers, shown by orange pointers. For object recognition and tracking, stereo camera of the robot is used. This is based on tag identification system, for which objects are required to have unique tag printed or pasted on them, pointed by blue arrows. For obstacle detection while navigating, the laser sensors mounted on the robot are used. Information from all these sensors are passed to Spark module, which then update the 3D model of the world, in our 3D representation and planning software *Move3D* [Simeon 2001]. The real time 3D reconstructions of the environment have been shown in A.2. This 3D model is then shared and used by different components for a variety of purposes ranging from trajectory planning to task planning, decision making and supervising to learning and so on.

Now, let us assume that we have obtained a solution for a task. Then it is the job of supervisor system *SHARY*, [Clodic 2009], to check the preconditions of the task and then to send the command to the low-level controller to execute a particular trajectory. And during the execution, *SHARY* also monitors the environment and if something goes *wrong*, it decides to stop the execution and send appropriate re-planning requests to other modules.

Appendix B

Human-Robot Competition Game

Contents

B.1 The Context and The Game 293

B.2 The Scenario 294

B.3 The Human's and The Robot's Explanations about the Observed Changes in the Environment and the Guessed Course of Actions 294

B.1 The Context and The Game

Here we will describe a live demonstration, which is planned to be broadcasted during EU Robotics week 2012. The title of the workshop is:

Workshop on Practical Advancements in Human-Robot Interaction and Social Robotics - EURobotics Week 2012

The title of the demonstration is "*Let's Compete, PR2 Competing with the Humans.*"

The entire demonstration is built upon the scientific components developed in this thesis and their integration with modules of other contributors in the group, such as supervisor system, dialogues module, text to speech generation, 3D remote world reconstruction and visualization system, parsing of the raw output of the planner, etc. (See appendix A for a brief overview of such components). However, the core of the game is basically built around the framework of grounding and explaining changes in the environment, which have been presented in chapter 8.

The game is as follows:

- Two humans (Human 1 and Human 2) are interacting in the living room of our apartment built for the robot in ADREAM building of LAAS-CNRS. It is a typical scenario with furniture and some objects, which could be manipulated by the agents.

- Third human (Human 3) and the robot PR2 are assessing the situation from the aspects of visuo-spatial perspective taking, ability, affordance and effort analyses of all the agents and objects in the scene.

- Then Human 3 and PR2 are blindfolded or asked to turn back.

- Human 1 and Human 2 are asked to make some random changes in the environment, about which Human 3 and PR2 will be oblivious.

- Now we remove the blindfold or ask Human 3 and PR2 to turn back. Then Human 3 and PR2 will be asked to observe the scene and identify the changes if any.

Both of them have to describe the effect of the changes on abilities and affordances. And also they have to explain the changes in terms of the object, agent and the potential course of actions, effort and cooperation if any.

Basically the robot and the human will compete to explain what might have happened when they were blindfolded in terms of:

- *What* has changed physically?
- *What* are the effects of those changes?
- *Who* might have done those changes and *How*?

This demonstration will interestingly conclude by comparing the human participant's and the robot's abilities and responses.

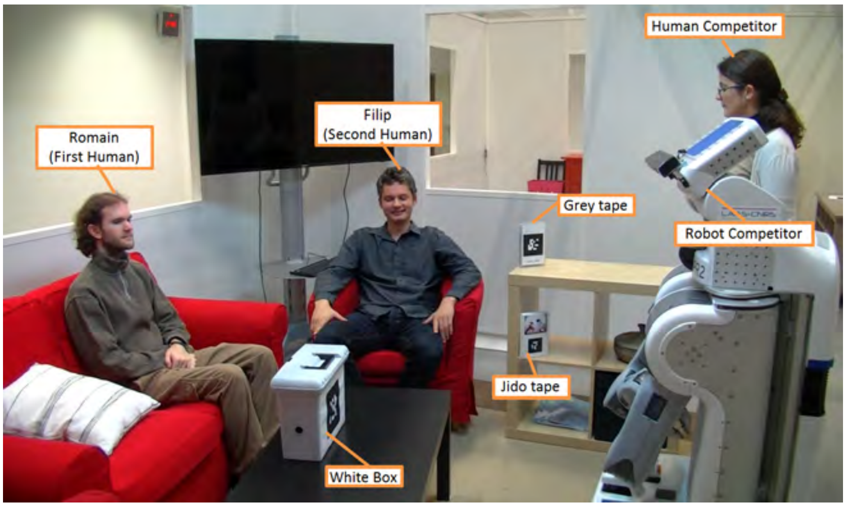
B.2 The Scenario

Below we will illustrate one test run with a human competitor. The humans have already been explained about the game and the type of explanation expected from them, such as physical changes, effect on visuo-spatial abilities and the potential cause or course of actions behind those changes. However, to be interactive and to get desired type of information, the person behind the camera also asks questions when the human participant explains.

Figure B.1 shows the initial scenario to be observed by the human and robot competitors. Then they are asked to look back. Figure B.2 shows the sequence of actions, which other two humans have decided together, to make changes in the environment. Now both the competitors have been asked to look back and observe the environment.

B.3 The Human's and The Robot's Explanations about the Observed Changes in the Environment and the Guessed Course of Actions

Figure B.3(a) shows that the human competitor is explaining the changes and their effects. During the course of interactive explanation, the participant asks whether she could move around to look for one missing object and then she went to look behind the box. Figure B.3(c) shows robot explaining the changes, after the human



(a) Initial scenario, which the human competitor and the robot competitor are observing. First human and second human are the players who will make some changes in the environment.

(b) After observing the initial scene both the competitors are looking back and other two humans are expected to make some changes in the environment by playing with the small objects.

Figure B.1: Human Robot Competition initial scenario.

partner has finished. The screen in this figure shows the 3D model of the environment, which the robot has constructed by observing the changed environment.

Below is the key explanation of the human competitor. Our remarks are put inside parentheses ():

==== Human competitor's explanation about the changes ====

- Grey tape was here, and moved over there . . .

(a)

(b)

(c)

(d)

(e)

(f)

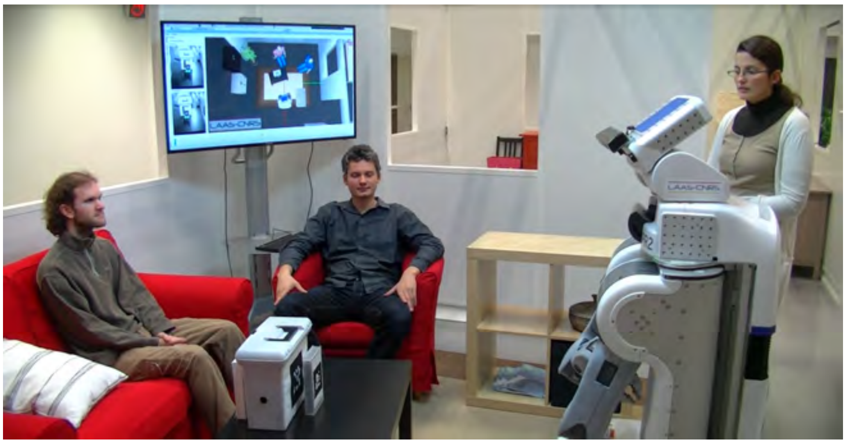
Figure B.2: Sequence of changes made in the environment, which both the humans are deciding together.

- I cannot see Jido tape...it was there but now not there...from here I cannot see it...
- ...and the box did not move...
- Can I move? (We allowed her to move)

  (She moved to look behind the white box on the table...and found the missing tape over there. It means her first guess was that the Jido tape would be behind the box.)
- Gray tape is now not visible to Romain...it is visible for Filip...and it is

(a) After observing the new state, the human competitor begins to explain the changes.

(b) During the interactive explanation, the human competitor asks for the permission to move around and looks for a missing object. Interestingly her first attempt was to look behind the white box.



(c) Now the robot is grounding the changes and explaining them. Robot's perception about the current 3D physical state of the environment has been displayed on the TV screen in the figure.

Figure B.3: Human Robot Competition changed scenario. The robot and the human competitors are now asked to again observe the environment and guess the changes, effects and potential cause.

even reachable by robot I guess...

- The Jido tape is not visible by the robot and for me. . . and I am not sure about Filip...and it is visible and reachable by Romain...
- Probably for both the tapes, Filip took them from here and passed them over to Romain...and he just made the final rotations...

Below is the verbal explanation of the robot about the changes. (Note that this verbal output is generated by parsing and mapping the strings of the raw output through a TCL script to make the verbalization more human understandable. The format of the raw output has been shown in section 8.8.4, where we have discussed our approach of grounding changes.)[1] Our remarks are inside parentheses ().

==== The PR2 robot competitor's explanation about the changes ====

- The Jido tape has moved... I cannot see it anymore...are you playing with me...

  (However, as explained in the chapter 8, the robot also guesses about where the lost object might be hidden. In the current scenario, it found a grounded position and orientation behind the box. As explained in section 8.8.4, such guesses are based on analyzing the *taskability graph* of hiding an object and the assumption that human will place something on the flat horizontal surfaces. However, in this case human seems to be guessing the same position of the lost object, as she first moved to look behind the box.)
- The Grey tape has moved...
- For the Grey tape, first human (Romain) will find it more difficult to see it...
- For the Grey tape, first human can reach it now...
- For the Gray tape, second human (Filip) can see it more easily...
- For the Grey tape, second human cannot reach it anymore...
- For the Grey tape, the robot can see it more easily...
- For the Grey tape...it was picked up by the second human and then it was given to the first human and then it was placed by the first human...
- For the Jido tape... it was picked up by the second human and then it was placed...

  (based on the guessed position of the Jido tape the robot found an explanation for it also)

Note that while explaining for the possible action behind the movement of the Jido tape, the robot's guess was different from the human, because it was based on the efforts in the corresponding affordance graph. However, in reality, Filip has handed-over the object to Romain and then Romain decided to place it somewhere. Hence, it would really be an interesting research challenge to incorporate probability of such behaviors in planning and decision-making.

[1] A high-level supervision script to parse and verbalize the string output of the game reasoner and to execute the robot's behavior has been provided by Lavindra De Silva (LAAS-CNRS).

Appendix C

Publications and Associated Activities

C.1 List of publications

Following publications are related to the research contribution of this thesis:

- **Amit Kumar Pandey** and Rachid Alami, "*Towards Effect-Based Autonomous Understanding of Task Semantics for Human-Robot Interaction*", International Journal of Social Robotics (**IJSR**) (To Appear).
- **Amit Kumar Pandey** and Rachid Alami, "*Mightability: A Multi-State Visuo-Spatial Reasoning for Human-Robot Interaction*" Springer Tracts in Advanced Robotics, Vol. 79, Khatib, Oussama; Kumar, Vijay; Sukhatme, Gaurav (Eds.), (**STAR**) (To Appear)
- Severin Lemaingan, Rachid Alami, **Amit Kumar Pandey**, Matthieu Warnier, Guitton, J, "*Towards Grounding Human-Robot Interaction*", in Bridges between the Methodological and Practical Work of the Robotics and Cognitive Systems Communities, From Sensors to Concepts, Amirat, T.; Chibani, A.; Zarri, G. P. (Eds.), series Intelligent Systems Reference Library, Springer Publishing, (To Appear)
- **Amit Kumar Pandey**, Muhammad Ali and Rachid Alami, "*Towards Task-Aware Proactive-Sociable Robot based on Multi-State Perspective-Taking*", International Journal of Social Robotics (**IJSR**). (Under revised Submission)
- **Amit Kumar Pandey** and Rachid Alami, "*Visuo-Spatial Ability, Effort and Affordance Analyses: Towards Practical Realization of Building Blocks for Robot's Complex Socio-Cognitive Behaviors*", 8th International Workshop on Cognitive Robotics in conjunction with AAAI-2012 (**CogRob-AAAI 2012**).
- **Amit Kumar Pandey**, Jean-Philippe Saut, Daniel Sidobre, Rachid Alami, "*Towards Planning Human-Robot Interactive Manipulation Tasks: Task Dependent and Human Oriented Autonomous Selection of Grasp and Placement*", IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (**BioRob 2012**).
- **Amit Kumar Pandey** and Rachid Alami, "*Towards Task Understanding through Multi-State Visuo-Spatial Perspective Taking for Human-Robot In-*

teraction", IJCAI Workshop on Agents Learning Interactively from Human Teachers (**ALIHT-IJCAI 2011**).

- **Amit Kumar Pandey**, Muhammad Ali, Matthieu Warnier and Rachid Alami, "*Towards Multi-State Visuo-Spatial Reasoning based Proactive Human-Robot Interaction*", 15th International Conference on Advanced Robotics (**ICAR 2011**), (***finalist for the best student paper award***).
- **Amit Kumar Pandey** and Rachid Alami, "*Mightability: A Multi-State Visuo-Spatial Reasoning for Human-Robot Interaction*", 12th International Symposium on Experimental Robotics **(ISER 2010)**
- **Amit Kumar Pandey** and Rachid Alami, "*Mightability Maps: A Perceptual Level Decisional Framework for Co-operative and Competitive Human-Robot Interaction*", IEEE/RSJ International Conference on Intelligent Robots and Systems (**IROS 2010**).
- **Amit Kumar Pandey** and Rachid Alami, "*A Framework towards a Socially Aware Mobile Robot Motion in Human-Centered Dynamic Environment*", IEEE/RSJ International Conference on Intelligent Robots and Systems (**IROS 2010**).
- Samir Alili, **Amit Kumar Pandey**, E. Akin Sisbot and Rachid Alami, "*Interleaving Symbolic and Geometric Reasoning for a Robotic Assistant*", ICAPS Workshop on Combining Action and Motion Planning (**CAMP-ICAPS 2010**).
- Raquel Ros, E. Akin Sisbot, Severin Lemaingan, **Amit Kumar Pandey** and Rachid Alami, "*Robot, tell me what you know about...?: Expressing robot's knowledge through interaction*", ICRA Workshop on Interactive Communication for Autonomous Intelligent Robots (**ICAIR-ICRA 2010**).
- **Amit Kumar Pandey** and Rachid Alami, "*A Framework for Adapting Social Conventions in a Mobile Robot Motion in Human-Centered Environment*", 14th International Conference on Advanced Robotics (**ICAR 2009**).
- **Amit Kumar Pandey** and Rachid Alami, "*A Step towards a Sociable Robot Guide which Monitors and Adapts to the Person's Activities*", 14th International Conference on Advanced Robotics (**ICAR 2009**).
- Luis F. Marin-Urias, Emrah Akin Sisbot, **Amit Kumar Pandey**, Riichiro Tadakuma and Rachid Alami, "*Towards Shared Attention through Geometric Reasoning for Human Robot Interaction*", IEEE-RAS International Conference on Humanoid Robots (**Humanoids 2009**).

C.2 Associated EU Projects

Following are the EU projects, where the thesis has actively contributed:

- SAPHARI (*Safe and Autonomous Physical Human-Aware Robot Interaction*) funded by European Community's 7th Framework Programme (http://www.saphari.eu/)
- CHRIS (*Cooperative Human Robot Interaction Systems*) funded by the E.C. Division FP7-IST. (http://www.chrisfp7.eu/)
- DEXMART (DEXterous and autonomous dual-arm/hand robotic manipulation with sMART sensory-motor skills: A bridge from natural to artificial cognition) funded under the European Community's 7th Framework Program. (http://www.dexmart.eu/)
- URUS (Ubiquitous Networking Robotics in Urban Settings) funded by the E.C. Division FP6-IST. (http://urus.upc.es/)

C.3 Associated Scientific Gathering Activities

Associated with the scientific contribution of the thesis, following are the workshops we have organized and contributed in partnership with other contributors in the scientific community:

- We are organizing a full day workshop titled **"Workshop on Practical Advancements in Human-Robot Interaction and Social Robotics"** during the EURobotics Week 2012.

  The idea is to broadcast the live demonstrations through EU platform throughout the Europe and worldwide as an attempt to enrich the general people's awareness about the domestic and service robots. The proposed demonstrations include the system developed within the thesis as well as the demonstrations integrating the developed system with the components of other contributors of our group, such as ontology, speech, supervision, etc.
- We have organized a two days workshop at LAAS-CNRS in connection with EU FP7 SAPHARI project (http://www.saphari.eu/) and presented **"Human- and Task- aware reasoning on Grasp and Placement for Basic HRI tasks."** May 29-30, 2012.
- We have contributed to a three days LAAS-TUM (Technical University Munich) joint workshop and given a talk on **"Mightability Analysis: A Multi-State Visuo-Spatial Perspective Taking for Human-Robot Interaction (HRI) and Applications."**July 11-13 2011.

Do we expect the existence of "*Robot Being*"? Will we accept that? ...

Index

3D world representation, 64
  perception, 290

abilities of agent, 46
affordance analysis, 17
  categories, 88
Affordance Graph, 185
  applications, 189
    generating shared cooperative plans, 190
    grounding changes, 198
    grounding interaction, 190
    supporting symbolic planners, 202
  computation time, 188
  construction rules, 186
agent state, 41
  hand in manipulation mode, 98
  hand in rest mode, 98
  hand mode, 44
  hand occupancy state, 43
  mental state, 45
  motion status, 44, 102
  physical state, 43
  posture, 43
agent-agent affordances, 91
  considering object dimension, 96
  least feasible effort based, 96
  Mightability based framework, 92

basic HRI tasks planning
  framework, 155
  give object, 159, 162
  grasp set, 151
  grasp-placement inter-dependency, 150
  hide object, 160, 166
  make object accessible, 159, 163
  object alignment constraint, 153
  placement, 150
  placement orientation, 151, 152
  show object, 159, 161
  simultaneous dual grasp, 153
  wrist alignment constraint, 154
bottom up social embodiment, 11

Causality of environmental changes, 40

effort analysis, 69
  factors deciding effort levels, 179
  in cooperative task planning, 178
  in grounding, 178
environment topology extraction, 109

fact variable, 40

geometric-symbolic task planning, 31
  geometric backtracking, 31
  geometric planner constraints' types, 203
  geometric planner layers, 202
  geometric planner task level back-tracking, 207
  two-way handshaking, 31, 202
geometrical state, 42
grasp placement interdependency, 6
grounded fact variable, 47
grounding
  changes, 198
  incorporating effort, 178
  interaction, 28, 190

HRI
  action definition, 48
  domain theory, 41
  environmental attributes, 41
  Environmental changes, 40
  generalized theory, 39
HRI environment state, 47
HRI environmental changes, 47
HRI environmental state space, 47
human-aware effort analysis, 70

arm effort, 70
displacement effort, 70
head effort, 70
torso effort, 70
whole body effort, 70
human-aware effort hierarchy, 71

Interesting Boundary Line (IBL), 110
Interesting Cell (IC), 110

learning task
confusing predicate, 260
domain theory, 256
effect based, 35
emulation, 35, 248
explanation based, 255
goal concept, 256
irrelevant predicate, 259
m-estimate based refinement, 257
predicate consistency factor, 259
relevant predicate, 260
semantics
applications, 274
comparative facts, 250
give object, 268
hide object, 265
hide object away, 270
hierarchical knowledge base, 249
make object accessible, 267
put object away, 269
qualitative facts, 251
quantitative facts, 249
show object, 262
semantics, challenges, 36
sub-action based, 34
Least Effort Analysis, 83
least feasible effort state, 83
local clearance, 110

m-estimate, 257
Manava robot, 1
Manipulability Graph, 183
Mightability Analysis, 72
estimation, 73
Mightability Maps, 76
milestone, 113

Object Flow Graph, 187
object manipulation, 25
Object Oriented Mightabilities, 79
object state, 42
closed inside, 103, 104
effect of, 104
covered by, 103, 104
effect, 104
enclosed by, 103, 104
effect, 104
lying inside, 103, 104
effect of, 104
obstructing object, 68
occluding object, 67

Partial Plan, 52
performing-agent, 256
pick-and-place tasks, 149
proactive action, 213
desired effects, 224
instantiation, 220
framework, 225
level-1, 215
level-2, 216
level-3, 217
level-4, 218
proactive reach out, 224
proactive suggestion, 224
spaces for synthesis, 213
user studies, 236
Proactive Behavior (PB) vs. Non-Proactive Behavior (NPB), 236
proactivity, 32
Proxemics relation, 42

sociable robot motion, 22
social bias of navigation, 7
social forces of navigation, 7
Social Intelligence Embodiment Pyramid, 8
social learning, 5
social robot guide, 131
desirable behavior, 132

goal-oriented re-engagement effort, 135
human activity analysis
leave taking, 135
non-leave taking, 133
Socially Intelligent Robot, 8
Socially-Aware Path
behavior analysis
with humans, 123
with other paths, 122
with reactive behavior, 129
Socially-Aware Path Planner, 109
clearance constraints (C-rules), 113
dealing with
dynamic humans, 116
group of people, 117
obstacles, 116
framework, 117
proximity guidelines (P-rules), 112
selective adaptation of rules, 113
social conventions (S-rules), 111
Spatial relation, 42
system architecture, 290

target-agent, 256
target-object, 256
Taskability Graph, 180

ungrounded fact variable, 47

visuo-spatial ability graph, 85
visuo-spatial perspective taking, 4, 15, 65
invisible, 63
obstructed, 63
occluded, 63
reachable, 63
unreachable, 63
visible, 63
Voronoi Diagram, 109

Bibliography

[Adorno 2011] B.V. Adorno, A.P.L. Bo, P. Fraisse and P. Poignet. *Towards a co-operative framework for interactive manipulation involving a human and a humanoid*. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 3777 –3783, may 2011. (Cited on page 26.)

