# Reusing A Robot's Behavioral Mechanisms to Model and Manipulate Human Mental States

by

Jesse Vail Gray

S.M., Massachusetts Institute of Technology (2004)
B.A., Tufts University (2002)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010



Author ______________________________

Program in Media Arts and Sciences
June 4, 2010

Certified by ______________________________

Cynthia Breazeal
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ______________________________

Pattie Maes
Associate Academic Head
Program in Media Arts and Sciences

**Reusing A Robot's Behavioral Mechanisms to Model and Manipulate Human Mental States**

by

Jesse Vail Gray

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on June 4, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

## Abstract

In a task domain characterized by physical actions and where information has value, competing teams gain advantage by spying on and deceiving an opposing team while cooperating teammates can help the team by secretly communicating new information. For a robot to thrive in this environment it must be able to perform actions in a manner to deceive opposing agents as well as to be able to secretly communicate with friendly agents. It must further be able to extract information from observing the actions of other agents.

The goal of this research is to expand on current human robot interaction by creating a robot that can operate in the above scenario. To enable these behaviors, an architecture is created which provides the robot with mechanisms to work with hidden human mental states. The robot attempts to infer these hidden states from observable factors and use them to better understand and predict behavior. It also takes steps to alter them in order to change the future behavior of the other agent. It utilizes the knowledge that the human is performing analogous inferences about the robot's own internal states to predict the effect of its actions on the human's knowledge and perceptions of the robot. The research focuses on the implicit communication that is made possible by two embodied agents interacting in a shared space through nonverbal interaction.

While the processes used by a robot differ significantly from the cognitive mechanisms employed by humans, each face the similar challenge of completing the loop from sensing to acting. This architecture employs a self-as-simulator strategy, reusing the robot's behavioral mechanisms to model aspects of the human's mental states. This reuse allows the robot to model human actions and the mental states behind them using the grammar of its own representations and actions.

Thesis Supervisor: Cynthia Breazeal

Title: Associate Professor of Media Arts and Sciences, Program in Media Arts and Sciences

**Reusing A Robot's Behavioral Mechanisms to Model and Manipulate Human Mental States**

by

Jesse Vail Gray

The following people served as readers for this thesis:

Thesis Reader ______________________________

Bruce Blumberg

Thesis Reader ______________________________

Deb Roy
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

# Acknowledgements

I want to thank my committee, Cynthia Breazeal, Deb Roy, and Bruce Blumberg, for all of their insights and guidance through this process, as well as their support and patience, which made this thesis possible.

I feel extremely fortunate that Bruce and Bill Tomlinson gave me my first chance back when, and that Cynthia then proceeded to keep me exceptionally well supplied with a surplus of good ideas, robots, and an unbelievable environment for combining the two.

Special thanks to Matt Berlin for getting me into the building that critical first time, all of the assistance and support after that, and then making sure I eventually finished!

I want to thank the entire robots group for being so unbelievably awesome: Mikey Siegal, Jun Ki Lee, Sonia Chernova, Dan Stiehl, Jeff Lieberman, Philipp Robbel, Angela Chang, Matt Berlin, Sigurður Örn, Kenton Williams, Ryan Wistort, Jin Joo Lee, David Robert, Andrea Lockerd Thomaz, Cory Kidd, Zoz Brooks, Matt Hancher, John Mcbean, Jason Alonso, Guy Hoffman, Heather Knight, Rob Toscano, and Polly Guggenheim!

I feel lucky to have met and worked with all of you. Guy, Andrea, Zoz, and Matt participated in an early collaboration that set the tone for my time at the lab (crew Leo! RL-BP/1). Thanks to Amy for the stats consult, Mikey for the 13k mile robot adventure, Daphna Buchsbaum for pushing through our first paper, and Marc Downie for demonstrating a new kind of extreme. Thanks in advance to Siggi and Jin Joo for pushing Nexi into the future!

Finally to my family - I love you all, and good work!

# Contents

**Abstract** **3**

**1 Introduction** **15**
1.1 Strategy . . . . . 18
1.1.1 Underlying Architecture Overview . . . . . 19
1.1.2 Integration with this architecture . . . . . 20
1.1.3 Research Structure . . . . . 20
1.1.4 Research Scope . . . . . 22

**2 Background and Related Work** **25**
2.1 Theory of Mind in Humans . . . . . 25
2.2 Why use Simulation Theory Ideas for Robots? . . . . . 27
2.3 Related Robots/Agents . . . . . 29
2.3.1 Embodied Architectures . . . . . 29
2.3.2 Motor Mapping and Imitation . . . . . 31
2.3.3 Statistical Activity Recognition . . . . . 32
2.3.4 Visual Perspective Taking . . . . . 32
2.3.5 Using Perspective to Resolve Ambiguity . . . . . 33
2.3.6 HAMMER . . . . . 34
2.3.7 Experience-Based Perspective Taking . . . . . 35
2.3.8 Soar . . . . . 36
2.3.9 BDI For Teamwork . . . . . 38
2.3.10 Causal Bayesian Networks . . . . . 40

**3 Determining Actions from Observation** **43**
3.1 The Motor System . . . . . 43
3.2 Body mapping . . . . . 44
3.3 Matching observed actions to motor repertoire . . . . . 47

**4 Visual Perspective Taking** **51**
4.1 The Perception and Belief Systems . . . . . 51
4.2 Percept modeling . . . . . 52
4.3 Belief modeling . . . . . 52
4.4 Belief inference and visual perspective simulation . . . . . 55

**5 Inferring Intentions** **61**
5.1 Task Representation using Schemas . . . . . 62
5.2 Generating Goal-Achieving Behavior . . . . . 64
5.3 Inferring Intent from Observed Behavior . . . . . 65
5.4 An Example: Goal Assistance . . . . . 68
5.5 Providing Assistance on a Physical Task . . . . . 71
5.5.1 Benchmark Tasks . . . . . 71
5.5.2 Human Subjects Study . . . . . 73
5.5.3 Robot Experiment . . . . . 77
5.5.4 Summary . . . . . 80

**6 Modifying Mental States** **81**
6.1 Improvements made in realtime monitoring . . . . . 82
6.1.1 Recursive Monitoring . . . . . 82
6.1.2 Re-Imagining from Beliefs . . . . . 83
6.1.3 Minimal Robots . . . . . 85
6.2 Making predictions about future mental states . . . . . 85
6.2.1 Differences From Realtime Robot . . . . . 89
6.2.2 Connecting Hypothetical Robot to Real robot . . . . . 90
6.2.3 Updating Object Beliefs . . . . . 92
6.3 Changing the future . . . . . 95
6.3.1 Search the Future . . . . . 97
6.3.2 Representation of Mental-State-Goals . . . . . 98
6.3.3 Mapping between Real/Hypothetical robot for planning, goals . . . . . 102
6.4 Evaluation . . . . . 104
6.4.1 Demonstration Scenario . . . . . 105
6.4.2 Human Subjects Study Design . . . . . 107
6.4.3 Study Results and Discussion . . . . . 110

**7 Conclusion** **117**
7.1 Contributions . . . . . 117
7.2 Insights and Future Work . . . . . 117
7.2.1 Embodiment . . . . . 118
7.2.2 Simulation and Re-use . . . . . 120
7.2.3 Uncertain World . . . . . 121
7.2.4 Scalability . . . . . 123

# List of Figures


3-1 Motor Mapping Training and Execution . . . . . . . . . . . . . . . . . . . . . . 44
3-2 Example Training set - Training poses for Leo's right arm. . . . . . . . . . . . 46
3-3 Posegraph Motor Representation . . . . . . . . . . . . . . . . . . . . . . . . . 48
3-4 Demonstration of motor mimicry . . . . . . . . . . . . . . . . . . . . . . . . . 49

4-1 Perspective Transform of Sensor Data for Belief Modeling . . . . . . . . . . . 56
4-2 Re-use of Robot's Belief Maintenance Architecture . . . . . . . . . . . . . . . 57
4-3 Timeline demonstrating belief modeling . . . . . . . . . . . . . . . . . . . . . 57
4-4 False belief task demonstration . . . . . . . . . . . . . . . . . . . . . . . . . 59

5-1 Example task representation . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
5-2 Section of action schema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
5-3 Example goal inference and helpful behavior . . . . . . . . . . . . . . . . . . . 70
5-4 The four collaborative benchmark tasks . . . . . . . . . . . . . . . . . . . . . 72
5-5 Setup of human subjects study . . . . . . . . . . . . . . . . . . . . . . . . . . 75
5-6 Setup of the human-robot study . . . . . . . . . . . . . . . . . . . . . . . . . . 78
5-7 Walkthrough of the human-robot study . . . . . . . . . . . . . . . . . . . . . . 79

6-1 Object belief representation . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
6-2 Motivating scenario for re-imagining sensor data . . . . . . . . . . . . . . . . 86
6-3 Recursive Sensory Data Propagation . . . . . . . . . . . . . . . . . . . . . . . 87
6-4 Using a hypothetical copy of robot for mental state predictions . . . . . . . . 88
6-5 Dataflow during reset and operation of hypothetical robot . . . . . . . . . . . 91
6-6 Expectation mechanism for object manipulation . . . . . . . . . . . . . . . . . 93
6-7 Generative Context/Goal Conditions . . . . . . . . . . . . . . . . . . . . . . . 96
6-8 Search through action space . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
6-9 Mental state configurations of other agents as goals . . . . . . . . . . . . . . 101
6-10 Use of context tree . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
6-11 Layout of Demonstration Scenario . . . . . . . . . . . . . . . . . . . . . . . . 105
6-12 Study Condition Goals and Behavior . . . . . . . . . . . . . . . . . . . . . . . 106
6-13 Robotic Demonstration of Mental State Manipulation . . . . . . . . . . . . . . 108
6-14 Visualizer of Robot's Planning . . . . . . . . . . . . . . . . . . . . . . . . . . 109
6-15 Object Choice Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
6-16 Teaming Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
6-17 Rating Scale Before and After Revealing Manipulation . . . . . . . . . . . . . 116

# List of Tables

5.1 Human subjects study results . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
5.2 Behavior of robot on benchmark tasks . . . . . . . . . . . . . . . . . . . . . . . . 79

# Chapter 1

# Introduction

Collaboration and competition are two fundamental modes of human interaction. The aim of this work is to endow a robot with new skills when participating in collaborative and competitive interactions with people. In particular, the robot should be able to gain insights into a human's behavior by observation, to predict how its own behavior affects the human's future knowledge and behavior, and to take steps to alter the human's knowledge and future actions.

Consider a robot collaborating with a human partner Bob, who together as a team are competing against a human adversary Eve to collect a number of objects from hidden boxes and deposit them in their respective goals. It is to the advantage of all players to watch each other, because any player might happen upon a box, and that information is valuable to all the players. Likewise it is to the advantage of the players to keep information about new boxes from the other team, so they don't have access to that source of objects. An even better strategy might be to proactively deceive the other team, indicating the presence of a box where there is none. During this secrecy and deception, however, it is to the advantage of the robot and Bob to privately communicate, so if either happens upon a cache of objects or observes an interesting behavior they can work together to exploit their knowledge.

To succeed in this environment, the robot must be able to utilize certain aspects of deception and communication. It is not enough for it to *determine* when to monitor Eve, what information to secretly communicate with Bob, and what falsehoods to broadcast to Eve. As a physical agent engaging in this task, it must *actually perform* these tasks. It must be able to watch Eve, and figure out how Eve's observable behaviors may indicate her actions and knowledge. It must be able to physically perform secret communications that are interpretable by Bob but not Eve, as well as be able to interpret Bob's communications intended for the robot. To give Eve false information, the robot must understand how Eve observes the robot, and how she generates her information from those observations.

The architecture presented in this thesis was generated to address these issues. The robot's strategy is to leverage the two-way connection between the observable world and the hidden internal mental states of the humans. Observing the human and the world around it gives the robot possible insight into the internal mental states of the human, and information about the mental states helps the robot predict the human's future behavior. By visibly performing actions or otherwise altering the world the human sees, the robot can also attempt to manipulate those same internal states for the purposes of altering that human's future behavior. This behavior can function as communication or deception, depending on the nature of the manipulation of the mental states. Finally, the robot must also factor in that the human is likely performing these same inference operations on the robot's internal states.

Indeed, humans frequently interpret the behavior of other humans and even themselves in terms of intentions and beliefs. While these concepts may or may not have neural correlates in the brain, they do have explanatory and predictive power and humans are able to use them to great effect for those purposes. This ability to attempt to infer the hidden mental states such as intentions and beliefs from observable behavior, sometimes called "mindreading," begins to develop at a young age.

People use these abilities to inform their plans and actions. Knowing a person's probable mental states helps predict their future actions. Alternatively, it may be important to alter their mental states; whether it is something as simple as showing them new information, or perhaps more complicated like changing their goal, humans are able to form these kinds of plans and take steps to carry them out. The robot is not intended to perform these inferences and actions identically to how a human performs them, but instead it is designed to be cognitively compatible, employing these skills in a way that correctly predicts human behavior and allows the human to correctly predict the robot's behavior.

The aim of this research is to design a robot that can function in the above scenario. The strategy is to implement an architecture that employs theory of mind techniques to allow a robot to include inference and manipulation of mental states in its interactions with other agents, including humans. This requires modeling the mental states, using them to predict future behavior, using our planned future behavior to predict resulting mental states, and even taking action to make a specific change in another agent's mental states. The architecture includes these critical features:

**Combining Knowledge and Intentional Inferences.** Combining the ability to infer the knowledge of an observed human with the ability to infer intention provides more power their either alone. Knowledge inferred through a process such as perspective taking can be used to inform the process of intention inference. Finally, combining intentions and knowledge is necessary to make predictions about the future actions of the agent.

**Recursive and Predictive Capabilities.** The robot not only models mental states, but also becomes an active participant in mental state formation and manipulation. This requires the robot to track it's own presence in mental states through recursive modeling (modeling mental states the humans have about the robot's mental states). Further, the robot must look past the present and forward into the future to judge how its actions affect not just the target objects but also the mental states of those around it, potentially changing their behavior.

**Self-As-Simulator Architecture.** The architecture uses self-as-simulator techniques where possible in its attempts to model the mental states of the human. This provides a mechanism to infer intentions from physical actions (reusing the robot's own action production system) and model knowledge (reusing the robot's own knowledge modeling systems via perspective taking). It also allows recursive models and predictions to refer to the robot and human mental states interchangeably, and in general causes any inferences it makes about the human to be in the language of its own generative systems.