[Agostini 2011] Alejandro Agostini, Carme Torras and Florentin Wörgötter. *Integrating task planning and interactive learning for robots to work in human environments*. In Proceedings of the 22nd International Conference on Artificial Intelligence (IJCAI), pages 2386–2391, 2011. (Cited on pages 60 and 257.)

[Alami 1990] Rachid Alami, Thierry Simeon and Jean-Paul Laumond. *A geometrical approach to planning manipulation tasks. The case of discrete placements and grasps*. In The fifth international symposium on Robotics research, pages 453–463, Cambridge, MA, USA, 1990. MIT Press. (Cited on page 31.)

[Alili 2009] Samir Alili, Rachid Alami and Vincent Montreuil. *A Task Planner for an Autonomous Social Robot*. In Hajime Asama, Haruhisa Kurokawa, Jun Ota and Kosuke Sekiyama, (eds.), Distributed Autonomous Robotic Systems 8, pages 335–344. Springer Berlin Heidelberg, 2009. (Cited on pages 18, 49, 92, 96, 177, 202, 205, 228, 275, 277 and 290.)

[Althaus 2004] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita and H.I. Christensen. *Navigation for human-robot interaction tasks*. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 1894 – 1900, 2004. (Cited on page 22.)

[Ansuini 2006] C. Ansuini, M. Santello, S. Massaccesi and U. Castiello. *Effects of End-Goal on Hand Shaping*. Neurophysiology, vol. 95, pages 2456–2465, Dec. 2006. (Cited on page 6.)

[Arechavaleta 2008] Gustavo Arechavaleta, Jean-Paul Laumond, Halim Hicheur and Alain Berthoz. *On the nonholonomic nature of human locomotion*. Autonomous Robots, vol. 25, pages 25–35, 2008. (Cited on pages 21, 116 and 138.)

[Argall 2009] Brenna D. Argall, Sonia Chernova, Manuela Veloso and Brett Browning. *A survey of robot learning from demonstration*. Robotics and Autonomous Systems, vol. 57, no. 5, pages 469 – 483, 2009. (Cited on page 34.)

[Batson 2003] C. Daniel Batson and Adam A. Powell. *Altruism and Prosocial Behavior*. In Handbook of Psychology. John Wiley & Sons, Inc., 2003. (Cited on page 9.)

[Berenson 2008] D. Berenson, J. Kuffner and H. Choset. *An optimization approach to planning for mobile manipulation*. In Proceedings of IEEE International

Conference on Robotics and Automation (ICRA), pages 1187–1192, may 2008. (Cited on pages 26 and 148.)

[Bischoff 1999] R. Bischoff and T. Jain. *Natural communication and interaction with humanoid robots.* 1999. (Cited on pages 26, 28 and 149.)

[Bobrow 1991] Daniel G. Bobrow. *Dimensions of interaction: a shift of perspective in artificial intelligence.* AI Mag., vol. 12, no. 3, pages 64–80, September 1991. (Cited on pages 3 and 334.)

[Bolstad 2001] C A Bolstad. *Situation awareness: Does it change with age.* In Human Factors and Ergonomics Society, vol. 45, pages 272–276. Human Factors and Ergonomics Society, 2001. (Cited on page 18.)

[Bootsma 1992] R. J. Bootsma, F. C. Bakker, F. J. Van Snippenberg and C. W. Tdlohreg. *The Effects of Anxiety on Perceiving the Reachability of Passing Objects.* Ecological Psychology, vol. 4, no. 1, pages 1–16, 1992. (Cited on page 67.)

[Breazeal 2002] Cynthia Breazeal. Designing sociable robots. MIT Press, Cambridge, MA, USA, 2002. (Cited on page 9.)

[Breazeal 2003] Cynthia Breazeal. *Toward sociable robots.* Robotics and Autonomous Systems, vol. 42, no. 3-4, pages 167 – 175, 2003. (Cited on pages 3 and 334.)

[Breazeal 2006] Cynthia Breazeal, Matt Berlin, Andrew G. Brooks, Jesse Gray and Andrea Lockerd Thomaz. *Using perspective taking to learn from ambiguous demonstrations.* Robotics and Autonomous Systems, vol. 54, pages 385–393, 2006. (Cited on page 15.)

[Breazeal 2009] Cynthia Breazeal. *Role of expressive behaviour for robots that learn from people.* Philosophical Transactions of the Royal Society B-Biological Sciences, vol. 364, pages 3527–3538, 2009. (Cited on page 223.)

[Broquère 2010] X. Broquère, D. Sidobre and Khoi Nguyen. *From motion planning to trajectory control with bounded jerk for service manipulator robots.* In IEEE International Conference on Robotics and Automation (ICRA), pages 4505–4510, 2010. (Cited on page 151.)

[Bruce 2002] Allison Bruce, Illah Nourbakhsh and Reid Simmons. *The Role of Expressiveness and Attention in Human-Robot Interaction.* 2002. (Cited on page 9.)

[Buss 2011] M. Buss, D. Carton, B. Gonsior, K. Kuehnlenz, C. Landsiedel, N. Mitsou, R. de Nijs, J. Zlotowski, S. Sosnowski, E. Strasser, M. Tscheligi, A. Weiss and D. Wollherr. *Towards proactive human-robot interaction in human environments.* In 2nd International Conference on Cognitive Infocommunications (CogInfoCom), pages 1 –6, july 2011. (Cited on page 33.)

[Cakmak 2011] M. Cakmak, S.S. Srinivasa, Min Kyung Lee, J. Forlizzi and S. Kiesler. *Human preferences for robot-human hand-over configurations*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1986 –1993, sept. 2011. (Cited on pages 26, 28, 149 and 226.)

[Calinon 2009] S. Calinon, F. Dhalluin, D.G. Caldwell and A.G. Billard. *Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework*. In Proceedings of 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pages 582 –588, dec. 2009. (Cited on page 34.)

[Cambon 2004] S. Cambon, F. Gravot, and R. Alami. *A robot task planner that merges symbolic and geometric reasoning*. In 16th European Conference on Artificial Intelligence, pages 895–899, 2004. (Cited on page 202.)

[Cambon 2009] Stéphane Cambon, Rachid Alami and Fabien Gravot. *A Hybrid Approach to Intricate Motion, Manipulation and Task Planning*. International Journal of Robotics Research, vol. 28, no. 1, pages 104–126, January 2009. (Cited on pages 31, 202 and 277.)

[Cantrell 2011] R. Cantrell, P. Schermerhorn and M. Scheutz. *Learning actions from human-robot dialogues*. In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 125 –130, 2011. (Cited on page 35.)

[Carello 1989] Claudia Carello, Alexis Grosofsky, Francene D. Reichel, H. Yosef Solomon and M.T. Turvey. *Visually Perceiving What is Reachable*. Ecological Psychology, vol. 1, no. 1, pages 27–54, 1989. (Cited on page 67.)

[Carlson 2008] T. Carlson and Y. Demiris. *Human-wheelchair collaboration through prediction of intention and adaptive assistance*. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 3926 –3931, may 2008. (Cited on page 33.)

[Caron 2002] Albert J. Caron, Elizabeth J. Kiel, Millie Dayton and Samantha C. Butler. *Comprehension of the Referential Intent of Looking and Pointing Between 12 and 15 Months*. Journal of Cognition and Development, vol. 3, no. 4, pages 445–464, 2002. (Cited on pages 4 and 336.)

[Carpendale 2006] Jeremy Carpendale and Charlie Lewis. How children develop social understanding. Wiley-Blackwell, 2006. (Cited on pages 4 and 336.)

[Carpenter 2002] Malinda Carpenter and Josep Call. *The chemistry of social learning*. Developmental Science, vol. 5, no. 1, pages 22–24, 2002. (Cited on page 5.)

[Cesta 2007] Amedeo Cesta, Gabriella Cortellessa, Federico Pecora and Riccardo Rasconi. *Supporting Interaction in the RoboCare Intelligent Assistive Environment*. In Proceedings of AAAI Spring Symposium on Interaction

Challenges for Intelligent Assistants, pages 18–25. AAAI, 2007. (Cited on page 33.)

[Cestnik 1990] Bojan Cestnik. *Estimating probabilities: A crucial task in machine learning*. In Proceedings of the Ninth European Conference on Artificial Intelligence, ECAI, pages 147–149, 1990. (Cited on page 258.)

[Chella 2006] A. Chella, H. Dindo and I. Infantino. *A cognitive framework for imitation learning*. Robotics and Autonomous Systems, vol. 54, no. 5, pages 403 – 408, 2006. (Cited on page 34.)

[Choi 2004] Hyeg Joo Choi and Leonard S. Mark. *Scaling affordances for human reach actions*. Human Movement Science, vol. 23, no. 6, pages 785 – 806, 2004. (Cited on pages 69, 70 and 338.)

[Clark 2003] Herbert H. Clark. *Pointing and placing*. In S. Kita, (ed.), Pointing: Where language, culture, and cognition meet, pages 243–268. Hillsdale NJ: Erlbaum, 2003. (Cited on page 240.)

[Clodic 2006] A. Clodic, S. Fleury, R. Alami, R. Chatila, G. Bailly, L. Brethes, M. Cottret, P. Danes, X. Dollat, F. Elisei, I. Ferrane, M. Herrb, G. Infantes, C. Lemaire, F. Lerasle, J. Manhes, P. Marcoul, P. Menezes and V. Montreuil. *Rackham: An Interactive Robot-Guide*. In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 502 –509, sept. 2006. (Cited on pages 22 and 24.)

[Clodic 2009] Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami and Raja Chatila. *SHARY: A Supervision System Adapted to Human-Robot Interaction*. In Oussama Khatib, Vijay Kumar and George Pappas, (eds.), 11th International Symposium on Experimental Robotics Experimental Robotics, vol. 54 of *Springer Tracts in Advanced Robotics*, pages 229–238. Springer Berlin / Heidelberg, 2009. (Cited on pages 18, 19, 96, 101, 102, 106, 188, 196, 228, 290 and 291.)

[Coradeschi 2003] Silvia Coradeschi and Alessandro Saffiotti. *An introduction to the anchoring problem*. Robotics and Autonomous Systems, vol. 43, pages 85–96, 2003. (Cited on page 28.)

[Cramer 2009] Henriette S.M. Cramer, Nicander A. Kemper, Alia Amin and Vanessa Evers. *The effects of robot touch and proactive behaviour on perceptions of human-robot interactions*. In Proceedings of the 4th ACM/IEEE international conference on Human Robot Interaction (HRI), pages 275–276, New York, NY, USA, 2009. ACM. (Cited on page 32.)

[Şahin 2007] Erol Şahin, Maya Çakmak, Mehmet R. Doğar, Emre Uğur and Göktürk Üçoluk. *To Afford or Not to Afford: A New Formalization of Affordances Toward Affordance-Based Robot Control*. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems, vol. 15, no. 4, pages 447–472, December 2007. (Cited on page 17.)

[Csibra 2008] Gergely Csibra and Agnes Volein. *Infants can infer the presence of hidden objects from referential gaze information.* British Journal of Developmental Psychology, vol. 26, no. 1, pages 1–11, 2008. (Cited on pages 4 and 336.)

[Dautenhahn 2007] K. Dautenhahn. *Socially intelligent robots: dimensions of human–robot interaction.* Philosophical Transactions of the Royal Society B: Biological Sciences, vol. 362, no. 1480, pages 679–704, 2007. (Cited on page 33.)

[Deak 2000] G. O. Deak, R. A. Flom and A D Pick. *Effects of gesture and target on 12- and 18-month-olds' joint visual attention to objects in front of or behind them.* Developmental Psychology, vol. 36, no. 4, pages 511–523, 2000. (Cited on pages 4 and 336.)

[Dehais 2011] F. Dehais, E. A. Sisbot, R. Alami and M. Causse. *Physiological and subjective evaluation of a human-robot object hand-over task.* Applied Ergonomics, vol. 42, no. 6, pages 785–792, 2011. (Cited on page 226.)

[Dejong 1986] Gerald Dejong and Raymond Mooney. *Explanation-based learning: An alternative view.* Machine Learning, vol. 1, pages 145–176, 1986. (Cited on page 255.)

[dictionary.reference.com] dictionary.reference.com. *Definition of Social.* `http://dictionary.reference.com/browse/social`. (Cited on pages 3 and 334.)

[Diftler 2004] M.A. Diftler, R.O. Ambrose, K.S. Tyree, S.M. Goza and E.L. Huber. *A mobile autonomous humanoid assistant.* In 4th IEEE/RAS International Conference on Humanoid Robots, pages 133–148, nov. 2004. (Cited on page 26.)

[Dillmann 2004] Rudiger Dillmann. *Teaching and learning of robot tasks via observation of human performance.* Robotics and Autonomous Systems, vol. 47, pages 109 – 116, 2004. (Cited on page 34.)

[Dragan 2011] A. Dragan, G. Gordon and S. Srinivasa. *Learning from Experience in Manipulation Planning: Setting the Right Goals.* In Proceedings of International Symposium on Robotics Research (ISRR), August 2011. (Cited on page 34.)

[Dunphy-Lelii 2004] Sarah Dunphy-Lelii and Henry M Wellman. *Infants' understanding of occlusion of others' line-of-sight: Implications for an emerging theory of mind.* European Journal of Developmental Psychology, vol. 1, no. 1, pages 49–66, 2004. (Cited on pages 4 and 336.)

[Duong 2005] T.V. Duong, H.H. Bui, D.Q. Phung and S. Venkatesh. *Activity recognition and abnormality detection with the switching hidden semi-Markov model.* In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pages 838 – 845, june 2005. (Cited on page 33.)

[Edsinger 2007] A. Edsinger and C.C. Kemp. *Human-Robot Interaction for Cooperative Manipulation: Handing Objects to One Another*. In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 1167–1172, aug. 2007. (Cited on page 26.)

[Eisenberg 1998] N. Eisenberg and R. A. Fabes. *Prosocial development*. Handbook of child psychology: Social, emotional, and personality development, vol. 3, pages 701–778, 1998. (Cited on page 5.)

[Eisenberg 2007] Nancy Eisenberg, Richard A. Fabes and Tracy L. Spinrad. *Prosocial Development*. In Handbook of Child Psychology. John Wiley & Sons, Inc., 2007. (Cited on page 9.)

[Ekvall 2008] S. Ekvall and D. Kragic. *Robot learning from demonstration: a task-level planning approach*. International Journal of Advanced Robotic Systems, vol. 5, no. 3, pages 223–234, 2008. (Cited on page 35.)

[Endsley 2000] M. R. Endsley. *Theoretical underpinnings of situation awareness: a critical review*. In M. R. Endsley and D. J. Garland, (eds.), Situation Awareness Analysis and Measurement. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2000. (Cited on pages 18 and 56.)

[Flanagan 2003] J. Randall Flanagan and Roland S. Johansson. *Action plans used in action observation*. Nature, no. 6950, page 769–771, 2003. (Cited on page 244.)

[Flann 1989] Nicholas S. Flann, Dietterich and Thomas G. *A Study of Explanation-Based Methods for Inductive Learning*. Machine Learning, vol. 4, pages 187–226, 1989. (Cited on pages 255 and 358.)

[Flavell 1977] J. H. Flavell. *The development of knowledge about visual perception*. In The Nebraska Symposium on Motivation, vol. 25, pages 43–76, 1977. (Cited on pages 4 and 336.)

[Flavell 1978] John H. Flavell, Susan G. Shipstead and Karen Croft. *Young Children's Knowledge about Visual Perception: Hiding Objects from Others*. Child Development, vol. 49, no. 4, pages 1208–1211, 1978. (Cited on pages 4, 5 and 336.)

[Flavell 1981] J. H. Flavell, B. A. Everett, K. Croft and E. R. Flavell. *Young children's knowledge about visual perception: Further evidence for the Level 1 - Level 2 distinction*. Developmental Psychology, vol. 17, no. 1, pages 99–103, 1981. (Cited on pages 4 and 336.)

[Fleury 1997] Sara Fleury, Matthieu Herrb and Raja Chatila. *GenoM: A Tool for the Specification and the Implementation of Operating Modules in a Distributed Robot Architecture*. In International Conference on Intelligent Robots and Systems (IROS), pages 842–848, 1997. (Cited on page 290.)

[Fong 2003] T. Fong, I. Nourbakhsh and K. Dautenhahn. *A Survey of Socially Interactive Robots.* Robotics and Autonomous Systems, vol. 42, no. 3-4, 2003. (Cited on pages 3, 9 and 334.)

[Fraichard 2007] T. Fraichard. *A Short Paper about Motion Safety.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 1140 –1145, april 2007. (Cited on page 20.)

[Friedman 2007] Stephen Friedman, Hanna Pasula and Dieter Fox. *Voronoi random fields: extracting the topological structure of indoor environments via place labeling.* In Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI), pages 2109–2114, 2007. (Cited on page 109.)

[Furnkranz 2003] J. Furnkranz and P.A. Flach. *An analysis of rule evaluation metrics.* In Proceedings of 20th International Conference on Machine Learning (ICML), pages 202–209. AAAI Press, January 2003. (Cited on page 257.)

[Gardner 2001] Douglas L. Gardner, Leonard S. Mark, John A. Ward and Heather Edkins. *How Do Task Characteristics Affect the Transitions Between Seated and Standing Reaches?* Ecological Psychology, vol. 13, no. 4, pages 245–274, 2001. (Cited on pages 69, 70 and 338.)

[Garrido 2006] S. Garrido, L. Moreno and D. Blanco. *Voronoi diagram and fast marching applied to path planning.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 3049 –3054, may 2006. (Cited on page 122.)

[Ghallab 2004] Malik Ghallab, Dana Nau and Paolo Traverso. Automated planning: theory and practice. Elsevier, Burlington, MA, 2004. (Cited on pages 40, 41, 48, 51, 52, 54, 58 and 60.)

[Gharbi 2008] M. Gharbi, J. Cortes and T. Simeon. *A sampling-based path planner for dual-arm manipulation.* In IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 383–388, july 2008. (Cited on pages 157 and 231.)

[Gibson 1986] James J. Gibson. *The Theory of Affordances.* In The Ecological Approach To Visual Perception, pages 127–143. Psychology Press, September 1986. (Cited on pages 17 and 45.)

[Gibson 2000] Eleanor J Gibson. *Perceptual Learning in Development: Some Basic Concepts.* Ecological Psychology, vol. 12, no. 4, pages 295–302, 2000. (Cited on page 17.)

[Gockley 2007] Rachel Gockley, Jodi Forlizzi and Reid Simmons. *Natural person-following behavior for social robots.* In Proceedings of the ACM/IEEE international conference on Human-Robot Interaction (HRI), pages 17–24, New York, NY, USA, 2007. ACM. (Cited on page 20.)

[Goodrich 2007] Michael A. Goodrich and Alan C. Schultz. *Human-robot interaction: a survey.* Foundations and Trends in Human–Computer Interaction, vol. 1, no. 3, pages 203–275, January 2007. (Cited on pages 3 and 39.)

[Gredeback 2010] Gustaf Gredeback and Olga Kochukhova. *Goal anticipation during action observation is influenced by synonymous action capabilities, a puzzling developmental study.* Experimental Brain Research, vol. 202, pages 493–497, 2010. (Cited on pages 223 and 240.)

[Gribovskaya 2011] E. Gribovskaya, S.M. Khansari-Zadeh and A. Billard. *Learning Non-linear Multivariate Dynamics of Motion in Robotic Manipulators.* International Journal of Robotics Research, vol. 30, no. 1, pages 80–117, January 2011. (Cited on page 34.)

[Gspandl 2011] S. Gspandl, I. Pill, M. Reip, G. Steinbauer and A. Ferrein. *Belief Management for High-Level Robot Programs.* In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI), pages 900–905, 2011. (Cited on page 45.)

[Guan 2006] Yisheng Guan and K. Yokoi. *Reachable Space Generation of A Humanoid Robot Using The Monte Carlo Method.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1984 –1989, oct. 2006. (Cited on page 15.)

[Guilamo 2005] L. Guilamo, J. Kuffner, K. Nishiwaki and S. Kagami. *Efficient prioritized inverse kinematic solutions for redundant manipulators.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3921 – 3926, aug. 2005. (Cited on page 15.)

[Gulati 2008] S. Gulati and B. Kuipers. *High performance control for graceful motion of an intelligent wheelchair.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 3932 –3938, may 2008. (Cited on page 24.)

[Hafner 2008] Verena V. Hafner and Frédéric Kaplan. *Interpersonal maps: how to map affordances for interaction behaviour.* In International conference on Towards affordance-based robot control, pages 1–15, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 18.)

[Hall 1966] Edward T. Hall. The hidden dimension. Garden City, N.Y., 1966. (Cited on pages 7, 42 and 112.)

[Harnad 1990] Stevan Harnad. *The symbol grounding problem.* Phys. D, vol. 42, no. 1-3, pages 335–346, June 1990. (Cited on page 28.)