**Embodied, Real-Time, Physical.** Finally, this architecture is implemented in a real-time manner running on a physical robot interacting in the real world. The mental state inference described here functions through exploiting the joint embodiment of the robot and human and how this embodiment maps their internal mental states to the world.

## 1.1 Strategy

As an embodied agent interacting with the world and with other agents, any cognitive processes that that agent employs eventually ground out in a physical interaction with the world and the other agents. Any cognitive processing that can have no effect on the agent's behavior provides no benefit to the agent. The motivation, then, for endowing an agent with theory of mind capabilities is because it can use these capabilities to affect its behavior, and in doing so affect the behavior of those around it in ways it can predict.

The focus here is on the use of theory of mind techniques as a tool to enable new kinds of behaviors and interactions with humans. Because of this focus, instead of developing a standalone "theory of mind" mechanism in a vacuum, this research focuses on integrating closely with an architecture designed for controlling interactive agents that must itself close the loop from perception of the world to actions in that world. This choice shapes the research described here.

Systems designed for logical inference can perform powerful theory of mind type analyses on their internal data; however, the connection between these structures and world, through sensing and action, is often quite thin, with the idiosyncrasies of these connection seen as distractions to be abstracted away. In contrast, the research described here focuses on those connections. Much of the cognitive effort in an embodied creature goes into the connection on the input side, where it takes in data and attempts to make sense of the world around it, and the output, where it tries to carry out the result of its behavioral decisions. Instead of trying to abstract these processes away and focus only on the middle, where decisions happen between input and output, these input and output processes themselves which serve as the connection between hidden mental states and the world can be a valuable source of clues about the hidden mental states of other agents.

The work described here is built by leveraging and expanding upon the c6 architecture. Originally developed for interactive graphical characters [Blumberg et al., 2002, Burke et al., 2001], then later adapted for robots [Gray, J. et al., 2005, Gray et al., 2010], this architecture is designed to support the development of interactive, realtime, embodied agents that need to perceive the world, process this information, and, through their control over their bodies, perform physical action to change the world. Subsystems relevant to this work will be described in greater detail in the sections which refer to them, however a brief description of how these subsystems fit together will help inform the rest of this section.

### 1.1.1 Underlying Architecture Overview

The system is designed to be modular, with modules functioning with some independence so they can be used in different configurations. In the standard configuration, incoming sensory data passes through a classifier tree called the *percept tree*. Each classifier is tasked with determining whether it "matches" each element of incoming data. The input to these classifiers varies based on the type of data in question, but the classifier tree forms a uniformly formatted output which includes, for each incoming piece of sensory data, the match values for the classifiers. The classifiers also have the option of storing processed data into the results of the tree, if that data is expected to be of use to the downstream systems.

The output of the classifier tree, which is a set of percept evaluations, becomes the common currency of information about the outside world. This data is merged into the agent's existing *object beliefs*. The object beliefs each represent the agent's sum of knowledge about a particular object in the world. The data inside the *object belief* consists of percept evaluations that are relevant to that object. These objects are persistent, and the manner in which the time sequence of new percept evaluations are incorporated into the *object belief* can be configured based on the desired behavior, from a simple strategy like only saving the most recent data, to a more complicated one such as creating a statistical model of the data.

The *object beliefs*, along with proprioceptive and other internal state information, are then used by the deliberative systems of the robot to make decisions about what behavior to pursue. The results of this process are continually relayed to the *motor system* which governs the physical behavior of the agent. At any point in time, the *motor system* has a desired joint configuration that indicates position (or angle) for each degree of freedom under its control. To calculate these positions, it employs a number of techniques, including blended combinations of hand made animations and inverse kinematics, selecting among the options available to it based on the requests made of it from the deliberative systems.

### 1.1.2 Integration with this architecture

The research described here reuses parts of this architecture to achieve theory of mind related goals. The motivation for this reuse comes in part from simulation theory (section 2.2), and in part it from efficiency of implementation. However, the largest motivation is to enable the agent to effectively use the information it is inferring. These agents lack any verbal language; they have no abstract format in which to store and access the mental state inferences they are making. However, they do have many systems and representations used for their own action production. Reusing the agent's internal systems where possible to detect and represent the mental states and activities of the agents means that these inferences produce information in the only vocabulary known to the agent - that of its own internal workings. These representations form the natural common ground for representing the perceived information about other agents. As the motivation for making these inferences is to drive behavior, representing them in the format of the robot's own goals and actions makes it possible to relate them to the robot's internal structures and act on them.

### 1.1.3 Research Structure

The description of the architecture created to address these goals is divided into four chapters. Three chapters discuss building blocks that reuse different parts of the robot's mechanisms to make various inferences about the mental states of nearby agents. Each of these building block by itself adds to the interactional capabilities of the robot, and the end of each chapter provides a demonstration of that capability. However, the fourth chapter describes how to bring these elements together into a system that can be used to interact using the inferred mental state information and take action to manipulate the mental states of other agents.

Chapter 3 focuses on understanding the physical actions of other agents. Embodied agents are present in the world, and therefore always displaying a particular joint configuration. The internal mental states and motivations inside another agent are hidden, however the physical motions that result from attempting to realize those goals are not. Working backwards from observed motions to underlying goals is something that humans are quite good at, and this inference ability is a critical part of making sense of the behavior of those around us. This section attempts to give the robot access to these types of inferences by reusing the motor production capabilities of the robot to detect and classify the motions it observes performed by others. This ability will be utilized in later chapters, however by itself it is also a valuable addition to the robot's interactive capabilities by allowing it to recognize motor actions and perform motor mimicry.

Another way of getting at hidden mental states is to keep track of the sensory input coming in to another agent. While the world modeling going on inside someone else's head is invisible, knowing what information they are receiving can provide some important clues. Chapter 4 focuses on this strategy; by reusing the robot's own methods for keeping track of the world state given its sensors, the robot can model the world from a second perspective to try to understand what the world model of another agent might be like. This ability is sufficient to pass a variant of the classic "false belief" task, which is demonstrated at the end of this chapter.

The previous two chapters approach the problem of "mindreading" from different angles: how does the physical behavior of an agent help us understand what action it is performing, and what does the sensory information passing in to an agent tell us about its model of the world around it. In Chapter 5, we combine the information from these two sources to try to access an agent's "intention" behind its actions. This chapter examines the reuse of the agent's goal directed actions, combining modeled world state with observed physical action to determine underlying goals. The agent's ability to infer goals (even in the case of failed actions) is evaluated against a set of benchmark tasks, with the results of the robot's performance validated in a human subject study.

Chapter 6 describes mechanisms for making use of the previously described ideas in order to proactively perform actions with the goal of achieving particular mental states configurations in other agents. This includes a very low level form of communication: trying to form a mental state inside another agent. It also includes competitive behavior such as deception, where the desired mental state configuration does not necessarily reflect reality. This ability functions by performing the "mindreading" abilities described above, but extending them in two directions. First, it extends them past the present and into the near future, to test out actions and predict their mental state consequences. Second it extends them past modeling a single agent's mental states and into a recursive structure where the robot not only maintains mental states of agents around it, but it maintains the mental states that they would be maintaining about the agents around them. These two extensions allow the robot to look into the near future to choose actions while tracking how its own actions will affect others' perceptions of it.

### 1.1.4 Research Scope

Theory of mind is a broad topic, with many interesting research directions. The research here focuses on leveraging the rich connection between embodied agents and the world to attempt to track and alter their mental states. The monitoring and simulations the robot performs are very rich in geometric detail, tracking visual perspectives, occlusions, and physical actions. As such, they are quite computationally intensive. These strategies are not intended to be used as the exclusive interaction and planning mechanisms for an agent that operates in a real world situation with many people and numerous actions; instead, they are designed to be a very high resolution mechanism that would be used for detailed analysis of the actions deemed most relevant and to model the handful of people currently present. The future work section addresses how this work could be integrated into a longer term, lower resolution but longer time scale system.

# Chapter 2

# Background and Related Work

This section provides some background for the motivation, goals, and related systems to this work. Section 2.1 provides a brief summary of some of the work done studying human theory of mind abilities that has inspired this research. Section 2.2 provides an argument for why simulation theory ideas are compelling for a robotic implementation of theory of mind. Section 2.3 provides details on related research and situates this work within that context.

## 2.1 Theory of Mind in Humans

In humans, the ability to infer the mental states behind observed actions has been a topic of much study. In adults this mental modeling is ubiquitous, but even in young children there is evidence of perceiving the behavior of others in terms of mental states rather than physical actions. In a series of experiments, [Meltzoff, 1995] was able to show that infants at 18 months understood the goal behind a simple, likely novel, motor action that was demonstrated to them. The infants were shown a failed attempt to perform an action, but when given the opportunity themselves they recreated not the failed attempt they observed, but the goal of the demonstrator that had never been successfully demonstrated. Another famous experiment (which has since been re-administered hundreds of times in different conditions [Wellman et al., 2001]) seeks to test the ability of children to keep track of not only their own understanding of the world, but also of the beliefs of others around them. In these experiment, which began with [Wimmer and Perner, 1983], children are presented with a scenario where the location of an object changes while one of the participants is away. To pass, the child must be able to identify where the participant, absent during the switch, believes the doll to be. Children are able to pass this test beginning at about 3 years old.

Many have suggested that these skills are enabled through Simulation Theory [Gordon, 1986]. This theory proposes that humans are able to make sense of the mental states of others by putting themselves in the shoes of the other and examining what their own mental states would be in that situation. While there is some disagreement about how much of the Theory of Mind activities can be explained by simulation theory and what other mechanisms are required (In [Nichols and Stich, 2003] an interesting boxology is proposed that goes into detail about which mechanisms could be reused and which additional mechanisms would be required to do so), the theory has gained traction due not only to its simplicity but also due to suggestive experimental evidence.

Meltzoff writes about the "Like-Me" hypothesis which states that humans see other humans in a special way: as an entity that is similar to them and which can be imitated. Moreover, he has collected evidence that the simplest forms of imitation may be innate - studies of infants (some as young as 42 minutes) show imitation of facial expressions [Meltzoff, 2005]. This evidence supports his Inter-Modal Mapping hypothesis, which describes how perception of actions and production of those actions must share a common coding in the brain; this commonality allows us to equate perceived motions with our own producible motions. Later evidence of Mirror Neurons, a set of neurons in the brain which fire both during production of an action and during its observation, supports this idea of a common coding for perception and production [Rizzolatti et al., 1996]. While Meltzoff's experiments would indicate that some amount of this skill must be innate, other evidence shows that this ability can be learned and refined. For example, a study of capoeira and ballet dancers showed that each had neurons in their brain specialized to detect movements from their own disciplines that did not fire in response to similar movements from the other discipline [Calvo-Merino et al., 2005]. In another study, [Gallese and Goldman, 1998] showed that sensitive instruments can detect slight movements in the muscles of an observer that correspond to the movement they are observing, further evidence that the observer is at least in some way activating their own representation of the task they are watching.

Other evidence suggestive of simulation theoretical mental modeling comes from examples where first person experience helps to inform 3rd person observations. A compelling example of this comes from a recent study [Meltzoff and Brooks, 2008] involving gaze following. In this study, 1 year old children are observed to follow the gaze of an adult even though the adult is wearing a blindfold. However, children who have experienced a blindfold blocking out their own sight are less likely to follow the blindfolded adult's gaze direction.

Finally, the developmental timeline of children also provides some evidence favorable of simulation theory - certain mental milestones come online in a timeframe that suggests the same mechanism might be used both for the child's processing of its own mental states and also for the mental states of others. A good example of this is the development of the ability to represent the false beliefs of others - this develops between the ages of 3 and 4, which is the same time that children are acquiring this ability relative to their own mental states [Astington and Gopnik, 1991].

The architecture described here, inspired by these ideas, attempts to perform these mind reading tasks through re-use of behavior generation system where possible.

## 2.2 Why use Simulation Theory Ideas for Robots?

The literature on Theory of Mind in humans is interesting for robot design for a number of reasons. First, humans are an existence proof for the kind of intelligence we intend to imbue our robots with. Anything we can learn about how humans are able to operate may be valuable information to consider when trying to design an intelligent system. Another reason to pay careful attention to how humans achieve these tasks is cognitive compatibility - we want our robots not only to be able to produce intelligent behavior, but to do so in a way that is compatible with the cognitive skills of the humans it will be interacting with.

One way to do this is to have the robot's skills operate in the same manner as the humans.

Therefore, information we can gather from how humans are able to achieve Theory of Mind tasks is of great interest when designing robots that we intend to have that same capability. There seems to be a significant amount of evidence that, at least on some level, mechanisms humans use for producing behavior are re-used to understand the behavior of others. Interestingly, some of the most significant criticisms of Simulation Theory in the psychological community do not necessary argue against it's adoption, in some form, in robots.

The chief competitor of Simulation Theory is Theory Theory - the idea that we reason about the mental states of others not by simulating their condition through putting ourselves in their place, but that instead we have a separate reasoning system that specializes in this kind of analysis. However, many of the proponents of Theory Theory believe that this ability is applied not only to understand and predict the behavior of others, but also to explain and predict one's own behavior [Gopnik and Wellman, 1994]. Thus, though the human is not exactly re-using their own behavior generation systems, they are re-using their own systems nomaly used for making sense of their own behavior (which presumably have some role in behavior generation, such as conscious planning). Further, evidence from mirror neuron studies indicates that perceived actions can immediately be equated to the performance of the same action - even if this turns out to be exclusively used as the input to a Theory Theory based reasoning process, a robotic implementation based on these ideas is starting to look like it has a number of similar features to one based on Simulation Theory. In both cases, the robot must be able to connect perceived actions to performed actions, and make similar inferences about the behavior of self and other using a single mechanism.

Another possible criticism of Simulation Theory, or more specifically of Folk Psychology which Simulation Theory draws on, comes from *Eliminative Materialism* [Churchland and Churchland, 1998]. Eliminativists believe that our folk psychology, based on such terms as belief, desire, intention, etc., is incorrect. That is, these are elements of a theory that we apply to human behavior, but they do not correspond to any particular aspect of the brain. It is interesting to consider whether these qualities do in fact correlate to actual brain function - however, for a robotic implementation it is not necessary to find the answer.

If we accept that people use these concepts in their use of folk psychology, and that folk psychology is effective at predicting behavior and helping people efficiently interact, then these concepts are sufficient for a robot to use, even if they end up not relating to a neural configuration in the brain.

Finally, a Simulation Theoretic approach has a number of advantages. Apart from being parsimonious from an implementation standpoint, it grounds the beliefs and behaviors of others against the robot's own beliefs and behaviors. Since we presumably desire the robot to be able to process and respond in some way to inferences it is making about human behavior, this grounding is important because it gives the human behavior meaning to the robot - it can fit the observed behavior into its own structure of actions, goals, and beliefs about the world which puts it in a position to use its own behavioral structures to make further inferences and take some action.

## 2.3 Related Robots/Agents

There has been a lot of interesting work exploring how robots can begin to work with theory of mind concepts to enhance interaction with humans. I will describe some of that work here, and relate the projects to the appropriate parts of the architecture described here.

### 2.3.1 Embodied Architectures

The work described here builds upon an architecture [Burke et al., 2001] developed by the Synthetic Characters group (MIT Media Lab). The system is designed to model an entire agent which perceives and changes its world. It is also designed to be flexible in allowing augmentation and modification, allowing researchers to explore a particular subsystem or combinations of subsystems in the context of a behaving creature.

In [Blumberg et al., 2002], a virtual dog could be trained to perform new motor skills in response to new (vocal) commands using clicker training, a standard dog training technique.

This demonstration integrated simultaneous learning in 3 different areas (motor, perception, and action selection) to enable the dog to learn. In contrast to the proposed research here, the trainer is explicitly trying to teach the dog the mapping from the trainer's behavior to their intention through a formal training technique.

[Isla and Blumberg, 2002] modeled object expectations and persistence by augmenting the properties of an object belief with its probability distribution over space given the agent's perspective and occlusions. This technique was used to generate lifelike search and attention behavior, but may fit well into the goals of this research if applied instead to help the robot model a human's object expectations.

Another related project in this framework is [Buchsbaum, 2004], which showed re-use of motor trajectories to infer unknown object affordances. In this work an agent can learn that a particular motor trajectory (and thus likely that trajectory's associated goal) is relevant for an unknown object by observing another agent perform that action on the object. (Section 3.1 includes parts jointly developed to be used in this project as well as the proposed research here)

The work proposed here differs from previous work in this architecture by its focus on re-use of all modules to try to make sense of, predict, and attempt to influence the behavior of others. The research described here attempts to take advantage of the rich interface between an embodied agent (be it human or artificial) and the world to allow an agent to infer, utilize, and manipulate the mental states of a human.

In [Roy, 2005], the author takes advantage of this interface and defines the term "grounding" to help describe a framework where concepts and words have meaning for a robot not only by their relation to other concepts or words, but instead by the relationship of that concept to the sensorimotor experience of the robot. In this way a robot grounds out its representation of a concept in the model of real interaction or expectations through its sensors and actuators.

I have borrowed that word in this document to describe a benefit of using a simulation theoretic approach to mental state inference (a benefit which I feel is analogous in some ways to the above usage). By perceiving mental states or intentions directly as elements of the robot's own behavior generation systems, those states assume a direct, functional meaning to the robot, including sensorimotor predictions or actions.

In [Hsiao, 2007], the author describes a framework and robotic implementation based on grounding language through a schema based representation of objects - objects are represented by the possible interactions the robot can have with them. Similarly to the architecture proposed here, this framework includes sensorimotor correlations, where the robot can model what perceptions it expects based on what motions it plans to perform. It uses them to a different end, however, managing its perceptions to accomplish linguistically described tasks verses manipulating a partner's perception.

### 2.3.2 Motor Mapping and Imitation

In Section 3.1, the robot maps perceptual data obtained from observing a human into its own motor space in order to match it against motor primitives. Many researchers have been doing interesting work in this area.

In [Matarić, 2002], researchers created a system that can learn common motions from motion capture data of human performers, then later classify observed motions as parameterized combinations of learned motions. The learned motions can be used to move the robot as well as for representing observed movements, making them useful for imitative learning.

[Jenkins et al., 2007] presents a system that can generate the basis set of primitives which are used to represent motions, then uses a single camera to classify observations against known motions using particle filters.

In [Chalodhorn et al., 2007], human motion capture data is mapped onto a robotic joint structure and used as a starting point for learning motions, including dynamically balancing walking, using a system that makes sensory predictions based on sensorimotor state and experience.

In contrast to these systems, our system requires the robot to already have a set of primitives, and focuses on the mapping problem between human and robot morphologies (towards the end of mapping mental states) by leveraging the mapping abilities of the human demonstrator.

### 2.3.3 Statistical Activity Recognition

There is a rich literature for recognizing human activities using training and statistical methods. Some researchers use audio/video feeds to try to pick out activities in a surveillance or other observational capacity [Ivanov and Hamid, 2006]. Others try to get at hidden internal states with this data, such as [Dong et al., 2007] where the authors are able to train a system to recognize the roles of participants in a meeting. In other applications [Lee and Mase, 2002] the authors seek to recognize activities such as walking from a minimal sensor setup such as a wearable accelerometer.

This work differs from the proposed research because they seek to learn activity classifiers based on training data in the target domain. The proposed research here instead aims to utilize the fact that the robot already has behavioral capabilities similar to the behaviors we wish to recognize, which we can levarage to help recognize activities. Additionally, in our domain the reason we wish to recognize any activity at all is in order to use that knowledge to behave more correctly with respect to it; anything we recognize, then, must have a functional meaning to the robot, so the mapping of observations into the robot's cognitive mechanisms is a main concern.

### 2.3.4 Visual Perspective Taking

Many systems perform an egocentric transform to put incoming data into the coordinate frame of another agent. In [Johnson and Demiris, 2005], the authors have created a system that actually re-renders a scene from the perspective of a nearby human and computes what they see using synthetic vision. While perhaps this is computationally intensive, it does provide a completeness that other methods may lack. For example, concepts such as obstruction do not need to be programmed in, because the rendering process will automatically cause obstructed objects to be invisible to the human. More complex concepts could also come about using this method - for example, perhaps two objects might be mistaken when seen from a certain angle; using this method the robot could predict the human's confusion.

However, putting the human's perspective transform into such a complex black box as a rendering engine has certain disadvantages as well - while it is helpful that concepts like calculating occlusion come for free, it is hard to introspect on these free properties - for example, if the robot wanted to hide an object from the human, it would need additional information in order to make a plan about how to move an occluder into the appropriate position.

This system most closely maps onto the "perspective transform" that takes place in section 4.4. It is a more sophisticated transform than is done by the architecture introduced in this thesis, in that it is actually re-rending the full scene from a new perspective, while instead the robot here performs a geometric transform/occlusion calculation based on the 3D locations of objects. Because it is a goal of the architecture here to introspect on this system and attempt to infer how to get an agent to see (or not see) an object, this simpler geometric transformation may end up being easier to use (and reverse) than a black-box rendering system.

### 2.3.5 Using Perspective to Resolve Ambiguity

The robot described in [Trafton et al., 2005] has the ability to resolve certain types of ambiguities in operator commands by taking the operator's perspective using their architecture *Polyscheme*.

They demonstrate this ability in a number of scenarios - in some scenarios the human's command is ambiguous because it refers to an object where in fact the robot can see multiple instances of the object. Taking the human's perspective, the robot finds out that the human can only see one of the objects and thus the command likely refers to that object. In another scenario the command refers to an object the robot cannot see - in this case, it must take the perspective of the human to find areas that the human can see that are occluded for the robot, and assume that the command must refer to an object in one of these areas.

Relating this to system described here, it can be seen that they are making a spatial model of objects in the environment and using this to calculate occlusion. This is similar to our perspective transform element, but they do not seem to be following that with a belief maintenance system as we do - thus their system could determine which objects are visible to which agent, but is not necessarily tracking other states like what a human knows about an object whose property changed while invisible to the human. However, their occlusion model is quite sophisticated, because not only does it cull invisible objects from the mental model of the observed human, it can also be used generatively to hypothesize objects in occluded locations, a feature which would be a nice addition to our system.

### 2.3.6 HAMMER

HAMMER is a system which uses a combination of forward and inverse models to relate observed motions to the robot's own repertoire of actions [Demiris and Khadhouri, 2006]. The inverse model is used in normal, independent operation to determine how the robot should proceed to achieve its task. When the robot is observing a motion, however, the forward and inverse models can operate together to select which of the robot's own tasks most closely resembles the observations. In this case, after an egocentric transformation, the robot knows the current state of the observed system (this is possible by direct observation because its models made no reference to any unobservable states). It can then, for each possible goal in the robot's repertoire, generate the appropriate command necessary to go from the observed current state to that goal state using the inverse model for each goal. The forward model associated with each goal can then make a prediction, using the observed current state and the generated command for each goal, about the next state of the observed agent were it pursuing that goal. These predicted goals can be checked against reality as time passes, and the robot's confidence in each goal is modulated by the accuracy of these predictions.

While it is noted that it would be possible to parameterize the actions, in the implementation described here it seems that each action is tied directly to its goal ("grab the can"), and therefore relates in part to chapter 3, which addresses physical actions, and in part to chapter

5, which addresses the goals of those actions. One nice feature of this forward/inverse model system is that each action can look at different and arbitrary state information to try to determine its relevance - the authors avoid perspective issues by expressing state in inherently relative terms (e.g. distance between hand and object). A challenge with this model, as the number of actions grows and becomes more varied, is to provide an egocentric transform for which this arbitrary state information is preserved (and directly observable) for all types of actions.

### 2.3.7 Experience-Based Perspective Taking

In [Kelley et al., 2008], the authors describe an interesting design in which the robot employs it's own history of perceptual experience to determine the behavior of an observed human. In their system, the robot has some initial mechanisms that allow it to successfully perform several actions. While performing these actions itself, the robot can use its vision system to track various objects over the course of the action and use the motions of the objects to build an HMM describing their behavior. When observing other agents, the robot can use a geometric egocentric transform on its perceptual data, then use it's HMM's and determine which was the most likely to produce the transformed data.

This technique relates to the intention recognition mechanisms (chapter 5), and like others mentioned here recognizes an action/goal pair together instead of allowing parameterization to reduce the search space. It also differs from the proposed architecture and from other techniques mentioned here in its use of a model learned during action production rather than the underlying action production structure. The paper mentions that the robot has a system which can perform the required actions, however this system is opaque to the action recognition system - instead of comparing observed actions against the robot's own action production structures, it compares them against the robot's perceptions during its production of those actions.

This difference has some advantages. The data used is already automatically in the same modality for both self and other. This differs from action production structures, which may include physical motions - transforming observed motions into the egocentric motion space may be difficult. However, perceptual effects, especially visual ones, that occur as a result of performing an action can be transformed into an egocentric space with a simple geometric transform, simplifying the mapping process. Additionally, the fact that these HMM's are learned during performance means that they can take into account perceptual details not strictly described in the implementation of the actions but which may frequently occur during an action and help disambiguate it from otherwise similar ones.

This approach does, however, ignore the structure of the action as used by the robot for action performance which may provide useful cues for detecting the action. Context and goal parameters are likely to be explicitly described in this structure, and though these may be partially learned through the self-demonstration process, it is unlikely that the robot could learn these with the same richness as they must be represented by the action for successful performance. Additionally, this technique does not take advantage of the fact that the robot most have some internal information about the physical motions required to perform an action.

### 2.3.8 Soar

In a demonstration using a quake bot, [Laird, 2001] has been able to add perspective taking capabilities to a Soar based agent. In soar, the behavior of the character is based on the firing of *rules*, both to help select a current *operator* and to carry out the operator once selected. It keeps track of current state in a stack-based working memory, allowing it to setup data at multiple levels of resolution during the execution of a hierarchical operator. Rules can operate on and modify data in the working memory, propose possible operators, and act as preferences to weight different possible operators. Based on this, an operator is selected (often the same one as selected last cycle since operators can last for a period of time), and then rules that match that operator can fire (possibly adding sub-operators to the stack). The operator concludes only when no more rules fire. A sequence of rule firings for one operator can encode a task which takes place over time and is modulated by inputs.

Anticipation, then, exists as an operator whose rules set up a special level on the working memory stack where the quake bot attempts to recreate the relevant working memory state of the observed player. Once it has simulated the internal state, it can attempt to apply it's own operators to the state, to see what it would do were it in that situation. In this context, however, the operation of these operators has been modified so that instead of actually sending motor commands (and waiting for the command to complete), each has been set up to modify the working memory into a state as if the action successfully completed. This allows the bot to, in one update cycle, simulate the execution of many sequential operators from the perspective of the player.

This rule based system operates differently in many ways from the proposed architecture, however parts of the system described by the authors have functionality designed with similar goals. Using the rules, it keeps track of both its own idea of the world as well as performing an egocentric transform to also model a potential world state from the opponents view. Then it can make a plan either from its own data, or starting from its opponent's data, and the predicted opponent's plan can be used to help shape its own plan.

The egocentric transform seems to be performed entirely based on data available at the time the opponent's behavior is to be predicted - in part this may be necessitated by operator selection and the working memory stack, where a single operator is selected at each level and the information produced by the egocentric transform can only last as long as the "anticipate" operator is active. This differs, however, from the strategy pursued by the proposed system which is constantly modeling the human's beliefs - this has the benefit of tracking what the human has seen or not seen, enabling complicated false belief possibilities. Of course, that comes at a complexity penalty both in terms of execution speed and error recovery - any errors made by the soar system will be flushed out when it is done with the prediction, but they will persist until corrected in our system.

The action classification and future predictions also operate differently. Action classification in the soar system takes place from context alone - given a situation that includes current perceptions, health, ammo, and similar state, the agent determines what it would most prefer to do and assigns that prediction to the player. Our system instead relies heavily on taking into account the physical motions of the human, then using the perceptual state to disambiguate the possible goals that could result in that motion. This seems like it may result in more accurate action recognition when observing agents with rich embodiment (although at the cost of possibly ignoring actions that the human performs in a manner sufficiently physically different from the robot's notion).

I believe that this difference illustrates a difficulty of applying rule based systems to this domain - the rules must ground out in a black box that actually performs the action chosen (drives the quake avatar through the chosen motion, for example). The black box action production will make it difficult to introspect into that motion and glean the information necessary to recognize it being performed by others.

### 2.3.9 BDI For Teamwork

Many Belief-Desire-Intention (BDI) frameworks exist and are used to allow agents and robots to represent their own mental states in terms of beliefs, desires, and intentions, as well as to allow those agents to work with the mental states of other agents around them.