[Helbing 1991] Dirk Helbing. *A Mathematical Model for the Behavior of Pedestrians.* Behavioral Science, vol. 36, pages 298–310, 1991. (Cited on pages 7 and 22.)

[Helbing 1995] Dirk Helbing. *Social force model for pedestrian dynamics.* Physical Review E, vol. 51, no. 5, pages 4282–4286, 1995. (Cited on pages 7, 8 and 22.)

[Hoeller 2007] F. Hoeller, D. Schulz, M. Moors and F.E. Schneider. *Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1260 –1265, 2007. (Cited on page 20.)

[Hoffman 2010] G. Hoffman. *Anticipation in Human-Robot Interaction*. In AAAI Spring Symposium Series, 2010. (Cited on page 33.)

[Hoffmann 2003] Jorg Hoffmann. *The Metric-FF planning system: Translating "ignoring delete lists" to numerical state variables*. Journal of Artificial Intelligence Research. Special issue on the 3rd International Planning Competition, vol. 20, pages 291–341, 2003. (Cited on page 31.)

[Holthaus 2010] Patrick Holthaus, Ingo Lutkebohle, Marc Hanheide and Sven Wachsmuth. *Can I Help You?* In Proceedings of International Conference on Social Robotics, pages 325–334. Springer Berlin / Heidelberg, 2010. (Cited on page 223.)

[Holthaus 2011] Patrick Holthaus, Karola Pitsch and Sven Wachsmuth. *How Can I Help?* International Journal of Social Robotics, vol. 3, pages 383–393, 2011. (Cited on page 222.)

[Hoogendoorn 2011] Lambalgen R. M. v. Hoogendoorn M. and J. Treur. *Modeling Situation Awareness in Human-Like Agents Using Mental Models*. In International Joint Conference on Artificial Intelligence(IJCAI), pages 1697–1704, 2011. (Cited on page 45.)

[Huang 2010] Kuo-Chen Huang, Jiun-Yi Li and Li-Chen Fu. *Human-oriented navigation for service providing in home environment*. In SICE Annual Conference, pages 1892 –1897, aug. 2010. (Cited on page 21.)

[Huber 2009] M. Huber, A. Knoll, T. Brandt and S. Glasauer. *Handing Over a Cube*. Annals of the New York Academy of Sciences, Basic and Clinical Aspects of Vertigo and Dizziness, vol. 1164, pages 380–382, 2009. (Cited on page 226.)

[Iio 2011] Takamasa Iio, Masahiro Shiomi, Kazuhiko Shinozawa, Takaaki Akimoto, Katsunori Shimohara and Norihiro Hagita. *Investigating Entrainment of People's Pointing Gestures by Robot's Gestures Using a WOZ Method*. International Journal of Social Robotics, vol. 3, pages 405–414, 2011. (Cited on page 223.)

[Imai 2003] M. Imai, T. Ono and H. Ishiguro. *Physical relation and expression: joint attention for human-robot interaction*. IEEE Transactions on Industrial Electronics, vol. 50, no. 4, pages 636 – 643, aug. 2003. (Cited on page 223.)

[Ingrand] Félix Ingrand. *OpenPRS is an open source version of PRS (Procedural Reasoning Systems)*. `http://www.openrobots.org/wiki/openprs/`. (Cited on page 290.)

[Janis 1996] Irving L. Janis and Leon Mann. *Satisficing.* The effective manager: Perspectives and illustrations, pages 157–159, 1996. (Cited on pages 157 and 347.)

[Johnson 2005] M. Johnson and Y. Demiris. *Perceptual Perspective Taking And Action Recognition.* International Journal of Social Robotics, vol. 2, pages 181–199, 2005. (Cited on page 15.)

[Jordan 2008] J. S. Jordan and M. Hunsinger. *Learned patterns of action effect anticipation contribute to the spatial displacement of continuously moving stimuli.* Journal of Experimental Psychology: Human Perception and Performance, pages 113–124, 2008. (Cited on page 223.)

[Kakas 2011] Antonis Kakas, Loizos Michael and Rob Miller. *Modular-E and the role of elaboration tolerance in solving the qualification problem.* Artificial Intelligence, vol. 175, no. 1, pages 49 – 78, 2011. John McCarthy's Legacy. (Cited on page 40.)

[Khatib 2009] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier and S. Delp. *Robotics-based synthesis of human motion.* Journal of Physiology-Paris, vol. 103, pages 211 – 219, 2009. (Cited on pages 71, 72 and 251.)

[Kim 2004] J. Kim, J. Park, Y. Hwang and M. Lee. *Advanced Grasp Planning for Handover Operation Between Human and Robot: Three Handover Methods in Esteem Etiquettes Using Dual Arms and Hands of Home- Service Robot.* pages 34–39, 2004. (Cited on pages 26 and 148.)

[Kirby 2007] Rachel Kirby, Jodi Forlizzi and Reid Simmons. *Natural Person-Following Behavior for Social Robots.* In Proceedings of Human-Robot Interaction, pages 17–24, March 2007. (Cited on page 21.)

[Kirby 2009] Rachel Kirby, Reid Simmons and Jodi Forlizzi. *COMPANION: A Constraint Optimizing Method for Person-Acceptable Navigation.* In IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 607–612, September 2009. (Cited on pages 20, 21 and 22.)

[Knickerbocker 2003] Roberta L. Knickerbocker. *Prosocial Behavior.* 2003. (Cited on page 9.)

[Kockler 2010] H. Kockler, L. Scheef, R. Tepest, N. David, B.H. Bewernick, A. Newen, H.H. Schild, M. May and K. Vogeley. *Visuospatial perspective taking in a dynamic environment: Perceiving moving objects from a first-person-perspective induces a disposition to act.* Consciousness and Cognition, vol. 19, no. 3, pages 690 – 701, 2010. (Cited on page 223.)

[Kopp 2004] Stefan Kopp and Ipke Wachsmuth. *Synthesizing multimodal utterances for conversational agents: Research Articles.* Computer Animation and Virtual Worlds, vol. 15, no. 1, pages 39–52, March 2004. (Cited on page 222.)

[Kozima 2001] H. Kozima and H. Yano. *In search of otogenetic prerequisites for embodied social intelligence.* In Proceedings of the Workshop on Emergence and Development on Embodied Cognition (EDEC-2001); International Conference on Cognitive Science (ICCS-2001), pages 30–34, 2001. (Cited on pages 223 and 226.)

[Krishna 2006] K. Madhava Krishna, R. Alami and T. Simeon. *Safe proactive plans and their execution.* Robotics and Autonomous Systems, vol. 54, no. 3, pages 244 – 255, 2006. (Cited on page 23.)

[Kruse 2010] Thibault Kruse, Alexandra Kirsch, E. Akin Sisbot and Rachid Alami. *Exploiting Human Cooperation in Human-Centered Robot Navigation.* In IEEE International Symposium in Robot and Human Interactive Communication (RO-MAN), pages 192 –197, 2010. (Cited on pages 20 and 22.)

[Kulyukin 2006] Vladimir Kulyukin, Chaitanya Gharpure, John Nicholson and Grayson Osborne. *Robot-assisted wayfinding for the visually impaired in structured indoor environments.* Autonomous Robots, vol. 21, pages 29–41, 2006. (Cited on page 24.)

[Kuniyoshi 1994] Y. Kuniyoshi, M. Inaba and H. Inoue. *Learning by watching: extracting reusable task knowledge from visual observation of human performance.* IEEE Transactions on Robotics and Automation, vol. 10, no. 6, pages 799 –822, dec 1994. (Cited on page 34.)

[Kwon 2010] W. Y. Kwon and I. H. Suh. *Probabilistic Temporal Prediction for Proactive Action Selection.* In IROS Workshop on Probabilistic Graphical Models in Robotics, GraphBot, 2010. (Cited on page 33.)

[Kwon 2011] Woo Young Kwon and I Hong Suh. *Towards proactive assistant robots for human assembly tasks.* In Proceedings of the 6th international conference on Human-Robot Interaction (HRI), pages 175–176, New York, NY, USA, 2011. ACM. (Cited on page 33.)

[L'Abbate 2007] Marcello L'Abbate. *Modelling Proactive Behaviour of Conversational Interfaces.* PhD thesis, TU Darmstadt, Darmstadt, November 2007. (Cited on page 33.)

[Lam 2011] Chi-Pang Lam, Chen-Tun Chou, Kuo-Hung Chiang and Li-Chen Fu. *Human-Centered Robot Navigation - Towards a Harmoniously Human-Robot Coexisting Environment.* Robotics, IEEE Transactions on, vol. 27, no. 1, pages 99 –112, feb. 2011. (Cited on page 21.)

[Lee 2009] Kyu Hwa Lee, Jinhan Lee, Andrea L. Thomaz and Aaron F. Bobick. *Effective robot task learning by focusing on task-relevant objects.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2551–2556, Piscataway, NJ, USA, 2009. (Cited on page 261.)

[Lemaignan 2010] S. Lemaignan, Ros R., Mösenlechner L., Alami R. and M. Beetz. *ORO, a knowledge management module for cognitive architectures in robotics.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3548–3553, 2010. (Cited on page 290.)

[Lemaignan 2011a] S. Lemaignan, R. Ros and Alami R. *Dialogue in situated environments: A symbolic approach to perspective-aware grounding, clarification and reasoning for robot.* 2011. (Cited on page 290.)

[Lemaignan 2011b] Séverin Lemaignan, Raquel Ros, Rachid Alami and Michael Beetz. *What are you talking about? Grounding dialogue in a perspective-aware robotic architecture.* In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), 2011. (Cited on page 29.)

[Lemaignan 2011c] Séverin Lemaignan, Raquel Ros, E. Akin Sisbot, Rachid Alami and Michael Beetz. *Grounding the Interaction: Anchoring Situated Discourse in Everyday Human-Robot Interaction.* International Journal of Social Robotics, pages 1–19, 2011. (Cited on page 29.)

[Lemaignan 2012] Severin Lemaignan, Raquel Ros, E. Sisbot, Rachid Alami and Michael Beetz. *Grounding the Interaction: Anchoring Situated Discourse in Everyday Human-Robot Interaction.* International Journal of Social Robotics, vol. 4, pages 181–199, 2012. (Cited on page 244.)

[Lempers 1977] J. D. Lempers, E. R. Flavell and J. H. Flavell. *The development in very young children of tacit knowledge concerning visual perception.* Genetic Psychology Monographs, vol. 95, no. 1, pages 3–53, 1977. (Cited on page 5.)

[Li 2011] Jamy Li and Mark Chignell. *Communication of Emotion in Social Robots through Simple Head and Arm Movements.* International Journal of Social Robotics, vol. 3, pages 125–142, 2011. (Cited on page 223.)

[Liebowitz 1976] Bernard Liebowitz. *Human Territories: How We Behave in Space-Time.* Family Process, vol. 15, no. 4, pages 447–451, 1976. (Cited on page 7.)

[Lopes 2007] Manuel Lopes, Francisco S. Melo and Luis Montesano. *Affordance-based imitation learning in robots.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1015–1021, 2007. (Cited on pages 17 and 35.)

[Lörken 2008] Christopher Lörken and Joachim Hertzberg. *Grounding planning operators by affordances.* In International Conference on Cognitive Systems, pages 79 –84, 2008. (Cited on page 29.)

[Louwerse 2005] M. Louwerse and A. Bangerter. *Focusing attention with deictic gestures and linguistic expressions.* In Proceedings of the 27th Annual Meeting of the Cognitive Science Society, 2005. (Cited on pages 240 and 241.)

[Lozano-Perez 1987] T. Lozano-Perez, J. Jones, E. Mazer, P. O'Donnell, W. Grimson, P. Tournassoud and A. Lanusse. *Handey: A robot system that recognizes, plans, and manipulates.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), vol. 4, pages 843 – 849, mar 1987. (Cited on page 31.)

[Luciw 2011] M. Luciw, V. Graziano, M. Ring and J. Schmidhuber. *Artificial curiosity with planning for autonomous perceptual and cognitive development.* In IEEE International Conference on Development and Learning (ICDL), vol. 2, pages 1 –8, aug. 2011. (Cited on pages 19 and 106.)

[Lunsky 1965] Louis L. Lunsky. *Learning and Instinct in Animals.* Archives of Internal Medicine, vol. 115, no. 6, pages 757–758, 1965. (Cited on page 5.)

[Mainprice 2011] J. Mainprice, E. Akin Sisbot, L. Jaillet, J. Cortes, R. Alami and T. Simeon. *Planning human-aware motions using a sampling-based costmap planner.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 5012–5017, may 2011. (Cited on pages 25 and 151.)

[Marin-Urias 2009a] L. F. Marin-Urias. *Reasoning About Space for Human-Robot Interaction.* PhD thesis, Universite de Toulouse, LAAS-CNRS, November 2009. (Cited on pages 20 and 26.)

[Marin-Urias 2009b] L.F. Marin-Urias, E.A. Sisbot, A.K. Pandey, R. Tadakuma and R. Alami. *Towards shared attention through geometric reasoning for Human Robot Interaction.* In Proceedings of 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pages 331 –336, dec. 2009. (Cited on page 15.)

[Martin 2004] C. Martin, H.J. Bohme and H.M. Gross. *Conception and realization of a multi-sensory interactive mobile office guide.* In IEEE International Conference on Systems, Man and Cybernetics, pages 5368 – 5373, oct. 2004. (Cited on pages 20 and 24.)

[Martinez-Garcia 2005] E.A. Martinez-Garcia, Ohya Akihisa and Shin'ichi Yuta. *Crowding and guiding groups of humans by teams of mobile robots.* In IEEE Workshop on Advanced Robotics and its Social Impacts, pages 91 – 96, june 2005. (Cited on pages 20 and 24.)

[Martinson 2007] E. Martinson. *Hiding the acoustic signature of a mobile robot.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 985 –990, 2007. (Cited on page 21.)

[merriam webster.com a] merriam webster.com. *Definition of Action, [Accessed 2011].* http://www.merriam-webster.com/dictionary/action. (Cited on page 48.)

[merriam webster.com b] merriam webster.com. *Definition of Proactive, [Accessed 2011]*. http://www.merriam-webster.com/dictionary/proactive. (Cited on pages 57 and 213.)

[Miceli 1995] Maria Miceli, Amedo Cesta and Paola Rizzo. *Distributed artificial intelligence from a socio-cognitive standpoint: Looking at reasons for interaction*. AI & Society, vol. 9, pages 287–320, 1995. (Cited on pages 3 and 334.)

[Michael 2011] Loizos Michael. *Causal Learnability*. In International Joint Conference on Artificial Intelligence (IJCAI), pages 1014–1020, 2011. (Cited on pages 35, 36 and 40.)

[Moll 2004] H. Moll and M. Tomasello. *12-and 18-month-old infants follow gaze to spaces behind barriers*. Developmental Science, vol. 7, no. 1, pages F1–F9, Feb. 2004. (Cited on pages 4 and 336.)

[Montesano 2007] L. Montesano, M. Lopes, A. Bernardino and J. Santos-Victor. *Modeling affordances using Bayesian networks*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4102 –4107, 2007. (Cited on page 35.)

[Muhlig 2009] Manuel Muhlig, Michael Gienger, Sven Hellbach, Jochen J. Steil and Christian Goerick. *Task-level imitation learning using variance-based movement optimization*. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 1177 –1184, may 2009. (Cited on page 34.)

[Müller 2008] J. Müller, C. Stachniss, K.O. Arras and W. Burgard. *Socially Inspired Motion Planning for Mobile Robots in Populated Environments*. pages 85–90, Karlsruhe, Germany, 2008. (Cited on pages 21 and 22.)

[Norman 1988] D.A. Norman. The psychology of everyday things. Basic Books, 1988. (Cited on page 17.)

[Novak 2011] P. Novak and Jamroga. *Agents, Actions and Goals in Dynamic Environments*. In International Joint Conference on Artificial Intelligence (IJCAI), pages 313–317, 2011. (Cited on pages 40, 49 and 60.)

[Ogawara 2003] K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi. *Extraction of essential interactions through multiple observations of human demonstrations*. IEEE Transactions on Industrial Electronics, vol. 50, no. 4, pages 667 – 675, aug. 2003. (Cited on page 35.)

[Olsen 2003] D. Olsen and M. Goodrich. *Metrics for evaluating human-robot interactions*. In Proceedings of NIST Performance Metrics for Intelligent Systems Workshop, 2003. (Cited on page 245.)

[Oudeyer 2007] Pierre-Yves Oudeyer, Frédéric Kaplan and Véréna Hafner. *Intrinsic Motivation Systems for Autonomous Mental Development*. IEEE Transactions on Evolutionary Computation, vol. 11, no. 2, pages 265 –286, 2007. (Cited on pages 19 and 106.)

[Pacchierotti 2005] E. Pacchierotti, H.I. Christensen and P. Jensfelt. *Human-robot embodied interaction in hallway settings: a pilot user study.* In Proceedings of IEEE International Workshop on Robot and Human interactive Communication (RO-MAN), pages 164 – 171, aug. 2005. (Cited on pages 20 and 112.)

[Pacchierotti 2006a] E. Pacchierotti, H. I. Christensen and P. Jensfelt. *Evaluation of Passing Distance for Social Robots.* In IEEE International Workshop on Robot and Human Interactive Communication, RO-MAN, pages 315–320, Hatfield, UK, 2006. (Cited on page 20.)

[Pacchierotti 2006b] E. Pacchierotti, H.I. Christensen and P. Jensfelt. *Design of an Office-Guide Robot for Social Interaction Studies.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4965 –4970, oct. 2006. (Cited on page 24.)

[Pardowitz 2007] M. Pardowitz and R. Dillmann. *Towards life-long learning in household robots: The Piagetian approach.* In IEEE 6th International Conference on Development and Learning, (ICDL), pages 88 –93, july 2007. (Cited on pages 3 and 34.)

[Paris 2007] Sebastien Paris, Julien Pettre and Stephane Donikian. *Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach.* Computer Graphics Forum, vol. 26, no. 3, pages 665–674, 2007. (Cited on pages 23 and 116.)

[Piaget 1945] Jean Piaget. Play, dreams, and imitation in childhood. Norton, New York, 1945. (Cited on page 5.)

[Project] CHRIS Project. *Cooperative Human Robot Interaction Systems.* `http://www.chrisfp7.eu/`. (Cited on page 3.)

[psychwiki Prosocial] psychwiki Prosocial. *Prosocial Behavior.* `http://www.psychwiki.com/wiki/Prosocial_Behavior`. (Cited on page 9.)

[Rieger 1976] Chuck Rieger. *An organization of knowledge for problem solving and language comprehension.* Artificial Intelligence, vol. 7, no. 2, pages 89 – 127, 1976. (Cited on page 48.)

[Rochat 1995] P. Rochat. *Perceived reachability for self and for others by 3 to 5-year old children and adults.* Journal of Experimental Child Psychology, vol. 59, pages 317–333, 1995. (Cited on pages 4 and 336.)

[Rochat 1997] P. Rochat and M. Wraga. *An account of the systematic error in judging what is reachable.* Journal of Experimental Psychology: Human Perception and Performance, vol. 23, no. 1, pages 199–212, 1997. (Cited on page 67.)

[Rome 2008] Erich Rome, Lucas Paletta, Erol Şahin, Georg Dorffner, Joachim Hertzberg, Ralph Breithaupt, Gerald Fritz, Jörg Irran, Florian Kintzler, Christopher Lörken, Stefan May and Emre Uğur. *The MACS project: an approach to affordance-inspired robot control.* In Proceedings of the 2006 international conference on Towards affordance-based robot control, pages 173–210, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 29.)

[Ros 2010] R. Ros, S. Lemaignan, E.A. Sisbot, R. Alami, J. Steinwender, K. Hamann and F. Warneken. *Which one? Grounding the referent based on efficient human-robot interaction.* In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 570 –575, sept. 2010. (Cited on page 244.)

[Rosenbaum 1992] D. A. Rosenbaum, J. Vaughan, H. J. Barnes and M. J. Jorgensen. *Time course of movement planning: selection of handgrips for object manipulation.* Experimental Psychology: Learning, Memory, and Cognition, vol. 18, pages 1058–1073, 1992. (Cited on page 6.)

[Rosenbaum 1995] David A. Rosenbaum, Loukia D. Loukopoulos, Ruud G. J. Meulenbroek, Jonathan Vaughan and Sascha E. Engelbrecht. *Planning reaches by evaluating stored postures.* Psychological Review, vol. 102, pages 28–67, 1995. (Cited on page 6.)

[Rosenbaum 2001] D. A. Rosenbaum, R. G. J. Meulenbroek, J. Vaughan and C. Jansen. *Posture-based motion planning: Applications to grasping.* Psychological Review, vol. 108, pages 709–734, 2001. (Cited on pages 6, 155 and 347.)

[Ruesch 2008] J. Ruesch, M. Lopes, A. Bernardino, J. Hornstein, J. Santos-Victor and R. Pfeifer. *Multimodal saliency-based bottom-up attention a framework for the humanoid robot iCub.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 962 –967, may 2008. (Cited on pages 19 and 106.)

[Sacerdoti 1974] Earl D. Sacerdoti. *Planning in a hierarchy of abstraction spaces.* Artificial Intelligence, vol. 5, no. 2, pages 115 – 135, 1974. (Cited on page 49.)