An interesting example of this kind of architecture is PsychSim [Marsella and Pynadath, 2005]. In PsychSim, each agent maintains beliefs about itself and other agents. These beliefs are subjective (even an agent's beliefs about its internal states are allowed to be incorrect), but they are constantly being updated as the agent observes actions and their results. In a school bullying example, the authors designed agents that had actions such as "laugh(performer, target)", and relationships such as "trusts(truster, trustee)". The goal of a bully might be to minimize the "power" of a victim while receiving laughs. However, the agents model that their success is dependent on their relationships to others, and those relationships are dependent on what the others observe and what beliefs they form.

The system models recursive beliefs two levels deep, allowing the agents to carefully plan a course of action that they predict will manipulate the situation, including the beliefs of others around them where necessary, to achieve their goals. Along with actions, agents can influence each other by sending messages which may refer (correctly or incorrectly) to facts, or even to beliefs about facts. Interestingly, the authors found that they could increase the efficiency of what might otherwise be an infeasible problem by allowing the agents to "stereotype" many other agents into a few different groups, greatly simplifying decision making.

Another interesting BDI architecture is Machinetta[Scerri et al., 2004]. Machinetta is designed to be a scalable, realtime architecture for very large teams. To allow for arbitrary robots or agents, each agent is managed by a proxy process that is designed to connect the arbitrary agent to the Machinetta system. Each proxy manages messages to and from the Machinetta system, and maintains beliefs, intentions, and goals as well as a sliding autonomy mechanism. Beliefs about joint goals can be explicitly shared with teammates through the system; however, to reduce the traffic in a large team setting the system tries to determine when communication of individual beliefs will help the team achieve a goal, and only communicate under these circumstances. Acknowledging that there will be domains where any kind of large team planning and execution will have problems, the authors have equipped the system with a meta-level controller that attempts to detect these failures - even failures in the overall management system, such as in role allocation, can be detected because the system is designed to make every such action explicit (for example, assigning roles is itself a role!).

These systems are quite sophisticated at using mental state information to perform inferences and plan out optimal courses of action in cooperative and competitive scenarios. However, their strength lies in the complex plans that can be made once the mental states of others are determined. In PsychSim an agent's model of the mental states of the other simulated agents is updated based on witnessing symbolic actions ("laugh(A,B)") or receiving belief-formated messages ("A says B believes C is weak"). In Machinetta the proxy agents have access to a network where beliefs are directly communicated.

The strength of the work proposed here, while it does not have the same depth of planning capabilities, is in its focus on what happens to infer and manipulate the mental states of others in the rich embodied interaction between agents, each other, and the world, which is skipped over by these architectures. They give agents symbolic access to observed action representations in the format of their own actions, and in some cases allow direct communication between agents. This architecture, in contrast, devotes much energy to the actual mechanisms to observe these actions and how to communicate intentions and information to another embodied agent (the "last mile" of the communication, where correctly modeling how agents perceive the world and each other's intentions can enhance cooperation). An interesting possibility in the future would be to try to combine this system with something like those above, and see if the sophisticated planning can be applied to these embodied techniques.

### 2.3.10 Causal Bayesian Networks

In [Goodman et al., 2007, Kushnir et al., 2003], the authors present a Bayesian framework where a computational agent can determine the structure of hidden causes for observed events by utilizing the difference between regular observations and interventions. Additionally, the authors show that the structures learned (and even in some cases the mistakes that are made) are similar to structures learned by humans.

The systems operate by taking advantage of the difference between observations and interventions. Making this distinction helps when trying to determine the structure of unseen causes within the system being observed. For example, if two events are observed in a short time period, it may be that the first caused the second, they are independent, or that one hidden cause caused both events. However, a human can "intervene" by causing one of the events themselves. In this case the cause of the event is known, it was the human's action. Thus if the second event still follows the first, it likely means that the second event is caused by the first rather than both being caused by a hidden third cause. The authors were able to show that their algorithm was able to determine the structure of the hidden causes in a system, and interestingly that humans determine the same structure.

The systems described in these papers are different from other work described here because they are about inferring general causes for generic observed events, rather than specific to inferring hidden mental states that bring about observed behavior. However, they do attempt to make these inferences in a manner consistent with human inference processes, so if these systems can get access to the perceptual input received by a human (perhaps through some egocentric transform) they could be employed to predict how a human will account for causes of observed events. This kind of reasoning could be an interesting addition to an autonomous robot that needs to learn about new systems it encounters, but also would be useful to a mindreading architecture, allowing a robot to predict the model a human is forming based on the human's observations of an object.

# Chapter 3

# Determining Actions from Observation

## 3.1 The Motor System

An important element of the robot's ability to predict and help with goals of people is to be able to make sense of their physical actions. The approach taken here is to re-use the physical actions the robot can perform to recognize the actions observed in the human. This is done in a two stage process. First we transform observed human movements into the same movement space as the robot. Once the observations are in a similar representation to the robot's own motor generation capabilities, we can match the robot-space motions against its own motion repertoire. This dual use of the same motor processes for both production and recognition is inspired by mirror neurons. This gives us a starting point towards understanding the overall activity being performed by the human (however this understanding will also be dependent on other mental states we are tracking within the human, see chapter 4). Not only does this structure provide a mechanism for classifying the human motions, the classifications generated through this process are inherently meaningful to the robot because they are exactly the same motions it uses, which are tied to higher level goals by their connections to its goal directed action system (Chapter 5).

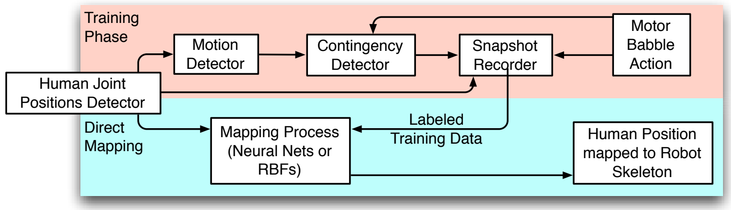


Figure 3-1: Mapping perceived human joints onto the robot's skeleton to allow for comparison between joint configurations of robot and human. In the training phase, the robot babbles and the human imitates the robot's pose. When the robot detects imitation (based on contingent motion timing), it saves a snapshot of the appropriate body part (a tuple of sensor data and current robot pose). Once a sufficient number of snapshots are recorded, this data can be used to create a mapping between the two spaces which is applied thereafter in realtime.

## 3.2 Body mapping

In order to compare observed human motions to the robot's motion repertoire, it is important for the human motions to be in the same representation as the robot's own motions. This can be difficult, because human morphology may not be the same as the robot's. Also, whatever sensing technology is used to provide data of human movements is unlikely to provide data in a way that can be related directly to the robot's representation of its own motions.

We use a mapping technique where the relation between sensed human body positions and robot's own body positions is learned through an imitative interaction [Gray, J. et al., 2005, Breazeal et al., 2005]. This technique allows the joint angle configuration of the human to be mapped efficiently to the geometry of the robot as long as the human has a consistent sense of how to mimic the poses of the robot and is willing to go through the quick, imitation-inspired process to help the robot learn this mapping. Figure 3-1 presents a schematic of this process.

Provided a robot has access to labeled training data (i.e., matched observations of human body position to the corresponding robot body position), a number of different mapping techniques could be applied to approximate a function that maps continuously between these spaces. To acquire this data, the robot engages the human in an intuitive "do as I do" imitation game, inspired by early facial imitation whereby human infants learn how to imitate the facial expressions of their caregivers [Meltzoff and Moore, 1997].

In this scenario, the robot first takes the lead and moves through a series of poses as the human imitates (the Motor Babble Action). Because the human is imitating the robot, there is no need to manually label the tracking data – the robot is able to self-label them according to its own pose. Instead of recording correlations between human pose and robot pose continually throughout this interaction, ideally the robot would monitor the human and record only those correlated poses where the human is actually imitating the robot. Granted, the robot cannot tell definitively whether the human is imitating it – this requires knowing the mapping it is currently trying to learn. Nevertheless, the robot can measure the amount of overall motion of the human's body to estimate contingent response (i.e., the Motion Detector and Contingency Detector). Namely, if the human starts and stops their movements contingently after the robot starts and stops its own, the robot assumes that the human is cooperating in the "imitation game". The robot then records training instances once the robot and the human complete each pose in the sequence (the Snapshot Recorder).

In prior work, we have found that it is often difficult for people to imitate the entire full-body pose of the robot at once [Breazeal et al., 2005]. To remedy this, we divide the training of each body region into separate "zones" (e.g., right arm, right hand, etc.). We make use of the symmetry in human and robot morphology to mirror and re-use training data, eliminating redundant training steps.

Once the robot finishes acquiring its training set, it creates a reverse map from human body position to robot body position. We use radial basis functions to perform this reverse mapping. While RBF's provide a very faithful mapping in the vicinity of the example poses, the error increases with inputs that are distant from the training poses. Additionally we found that if the robot used closely neighboring examples during training, the human could not distinguish them adequately to provide different training poses, and significant warping of the resulting map could occur. The robot must therefore use a carefully selected set of training poses to cover the movement space – an example set of poses for training an arm

Figure 3-2: Example Training set - Training poses for Leo's right arm.

mapping is shown in Figure 3-2.

When the robot relinquishes the lead to the human, the robot then tries to imitate the human's pose. The human can then take a reinforcement learning approach, verifying whether the robot has learned a good mapping or whether a further iteration of the imitative interaction is required.

We have used this technique to learn facial imitation based on facial features tracked using the AxiomFFT system. In this case, the human imitates facial expressions of the robot until the robot has enough samples to train a neural network that maps between perceived 2D locations of human facial features in image coordinates to robot's facial joint space [Breazeal et al., 2005]. We have also used it to learn a mapping from arms and torso of an observed human to corresponding body regions of the robot using a motion capture suit [Gray, J. et al., 2005], and later using an optical motion tracking system [Brooks, A.G. et al., 2005] as the input observations of the human pose. Figure 3-4 shows images captured during these two types of mimicry.

## 3.3 Matching observed actions to motor repertoire

Once the mapping is created, perceived data can be transformed into the same *pose* format the robot uses to store its own motion plans. The robot's motor repertoire is represented as a directed graph of connected poses, called the *pose graph*. The nodes represent specific body poses, and the arcs represent allowed transitions between them. Families of poses can be represented as a sub-graph of actions (e.g., different kinds of reaching, pointing, waving, etc.) and links between sub-graphs represent allowable transitions between families of actions. In addition, weighted blends of either discrete poses or full trajectories can be generated to enlarge the repertoire of possible movements [Downie, 2000]. For instance, the robot may have six explicit reaching movements represented in its pose graph (primitives), but can generate a new reaching movement using a weighted blend of reaching primitives to span its entire workspace. The goal of this example based technique is to satisfy the dual goals of having the robot produce lifelike, expressive motion characteristic of human-made animations while still having the flexibility to behave autonomously. In some cases if we need exact positioning (such as flipping a switch) we will start with the blended solution and augment it slightly using inverse kinematics to achieve the end effector position while attempting to preserve what we can of the animated motion.

This structure is quite a useful way to represent the motions of the robot. In practice, we overlay multiple motor systems for different body regions that can run simultaneously. This allows the robot to perform multiple motions simultaneously, such as pointing to an object, directing its gaze towards it, nodding, and expressing an emotional state such as interest. Individual trajectories specify the joints they require which allows the system to determine which motions are compatible with others (can be run simultaneously) .

Once the observed movements are represented within the pose graph, strung together into a trajectory through this space, the next challenge is to determine whether this trajectory is similar to (or can be generated by) any that exist within the robot's motor repertoire. Many interesting techniques exist and others are being developed to determine the match between trajectories based on the relative importance of spacial errors, timing errors, etc

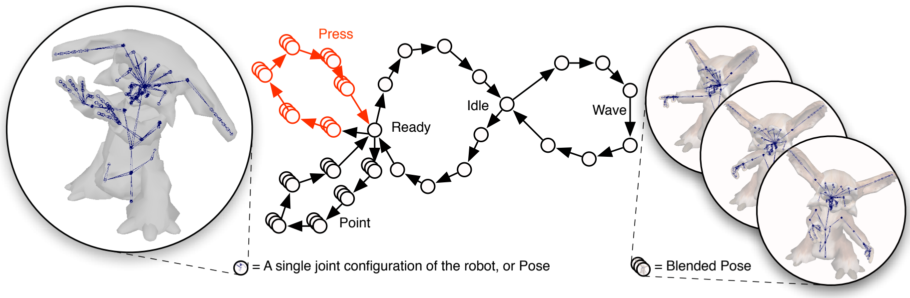


Figure 3-3: Example Posegraph. Each circle represents a single joint configuration of the robot. The Posegraph is a directed graph indicating allowed transitions between these nodes. Certain nodes (Blended Nodes) contain multiple joint configurations - in this case, a dynamic weight vector indicates how to blend these together into a single resultant pose. Section 3.2 describes how the robot maps an observed human pose to one of its graph nodes, then Section 3.3 discusses going from a sequence of these mapped poses to a labeled trajectory (eg, "Press" in red).

(e.g., [Jenkins and Matarić, 2002, Demiris and Hayes, 2002]). In the interaction described here we were able to use a simple heuristic to provide a goodness of fit measure: a voting system that chooses trajectories based on a running best-overall-matching-pose measure. However in interactions with more motions that need to be classified, we have also explored the use of Morphable Models to provide a more general solution [Brooks, A.G. et al., 2005].

Representing observed human's movements as one of the robot's own movements is useful for further inference using the Intention System. Rather than trying to recognize human behavior purely from a collection of joint angle trajectories, the Intention System integrates this motor information with other context provided by task schemas (that link environmental conditions with actions to achieve expected outcomes). This is described in the following sections.

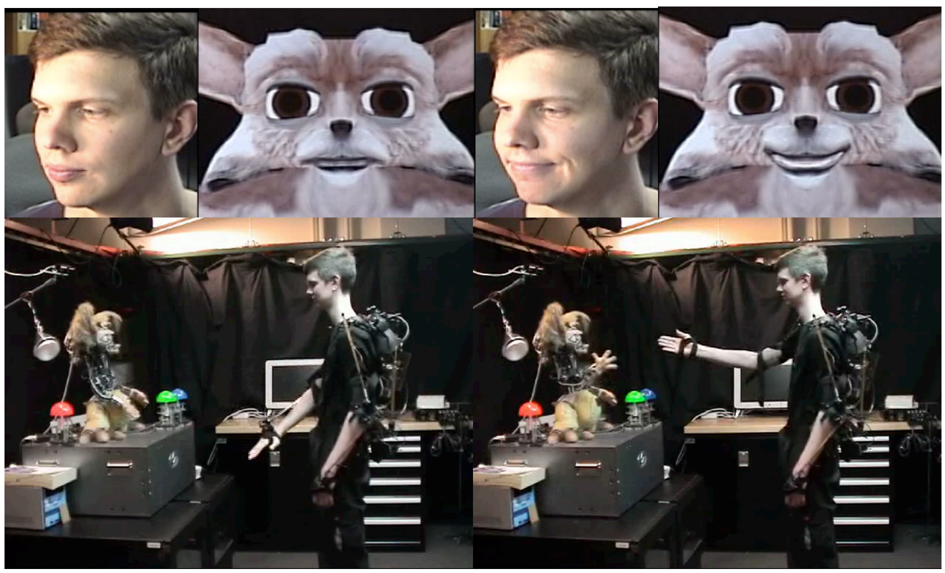

Figure 3-4: Demonstration of motor mimicry. These pictures show two different domains where the motor mapping techniques from this chapter have been demonstrated. The top row shows two human facial expressions and the resulting mimicry performed by the robot. In these examples, the output shown is the result of the process in section 3.3 - once the learning phase is over, the human's facial points are mapped into joint angles, then those joint angles are used as an input to search the robot's pose-graph for the closest animation (or in this case, the closest weighted blend of facial expressions animations). The bottom row shows the robot mimicking the position of the human's arm. In this case, the direct result of the joint mapping 3.2 is being performed.

# Chapter 4

# Visual Perspective Taking

The last chapter covers mechanisms that will help make mental state inferences by looking at the physical actions an agent makes, and trying to work back from there. This chapter looks at the opposite approach, monitoring the perceptual input available to an agent and trying to simulate the world model they will internally be constructing based on that input.

## 4.1 The Perception and Belief Systems

In order to convey how the robot interprets the environment from the human's perspective, we must first describe how the robot understands the world from its own perspective. This section presents a technical description of two important components of the underlying cognitive architecture: the *Perception System* and the *Belief System*. The Perception System is responsible for extracting perceptual features from raw sensory information, while the Belief System is responsible for integrating this information into discrete object representations. The Belief System represents the architecture's approach to sensor fusion, object tracking and persistence, and short-term memory.

## 4.2 Percept modeling

On every time step, the robot receives a set of sensory observations $O = \{o_1, o_2, ..., o_N\}$ from its various sensory processes. *As an example, imagine that the robot receives information about buttons and their locations from an eye-mounted camera, and information about the button indicator lights from an overhead camera. On a particular time step, the robot might receive the observations $O = \{$(red button at position (10,0,0)), (green button at (0,0,0)), (blue button at (-10,0,0)), (light at (10,0,0)), (light at (-10,0,0))$\}$.* Information is extracted from these observations by the Perception System. The Perception System consists of a set of *percepts* $P = \{p_1, p_2, ..., p_K\}$, where each $p \in P$ is a classification function defined such that

$$p(o) = (m, c, d), \tag{4.1}$$

where $m, c \in [0, 1]$ are match and confidence values and $d$ is an optional derived feature value. For each observation $o_i \in O$, the Perception System produces a *percept snapshot*

$$s_i = \{(p, m, c, d) | p \in P, p(o_i) = (m, c, d), m * c > k\}, \tag{4.2}$$

where $k \in [0, 1]$ is a threshold value, typically 0.5. *Returning to our example, the robot might have four percepts relevant to the buttons and their states: a location percept which extracts the position information contained in the observations, a color percept, a button shape recognition percept, and a button light recognition percept. The Perception System would produce five percept snapshots corresponding to the five sensory observations, containing entries for relevant matching percepts.*

## 4.3 Belief modeling

These snapshots are then clustered into discrete object representations called *beliefs* by the Belief System. This clustering is typically based on the spatial relationships between the various observations, in conjunction with other metrics of similarity. The Belief System maintains a set of beliefs $B$, where each belief $b \in B$ is a set mapping percepts to history functions: $b = \{(p_x, h_x), (p_y, h_y), ...\}$. For each $(p, h) \in b$, $h$ is a history function defined such that

$$h(t) = (m'_t, c'_t, d'_t) \tag{4.3}$$

represents the "remembered" evaluation for percept $p$ at time $t$. History functions may be lossless, but they are often implemented using compression schemes such as low-pass filtering or logarithmic timescale memory structures.

A Belief System is fully described by the tuple $(B, G, M, d, q, w, c)$, where

- $B$ is the current set of beliefs,
- $G$ is a generator function map, $G : P \rightarrow \mathcal{G}$, where each $g \in \mathcal{G}$ is a history generator function where $g(m, c, d) = h$ is a history function as above,
- $M$ is the belief merge function, where $M(b_1, b_2) = b'$ represents the "merge" of the history information contained within $b_1$ and $b_2$,
- $d = d_1, d_2, ..., d_L$ is a vector of belief distance functions, $d_i : B \times B \rightarrow \mathcal{R}$,
- $q = q_1, q_2, ..., q_L$ is a vector of indicator functions where each element $q_i$ denotes the applicability of $d_i$, $q_i : B \times B \rightarrow \{0, 1\}$,
- $w = w_1, w_2, ..., w_L$ is a vector of weights, $w_i \in \mathcal{R}$, and
- $c = c_1, c_2, ..., c_J$ is a vector of culling functions, $c_j : B \rightarrow \{0, 1\}$.

Using the above, we define the Belief Distance Function, $D$, and the Belief Culling Function, $C$:

$$D(b_1, b_2) = \sum_{i=1}^{L} w_i q_i(b_1, b_2) d_i(b_1, b_2) \tag{4.4}$$

$$C(b) = \prod_{j=1}^{J} c_j(b) \tag{4.5}$$

The Belief System manages three key processes: creating new beliefs from incoming percept snapshots, merging these new beliefs into existing beliefs, and culling stale beliefs. For the first of these processes, we define the function $N$, which creates a new belief $b_i$ from a percept snapshot $s_i$:

$$\begin{aligned} b_i = N(s_i) = \{(p,h) | &(p,m,c,d) \in s_i, \\ & g = G(p), h = g(m,c,d)\} \end{aligned} \tag{4.6}$$

For the second process, the Belief System merges new beliefs into existing ones by clustering proximal beliefs, assumed to represent different observations of the same object. This is accomplished via bottom-up, agglomerative clustering as follows. For a set of beliefs $B$:

**while** $\exists b_x, b_y \in B$ such that $D(b_x, b_y) < thresh$ **do**
  find $b_1, b_2 \in B$ such that $D(b_1, b_2)$ is minimal
  $B \leftarrow B \cup \{M(b_1, b_2)\} \setminus \{b_1, b_2\}$

We label this process $merge(B)$. Finally, the Belief System culls stale beliefs by removing all beliefs from the current set for which $C(b) = 1$. *In summation, then, a complete Belief System update cycle proceeds as follows:*

begin with current belief set $B$
receive percept snapshot set $S$ from the Perception System
create incoming belief set $B_I = \{N(s_i) | s_i \in S\}$
merge: $B \leftarrow merge(B \cup B_I)$
cull: $B \leftarrow B \setminus \{b | b \in B, C(b) = 1\}$

*Returning again to the example, the Belief System might specify a number of relevant distance metrics, including a measure of Euclidean spatial distance along with a number of metrics based on symbolic feature similarity. For example, a symbolic metric might judge observations that are hand-shaped as distant from observations that are button-shaped, thus separating these observations into distinct beliefs even if they are collocated. For the example, the merge process would produce three beliefs from the original five observations: a red button in the ON state, a green button in the OFF state, and a blue button in the ON state.*

*The Belief System framework supports the implementation of a wide range of object tracking methods, including advanced tracking techniques such as Kalman filters [Kalman, 1960] and particle filters [Carpenter et al., 1999]. The ability to specify multiple distance metrics allows sophisticated, general-purpose tracking methods such as these to operate side-by-side with hand-crafted rules which encode prior domain knowledge about object categories, dynamics and persistence.*

## 4.4 Belief inference and visual perspective simulation

When collaborating on a shared task, it is important for all parties involved to have a consistent representation of the task context. However, in complex and dynamic environments, it is possible for one collaborator's beliefs about the context surrounding the activity to diverge from those of other collaborators. For example, a visual occlusion could partially block one person's viewpoint of a shared workspace but not that of the other. Or, an event could occur that one person witnesses, but the other does not. There are many situations where the knowledge that two or more people have of a shared scenario can differ over time. The ability for an agent to estimate what others do and do not know based on their perceptual experience is at the crux of many false belief tasks. In this section we describe our method of modeling the knowledge of nearby humans based on their visual experience by taking their visual perspective.

As described in the previous section, belief maintenance consists of incorporating new sensor data into existing knowledge of the world. The robot's sensors are all in its own reference frame, so objects in the world are perceived relative to the robot's position and orientation. In order to model the beliefs of the human, the robot re-uses the same mechanisms used for its own belief modeling, but first transforms and filters the incoming data stream (see Figure 4-1). In this way, the beliefs modeled for the human are handled with the same tracking and maintenance systems that the robot uses for its own world model — however, the data is manipulated to simulate the first person experience from the perspective of the human

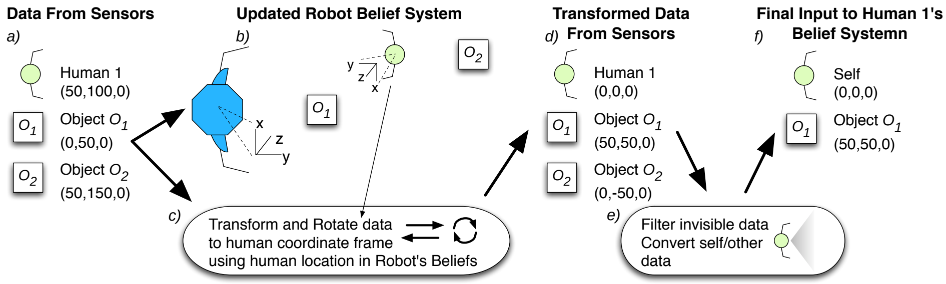


Figure 4-1: Perspective Transform of Sensor Data for Belief Modeling. *(a)* Data from sensors is used to update the robot's own model of the world (shown in *(b)*) via the normal Belief System update. *(b)* The the real world scenario and corresponding model – the robot (shown as an octagon) can see the human (shown as a circle) and two objects. The human can only see object $O_1$. Coordinate system orientation is shown next to the human and the robot where the origin is centered on each agent. *(c)* The human's position and orientation from this model are used to transform incoming sensor data to data that is relative to the human's coordinate system. ($d$) The result of the transformed data. *(e)* Next, objects that are out of sight of the human (estimated by an "attentional cone") are filtered out, and the data is transformed to human centric format. *(f)* This data is now ready to be presented to the Belief System that models the human's beliefs.

being modeled.

The robot can also filter out incoming data that it believes is not perceivable to the human, thereby preventing that new data from updating the model of the human's beliefs. If the inputs to the robot's perceptual-belief pipeline are the sensory observations $O = \{o_1, o_2, ..., o_N\}$, then the inputs to the secondary pipeline that models the human's beliefs are $O'$, where:

$$O' = \{P(o')|o' \in O, V(o') = 1\} \tag{4.7}$$

where:

$$V(x) = \begin{cases} 1 & \text{if } x \text{ is visible to human} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

and:

$$\begin{aligned} P :&\{\text{robot local observations}\} \\ &\rightarrow \{\text{person local observations}\} \end{aligned} \tag{4.9}$$

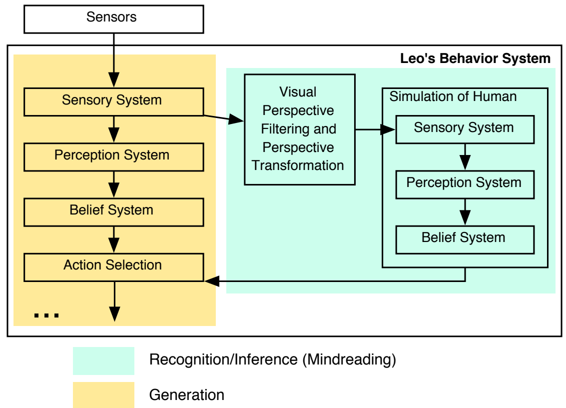


Figure 4-2: Architecture for modeling the human's beliefs re-uses the robot's own architecture for belief maintenance.

Visibility is determined by a cone calculated from the human's position and orientation. The robot also filters out objects whose view is blocked by occlusions (for any occlusions that it can detect).

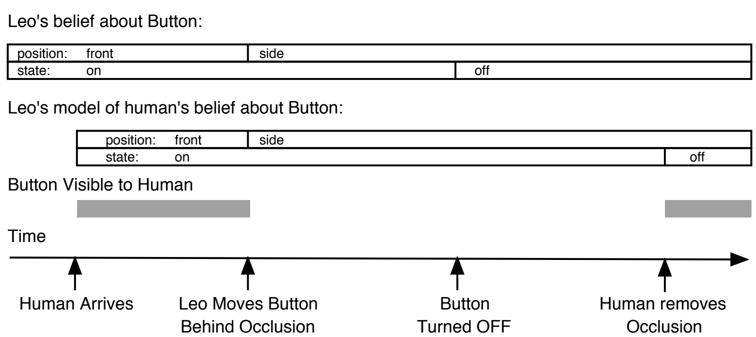


Figure 4-3: Timeline following the progress of the robot's beliefs for one button. The robot updates its belief about the button with any sensor data available - however, the robot only integrates new data into its model of the human's belief if the data is available when the human is able to perceive it.

Maintaining this parallel set of beliefs is different from simply adding "is-visible-to-human" metadata to the robot's original beliefs because it reuses the entire architecture which has mechanisms for object permanence, history of properties, etc. This allows for a more sophisticated model of the human's beliefs. For instance, Figure 4-3 shows an example where this approach keeps track of the human's false beliefs about objects that have changed state while out of the human's view. This method has the advantage of keeping the model of the human's beliefs in the same format as the robot's own, allowing both for direct comparison between the two and operating on these beliefs with the same mechanisms that operate on the robot's own. This is important for establishing and maintaining mutual beliefs in time-varying situations where beliefs of individuals can diverge over time.

These capabilities allow the robot to pass a variant of the classic "Sally-Anne" false belief task [Wimmer and Perner, 1983], originally created as a diagnostic test to determine how children develop the ability to track the beliefs of other people as different from their own. In the test, an object is moved from one hiding place to another while an actor is outside the room. When the actor returns, successfully passing the test requires the child to know that the actor is likely to think that the object remains where they last saw it (thus, that actor has a false belief about the location of the object). Figure 4-4 shows a demonstration of leo using the techniques described here to pass a similar test.