[Salichs 2006] M.A. Salichs, R. Barber, A.M. Khamis, M. Malfaz, J.F. Gorostiza, R. Pacheco, R. Rivas, A. Corrales, E. Delgado and D. Garcia. *Maggie: A Robotic Platform for Human-Robot Social Interaction.* In IEEE Conference on Robotics, Automation and Mechatronics, pages 1 –7, june 2006. (Cited on page 32.)

[Sapio 2006] V. Sapio, J. Warren and O. Khatib. *Predicting reaching postures using a kinematically constrained shoulder model.* In Jadran Lennarcic and B. Roth, (eds.), Advances in Robot Kinematics, pages 209–218. Springer Netherlands, 2006. (Cited on pages 71, 72 and 251.)

[Sartori 2011] Luisa Sartori, Elisa Straulino and Umberto Castiello. *How Objects Are Grasped: The Interplay between Affordances and End-Goals.* PLoS ONE, vol. 6, no. 9, Nov. 2011. (Cited on page 6.)

[Saut 2012] Jean-Philippe Saut and Daniel Sidobre. *Efficient models for grasp planning with a multi-fingered hand.* Robotics and Autonomous Systems, vol. 60, pages 347–357, 2012. (Cited on pages 88 and 89.)

[Scassellati 2002] Brian Scassellati. *Theory of Mind for a Humanoid Robot.* Autonomous Robots, vol. 12, pages 13–24, 2002. (Cited on page 223.)

[Schmid 2007] A.J. Schmid, O. Weede and H. Worn. *Proactive Robot Task Selection Given a Human Intention Estimate.* In Proceedings of 16th IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 726 –731, aug. 2007. (Cited on page 33.)

[Schmidt-Rohr 2010] S.R. Schmidt-Rohr, M. Losch and R. Dillmann. *Learning flexible, multi-modal human-robot interaction by observing human-human-interaction.* In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 582 –587, sept. 2010. (Cited on page 35.)

[Scholtz 2003] Jean Scholtz. *Theory and Evaluation of Human Robot Interactions.* In Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS), Washington, DC, USA, 2003. IEEE Computer Society. (Cited on page 39.)

[Schrempf 2005] O.C. Schrempf, U.D. Hanebeck, A.J. Schmid and H. Worn. *A novel approach to proactive human-robot cooperation.* In Proceedings of IEEE International Workshop on Robot and Human interactive Communication (RO-MAN), pages 555 – 560, aug. 2005. (Cited on page 33.)

[Schubö 2007] Anna Schubö, Cordula Vesper, Mathey Wiesbeck and Sonja Stork. *Movement coordination in applied human-human and human-robot interaction.* In Proceedings of the 3rd Human-computer interaction and usability engineering of the Austrian computer society conference on HCI and usability for medicine and health care (USAB), pages 143–154, Berlin, Heidelberg, 2007. Springer-Verlag. (Cited on page 6.)

[Sciutti 2012] Alessandra Sciutti, Ambra Bisio, Francesco Nori, Giorgio Metta, Luciano Fadiga, Thierry Pozzo and Giulio Sandini. *Measuring Human-Robot Interaction Through Motor Resonance.* International Journal of Social Robotics, vol. 4, pages 223–234, March 2012. (Cited on page 244.)

[Sebanz 2009] Natalie Sebanz and Guenther Knoblich. *Prediction in Joint Action: What, When, and Where.* Topics in Cognitive Science, vol. 1, no. 2, pages 353–367, 2009. (Cited on page 220.)

[Shi 2008] Dongqing Shi, E.G. Collins, A. Donate, Xiuwen Liu, B. Goldiez and D. Dunlap. *Human-aware robot motion planning with velocity constraints.* In

International Symposium on Collaborative Technologies and Systems (CTS), pages 490 –497, may 2008. (Cited on pages 20 and 23.)

[Simeon 2001] T. Simeon, J. P. Laumond and F. Lamiraux. *Move3D: a generic platform for path planning*. In Proceedings of 4th International Symposium on Assembly and Task Planning, pages 25–30, 2001. (Cited on pages 63, 97, 160, 260 and 291.)

[Siméon 2004] T. Siméon, J.P. Laumond, J. Cortés and A. Sahbani. *Manipulation planning with probabilistic roadmaps*. The International Journal of Robotics Research, vol. 23, no. 7-8, pages 729–746, 2004. (Cited on page 31.)

[Sisbot 2007a] E.A. Sisbot, L.F. Marin-Urias, R. Alami and T. Simeon. *A Human Aware Mobile Robot Motion Planner*. IEEE Transactions on Robotics, vol. 23, no. 5, pages 874 –883, oct. 2007. (Cited on page 21.)

[Sisbot 2007b] E.A. Sisbot, L.F. Marin-Urias, R. Alami and T. Simeon. *A Human Aware Mobile Robot Motion Planner*. Robotics, IEEE Transactions on, vol. 23, no. 5, pages 874 –883, oct. 2007. (Cited on pages 23 and 25.)

[Sisbot 2008] Emrah Akin Sisbot. *Towards Human Aware Robot Motions*. PhD thesis, Universite de Toulouse (Toulouse III), October 2008. (Cited on page 20.)

[Sisbot 2010] EmrahAkin Sisbot, LuisF. Marin-Urias, Xavier Broquère, Daniel Sidobre and Rachid Alami. *Synthesizing Robot Motions Adapted to Human Presence*. International Journal of Social Robotics, vol. 2, pages 329–343, 2010. (Cited on page 25.)

[Song 2010] D. Song, K. Huebner, V. Kyrki and D. Kragic. *Learning task constraints for robot grasping using graphical models*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1579–1585, oct. 2010. (Cited on pages 26 and 148.)

[Steinfeld 2006] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan Schultz and Michael Goodrich. *Common metrics for human-robot interaction*. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-Robot Interaction (HRI), pages 33–40, New York, NY, USA, 2006. ACM. (Cited on page 245.)

[Stoytchev 2005] A. Stoytchev. *Behavior-Grounded Representation of Tool Affordances*. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 3060 – 3065, april 2005. (Cited on page 17.)

[Svenstrup 2010] M. Svenstrup, T. Bak and H.J. Andersen. *Trajectory planning for robots in dynamic human environments*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4293 –4298, oct. 2010. (Cited on page 21.)

[Svetlova 2010] Margarita Svetlova, Sara R. Nichols and Celia A. Brownell. *Toddlers' Prosocial Behavior: From Instrumental to Empathic to Altruistic Help-*

ing. Child Development, vol. 81, no. 6, pages 1814–1827, 2010. (Cited on page 5.)

[Tenorth 2009] M. Tenorth and M. Beetz. *KNOWROB- knowledge processing for autonomous personal robots*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4261 –4266, oct. 2009. (Cited on page 35.)

[Thrun 1998] Sebastian Thrun. *Learning metric-topological maps for indoor mobile robot navigation*. Artificial Intelligence, vol. 99, no. 1, pages 21 – 71, 1998. (Cited on page 109.)

[Thrun 2000] Sebastian Thrun, M. Beetz, Maren Bennewitz, Wolfram Burgard, A.B. Creemers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Chuck Rosenberg, Nicholas Roy, Jamieson Schulte and Dirk Schulz. *Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva*. International Journal of Robotics Research, vol. 19, no. 11, pages 972–999, November 2000. (Cited on page 3.)

[Tipaldi 2011] Gian Diego Tipaldi and Kai O. Arras. *Please do not disturb! Minimum interference coverage for social robots*. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1968 –1973, sept. 2011. (Cited on page 21.)

[Tomasello 1990] Michael Tomasello. *Cultural Transmission in the Tool Use and Communicatory Signaling of Chimpanzees?* In Sue Taylor Parker and Kathleen Gibson, (eds.), Language and Intelligence in Monkeys and Apes, pages 274–311. Cambridge University Press, 1990. (Cited on page 5.)

[Tomasello 2005] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne and Henrike Moll. *Understanding and sharing intentions: the origins of cultural cognition*. Behavioral and Brain Sciences, vol. 28, no. 5, pages 675–691, 2005. (Cited on pages 5 and 223.)

[Trafton 2005a] J. Gregory Trafton, Alan C. Schultz, Magdalena Bugajska and Farilee Mintz. *Perspective-taking with Robots: Experiments and models*. In IEEE International Workshop on Robots and Human Interactive Communication (RO-MAN), pages 580–584, 2005. (Cited on pages 15 and 29.)

[Trafton 2005b] J.G. Trafton, N.L. Cassimatis, M.D. Bugajska, D.P. Brock, F.E. Mintz and A.C. Schultz. *Enabling effective human-robot interaction using perspective-taking in robots*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 35, no. 4, pages 460 – 470, july 2005. (Cited on pages 15 and 29.)

[Tranberg Hansen 2009] S. Tranberg Hansen, M. Svenstrup, H.J. Andersen and T. Bak. *Adaptive human aware navigation based on motion pattern analysis*. In Proceedings of IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pages 927 –932, 2009. (Cited on page 21.)

[Trautman 2010] P. Trautman and A. Krause. *Unfreezing the robot: Navigation in dense, interacting crowds.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 797 –803, oct. 2010. (Cited on page 22.)

[Tversky 1972] Amos Tversky. *Elimination by aspects: a theory of choice.* Psychological Review, vol. 79, no. 4, pages 281–299, 1972. (Cited on pages 155 and 347.)

[Ugur 2007] Emre Ugur, Mehmet R. Dogar, Maya Cakmak and Erol Sahin. *Curiosity-driven learning of traversability affordance on a mobile robot.* In IEEE 6th International Conference on Development and Learning (ICDL), pages 13–18, july 2007. (Cited on page 17.)

[Van Zwynsvoorde 2001] D. Van Zwynsvoorde, T. Simeon and R. Alami. *Building topological models for navigation in large scale environments.* In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), vol. 4, pages 4256 – 4261, 2001. (Cited on page 109.)

[Victorino 2003] Alessandro Correa Victorino, Patrick Rives and Jean-Jacques Borrelly. *Safe Navigation for Indoor Mobile Robots. Part I: A Sensor-Based Navigation Framework.* The International Journal of Robotics Research, vol. 22, no. 12, pages 1005–1118, 2003. (Cited on page 122.)

[Walters 2005] M.L. Walters, K. Dautenhahn, R. te Boekhorst, Kheng Lee Koay, C. Kaouri, S. Woods, C. Nehaniv, D. Lee and I. Werry. *The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment.* In Proceedings of IEEE International Workshop on Robot and Human interactive Communication (RO-MAN), pages 347 – 352, aug. 2005. (Cited on page 112.)

[Warneken 2007] Felix Warneken and Michael Tomasello. *Helping and Cooperation at 14 Months of Age.* Infancy, vol. 11, no. 3, pages 271–294, 2007. (Cited on page 5.)

[WillowGarage] WillowGarage and ROS Community. *ROS: Robot Operating System.* `http://www.ros.org/wiki/`. (Cited on page 290.)

[Wood 1998] David J. Wood. How children think and learn: The social contexts of cognitive development. Blackwell (Oxford, UK and Malden, Mass.), 1998. (Cited on page 5.)

[Wusteman 1992] Judith Wusteman. *Explanation-Based Learning: A survey.* Artificial Intelligence Review, vol. 6, pages 243–262, 1992. (Cited on pages 255 and 358.)

[Xue 2008] Z. Xue, J.M. Zoellner and R. Dillmann. *Planning regrasp operations for a multifingered robotic hand.* In IEEE International Conference on Automation Science and Engineering (CASE), pages 778–783, aug. 2008. (Cited on pages 26 and 148.)

[Yamaoka 2008] Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro and Norihiro Hagita. *How close?: model of proximity control for information-presenting robots.* In Proceedings of the 3rd ACM/IEEE international conference on Human Robot Interaction (HRI), pages 137–144, New York, NY, USA, 2008. ACM. (Cited on page 112.)

[Yanco 2004] H. A. Yanco and J. Drury. *Classifying human-robot interaction: an updated taxonomy.* In Proceedings of IEEE International Conference on Systems, Man and Cybernetics, pages 2841 –2846, 2004. (Cited on page 39.)

[Ye 2011] G. Ye and R. Alterovitz. *Demonstration-Guided Motion Planning.* In Proceedings of International Symposium on Robotics Research (ISRR), August 2011. (Cited on page 34.)

[Yoda 1997] M. Yoda and Y. Shiota. *The mobile robot which passes a man.* In Proceedings of IEEE International Workshop on Robot and Human interactive Communication (RO-MAN), pages 112 –117, sep-1 oct 1997. (Cited on page 112.)

[Zacharias 2007] F. Zacharias, C. Borst and G. Hirzinger. *Capturing robot workspace structure: representing robot capabilities.* In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3229 –3236, 2007. (Cited on pages 15 and 67.)

[Zacharias 2009] F. Zacharias, C. Borst and G. Hirzinger. *Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace.* In International Conference on Advanced Robotics (ICAR), pages 1 –8, june 2009. (Cited on page 15.)

[Zhang 2008] Wei Zhang and David Rosenbaum. *Planning for manual positioning: the end-state comfort effect for manual abduction-adduction.* Experimental Brain Research, vol. 184, pages 383–389, 2008. (Cited on page 6.)

[Zulueta 2010] A. Garrell Zulueta and Sanfeliu A. *Local optimization of cooperative robot movements for guiding and regrouping people in a guiding mission.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3294–3299, 2010. (Cited on page 24.)

[Zwickel 2009] Jan Zwickel. *Agency attribution and visuospatial perspective taking.* Psychonomic Bulletin and Review, vol. 16, pages 1089–1093, 2009. (Cited on page 223.)

Appendix D

Résumé en français

Vers des robots socialement intelligents en environnement humain

Les robots feront bientôt partie de notre vie quotidienne. Que ce soit dans la rue, au bureau, à la maison ou au supermarché, les robots seront là pour nous aider et nous servir. Pour que de tels robots soient acceptés et appréciés, ils doivent explicitement considérer la présence de l'homme et prendre en compte les facteurs sociaux inhérents dans toutes les étapes de planification et de prise de décision, que ce soit pour le mouvement, la manipulation, l'interaction, etc.

Cette thèse se concentre sur ces questions qui soulèvent de nouveaux défis qui ne peuvent pas être traitées de façon appropriée par une simple adaptation de l'état de l'art existant en robotique tant en terme de planification, de commande ou de prise de décision. Pour traiter ces défis, cette thèse introduit de nouveaux concepts pour la prise en compte de l'interaction Homme-Robot et montre leurs applications à travers plusieurs systèmes. Les résultats sont mis en œuvre sur différents types de robots réels (PR2, HRP2, Jido, ...). Notre vision est de construire une base pour le développement de robots avec des comportements socio-cognitifs plus complexes, ce qui permettra la future coexistence entre l'homme et les robots en parfaite harmonie. Cette thèse est une étape vers la mise en œuvre de robots socialement intelligents.

Mots clés: *Interaction Homme-Robot (HRI), Robot socialement intelligent, Prise de perpective spatio-visuel multi-états, Analyse d'Affordance, Analyse de Mightability, Carte de Mightability, Graphe D'Affordance, Manipulation dans le cadre d'une tâche d'interaction Homme-Robot, Navigation socialement adaptée, Robot Guide Social, Robot Coopératif, Comportement pro-actif, Plan partagé, Apprentissage par démonstration*

Keywords: *Human Robot Interaction (HRI), Socially Intelligent Robot, Multi-State Visuo-Spatial Perspective Taking, Affordance Analysis, Mightability Analysis, Mightability Maps, Affordance Graph, Human-Robot Interactive Manipulation, Socially Aware Navigation, Social Robot Guide, Cooperative Robot, Proactive Behavior, Shared Planning, Learning from Demonstration*

Appendix E

Vers des robots socialement intelligents en environnement humain

Contents

E.1 Introduction **332**

E.2 Pourquoi un robot social ? **334**

E.2.1 Les ingrédients de l'intelligence sociale 334

E.2.2 Le robot social/sociable 335

E.2.3 Pyramide de l'incarnation de l'intelligence sociale 336

E.2.4 Notre approche de l'incarnation sociale 336

E.3 Travaux Connexes, Challenges et Contribution **337**

E.4 Un cadre conceptuel pour l'Interaction Homme-Robot . . . **337**

E.5 Analyse de "Mightability": Prise de perspective spatio-visuel multi-états **338**

E.5.1 Hiérarchie des efforts 338

E.5.2 Analyse de la Mightability 339

E.6 Analyse d'affordance et Evaluation de la situation **339**

E.7 Navigation et Guidage socialement adaptés en environnement humain **342**

E.7.1 Planificateur de trajectoire socialement acceptable 342

E.7.2 Robot guide 342

E.8 Planification de tâches basiques pour l'interaction homme-robot **347**

E.9 Graphe d'affordance: Un cadre basé sur les efforts pour établir l'interaction et la génération de plan partagée . . . **348**

E.9.1 Taskability Graph 348

E.9.2 Manipulability Graph 350

E.9.3 Affordance Graph 351

E.10 Comportement pro-social pro-actif **352**

E.10.1 Proposition de niveaux de comportements pro-actifs 353

E.10.2 Instanciation de comportement pro-actifs 354

E.10.3 Etudes utilisateur 354

E.11 Compréhension de tâche par démonstration **355**

E.11.1 Apprentissage via l'explication et l'utilisation d'un arbre d'hypothèses initiales 358
E.11.2 Facteur de cohérence 360
E.11.3 Bénéfices et applications possibles 364
E.12 Conclusion 365

E.1 Introduction

Des robots interagissant avec nous dans notre vie de tous les jours, ce ne sera bientôt plus de la science-fiction. Dans la rue, au bureau, à la maison ou au supermarché, ils seront présents pour nous aider et nous assister dans notre quotidien. Pour cela, ils ne doivent plus être ces machines évoluant de manière isolée dans les usines ou dans l'environnement contrôlé d'un laboratoire de recherche, ils doivent prendre en compte explicitement la présence de l'homme dans leurs programmes et stratégies de prise de décision, que ce soit pour le mouvement, la manipulation ou l'interaction. Cette thèse explore divers aspects socio-cognitifs pour le permettre comme la prise de perspective spatiale, la navigation acceptable par l'homme, la planification coopérative, l'apprentissage par démonstration de comportements pro-actifs ou de la sémantique d'une tâche. Ainsi, en identifiant les ingrédients clés de ces différents aspects, nous dotons le robot des prémices d'une intelligence socio-cognitive, marquant une étape vers la coexistence avec l'homme.

Pour qu'un robot navigue au milieu des hommes de manière satisfaisante (ce que nous nommons *socially acceptable navigation of a robot*), il est clair que le robot ne doit plus seulement nous traiter, nous les hommes, comme des obstacles dynamiques de l'environnement. Par exemple, le robot doit pouvoir être capable de décider de prendre un chemin plus long si cela correspond à nos attentes et permet d'éviter une incompréhension/un énervement/une crainte/une surprise de notre part lors de l'exécution de la trajectoire.

De même, si le robot doit guider une personne d'un endroit à un autre, il doit être capable de prendre en compte les autres tâches de la personne et la guider comme elle le souhaite. Ainsi, il est naturel de penser que de manière intentionnel ou non, il y aura des déviations dans la trajectoire de la personne par rapport à celle calculée par le robot. En outre, le robot peut être amené, si la tâche de guidage est interrompue, à demander à l'homme de se réengager dans la tâche. Cependant, un robot (*human friendly robot*) pour être accepté ne devra jamais sur-réagir ou à l'inverse ne rien faire.

D'autre part, lorsque le robot doit travailler explicitement avec l'homme dans le cadre d'un scénario coopératif de manipulation, il doit être capable d'analyser les aptitudes et moyens de la personne avec laquelle il interagit. Cette capacité d'analyser les choses d'un autre point de vue, qui se nomme, prise de perspective (*perspective taking*), est essentielle pour plusieurs raisons. Si l'on considère les exemples suivants : où poser un objet si l'on souhaite que l'homme l'attrape avec le moindre effort, où et comment montrer un objet à l'homme, comment attraper un objet pour que l'homme soit capable de l'attraper également lors d'un échange, etc. Ils requièrent tous que le robot ait des capacités de raisonnement qui dépassent la stabilité de la prise d'un objet ou son placement (dans le cadre de tâches pourtant simple de manipulation d'un objet tel que : montrer, cacher, rendre accessible, etc.)

Pouvoir générer un plan en commun (*shared-plan*) pour exécuter des tâches quotidiennes, tel que nettoyer la table, est un autre aspect important lorsque l'on considère la coopération. Cela requiert que le robot ait des capacités de raisonnement à différents niveaux pour planifier une tâche : au niveau symbolique pour décider comment réaliser la tâche et attribuer un rôle à chacun des agents; au niveau géométrique pour s'assurer de la faisabilité des actions. De plus, les raisonnements fait sur les efforts et l'état courant peuvent être utilisés pour valider la quantité, l'étendue de la coopération de même que la méthode de coopération choisie.

Un autre aspect de l'interaction socio-cognitive est le comportement pro-actif, i.e. planifier et agir en anticipant les futures besoins, problèmes, changements. Cela nécessite que le robot soit capable de savoir comment et quand agir de manière pro-active de façon à réaliser l'interaction ou la tâche en cours.