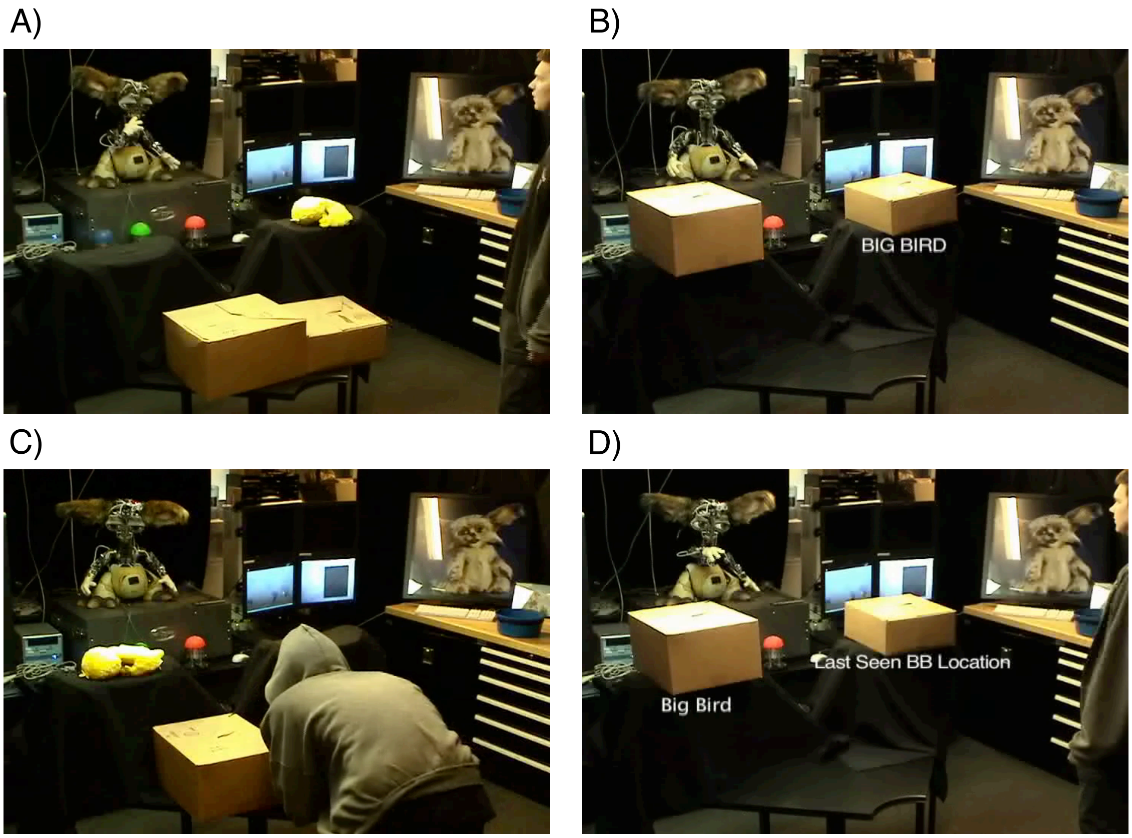


Figure 4-4: Excepts from a video demonstrating the robot's ability to perform a variant of the Sally-Anne false belief task. In $(A)$, big bird is placed on the right stand. In $(B)$ the subject has covered both stands and left the room. A different (as far as the robot is concerned – the hood prevents the robot from seeing the same person) human moves the puppet and replaces the covers in $(C)$. In $(D)$ the robot is asked to point to the location where the subject believes the puppet to be, and points to where that subject last saw it - the rightmost stand.

# Chapter 5

# Inferring Intentions

The Intention System is responsible for generating the goal-achieving behavior of the robot. This representation for goal-directed action enables the robot to plan a set of actions under particular circumstances to achieve a desired result. This chapter describes how the robot can combine inferences from the previous chapters with reuse of this system to move into higher level inferences about an agent's mental states, trying to determine the person's plans and goals based on what the robot's would be if it were performing the same action in the human's situation. The reuse of the robot's own task structures allows the robot to achieve certain types of helpful behavior. First the robot determines where the human's actions fit within its own task structures, and what the human's particular goals are given that structure and the human's model of the world. Then the robot can determine how it could achieve that goal, given the same task structure but using it's own (possibly divergent) model of the world.

The following sections describe the task representation and the processes that operate on this representation to generate behavior and to produce intentional inferences. Then, in section 5.4, we work through a detailed example of how these processes function, providing additional technical details of the fielded system. Section 5.5 describes a set of benchmark tasks designed to demonstrate the performance of this system in different situations. It includes the results of a human subject study used to validate the performance of the robot against that of humans in the same position.

## 5.1 Task Representation using Schemas

Within the deliberative system of the robot, the atomic-level representation of a goal-directed behavior is a schema that associates its necessary perceptual preconditions with a specific action (optionally performed on a particular object, or with other parameters) to achieve an expected outcome – its goal.

As such, it resembles STRIPS operators within classic planning literature. Schemas can be organized sequentially and/or hierarchically to create larger structures to represent tasks and execute them. When chaining sequential schemas, the goal of one schema becomes the precondition of the subsequent schema. Compound tasks are specified as a hierarchy of schemas, where the expected result of multiple schemas are the inputs (i.e., listed in the preconditions) of the subsequent schema. To achieve some desired task goal, only the relevant schema need be activated and all necessary preconditions will be fulfilled. Figure 5-1 shows an example schema structure.

Each schema has a number of individual components. It has a Motor Action, which causes the robot to physically perform some sort of movement trajectory by activating the corresponding path within the pose graph (described in Section 3.3). It also has an Evaluative Mechanism to determine the success of the action.

Both the Motor Action and the Evaluative Mechanism may depend on additional parameters (for instance, specifying one or more target objects in the world). These parameters may be provided externally from a higher level deliberative processes if the schema is directly activated to produce the robot's own behavior (see Section 5.2). If the schema is activated in simulation-mode in response to an observed action, then the parameters must be discovered based on observation (see Section 5.3). If the schema is activated as a precondition for a sequence of schemas, however, the associated parameters must be automatically determined based on the parameters of the downstream schema.

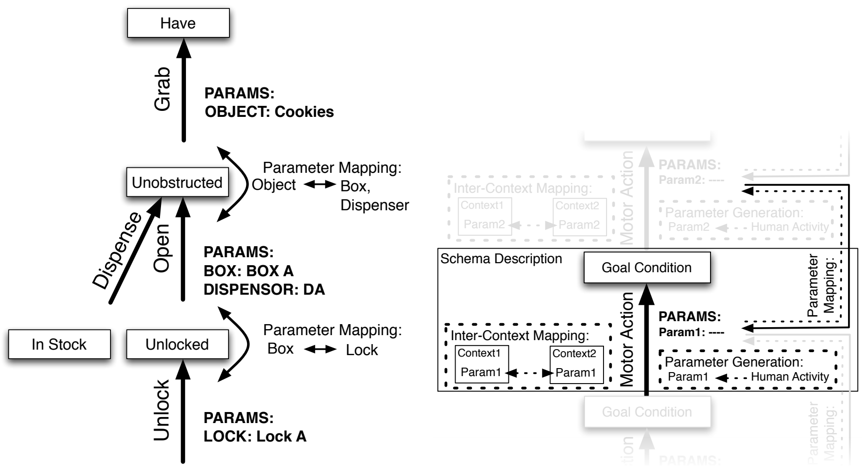


Figure 5-1: Example of a task representation. In the simplified task representation to the left, the agent intends to get cookies. There are two possible behaviors to attain cookies: open a locked box that contains cookies, or operate a dispenser to release cookies. In the more detailed figure to the right, the highlighted schema acts as the precondition for the upper schema, while the lowest schema is the precondition for the highlighted schema. In order for the schema to activate a necessary precondition (or to evaluate whether it is necessary using that precondition schema's goal condition) it may need to compute the necessary parameters relevant for that schema based on its own parameters. The downward mapping (solid line) in generation-mode is necessary to perform and evaluate precondition schemas based on the parameters of an upper schema. The upward mapping for simulation-mode (dashed line) is used to populate later schemas with potential parameters based on known precondition parameters (used during an attempt to predict an ultimate goal for an observed action). Finally the Inter-Context Mapping module is necessary when the robot is trying to compare observed goals to its own world knowledge in order to formulate a helpful plan. It must have a metric to determine how parameters that a human is using (that often relate to his or her possibly differing beliefs about the world) can be expressed in terms of the robot's own world knowledge.

## 5.2 Generating Goal-Achieving Behavior

In behavior generation-mode, schema structures can be traversed top-down to achieve goals and automatically satisfy preconditions in the process. In this process, the robot activates the top level schema which in turn may need to activate other supporting schemas before it can execute fully. When schemas are activated to generate the robot's own behavior, most schemas are parameterized by a set of arguments that adapt how the motor action operates to suit the situation at hand. For instance, the style of the action may be adjusted to express the robot's affective state, or a particular object could be set as the target for a given action. When this first schema is activated, the parameters (or target) for this schema are based on its goal. It is then up to the schema hierarchy to automatically generate schema parameters for any other required schemas based on the initial parameters provided to this first top-level schema.

At every juncture between schemas in a hierarchy, there exists a Parameter Mapping Module. This module is designed to generate the necessary parameters for precondition schemas based on the existing parameters for the parent schema. For example, in Figure 5-1 the robot's goal is to get cookies. Grasping cookies requires an unobstructed path. In this case, however, cookies are believed to be in a Box or in a Dispenser that the robot can perceive. The Unobstructed Schema is activated to reveal a clear path to cookies. One strategy is to Open the target box, Box A. As it turns out, Box A is locked. Accordingly, the robot should attempt the Unlocked Schema to Unlock the correct lock, namely Lock A.

Note that for any given situation, many schemas within the robot's task repertoire will not be relevant. In these cases, the Parameter Mapping Module will not be able to assign parameters to instantiate the corresponding schema. This indicates that the current context is not appropriate for that precondition schema to be performed. Hence, this serves an important filtering process whereby the robot only entertains executing schemas that are relevant and performable.

The system currently uses this filtering process to select one goal schema that describes the human's behavior. However, for future tasks in more complex environments, it will be important to revise this system to maintain multiple, probabilistic hypotheses. Another important future addition to the system is in the area of probabilistic actions. The robot's goals are specified relative to perceived world state which gives it some persistence in the face of failed actions, however a more explicit modeling would be required to use that information for planning and re-planning in the face of motor failure and other uncertain outcomes.

## 5.3 Inferring Intent from Observed Behavior

In simulation-mode, the robot tries to infer the intention of a person's observed course of action. To do so, the robot traverses schemas in the reverse direction. As schemas are traversed bottom-up, each schema's Parameter Mapping Module is applied to the robot's model of the human's beliefs — mapping parameters relevant to a precondition upwards to parameters necessary for the next higher-level schema. In general, the reverse mapping may be ambiguous (for instance, if someone is opening a box containing multiple items, which one might they want to grab?), and it may also by arbitrarily complex. For this reason, the architecture allows for each action to specify its own mapping function which handles both forward and reverse mapping. The actions used in this demonstration employ a mapping function based on object types and spatial relationships, which can operate similarly in either forward or reverse operation.

For instance, if schema $S_1$ (operating in relation to belief $b_1$) is a precondition to schema $S_2$ (operating in relation to $b_2$), then if either $b_1$ or $b_2$ is known the other can be determined according to the following:

$$b_1.isTypeForS_1 \wedge b_2.isTypeForS_2 \wedge r(b_1, b_2), b_1, b_2 \in \text{Beliefs}$$

where $r$ is the relation that must hold between the beliefs.

*For the schemas described here, $r$ is a position-based relationship. For example, in the case* *of a lock and a box:*

$$r(b_1, b_2) = |b_1.location - b_2.haspLocation| < 20cm$$

In simulation-mode, some schema parameters must be detected through direct observation, such as the target object of an observed action. In this case, a Parameter Generation Module (associated with each schema) computes the specific arguments necessary to simulate an observed schema in the manner it is being performed by the human. For instance, a person's arm trajectory for a reaching movement has different end purposes depending on what is being reached for: to grasp cookies, to open a lid, to unlock a lock, etc. In this case, the Parameter Generation Module for the Reaching Schema produces its values based on the robot's models of the beliefs of the person as estimated by the Belief System — namely, an object near the person's hand that he or she can see.

$$\begin{aligned} target = b \text{ iff } \exists h, b \in \mathrm{B}\ \big\{&b.isCorrectType \wedge h.isOwnHand \\ &\wedge |b.position - h.position| < thresholdDistance\big\} \\ &\text{where B is the subject's Beliefs} \end{aligned}$$

If there exists such a $b$, then the Parameter Generation Module has determined the relevant target $b$, and the robot concludes that the attached schema may be relevant to the observed action.

Analogous to the filtering role of the Parameter Mapping Module described previously, these Parameter Generation Modules also serve an important filtering function that narrows the relevant candidate schemas that may describe the human's observed behavior. If the Parameter Generation Module is unable to populate its schema with the appropriate arguments for the current situation, the robot concludes that that schema does not describe the human's current behavior.

To summarize, the Intention System runs in simulation-mode to enable the robot to observe the human and infer their goal. This is achieved by first determining which schema in the robot's own repertoire matches the human's activity by finding a schema whose motor action matches the observed action of the human and whose Parameter Generation module indicates that it is a relevant schema in the human's current context. From there, the robot can traverse upwards in the schema hierarchy to try to determine the ultimate goal of the observed behavior. At each step, the robot must attempt to predict the relevant parameters of the higher (temporally later) schema based on the parameters of the lower (preceding) schema using the connecting Parameter Mapping Module. Once it comes to a point where there are no more unique, valid, higher schemas, (this can happen because the schema structure has no further schemas, because a Parameter Mapping Module cannot map parameters any further, or because there are more than one valid schemas or parameters for the next step) then it has found the farthest goal it can predict into the future without being ambiguous (See Algorithm 1).