Enfin, dans une perspective d'autonomie à long terme du robot, l'apprentissage par l'interaction avec l'humain et par l'observation au jour le jour des tâches sont des aspects importants. Ce processus, qualifié d'apprentissage social (*social learning*), rendra possible l'exécution efficace de ses tâches par le robot dans des contextes différents. En effet, même les tâches de base telles que donner, cacher, rendre accessible, montrer, etc, pourraient être réalisées différemment, en fonction de la situation et des contraintes. Cependant, on ne peut pas s'attendre à ce que le robot dispose d'un exemple pour chaque tâche et sous-tâche et donc le simple fait d'imiter les actions d'une démonstration ne peut être suffisant. Le robot doit être capable de comprendre le but de la démonstration, i.e. quels sont les effets attendus de la tâche. Le robot devra être capable d'apprendre cela de manière autonome à différents niveaux d'abstraction.

Cette thèse se concentre sur ces questions, qui posent de nouveaux défis qui ne peuvent être traités de façon appropriée par une simple adaptation de l'état de l'art en robotique sur la planification, l'automatique ou la prise de décisions. La thèse identifie tout d'abord des ingrédients socio-cognitifs fondamentaux provenant des domaines de la psychologie du comportement et du développement de l'enfant. Elle présente ensuite une architecture générale pour une interaction homme-robot socialement intelligente où nous introduisons ces nouveaux termes et concepts dans le cadre de l'interaction homme-robot et montrons leurs intégrations au niveau des mouvements du robot et de son interaction avec l'homme. Des résultats expérimentaux sur plusieurs robots réels (PR2, HPR2, Jido,...) permettront de montrer la pertinence de l'approche. Approche, qui est une étape vers des robots socialement intelligent (*Socially Intelligent Robots*) avec l'ambition de créer un terrain pour construire des comportements socio-cognitifs plus complexes pour la coexistence future de l'homme et du robot en parfaite harmonie.

E.2 Pourquoi un robot social ?

Dans cette thèse, le robot est considéré comme "...vivant ou prêt à vivre en compagnie d'autres personnes ou en communauté plutôt que seul...". "...living or disposed to live in companionship with others or in a community, rather than in isolation..." (definition of social, [dictionary.reference.com])

E.2.1 Les ingrédients de l'intelligence sociale

Vu à travers la fenêtre socio-cognitive, l'*IA (Intelligence Artificielle)*, et par la même, les *agents artificiels* devraient être en mesure de prendre en compte des éléments de haut niveau d'autres agents tels que l'aide et de la dépendance, [Miceli 1995]. Ici, le comportement social et le raisonnement social des agents sont décrits comme leur *capacité à recueillir des informations sur les autres et d'agir sur eux pour atteindre un objectif.* Ce qui signifie donc que ces agents ne peuvent pas subsister de manière isolée mais au contraire doivent s'adapter (*fit*) aux habitudes des autres agents (humain et/ou informatique (robotique)), [Bobrow 1991]. Si l'on explore cette adaptation (*'fit'*), les travaux sur les robots sociaux comme ceux de [Breazeal 2003], ou l'état de l'art sur la robotique interactive telle que [Fong 2003] distinguent plusieurs types d'incarnation :

- interface sociale (*social interface*) pour communiquer ;
- robot social/sociable (*sociable*), qui interragisse avec l'homme pour satisfaire leur but social ;
- robot socialement situé (*socially situated*), qui doit être capable de distinguer d'une part les agents et d'autres part les objets de l'environnement ;
- robot socialement adapté (*socially aware*), robot socialement situé qui est qui capable de s'adapter à l'homme ;

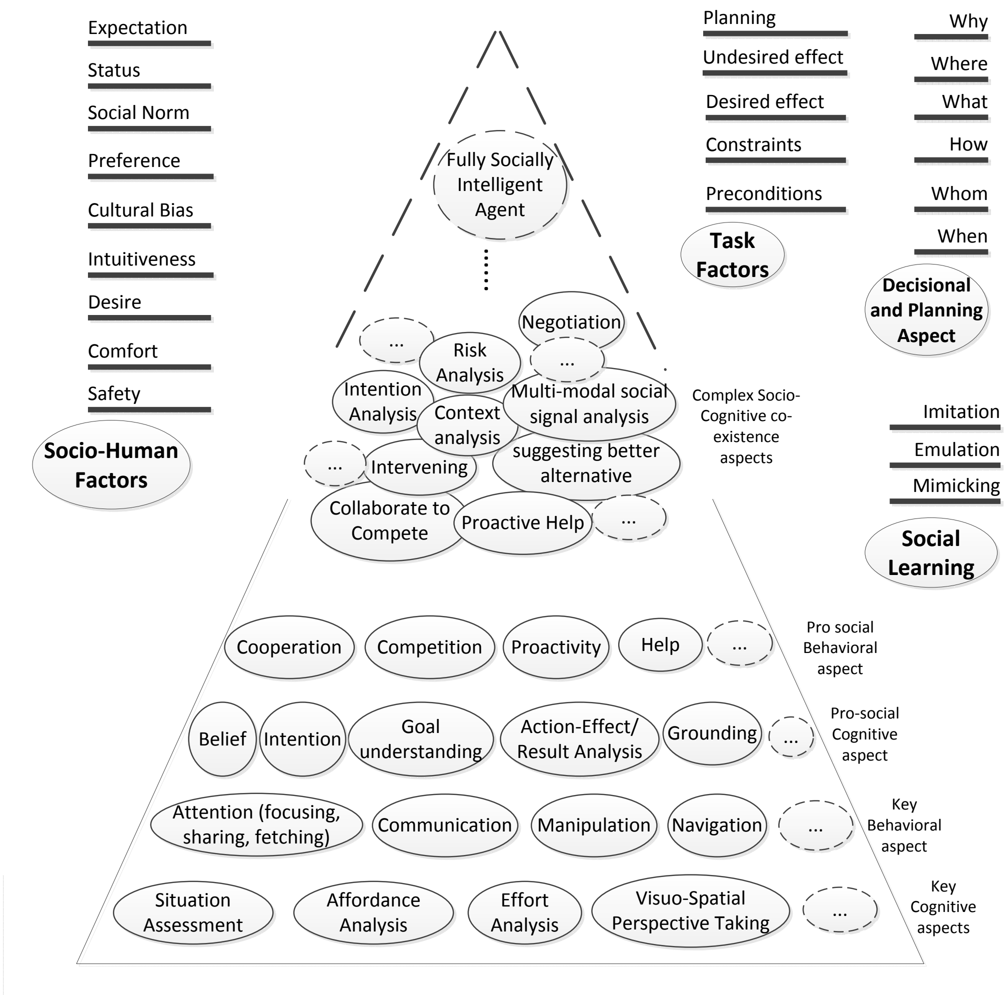


Figure E.1: Pyramide de l'incarnation de l'intelligence sociale *Social Intelligence Embodiment Pyramid*: Le point de départ sont les capacités socio-cognitives basiques qui amènent à des comportements socio-cognitifs plus complexes pour finalement donner un agent artificiel socialement intelligent.

- robot socialement intelligent (*socially intelligent*), robot qui montre des aspects d'intelligence sociale humaine.

E.2.2 Le robot social/sociable

En intégrant tout les éléments vus précédemment, nous définissons un robot socialement intelligent comme ceci: *"Un robot socialement intelligent dispose de la capacité cognitive clé de comprendre et d'évaluer : la situation, l'environnement, les agents et les capacités des agents. A partir de cela, il* *est capable de démontrer des comportements sûrs, compréhensibles et acceptables par l'homme et socialement attendus." "A socially intelligent robot is equipped with the key cognitive capabilities to understand and assess the situation and the environment; the agents and agents' capabilities; and exhibits behaviors, which are safe, human understandable, human acceptable and socially expected."*

Cette définition inclut les caractéristiques essentielles d'interface sociale, de prise de perspective sociale et d'attention à l'homme, comme discuté précédemment. Elle permet également la prise en compte d'un mélange de facteurs sociaux tel que : confort, intuition, etc.

E.2.3 Pyramide de l'incarnation de l'intelligence sociale

Différentes études en psychologie et développement de l'enfant [Carpendale 2006], [Dunphy-Lelii 2004], [Caron 2002], [Deak 2000], [Moll 2004], [Csibra 2008], [Flavell 1977], [Flavell 1981], [Flavell 1978], [Rochat 1995] ont analysé les capacités socio-cognitives et leurs évolutions. Ainsi, les formes les plus basiques d'interaction sociale comme le suivi du regard, la capacité de pointer un éléments ou de capter l'attention en pointant quelque chose apparaissent dès l'age de 12 mois. Entre 12 et 15 mois, commence à apparaître les preuves d'une compréhension de l'occlusion de la ligne de visée d'une autre personne. A partir de l'age de 3 ans, les enfants sont capables de percevoir quels sont les endroits atteignables par eux-mêmes et par les autres, montrant les premiers signes du développement de la capacité d'allocentrisme (*allocentrism*) i.e. décentrage spatial (*spatial decentration*) et prise de perspective (*perspective taking*). L'évolution de ces capacités socio-cognitives de raisonnement spatio-visuel chez l'enfant lui permet d'aider, de coopérer et de comprendre les intentions d'une personne avec qui il interagit. En se basant sur ces indications, nous avons construit une hiérarchie de capacités cognitives et comportementales permettant à un agent d'être socialement situé (*socially situated*) et socialement intelligent (*socially intelligent*). La figure E.1 montre cette pyramide en en identifiant les différents blocs qui composent cette thèse.

E.2.4 Notre approche de l'incarnation sociale

En nous inspirant des recherches sur le développement de l'enfant et de l'émergence des comportements sociaux, nous amenons le robot à devenir *"social"* en développant des composants de base (approche bottom-up) plutôt qu'en essayant de réaliser un ultime comportement social complexe (approche top-down). Ce choix d'une approche *bottom-up* sert un des objectifs de la

Figure E.2: Taxonomy des actions pour atteindre quelque chose :(a) arm-shoulder reach, (b) arm-torso reach, (c) standing reach.

thèse: *building a foundation for designing more complex socio-cognitive behaviors by exploring and realizing open 'nodes' to diversify and build upon.*

Nous allons maintenant décrire chapitre par chapitre les contributions de cette thèse.

E.3 Travaux Connexes, Challenges et Contribution

Le chapitre 2 présente l'état de l'art, identifie les challenges et met en avant les contributions de la thèse au sein de la hiérarchie présentée dans la figure E.1.

E.4 Un cadre conceptuel pour l'Interaction Homme-Robot

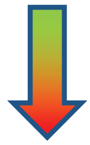
Le chapitre 3 rapporte la première contribution de cette thèse qui est une théorie unifiée de l'interaction homme robot basée sur la nature causale des changements dans l'environnement (*causal nature of environmental changes*). Nous présentons une définition généralisée de l'environnement, de ces attributes et des actions à partir de la perspective et des exigences du domaine de l'interaction homme-robot.

Nous expliquons comment cela peut servir d'infrastructure unifiée pour percevoir les différents aspects de l'interaction homme-robot selon l'état des connaissances sur l'état du monde et des actions: Pour agir de manière proactive; pour apprendre (émuler, imiter); pour valider les changements concernant les actions, les agents, les objets; pour prédire les conséquences; etc.

Les autres chapitres illustrent plusieurs éléments clés de cette infrastructure et présentent les algorithmes utilisés développés.

Effort to Reach	Effort to See
No_Effort	*No_Effort*
Arm_Effort	*Head_Effort*
Arm_Torso_Effort	*Head_Torso_Effort*
Whole_Body_Effort	*Whole_Body_Effort*
Displacement_Effort	*Displacement_Effort*
No_Possible_Known_Effort	*No_Possible_Known_Effort*

(a)

Effort Level
Minimum: 0

Maximum: 5

(b)

Figure E.3: Analyse d'effort avec prise en compte de l'homme, hiérarchie des efforts: a)**Analyse des efforts avec prise en compte de l'homme**: Qualifier les efforts pour voir et atteindre un objet ou un lieu à des niveaux d'abstraction compréhensibles par l'homme. (b)**Hiérachie des efforts**: Une manière de faire une analyse d'effort comparée. Ces deux éléments facilitent l'établissement, la comparaison et le raisonnement sur les efforts d'une manière compréhensible par l'homme.

E.5 Analyse de "Mightability": Prise de perspective spatio-visuel multi-états

Dans le chapitre 4, nous définissons le concept d'analyse de *Mightability*[1] (*Mightability Analysis*) qui va permettre au robot de raisonner sur les capacités spatio-visuelles de l'agent à partir des états qu'il est capable d'atteindre et d'analyser non seulement ce qui lui est visible et atteignable mais également ce qui ne l'est pas et pourquoi.

E.5.1 Hiérarchie des efforts

Nous avons présenté un nouvel ensemble de niveaux d'abstraction pour analyser les efforts et développer une hiérarchie des efforts basée sur les parties du corps impliquées dans l'effort. Cela permet au robot de mieux comprendre la signification de l'effort réalisée par l'homme et simplifie la communication. Cette hiérarchie a été développé en s'inspirant des recherches sur le mouvement humain et la psychologie du comportement, [Gardner 2001], [Choi 2004], où différents types d'action pour atteindre ("reach") ont été identifiés et analysés, figure E.2.

[1]Pour ce terme et quelques autres, nous avons préféré ne pas traduire le terme car il n'y a pas d'équivalent français acceptable; en effet, on pourrait traduire ici par "capabilité" mais çà n'a pas d'intérêt

E.5.2 Analyse de la Mightability

La *Mightability* signifie *"qui peut être capable de... "* ("*might be able to*"). Elle est calculée en fusionnant les informations obtenues via la prise de perspective et l'analyse des efforts. Elle permet au robot d'effectuer des raisonnements à propos des différentes capacités d'un agent, non-seulement à partir de son état actuel mais également à partir de différents états atteignables par lui pour un niveau d'effort donné. Elle est calculable en temps réel.

Elle se présente sous forme de cartes:

- *carte de Mightability vis-à-vis des lieux* (*mightability map*) (MM) : définit quels sont les lieux qu'un agent peut voir et/ou atteindre si il déploie un effort donné et réalise une action. Elle peut être utilisée par exemple pour trouver l'endroit où un agent peut réaliser une tâche pour/avec un autre agent avec un effort donné.
- *carte de Mightability vis-à-vis des objets* (*object oriented mightability*) (OOM) : réalise la même chose pour les objets. Elle peut être utilisée par exemple pour détecter qu'un agent cache un objet à un autre agent.

La figure 4.8 montre des exemples de ces cartes.

Avec ces informations, le robot peut raisonner sur les lieux atteignables et/ou visibles par les différents agents pour réaliser les tâches avec différents niveaux d'effort (e.g. le moins d'effort pour voir ou pour atteindre un lieu ou un objet). Cette brique de base va servir tout au long de la thèse à prendre en compte l'analyse des capacités des agents vis à vis de différents efforts dans le cadre de prise de décisions ou de planification.

E.6 Analyse d'affordance et Evaluation de la situation

Le chapitre 5 (Affordance Analysis and Situation Assessment) présente la contribution de cette thèse en terme d'analyse des affordances et d'estimation de la situation tel qu'introduit dans le chapitre 3. Ces éléments sont réalisés par l'intermédiaire d'un raisonnement géométrique sur le modèle 3D du monde et peuvent être mis à jour en temps réel.

Nous introduisons le concept d'affordances agent-agent (*agent-agent affordances* qui enrichit la notion d'affordance en incluant l'exécution des tâches entre agent en plus des affordances agent-objet.

Notre notion d'affordance comprend ce qu'un agent peut faire pour les autres agents (donner, montrer, ...); ce que l'agent peut faire avec un objet (prendre, porter, ...); ce qu'il est possible de faire pour un agent vis à vis des lieux (se

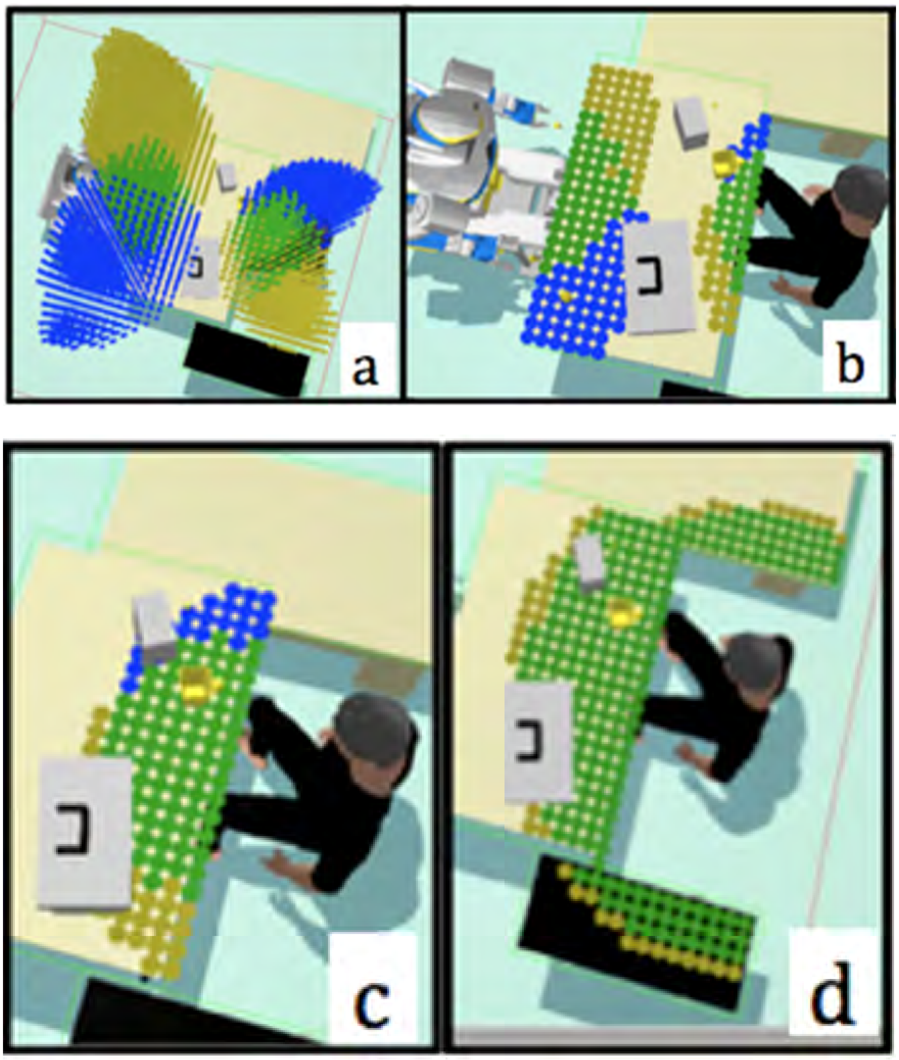


Figure E.4: Carte d'atteignabilité pour un Homme et HRP2 à partir de leur état courant: (a) en 3D et (b) sur le plan de la table. On voit ce qui est atteignable uniquement par la main gauche (jaune), par la main droite (bleu) et par les deux mains (vert). Dans le cas présent, il n'y a pas d'espace commun si aucun des agents ne s'implique et réalise un effort. (c) montre ce qui est atteignable par l'homme si il tend les bras (d) montre ce qui est atteignable par l'homme si il lui est possible de tendre les bras et de se tourner.

déplacer vers, . . .); ce qu'il est possible de faire pour un agent vis à vis des objets (poser dessus, mettre dans, . . .).

La figure E.5 montre les calculs réalisés sur la base des cartes de Mightability. On voit sur le tableau de la figure 5.9 qu'ils permettent une réduction significative de l'espace de recherche (ici dans le cas d'une interaction homme-robot illustrée figure 5.8).

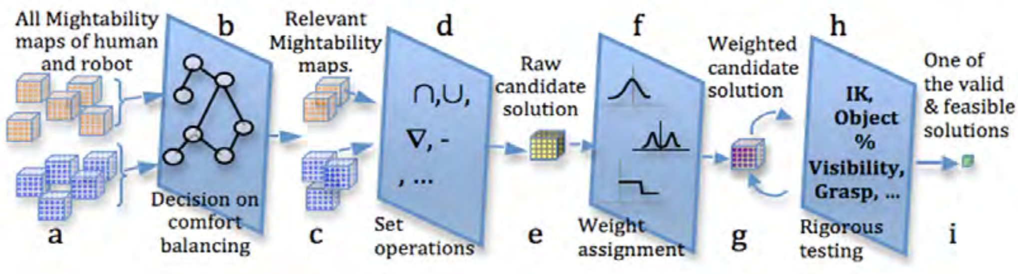


Figure E.5: Etapes pour extraire les lieux possibles pour une affordance agent-agent. (a) cartes de Mightability initiales (b) prise de décision sur le choix de la carte en fonction de la tâche et du niveau de confort choisi pour les agents (c) cartes de Mightability retenues (d) ensemble des opérations spécifiques à la tâche (e) ensemble des solutions possibles (f) ensemble des solutions pondérées par les préférences spatiales (g) ensemble de points candidats pondérés (h) application d'un ensemble complet et rigoureux de tests sur cet espace de recherche réduit (i) solution choisie)

Figure E.6: Interaction face à face entre HRP2 et un homme dans le cadre de réalisation de tâches interactives

Grâce à cela, le robot est capable d'inférer divers états physiques pour chacun des agents et divers états concernant la situation des objets les uns par rapport aux autres.