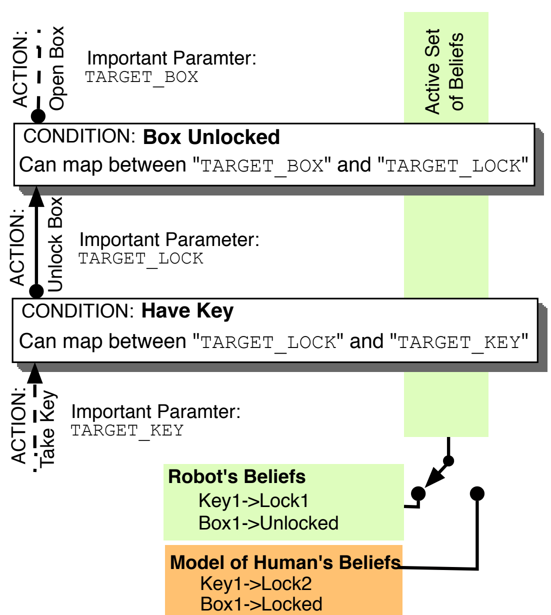


Figure 5-2: Section of action schema. Schemas can be traversed in either direction, upwards to infer goals and downwards to find ways to achieve those goals. Parameters for the actions are converted by the Conditions, using data in the model of the human's beliefs (for goal inference) or the robot's own beliefs (for task accomplishment)

**Algorithm 1** Finding the Human's Goal

**abbreviations:** PGM = Parameter Generation Module, MA = Motor Action
PMM = Parameter Mapping Module

```
ObservedSchema():
  loop
      for all s ∈ SCHEMAS do
          if s's PGM succeeds and
            s's motor action matches the human's then
              return s

FindGoal():
  OS ⇐ ObservedSchema() {Blocks until Schema is Observed}
  Params ⇐ OS's PGM operating on humansBeliefs
  loop {climb schema tree through unique valid schemas}
      matchingSchemas ⇐ Empty List
      for all s ∈ SCHEMAS such that OS is precondition of s do
          if OS's PMM can map Params to s using humansBeliefs then
              add s to matchingSchemas
      if matchingSchemas contains exactly 1 element then
          OS ⇐ first element of matchingSchemas
          Params ⇐ OS's PMM maps Params to OS using humanBeliefs
      else
          break {no higher unique schemas found}
  return OS, Params {goal is schema populated with parameters}
```

## 5.4 An Example: Goal Assistance

Using all the parts described above, the robot can infer what the human is intending to do even if their beliefs about the situation are false or incomplete and their resulting course of action will fail to accomplish their goal. How might a robot help a person in this situation? We consider the case where the robot has true beliefs about the situation at hand. The robot can assist the human by first adopting the same goal and then computing a course of action that resolves the errors the human has encountered.

To adopt the human's goal, the robot maps goal information from the context of the human's beliefs into its own set of beliefs. The most common mapping is simply to find a belief in the human's estimated context that preserves a set of properties from the first belief.

Here a belief $B_{bs1}$ from one belief system maps to $B_{bs2}$ in another based on the properties $P$ if:

$$\forall p \in P, B_{bs1}\text{'s } p = T(B_{bs2}\text{'s } p)$$

$$T(p) = \begin{cases} \text{perspective transform of } p & p \text{ has location data} \\ \text{p} & p \text{ has no location data} \end{cases}$$

This goal can then be used to provide assistance. Consider the case where the robot can use the same schema hierarchy computed according to Section 5.3. Algorithm 2 summarizes this process. For readability the algorithms shown here are simplified to refer to only a single human, however the architecture supports multiple humans.

**Algorithm 2** Providing Goal-Based Assistance

**ProvideAssistance():**
  $GS$, $GP \Leftarrow$ FindGoal()
  $GP \Leftarrow GS$ maps goal params from $humansBeliefs$ to $robotsBeliefs$
  **if** $GP =$ NULL **then**
    **return** NULL {cannot help if robot cannot understand goal}
  $TGS$, $TGP \Leftarrow$ FindATerminalSchema($GS$, $GP$, $robotsBeliefs$)
  **if** CanPerform($TGS$, $TGP$) **then**
    perform($TGS$, $TGP$)
  **else if** ($TGS \neq GS$) or ($TGS = GS \wedge TGP \neq GP$) **then**
    pointTo($TGP$)

**FindATerminalSchema($GS$,$GP$,$B$):**
  **for all** $s \in$ SCHEMAS such that $s$ is precondition of $GS$ **do**
    $sGP \Leftarrow s$'s PMM mapping $GP$ to $s$ using $B$
    **if** $sGP \neq$ NULL
      and $s$ is not accomplished according to $sGP$ and $B$ **then**
        **return** FindATerminalSchema($s$, $sGP$, $B$)
  **return** $GS$, $GP$

Figure 5-3 illustrates this process, following the example in Figure 5-1, where the human wants a bag of cookies he believes is contained in Box A. However, the human's beliefs are false as the cookies were moved to Box B while the human was not looking. The robot saw this switch take place and therefore has true beliefs of the situation. The robot uses this knowledge to help the human get the object of his desire. This is the general premise for our cooperative behavior experiments in Section 5.5.

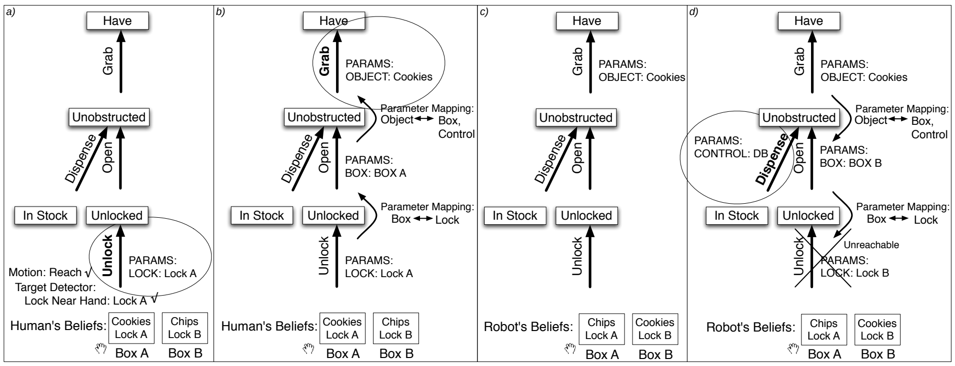


Figure 5-3: **Example goal inference and helpful behavior**. In this example, the human is trying to gain access to a bag of cookies which he believes is locked in Box A. The robot saw the cookies moved to Box B without the human seeing this event, so the human has false beliefs of the true location of the cookies. The schema hierarchy shown here describes two possible solutions that the robot knows to produce a food item: either unlocking and opening the correct box, or dispensing a matching food item from a dispenser that it can operate. Flow diagrams *(a–d)* represent the corresponding schema hierarchy that are evaluated in the context of a particular set of beliefs (either the human's or the robot's) shown at the bottom. *(a)* The robot detects a "Reach" motion and the relevant context for the "Unlock" segment ("own-hand-near-lock" from the human's perspective). This corresponds to the human reaching for the lock on Box A. *(b)* The process traverses up the hierarchy, using a model of the human's beliefs as input to the parameter mapping functions to predict targets for the potential human actions that are likely to follow the current action. In this example, the robot determines that the human's desire is to get the cookies. *(c)* Once a final goal is calculated, the process switches to the robot's own belief context. The robot knows that chips are actually in Box A and cookies are in Box B. *(d)* Again, the system uses parameter mapping to determine the targets of relevant actions necessary towards the goal, but this time starting from the goal and working backwards using knowledge from the robot's own beliefs. The robot can then choose an action that helps the human attain his goal: either unlocking Box B (the robot realizes the human is looking in the wrong box), or dispensing a bag of chips from the robot's dispenser. For instance, a principle of "least effort" can be applied to decide between the two.

## 5.5 Providing Assistance on a Physical Task

In order to evaluate our cognitive architecture, we have developed a novel set of benchmark tasks that examines the use of belief reasoning and goal inference by robots and humans in a collaborative setting. Our benchmark tasks are variants of the classic Sally-Anne false belief task from developmental psychology, but embedded within a live, cooperative setting. Subjects interact face-to-face with a partner (an experimental confederate), and are prompted to assist their partner in any way they see fit. Language is not required to perform these tasks. Instead of probing the participant with an explicit prompt (e.g. "where will your partner look for the cookies?"), we observe their behavior as they attempt to assist their partner. Our objective is to examine the spontaneous use of goal inference and false belief reasoning in collaborative activity.

### 5.5.1 Benchmark Tasks

A schematic of four benchmark tasks is shown in Figure 5-4. In each task, the Subject (i.e., a human or robot) interacts with a collaborative partner (Actor) who is an experimental confederate. The Subject has access to a collection of food objects (cookies in a small red package or chips in a larger blue package) that are identical to hidden target objects locked away in opaque boxes that their partner (Actor) may be searching for. It is thus possible for the Subject to assist their partner (Actor) by giving them the food item that matches the target of their search without requiring the Actor to figure out how to unlock the appropriate box. Or the Subject can communicate relevant information, such as gesturing to the location of the target item.

In those tasks that call for boxes to be sealed, color-coded combination locks are used. Two of the lock's four numeric dials are covered up and fixed in place by electrical tape, leaving only two dials free for manipulation. This lock mechanism served an important timing function in our study, introducing a delay in the Actor's process of opening any sealed box. This gives the Subject sufficient time to consider the Actor's goal and beliefs and then perform

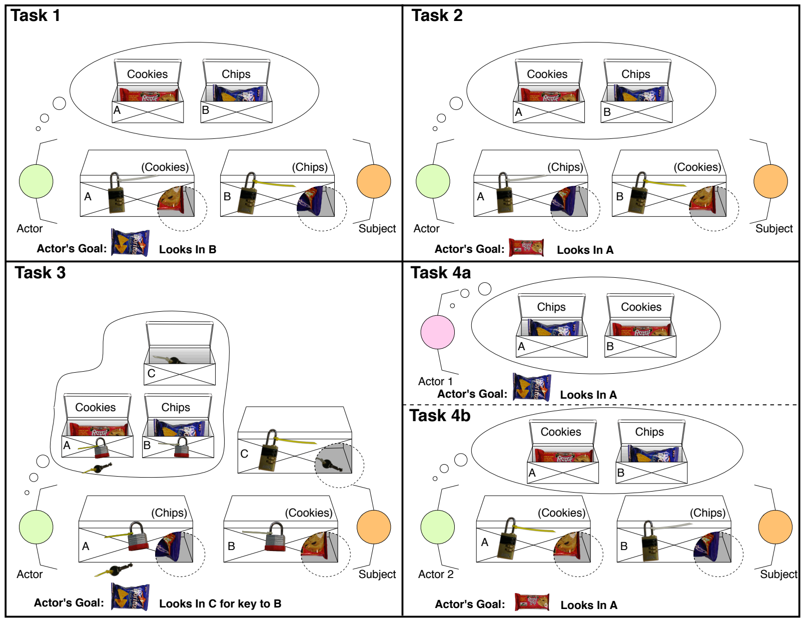


Figure 5-4: The four collaborative benchmark tasks using two different target objects, a small package with cookies and a larger package with chips. See text. Shown are the actual world state and the Actor's belief state depicted by a "thought bubble" at the moment when the Subject's behavior is classified.

potential helpful actions before the Actor unlocks the box.

**Task 1** is a control task examining simple goal inference. The Subject and Actor both watch as the experimenter hides a package of cookies in Box A and a bag of chips in Box B. The experimenter then seals both boxes. The Actor receives instructions written on a notecard to deliver a bag of chips to the experimenter. The Actor proceeds to attempt to open Box B, and the Subject's subsequent behavior is recorded. In order to successfully assist the Actor, the Subject must infer that because the Actor is attempting to open Box B, the Actor's goal is to acquire the chips contained within the box.

**Task 2** examines goal inference with false beliefs. The setup proceeds as in Task 1, with Subject and Actor both observing cookies hidden in Box A and chips hidden in Box B. After the boxes are sealed, the Actor is asked to leave the room, at which point the experimenter swaps the contents of the boxes. The Actor returns, receives instructions to get the cookies, and attempts to open Box A. In order to successfully assist the Actor, the Subject must infer that the Actor's goal is to acquire the cookies, even though Box A currently contains the chips.

**Task 3** examines goal inference with false beliefs and indirect, dislocated action. The setup proceeds as in Task 2, however, in this case, the experimenter locks both Box A and Box B with color-coded padlocks. The key to Box A is left in plain view, but the key to Box B is sealed inside of a third box, Box C. The Actor is then asked to leave the room, at which point the experimenter, using a master key, swaps the contents of Box A and Box B, leaving both boxes locked. The Actor returns, receives instructions to get the chips, and attempts to open Box C to get the key that unlocks Box B. In order to successfully assist the Actor, the Subject must infer that the Actor's goal is to acquire the chips, even though the immediate target of the Actor's actions, Box C, contains neither the chips nor even the key to a box containing chips.

**Task 4** examines goal inference with multiple agents and false beliefs. In this task, the Subject is introduced to two collaborative partners, Actor 1 and Actor 2. All three watch as Actor 2 hides chips in Box A and cookies in Box B, and then seals both boxes. Actor 1 is then asked to leave the room, at which point Actor 2 swaps the contents of Box A and Box B in view of the Subject. Actor 2 is then asked to leave, and Actor 1 returns. Actor 1 receives instructions to get the chips and attempts to open Box A. The Subject's subsequent behavior is recorded (Task 4a). Finally, Actor 1 leaves, and Actor 2 returns, receives instructions to get the cookies, and also attempts to open Box A. The Subject's behavior is recorded (Task 4b). In order to successfully assist both actors, the Subject must keep track of Actor 1's false beliefs about the object locations as well as Actor 2's correct beliefs about these locations.

### 5.5.2 Human Subjects Study

We conducted a human subjects study to gather human performance data on our collaborative benchmark tasks.

Figure 5-7 shows some of the essential elements of our study setup. Target objects were hidden in three flight cases (A), (B), and (C). Our experimental confederate and the study participant were seated opposite each other at locations (D) and (E), respectively. The participant's stock of food objects was located on a stool, (F), adjacent to their chair and out of the reach and view of the confederate. The target objects, (H), were a bright red package of chocolate-chip cookies and a bright blue bag of corn chips. Also shown are the viewpoint from the participant's location, (I), and the viewpoint from the confederate's location, (J) - note that the stock of food objects is not visible from this location.

A detail of our box-sealing mechanism is shown in (G). When attempting to open a sealed box, the Actor (experimental confederate) systematically tries two digit combinations in numeric order, starting at zero and tugging at the lock with each iteration. The correct code was always 21, so the Actor could open the lock within 30 to 45 seconds, giving the Subject sufficient time to consider the Actor's goal and contemplate potential helpful actions, while keeping the experiment running at a reasonable pace.