Dans les chapitres 4 et 5 nous avons instancié les attributs clés pour développer les capacités spatio-visuel du robot, tel que présenté chapitre 3 dans la théorie générale de l'interaction homme-robot. Ces attributs correspondent à la couche inférieure de la pyramide d'incarnation sociale (figure E.1) et vont servir de base aux autres contributions de cette thèse.

Initial Total Number of Cells in Workspace = **144000**	
Task (HRP2 for Human)	**Significantly Reduced Search Space (final number of candidate cells)**
Make bottle accessible	8
	33
Show bottle	414
Hide bottle	42

Figure E.7: Réduction importante de l'espace de recherche dans le cadre des tâches du scénario de la figure5.8

E.7 Navigation et Guidage socialement adaptés en environnement humain

Le chapitre 6 présente notre contribution quand à la navigation du robot.

E.7.1 Planificateur de trajectoire socialement acceptable

Nous définissons un planificateur de trajectoire socialement acceptable (*socially acceptable path planner*) qui permet également la prise en compte de l'aspect dynamique de l'environnement. La figure E.8 montre un scénario qui compare les trajectoires générées par notre planificateur, un planificateur A^* standard et un diagramme de Voronoi. On voit que notre algorithme tire partie des meilleurs caractéristiques du A^* et du diagramme de Voronoi tout en maintenant globalement une trajectoire fluide et une prise en compte des conventions sociales. De plus, comparé à un mouvement réactif, la trajectoire calculée par notre planificateur réduit les conflits et l'inconfort de l'homme.

La figure E.9 montre un exemple d'adaptation des différentes règles sociales dans un autre scénario où le robot navigue dans un environnement humain. Il est à noter que notre implémentation est générique et permet facilement d'adapter la trajectoire par exemple pour un droitier ou un gaucher.

E.7.2 Robot guide

Il s'agit maintenant pour le robot de guider une personne vers un endroit où elle désire se rendre. Pour cela, nous distinguons lors de la réalisation de la tâche les comportements de prise de congé (*leave-taking*) des comportements inverses. En se basant sur ces informations, le robot peut produire les comportements appropriés. Ainsi, il peut prendre en compte les écarts effectués par la personne qui est guidée tandis qu'il permet au robot d'exhiber un comportement de réengagement lorsque cela est nécessaire. Une autre de nos contributions est que dans notre cadre, c'est le but de la tâche qui guide les

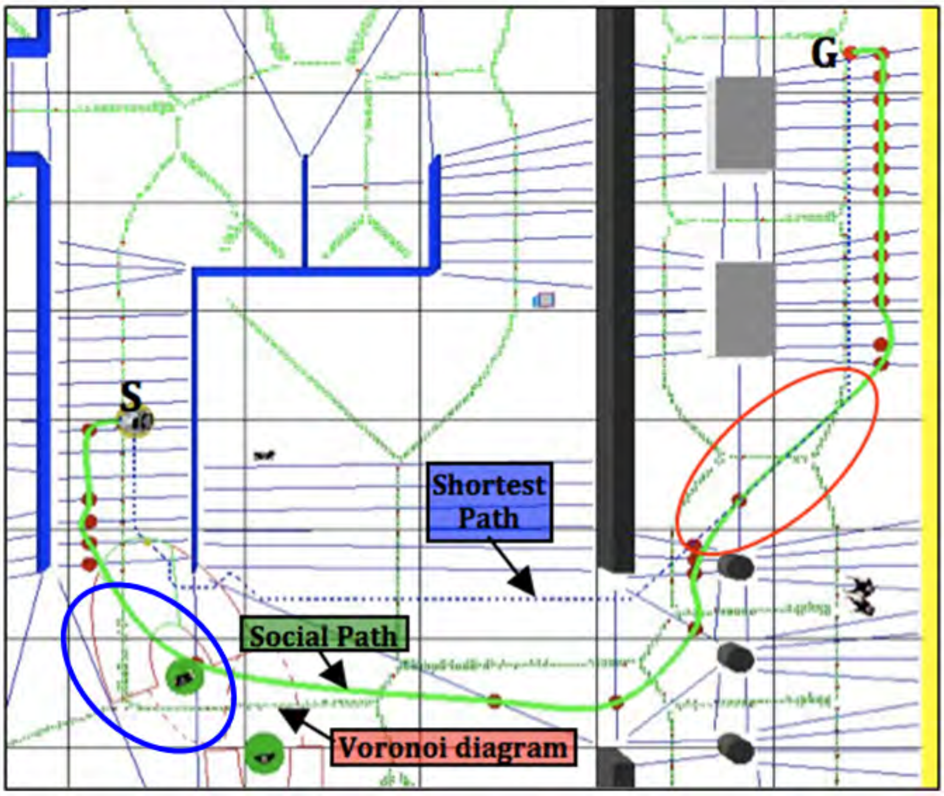


Figure E.8: S est la position de départ et G la position d'arrivée. La trajectoire en vert est celle calculée par notre planificateur, celle en bleue par un planificateur A^* (*shortest path planned by cost grid based A* planner*). Les points verts en pointillés montre le diagramme de Voronoi. Notre planificateur évite les routes les plus longues proposées par le diagramme de Voronoi (par exemple, dans le segment entouré en bleu). De plus, quand cela est possible, le planificateur choisi le plus court chemin (par exemple, dans le segment entouré en rouge). En cas de doute, le planificateur semble suivre le diagramme de Voronoi lui assurant ainsi un maximum d'espace autour de lui.

comportements de réengagements, exerçant ainsi une "pression sociale" sur la personne pour que le but soit réalisé.

La figure E.10 montre la trajectoire initialement calculée pour guider l'homme *H1* vers le but G. La figure E.11 montre l'homme commençant à se déplacer vers un nouveau lieu. Classant cela comme un comportement de prise de congés, le robot planifie une autre trajectoire où il essaie de réengager l'homme vers son but.

Les figures E.12 et E.13 montrent différents cas où l'homme décide de switcher de la droite vers la gauche du robot et décide d'accompagner le robot plutôt que de la suivre. Le robot classifie avec succès cela comme un comportement

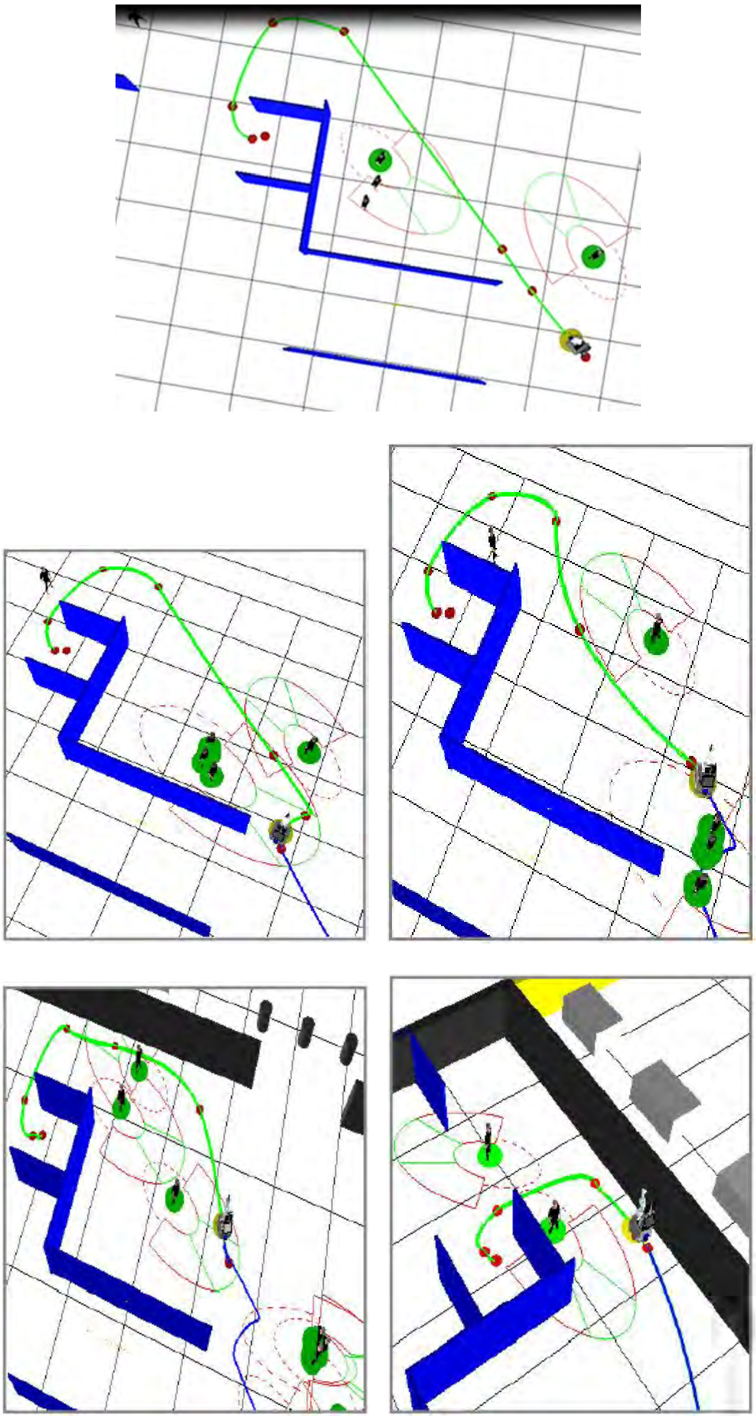


Figure E.9: (a) plan initialement calculé par notre planificateur. (b) Détection d'un groupe, contournement par la gauche du groupe. (c) Détection d'une personne, contournement par la gauche de la personne. (c) Overtaking a person from his left (d) Détection de plusieurs personnes. Contournement par leur droite. (e) Détection d'une personne dans un couloir. Contournement. On notera que l'aspect lisse et régulier des trajectoires recalculées.

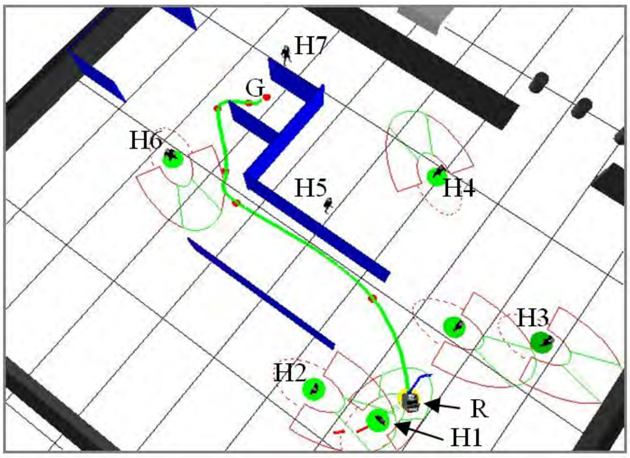


Figure E.10: Le robot Jido (R) doit guider l'homme (H1) vers le but (G). La trajectoire en vert montre le plan calculé par notre planificateur.

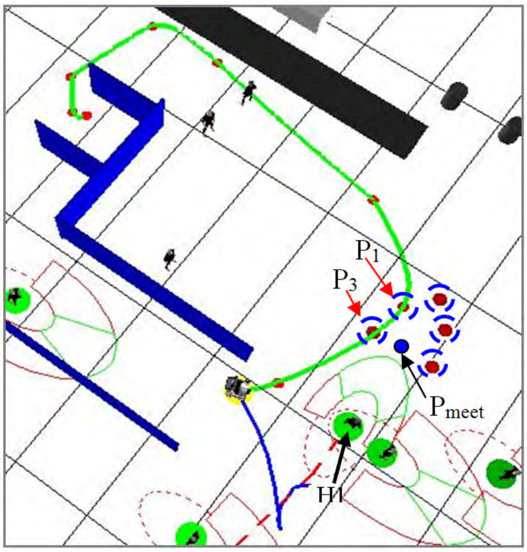


Figure E.11: L'homme suspend la tâche de guidage et le robot calcule un nouveau chemin pour tenter de réengager l'homme en essayant de l'attirer vers le but.

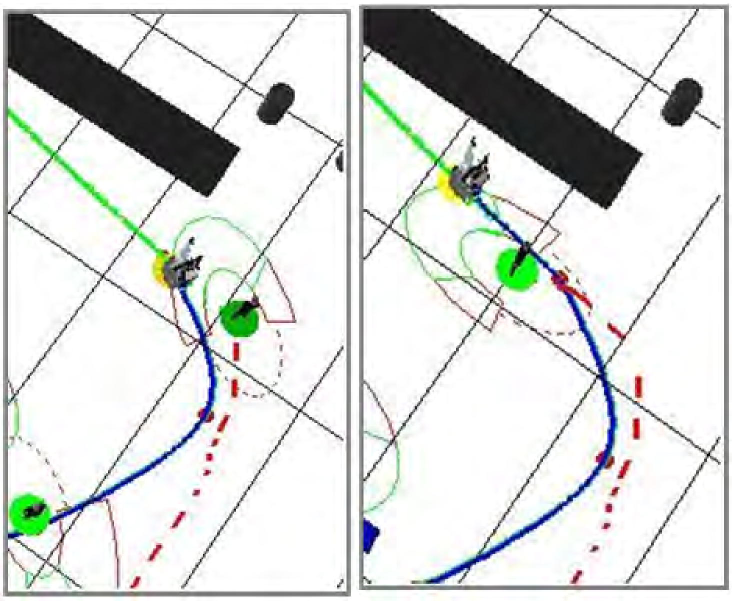


Figure E.12: L'homme switch de la droite vers la gauche du robot. Le robot ne réagit pas.

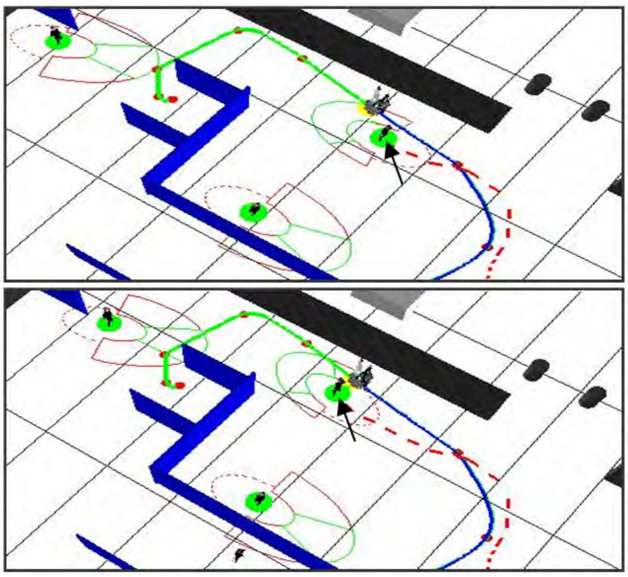


Figure E.13: L'homme indiqué par la flèche noir, passe de derrière le robot à son côté gauche, cela pour accompagner le robot plutôt que le suivre. Le robot ne réagit pas.

de suivi de la tâche (i.e. l'inverse d'un comportement de prise de congés) et ne montre pas de réaction inapproprié. Il fournit ainsi à l'homme la possibilité de le suivre de manière souple.

E.8 Planification de tâches basiques pour l'interaction homme-robot

Le chapitre 7 présente notre contribution quand à la réalisation de tâches de manipulation interactives. Pour cela, nous exploitons l'aspect important de l'interdépendance des positions de saisie.

L'élement clé de notre approche est : *introduire la bonne contrainte au bon niveau.* Elle prend en compte les modèles de planification de mouvement basé sur la posture de l'homme [Rosenbaum 2001], qui suggère d'évaluer chacune des postures et de les éliminer en fonction d'une liste d'exigences (*contraint hierarchy*). Cette méthode par élimination [Tversky 1972] a montré sa pertinence pour la représentation de prise de décision adaptative avec des contraintes multiples [Janis 1996].

Cela sert un autre but important : Au lieu d'introduire toutes les contraintes au début dans un large espace de recherche, cette approche introduit les contraintes petit à petit au moment approprié ce qui réduit significativement l'espace de recherche quand on introduit des contraintes importantes/coûteuses. *Instead of introducing all the constraints at once initially, in the large search space, this approach holds the constraints to be introduced successively at appropriate stages of planning; hence significantly reducing the search spaces before introducing expensive constraints.*

Nous avons choisi attentivement la hiérarchie des contraintes (*contraint hierarchy*) en prenant en compte pour chaque contrainte : son importance, la complexité de son calcul et sa contribution à la réduction de l'espace de recherche. La figure E.14 montre une vue globale du système de planification. La priorité la plus haute est donnée à la minimisation de l'effort de l'homme. Le planificateur extrait la liste des prises candidates GL, des positions possibles (*to-place positions*) PL, des orientations possibles (*to-place orientation*) OL avec un effort de l'homme minimal. Nous introduisons ensuite différentes contraintes *environment*-, *planning*-, *human*- and *task*-oriented à différents niveaux. La figure E.15 détaille l'intérieur du bloc de la figure E.14 et montre la liste des candidats GL (*block 1-A*), PL (*block 4-A*) et PO (*block 5-A*) qui sont extraites.

Nous avons appliqué ce cadre de planification à différentes tâches collaboratives comme : montrer, donner, rendre accessible ainsi que des tâches de

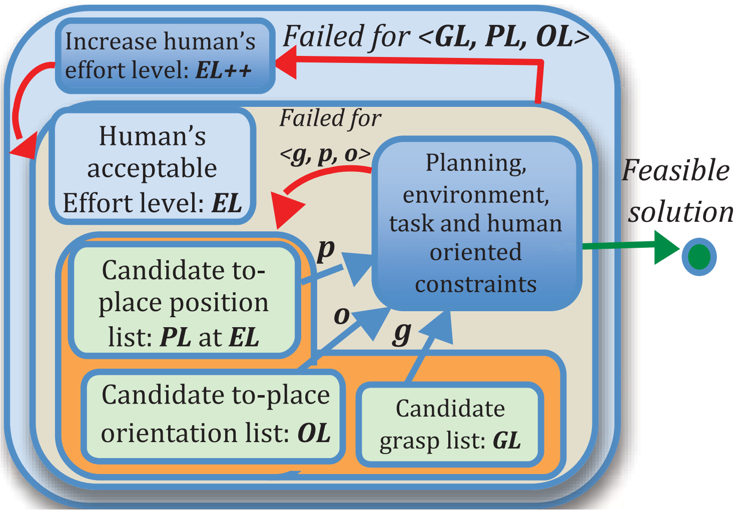


Figure E.14: Vue d'ensemble du système de planification : il y a une itération sur une liste de 3 candidats en cohérence avec les efforts de l'homme pour extraire une solution adaptée.

mise en compétition comme cacher un objet ou le rendre (in)accessible. Ces résultats ont été démontré sur 3 robots : Jido, PR2 et HRP2.

Les figures E.18, E.17, E.16, E.19, E.20 et E.21 montrent différents scénarios dans lesquels le robot réalise différentes tâches.

E.9 Graphe d'affordance: Un cadre basé sur les efforts pour établir l'interaction et la génération de plan partagée

Le chapitre 8 introduit le concept de graphe d'affordance (*Affordance Graph*) qui va permettre de répondre plus rapidement à différents types d'interrogation comme : qui peut faire cela, dans quel but, où, pour qui, avec quel niveau d'effort, etc. Pour cela nous convertissons le problème de recherche d'un plan partagé pour des tâches de manipulation coopérative en un problème de recherche dans un graphe.

E.9.1 Taskability Graph

Le *Taskability Graph* définit ce qu'un agent pourrait être capable de faire pour un autre agent, avec quel niveau d'effort pour chacun des agents et à quel endroit.

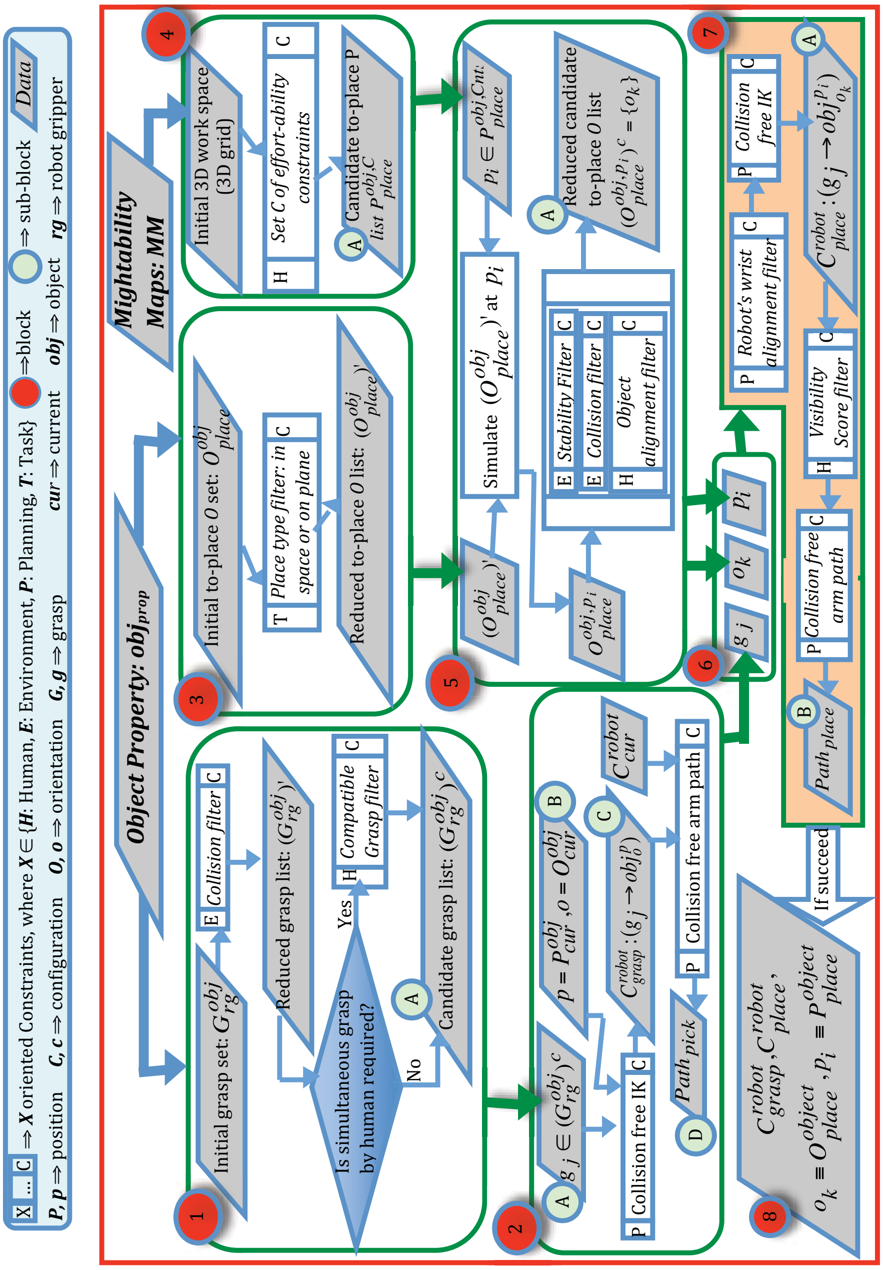


Figure E.15: La partie centrale du planificateur générique présenté, montrant les 4 aspects: (*i*) explique comment la liste des candidats de la figure E.14 sont extraites en *blocks 1-A, 4-A and 5-A*. (*ii*) explique comment le triplet candidat $< \mathbf{grasp} : \mathbf{g}, \mathbf{orientation} : \mathbf{o}, \mathbf{position} : \mathbf{p} >$ (*blocks 6*), est extrait. (*iii*) **utilisation de la hiérarchie de contraintes**: insertion de différentes contraintes à différents niveaux de planification (où l'espace de recherche est réduit significativement). (iv) montre toutes les **Pose** & **Config** requises pour la planification du *pick-and-place* de la figure 7.2 comme résumé dans *block 8*.

Figure E.16: **Tâche : montrer un objet:maximiser la visibilité, maintenir l'objet devant et en haut**: PR2 montrant l'objet avec une orientation qui permet de le rendre le plus visible possible vis à vis de l'homme.

La figure E.23 montre les *taskability graphs* pour 4 tâches: Make Accessible (rendre accessible), Show (montrer), Give (donner) and Hide (cacher) pour le scénario de la figure E.22 pour tous les agents présents dans l'environnement.

La figure E.24 montre une arête d'un *taskability graph* tandis que la figure E.25 en donne l'explication. C'est une arête orientée de l'agent réalisant l'action vers l'agent cible. Les sphères montrent les niveaux d'effort de chacun des agents et le nuage de points montre les lieux candidats où la tâche pourrait se dérouler selon les niveaux d'effort. Dans cet exemple, le *taskability graph* a été utilisé pour équilibrer les efforts. Le fait que l'action se déroule autour de la table a été utilisé pour restreindre les efforts individuels sur Arm_Torso_Effort. C'est pourquoi entre l'homme à droite et le robot il n'y a pas de possibilité de donner *give* ou rendre accessible *make accessible* comme on le voit dans les arêtes manquantes entre les deux agents au niveau du *taskability graph*.

E.9.2 Manipulability Graph

Le *Manipulability Graph* encode ce que l'agent est capable de faire avec un objet, selon un niveau d'effort particulier et à quel endroit (si c'est applicable). Il est complémentaire du *Taskability Graph* qui encode les affordances agent-agent, le *Manipulability Graph* représente les affordances agent-objet.

La figure E.26(a) représente un *Manipulability Graph* qui montre les capacités et les efforts des agents pour prendre *take* les objets et les mettre quelque part *putinto*. Chaque arête du *Manipulability Graph* montre les efforts des agents pour voir et atteindre les objets.

Figure E.17: **Tâche: montrer un objet**:**Assurer le minimum d'effort**: HRP2 montre l'objet à un endroit qui demandera un minimum d'effort à l'homme pour le voir.

E.9.3 Affordance Graph

En combinant le *Taskability Graph* et le *Manipulability Graph*, nous avons développé le concept de *Affordance Graph (AG)*. Il encode toutes les possibilités de manipulation d'un objet par les agents et vers les lieux, avec des informations sur les niveaux d'effort des agents et sur les lieux candidats. La figure E.27 montre le *Affordance Graph* du scénario. Chaque arête a un poids qui dépend des efforts contenus dans les graphes parents: *Taskability Graph* and *Manipulability Graph*.

Ce qui est nouveau avec le *Affordance Graph* c'est que: *(i)* il transforme la planification d'une tâche de manipulation coopérative dans le cadre d'une interaction homme-robot en un problème de recherche dans un graphe, *(ii)* il fournit la possibilité de raisonner sur les niveaux d'effort homme/agents, et *(iii)* il permet l'incorporation de préférences et de contraintes sociales en terme de désir et d'effort acceptable. Ce chapitre a montré également l'exploitation

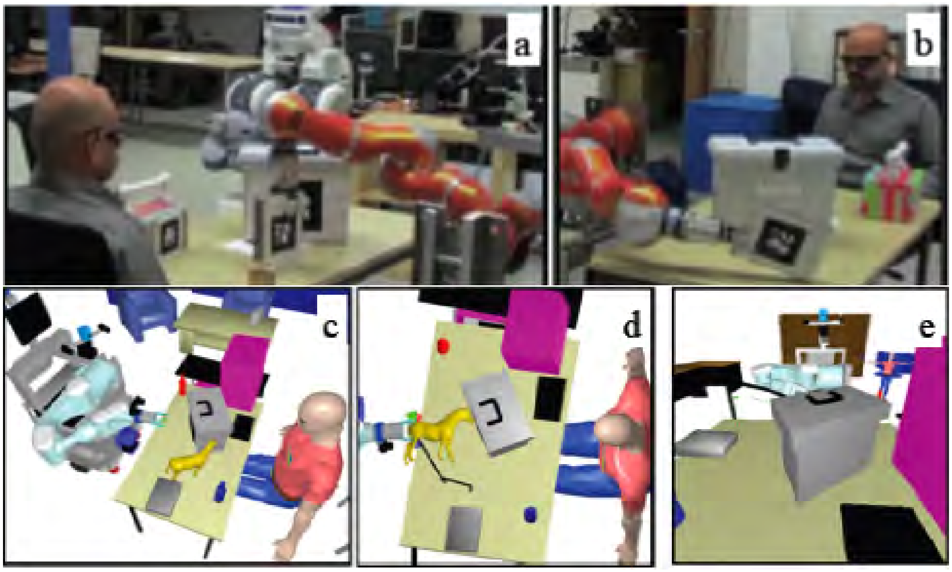


Figure E.18: **Tâche: montrer un objet**: Interdépendance des positions de saisie : (a) et (b) Le planificateur trouve une nouvelle orientation qui rend l'objet invisible pour l'homme. (c)(d)(e) autre scénario, le jouet en forme de cheval est debout et le robot le pose sur le flanc pour le rendre invisible à l'homme.

du lien entre des planificateurs symboliques et des planificateurs géométriques pour l'interaction homme-robot. Il introduit le concept de geometric action level backtracking pour résoudre une tâche planifiée par le planificateur symbolique.

E.10 Comportement pro-social pro-actif

La chapitre 9 présente nos travaux sur la possibilité de doter le robot de comportements pro-actifs *proactivity* et sur le moyen de réguler ce qui serait une pro-activité autorisée *allowed proactivity*.

Les comportements proactifs sont des éléments essentiels pour gérer l'aspect socio-cognitif et son évolution. Ansi équippé, le robot peut réaliser des interactions multi-modales et montrer des capacités de coopération plus riche, il peut développer des comportements sociaux plus complexes en environnement humain. Ces comportements sont variés et posent de nouveaux challenges concernant leur synthèse ou leur exécution. Cependant, nous manquons encore d'un cadre analytique fondant les bases d'une synthèse de ce genre de comportement.

(a)

(b)

(c)

(d)

(e)

Figure E.19: **Tâche : cacher un objet**: **Différentes positions pour cacher, avec un maximum d'effort pour l'homme.**:*(a)-(e) top-down*: (a)-(c) PR2 cache un objet à l'homme assis au milieu. Note, comme on le montre (c), l'objet est complètement caché à la vue de l'homme. (d)-(e) Montre le cas où l'on cache l'objet à l'homme par la droite. Le robot trouve un placement stable pour cacher dans le meuble et l'homme doit mettre beaucoup d'effort pour le voir.

E.10.1 Proposition de niveaux de comportements pro-actifs

Dans ce chapitre, nous proposons tout d'abord un moyen de synthétiser des comportements pro-actifs via la catégorisation de l'espace géométrique et de l'espace d'action. Nous présentons 4 niveaux de comportement pro-actifs, comme le montre la figure E.28. Cette catégorisation peut être utilisée pour:

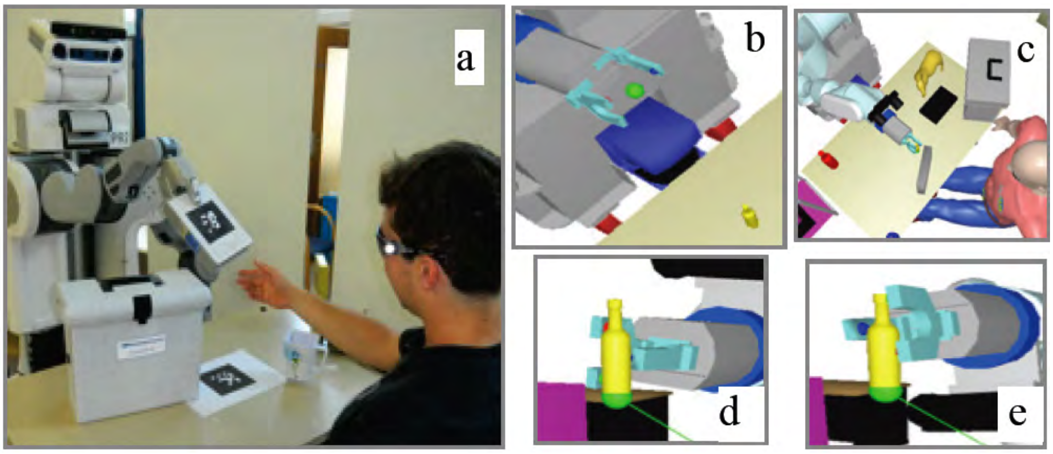


Figure E.20: **Tâche donner : Maintenir les features symboliques**: (a) PR2 donne un objet en maintenant l'objet devant l'homme. (b)-(e) **Double saisie simultanée**: Jido donne la bouteille jaune. (b) scenario initial. (c) configuration finale. A partir de la vue de l'homme: (d) sans introduire la contrainte de la double saisie et (e) avec. Note dans (e) Jido saisit la partie haute de la bouteille car il a analysé la géométrie de la bouteille et calculé la position de grasp pour l'homme.

(i) raisonner sur les lieux où il est possible de réaliser des comportements pro-actifs, (ii) réguler la productivité autorisée (*allowed proactivity*) de l'agent, (iii) fournir un moyen de mesurer l'influence du comportement pro-actif.

E.10.2 Instanciation de comportement pro-actifs

Nous avons également travaillé sur l'instanciation de comportements pro-actifs basée le lieu où la tâche peut être réalisée. Cela nous a permis de développer les comportements suivants: (i) le robot adopte une démarche proactive dans le cadre d'une tâche où l'homme doit donner un objet au robot, (ii) le robot suggère de manière proactive l'endroit où poser un objet dans le cadre d'une tâche où l'homme doit rendre un objet accessible à un autre homme. Les figures E.29 et E.30 montrent les résultats de ces scénarios sans et avec comportement pro-actifs. Dans ce dernier cas, on voit la diminution de l'effort de l'homme.

E.10.3 Etudes utilisateur

Nous avons également validé à travers des études utilisateurs que ces comportements pro-actifs minimisent les efforts de l'homme et les possibilités de confusion dans le cadre d'une tâche collaborative. Les figures E.31, E.32, et E.33 résument ces études. Une majorité des utilisateurs a trouvé que les

Figure E.21: **Tâche : rendre accessible**:**Placement stable sur un autre objet, assurant un effort minimal de l'homme**:*(a)-(f) row-wise*: PR2 rend accessible de manière séquentielle deux objets à l'homme. Le premier objet sur la droite n'est pas accessible par l'homme de sa position et le second objet est derrière la boîte d'un point de vue de l'homme et visible de l'homme à partir de sa position courante. La planificateur trouve de manière autonome un placement pour le 1er objet (b). Il trouve également un placement stable pour le second sur la boîte (e) ce qui assure un effort minimum pour l'homme pour le prendre Arm_Torso_Effort (e) et (f).

comportements proactifs du robot ont réduit leur confusion et leurs efforts et permet un meilleur accomplissement de la tâche.

E.11 Compréhension de tâche par démonstration

Dans le chapitre 10 nous tentons de comprendre les tâches effectuées au jour le jour par le robot en terme d'effets désirés et cela à différents niveaux

Figure E.22: Scénarion d'interaction homme-robot autour d'une table.

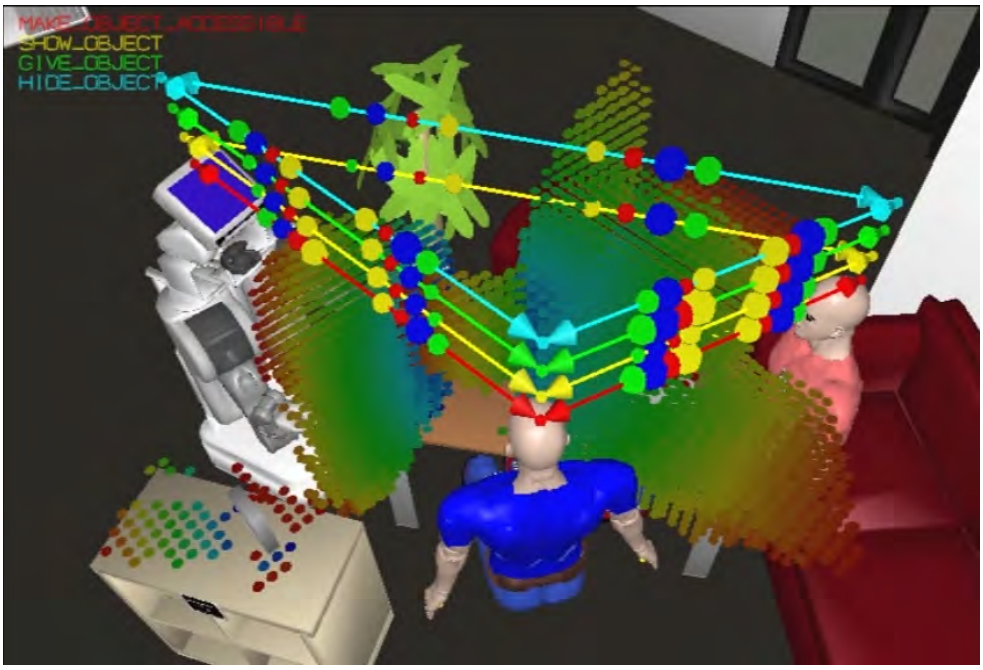


Figure E.23: Taskability Graph pour différents tâches équilibrant les efforts entre les statuts sociaux et les niveaux d'effort comme *Arm Torso Effort*

d'abstraction. Le but serait de permettre au robot de réaliser la même tâche dans différentes situations et même de transmettre ce savoir à un autre agent.

Premièrement, nous enrichissons la base de connaissances du robot avec une hiérarchie de faits : quantitatifs, qualitatifs, comparatifs liés à l'homme et à l'objet. Nous avons montrés que sans ces éléments clés, beaucoup de tâches

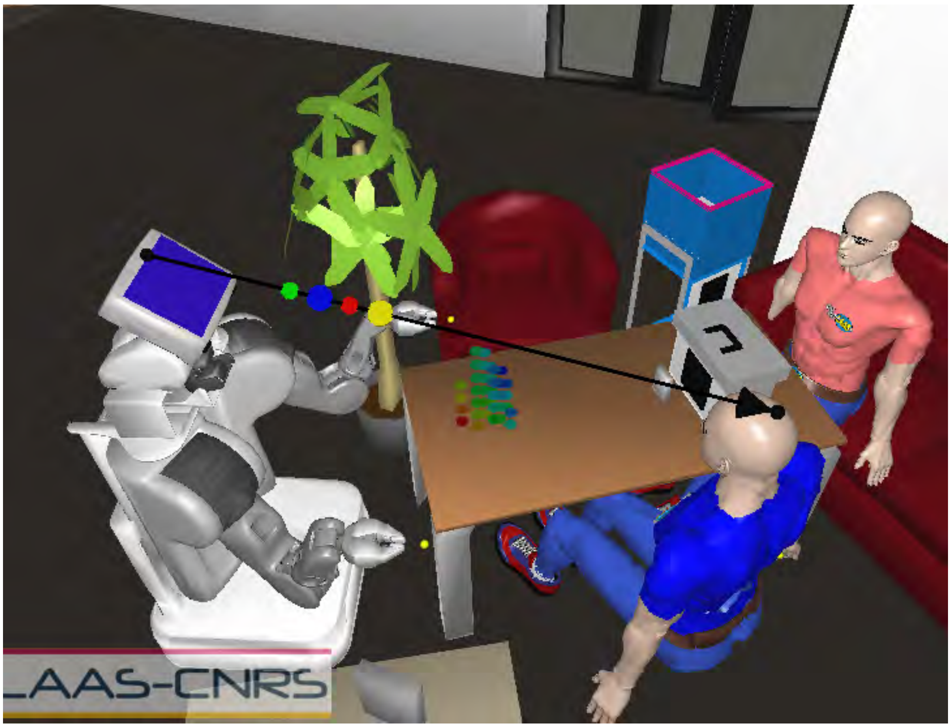
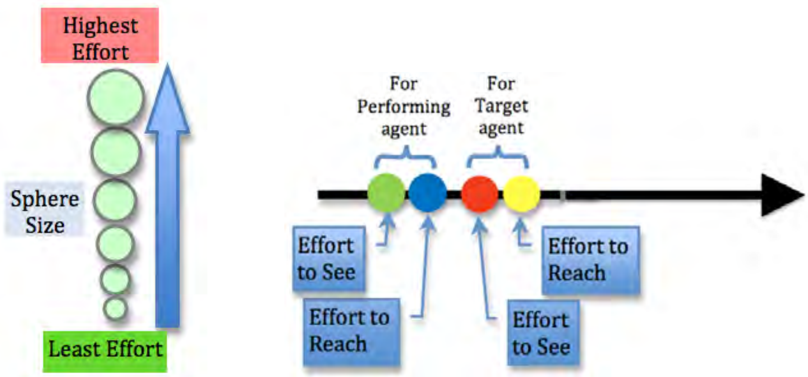


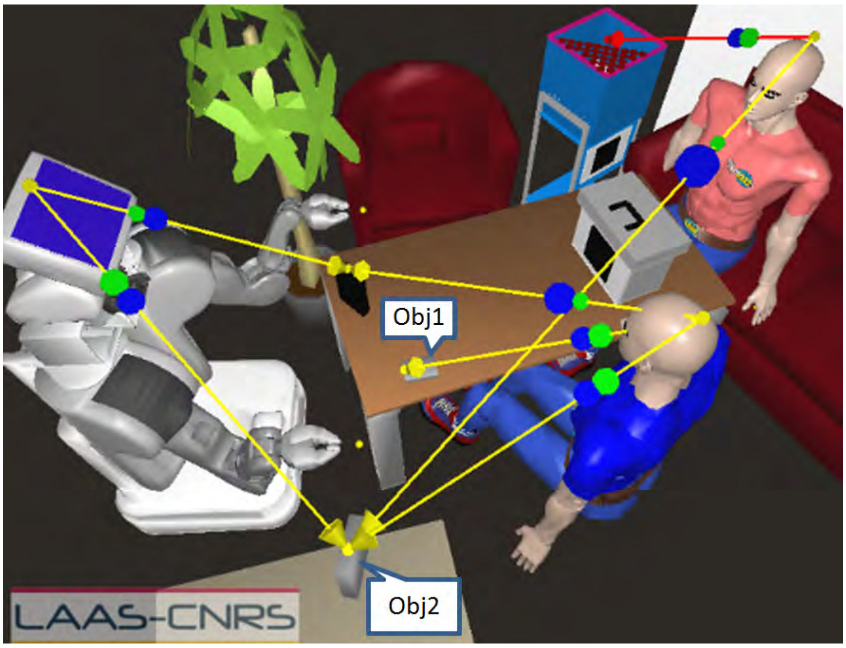
Figure E.24: Exemple d'une arête d'un taskability graph



(a) Effort Sphere (b) Edge Description

Figure E.25: Explication d'une arête d'un taskability graph

effectués au jour le jour ne peuvent pas être comprise correctement. En utilisant ces éléments, nous pouvons construire un arbre d'hypothèses qui pourra être utilisé pour l'apprentissage.



(a) Manipulability Graph pour saisir un objet

(b) Description d'une arête

Figure E.26: Manipulability Graph pour saisir des objets avec un niveau d'effort maximal basé sur les mêmes règles sociales utilisés pour fig.8.4

E.11.1 Apprentissage via l'explication et l'utilisation d'un arbre d'hypothèses initiales

Nos travaux se basent sur les notions d'apprentissage par l'explication *explanation-based learning*, [Wusteman 1992], [Flann 1989] et *m-estimate based refinement*. Notre cadre intègre une hiérarchie de faits et apprend automatiquement la sémantique des tâches à un niveau correcte d'abstraction de la manière suivante :

- (i) Contruction d'un arbre d'explication pour chaque exemple de la tâche
- (ii) Comparer les arbres pour trouver le sous-arbre le plus important
- (iii) Construire la clause de Horn utilisant les feuilles du sous-arbre le plus large pour trouver la règle la plus générale.

Une fois déplié, le domaine devient un *general initial hypothesis space* comme montré E.34.

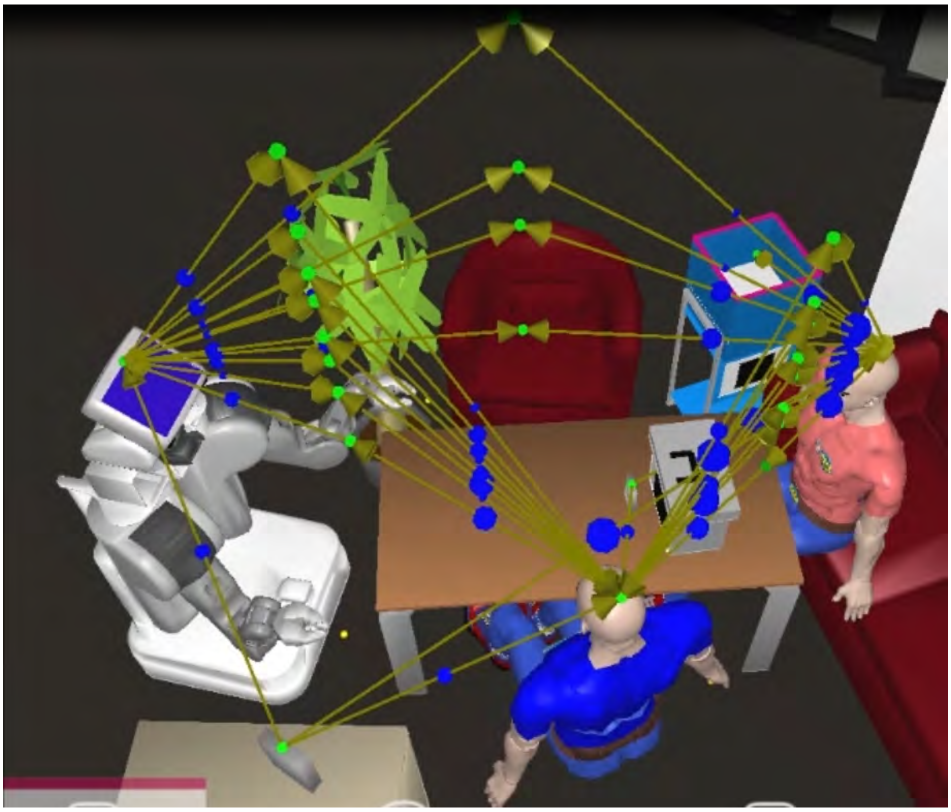


Figure E.27: Affordance Graph

Level of Proactive Action A^{pro}	Allowed changes (*s*: set of grounded facts, *S*: space in which the fact variables should be grounded, *A*: action)				Intentions from task's perspective	Intention/ nature of A^{pro}	Influence on *environment* sub-spaces	Desirable level of reliability of the proactive agent	Severity of A^{pro}
	can alter the goal state part	can alter the current plan part	can expand the goal state space and action space parts	can reduce the goal state space and action space parts					
Level-1	s^g_{cn} , s^g_{uc}	A_{pa}			smooth, problem free achievement	responsive, alterning	better value for un-grounded part	Average	Lowest
Level-2		$parm(A_{oa})$	S_{ud} , A_{res}	S_{uc} , S_{cn}	better achievement	suggestive, influencing, adjusting, fine-tuning	better specify ungrounded space		
Level-3	s^g_{gr}	A_{pa} , A_{oa}	A_{res}		better alternative	imposing	alter desired part		
Level-4				S_{ud} , A_{res}	avoid severer problem	obligating, risking	alter some undesired parts	Highly reliable	Highest

Figure E.28: Résumé des différents niveaux d'actions pro-actives présentées dans ce chapitre.

Figure E.29: Scénario pour une tâche où l'homme doit donner un objet au robot. (a) En l'absence de comportement pro-actif, l'homme est debout et vient vers le robot (*Whole_Body_Effort*) pour donner l'objet. (b) Avec un comportement pro-actif, l'homme donne l'objet au robot en faisant uniquement un mouvement du bras *Arm_Effort*.

Figure E.30: Tâche de rendre un objet accessible pour le robot par l'homme (en le déplaçant au bon endroit pour que le robot puisse l'attraper. (a) En l'absence de comportement pro-actif, l'homme le pose près du robot sur la table, cela nécessite *Whole_Body_Effort* pour l'homme. (b) Le robot utilise le comportement pro-actif pour trouver une solution où l'homme peut mettre un minimum d'effort. Ainsi, l'homme pose l'objet sur la boîte blanche comme suggéré par le robot et les efforts de l'homme passe de *Whole_Body_Effort* à *Arm_Torso_Effort*.

Reduction in the human's confusion because of the robot's proactive behaviors	
For give task by the human	70 %
For make accessible task by the human	62 %
Overall by combining both tasks	***66 %***

Figure E.31: Diminution du trouble des utilisateurs dans le cas d'un comportement pro-actif.

E.11.2 Facteur de cohérence

Le robot trouve automatiquement si un prédicat p est pertinent ou non, cette analyse se base sur la valeur du prédicat. Si les valeurs sont tout le temps différentes, cela signifie que le prédicat n'est peut être pas pertinent pour la tâche. De plus, nous statuons que v_h est la valeur du prédicat p ayant le m-

Reduction in the human's effort because of the robot's proactive behaviors	
For give task by the human	70 %
For make accessible task by the human	60 %
Overall by combining both tasks	***65 %***

Figure E.32: Diminution des efforts des utilisateurs dans le cas d'un comportement pro-actif.

Total % of users explicitly reported that the robot has better communicated its capabilities and was more supportive to the task and the user in proactive behaviors	***85*** %

Figure E.33: Obtention d'une meilleure communication et d'un meilleure accompagnement de l'homme.

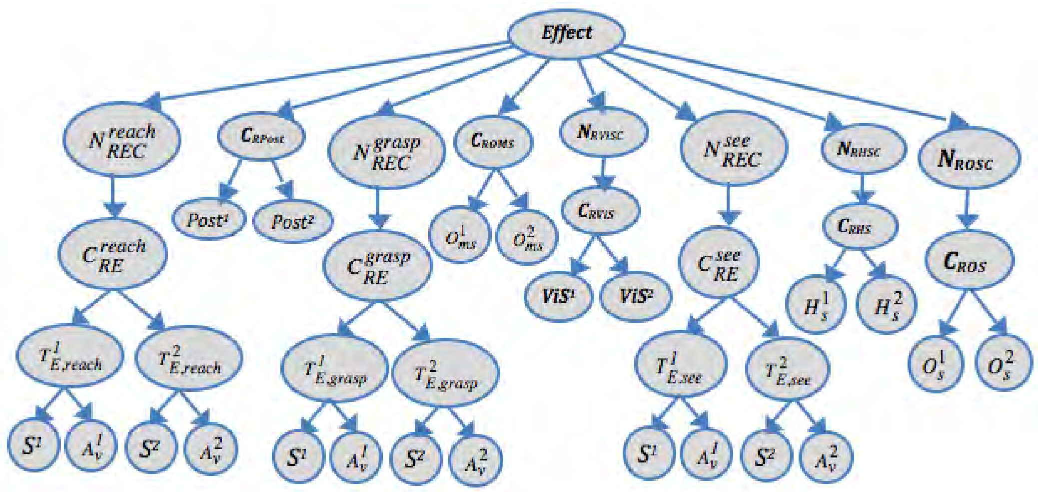


Figure E.34: Espace initial des hypothèses pour la compréhension de la sémantique de tâches.

estimate le plus élevé. Ainsi, pour un prédicat p, pris sur N démonstrations, N_p valeurs différentes $\{v_1, v_2, v_3, ...v_{Np}\}$ ont été observées. Nous définissons le facteur de cohérence *consistency factor (CF)* de p pour la tâche T pour décider de la pertinence de p ainsi :

$$CF_p^T = \overbrace{Q_p^{v_h,T}}^{\text{relevance evidence}} - \underbrace{\sum_{i=1 \wedge i \neq h}^{Np} Q_p^{v_i,T}}_{\text{non-relevance evidence}} \tag{E.1}$$

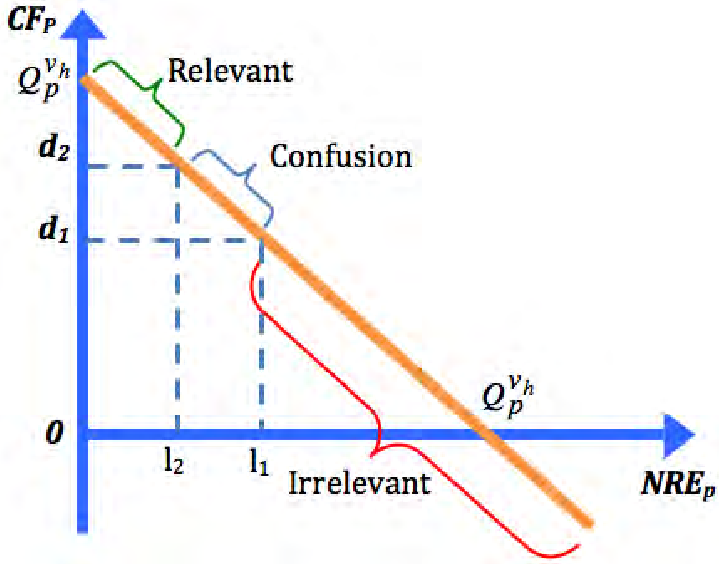


Figure E.35: Choix de la pertinence ou non d'un prédicat (ou sinon de la confusion associée).

A droite de l'équation, nous trouvons la pertinence de p. Plus sa valeur est élevée, plus la possibilité que la valeur la plus observée v_h, pour p fait partie de l'effet désiré pour la tâche T. La seconde partie donne la possibilité d'obtenir une valeur observée autre que v_h. Cela représente en fait, la *non-relevant evidence* de p, NRE_p, parce que plus cette valeur est élevée, plus faible est la probabilité que p ait une valeur cohérente.

En se basant sur ce facteur de cohérence (*consistency factor* après chaque démonstration, un prédicat p pour une tâche T peut prendre 3 valeurs (voir figure E.35): *Contradiction, so irrelevant p*, *Consistency, so relevant p* and *Confusion, so ask for clarification.*

La première tâche montré au robot a été une tâche où un homme devait montrer *show* un objet à un autre homme. Les figures E.36(a)-(d) montre les scénarios de quatre démonstrations de la tâche. Nous obtenons alors la perspective de l'agent pour lequel la tâche est réalisée de la manière suivante

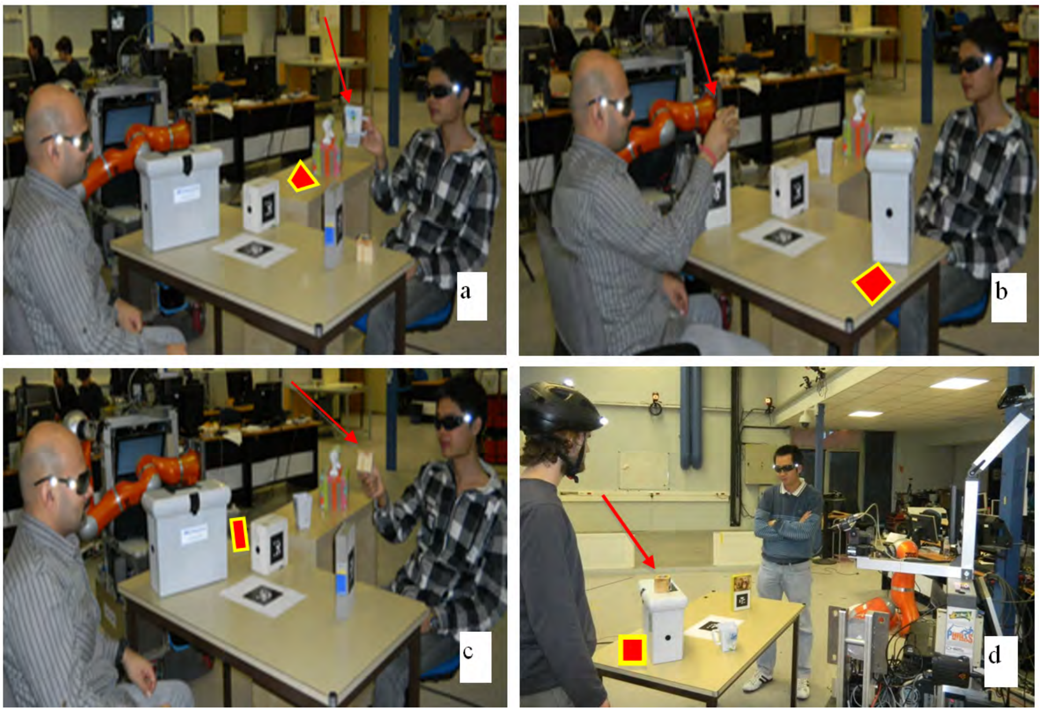


Figure E.36: tâche de montrer un objet dans un cadre collaboratif homme-homme. Les positions initiales de l'objets sont représentés en rouge. (a) l'humain de droite montre la tasse en la portant (b) L'homme de gauche montre le cube de bois en le portant. (c) L'homme de droite montre le cube en bois en le portant. (d) l'homme de gauche rend l'objet visible en le posant sur la boîte blanche.

:

$$\begin{aligned} Task\,(Show\_Object) \leftarrow (Relative\_Posture = Maintained)\wedge \\ (Object\_Initial\_Motion\_Status = Static)\wedge \\ (Object\_Final\_Motion\_Status = Static)\wedge \\ (Object\_Relative\_Visibility\_Score = Increased)\wedge \\ (Action\_to\_See = No\_Action\_Required)\wedge \\ (Initial\_Hand\_Status = Object\_Free)\wedge \\ (Final\_Hand\_Status = Object\_Free) \end{aligned} \tag{E.2}$$

Les tâches cacher *hide*, rendre accessible *make accessible*, donner *give*, put-away *éloigner*et hide-away *cacher* ont également été étudié dans ce cadre.

Par exemple dans le cas de la tâche cacher *hie*, nous obtenons :

$$\begin{aligned}
Task\,(Hide_Object) \leftarrow (Human_Initial_Posture = Sitting)\wedge \\
(Human_Final_Posture = Sitting)\wedge \\
(Object_Initial_Motion_Status = Static)\wedge \\
(Object_Final_Motion_Status = Static)\wedge \\
(Object_Final_Visibility_Score \approx 0)\wedge \\
(Relative_Effort_Class_to_See = Becomes_Difficult)\wedge \\
(Initial_Hand_Status = Object_Free)\wedge \\
(Final_Hand_Status = Object_Free)\wedge \\
(Object_Initial_Status = On_Support)\wedge \\
Object_Final_Status = On_Support)
\end{aligned} \tag{E.3}$$

Nous obtenons le bon niveau d'abstraction et nous notons que les principales différences entre les tâches ont été capturées, ainsi pour la tâche de cacher *hide*, il devient plus difficile de voir l'objet pour l'agent qui doit le recevoir.

Pour la tâche rendre accessible *make accessible*:

$$\begin{aligned}
Task\,(Make_Accessible) \leftarrow (Relative_Effort_to_Reach = Becomes_Easier)\wedge \\
(Relative_Posture = Maintained)\wedge \\
(Object_Initial_Motion_Status = Static)\wedge \\
(Object_Final_Motion_Status = Static)\wedge \\
(Relative_Effort_to_Grasp = Becomes_Easier)\wedge \\
(Object_Relative_Visibility_Score = Increased)\wedge \\
(Nature_Effort_Class_to_See = Supportive)\wedge \\
(Initial_Hand_Status = Object_Free)\wedge \\
(Final_Hand_Status = Object_Free)\wedge \\
(Object_Initial_Status = On_Support) \wedge (Object_Final_Status = On_Support)
\end{aligned} \tag{E.4}$$

Une observation intéressante dans ce cas, c'est qu'il ne filtre pas les prédicats d'atteignabilité *reachability* ou la capacité de saisie *graspability* comme non-pertinent comme se fut le cas pour les tâches précédentes. Au contraire, il trouve qu'atteindre ou saisir l'objet par l'agent qui doit le recevoir est facilité. Ainsi, comme pour les autres tâches, l'effet désiré à été compris à un niveau d'abstraction approprié avec les prédicats pertinents et leurs valeurs désirées.

E.11.3 Bénéfices et applications possibles

(i) *Généralisation à un nouveau scénario*: La compréhension d'une tâche dépend de la forme et de la taille de l'objet, de la trajectoire et des distances absolues et relatives entre les agents et les objets. Cela permet au robot de réaliser la tâche dans un scénario complètement différent.

(ii) *Une plus grande flexibilité du planificateur symbolique et des plans coopératifs*: La compréhension de la sémantique des tâches est réalisée à un niveau symbolique, indépendant de l'exécution, et donc il peut planifier une tâche de différentes manières. Par exemple, si il "comprend" que cacher un objet signifie qu'il devient difficile à voir pour l'agent qui doit le recevoir, il peut décider en fonction de la situation de cacher l'objet dans un meuble par ex pour le rendre invisible sans avoir à manipuler l'objet lui-même.

(iii) *Transfert de savoir entre des agents hétérogènes* Etant donné que la compréhension de la tâche est indépendante de la structure cinématique de l'agent, cette connaissance peut facilement être transféré à un autre robot ayant un structure cinématique ou une forme différente.

(iv) *Généralisation à plusieurs agents* Un tel niveau de compréhension symbolique peut permettre de généraliser la tâche à plusieurs agents et ainsi e.g. cacher un objet à deux hommes en même temps.

(v) *Faciliter la reconnaissance de tâche/d'action et comportement pro-actif* Etant donné que le robot connaît la "signification" de la tâche, il peut inférer la tâche grâce à une observation même partiel des activités et peut même calculer des comportements pro-actifs pour faciliter l'accomplissement des effets désirés de la tâche.

(vi) *Enrichir l'interaction homme-robot* Cette connaissance de la tâche peut également enrichir l'interaction en terme de verbalisation de l'interaction avec l'homme, pour cela, il faut que le robot soit capable de communiquer le niveau d'abstraction de la tâche compréhensible par l'homme.

E.12 Conclusion

The focus of the thesis is bottom up embodiment of social and human aware factors and abilities for the robot's 'development' i.e. Cette thèse présente une approche incrémentale pour la prise en compte des facteurs sociaux et humains dans le développement des capacités et comportements d'un robot:

Nous avons commencé par identifier les capacités cognitives et comportementales nécessaires pour que le robot co-existe avec l'homme dans son environnement d'une manière socialement intelligente (i.e. socialement acceptable) et attendue. Pour cela, nous nous sommes inspirés des recherches sur le développement de l'enfant et en psychologie comportementale. Ensuite, nous avons présenté une théorie pour l'interaction homme-robot et l'avons dérivé et instancié selon différents aspects socio-cognitifs et human-aware pour développer des cadres pour la navigation, la manipulation, la coopération, la pro-activité et l'apprentissage pour les robots.

A côté de ces développements à différents niveaux de la pyramide présentée dans E.1, nous avons développé de nouveaux concepts comme : l'analyse de Mightability, les affordances Agent-Agent, les graphes d'Affordance, Geometric task space backtracking, le mélange d'un planificateur symbolique et d'un planificateur géométrique, la notion de pro-activité pour le robot, et montré comment ces éléments étaient importants et utiles pour une interaction homme-robot efficace.

Le noyau de cette thèse est la construction incrémentale d'une incarnation sociale pour le robot. La motivation est de fournir les bases pour le développement de comportements socio-cognitifs plus complexes pour le robot, avec l'ambition qu'un jour la vision de *Manava* introduite au début de ce chapitre devienne réalité. Notre vision incrémentale sert cette vision et fournie des éléments pour explorer et faire grandir cet agent intelligent qu'est le robot.

Do we expect the existence of "*Robot Being*"? Will we accept that? ...