We gathered data from 20 participants: 11 females and 9 males, with ages ranging from 18 to 65. Our participants were a mix of undergraduates, graduate students, and staff from the MIT community. Participants were each presented with the four benchmark tasks in randomized order. Participants were instructed not to talk to their partner, but were told that they were otherwise free to perform any action or gesture that might help their partner achieve the goal. Participants were instructed that they might find the objects on the stool next to their chair useful, but that they could only use one of these objects per task.

The results of the study are summarized in Table 5.1. Participant behavior was partitioned into six categories, from most helpful to least helpful: correct object presented, guidance gesture presented, grounding gesture presented, other, no action, incorrect object presented. Behavior was classified as follows. If the participant presented the correct target object to their partner, they were tallied as "correct," and if they presented the wrong object, they were tallied as "incorrect."

Participants who did not present either object were classified according to the gestures that they displayed. "Guidance" gestures included only direct pointing or manipulation towards

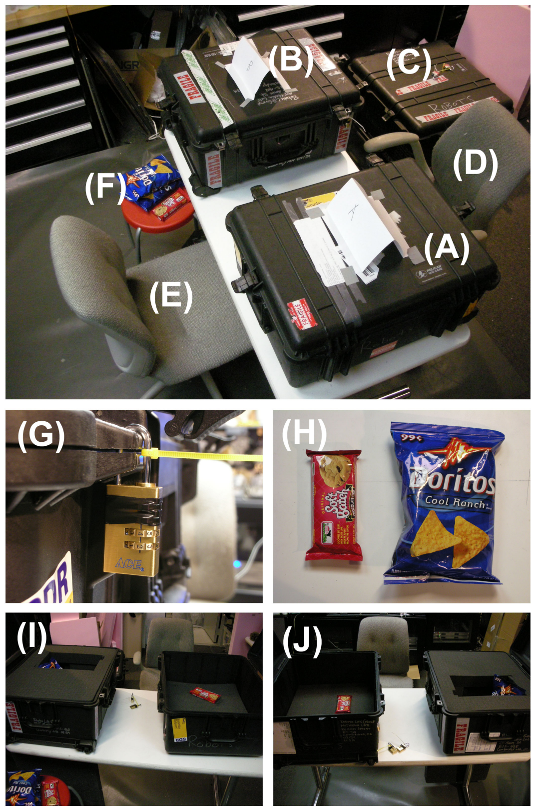


Figure 5-5: Setup of the human subjects study. (A,B,C) Boxes in which target objects were hidden. (D) Confederate's chair. (E) Participant's chair. (F) Objects available to participant. (G) Detail of box with combination lock. (H) Target objects. (I) Participant's viewpoint. (J) Confederate's viewpoint.

Table 5.1: Participants were instructed not to talk to their partner, but were told that they were otherwise free to perform any action or gesture that might help their partner achieve the goal. If the participant presented the correct target object to their partner, they were tallied as "Correct," and if they presented the wrong object, they were tallied as "Incorrect." Participants who did not present either object were classified according to the gestures that they displayed. "Guidance" gestures included only direct pointing or manipulation towards the correct target box, lock, or key. "Grounding" gestures included bi-directional pointing gestures indicating that the box contents had been swapped, as well as the use of the matching food objects as a "map" to indicate the correct contents of the various boxes. In the absence of such gestures, behavior was tallied as "No Action." Finally, two unexpected cases were tallied as "Other" as described in the table notes. It should be noted that in the case of Task 3, Guidance gestures were almost as helpful as producing the correct object, since indicating the correct padlocked box or its readily-available key resulted in the rapid acquisition of the contents of the box.

| Task | Correct Object | Guidance Gesture | Grounding Gesture | Other | No Action | Incorrect Object |
|---|---|---|---|---|---|---|
| Task 1 | 16 | 0 | 0 | 1† | 1 | 2 |
| Task 2 | 14 | 1 | 2 | 0 | 0 | 3 |
| Task 3 | 13⋆ | 5 | 2 | 0 | 0 | 0 |
| Task 4a | 14 | 2 | 1 | 0 | 3 | 0 |
| Task 4b | 13 | 0 | 1 | 1‡ | 1 | 4 |

⋆ One participant produced the object after the key was retrieved from box C.

† Participant successfully pried open the locked target box.

‡ Participant discovered the combination lock code and revealed it gesturally.

the correct target box, lock, or key. “Grounding” gestures included bidirectional pointing gestures indicating that the box contents had been swapped, as well as the use of the matching food objects as a “map” to indicate the correct contents of the various boxes. In the absence of such gestures, behavior was tallied as “no action.”

Finally, two unexpected cases were tallied as “other” as described in the table notes. It should be noted that in the case of Task 3, guidance gestures were almost as helpful as producing the correct object, since indicating the correct padlocked box or its readily-available key resulted in the rapid acquisition of the contents of the box.

These results indicate that participants were largely successful at inferring the goals of their collaborative partners and engaging in helpful behaviors even in the presence of false beliefs, multiple agents, and indirect goal cues.

It should also be noted, however, that success was not uniform. Many participants found some of the tasks to be quite challenging. Many reported difficultly in remembering the locations of the hidden objects and the divergent beliefs of their collaborative partners.

### 5.5.3 Robot Experiment

In the robot version of the experiment, the robot, Leonardo, interacts face-to-face with one or more human partners. See Figure 5-6. The physical robot in this demonstration (and its virtual counterpart) is able to exhibit a large repertoire of non-verbal communication cues such as facial expressions, gestures, and gaze shifts. Leonardo can perform simple manipulation tasks in a small workspace with objects specifically designed for the robot. While it has a number of camera systems to perceive events, objects, and people in its workspace, in this demonstration the robot uses a 10 camera Vicon Motion Capture system to robustly track specific objects and particular human features (tagged with reflective markers) in real-time to millimeter accuracy.

In this set of experiments, the robot’s goal is to assist the human (or humans) given the Actor’s goal of obtaining a desired food items. The robot study followed the same protocol

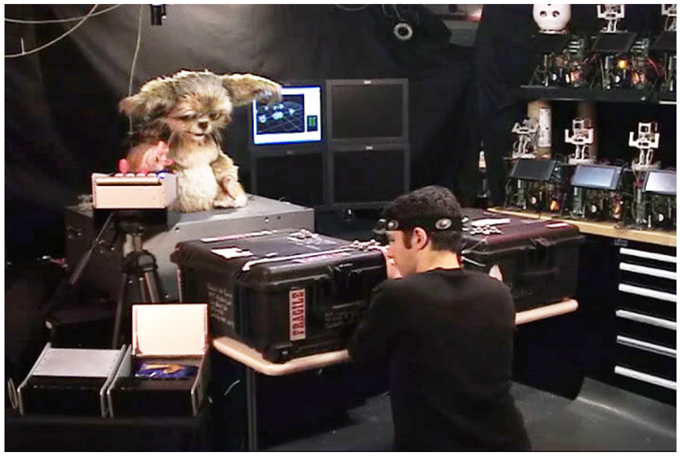

Figure 5-6: Leonardo can operate a remote control box to reveal the contents of two boxes located near the human.

as in the human study. Similar to the human study, language was not involved. The same objects and Actors were used in both studies, with one exception. Because the robot lacks sufficient dexterity to pick up and hand objects to the human, the robot was given a remote control panel that it could use to open either of two small metal boxes (one containing chips and the other cookies) near the Actor as shown in Figure 5-6.

The Actors wore a headband and gloves with a distinct pattern of markers so that the robot could distinguish between the different Actors as well as track their behavior (using the Vicon system described above). Distinct patterns of markers were also placed on each object used in the study. We developed customized tracking software to enable the robot to uniquely identify each rigid and near-rigid object (via their pattern of markers) to track their position and orientation. The robot must ascribe meaning to these trajectories: e.g., what food items are in which boxes over time, who is witness to which events, who is performing what actions, etc. The robot's perceptual and belief systems are responsible for constructing the robot's cognitive understanding of the scenario as it unfolds in real-time.

Table 5.2 displays the robot's behavior generated by our architecture on the various benchmark tasks in two conditions. In the first condition, the robot can offer the human a matching target object by operating its remote control box to reveal the correct item inside.

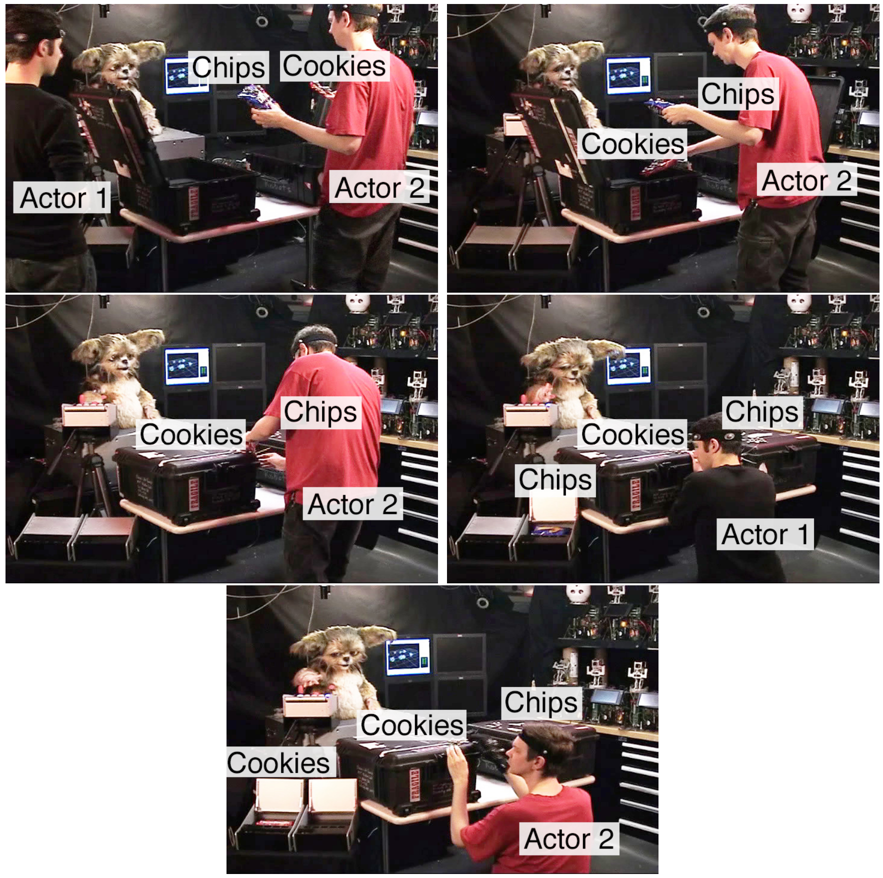


Figure 5-7: Setup of the human-robot study for Task 4a and 4b. The scenario proceeds from upper left image to bottom right. First, Actor 2 places chips in the left box and cookies in the right box for all to see. While Actor 1 is absent, Actor 2 switches the food items. Actor 1 returns looking for chips, but goes to the wrong box. The robot realizes the false belief and invalid plan of Actor 1, and gives him the chips he desires. Actor 2 (with true beliefs and a valid plan) returns looking for cookies in the same box, and the robot opens the small box revealing matching cookies.

Table 5.2: In the "Remote Control" condition the robot operates its remote control box interface to reveal the correct matching item for all tasks. In the "Deictic Gesture" condition, the robot can only help the Actor by pointing to the correct location of the needed item.

| Task | Remote Control | Deictic Gesture |
|---|---|---|
| Task 1 | open chips box (correct) | no action |
| Task 2 | open cookie box (correct) | points to target location |
| Task 3 | open chips box (correct) | points to key |
| Task 4a | open chips box (correct) | points to target location |
| Task 4b | open cookies box (correct) | no action |

In the second condition, the robot does not have its remote control box, so it cannot provide access to matching items. In this case, the robot can help the person by pointing to the location where the desired object really is, if the human is pursuing a failing strategy.

### 5.5.4 Summary

While the robot is not able to generate the full range of gestures and actions observed in our human study participants, the self-as-simulator cognitive architecture nevertheless allows the robot to produce helpful behaviors on a number of sophisticated collaborative tasks requiring goal inference in the presence of potentially divergent beliefs. Further, when given its remote control panel, the robot successfully opens the correct box to reveal the matching target item.

Our original objective of performing human subjects experiments was to gather data on the range of human behavior as they solved each task. We then wanted to compare the robot's performance on these tasks to human performance — fully expecting humans to be better. We were surprised at the number of people who did not perform the tasks correctly, and that people found some of these tasks to be difficult. These tasks are not as simple as one might initially think.

In light of our human performance data, it is interesting that our robot can successfully perform these tasks in both conditions.

# Chapter 6

# Modifying Mental States

Previous chapters describe techniques for re-using three parts of the robot's generative behavior architecture to characterize the behavior of an observed human. The robot's motor production graph is used as a common vocabulary to produce motions and to classify observe motions. The robot's belief maintenance system is re-used to model the world state as observed by a nearby human. Together these individual inferences allow the robot to characterize observed behavior in terms of its own goal directed actions.

While the robot uses its behavior generation mechanisms (and so, in a sense, is taking advantage of its embodiment) as a common language between its behavior and the human's, the robot's role is that of a passive observer, watching the human as an isolated actor and taking independent action only once it has completed a particular inference.

This chapter describes improvements made to the above techniques allowing the robot to move out of the role of an observer and instead become an active participant in theory of mind activities. Firstly, the robot's presence in an interaction with a human cannot be ignored. The robot must take into account not only first, but also second order mental states of the human because the human will have thoughts about the robot's knowledge, activities, and goals. Second, it is important to move beyond the present. While monitoring the mental states of a human *right now* is useful, it is also critical to make short term predictions. Finally, the robot needs to be able to modify the future mental states of the human. This can be thought of as a very low level type of communication - deciding how the robot should perform to cause the human to form a desired mental state.

## 6.1 Improvements made in realtime monitoring

This section describes improvements made to the system for modeling beliefs through perspective taking (Described in chapter 4). The changes here are to support the goal of interactive theory of mind behavior, which requires factoring in the robot's presence - including tracking a human's possible inferences about the robot's mental states.

### 6.1.1 Recursive Monitoring

This multi-level theory of mind inference (tracking mental states the human may have regarding the robot's mental states) is achieved through recursive belief modeling. Recursive belief modeling allows the robot not only to keep track of the human's beliefs about the world, but also the human's beliefs about the robot's beliefs about the world. This depth of modeling allows for much richer types of behavior. For example, in a two level recursive system, the robot could represent the goal of *demonstrating that it knows something* - a concept impossible to represent in the simpler, one level system.

The robot forms object beliefs based on its sensory input, attempting to track and identify the properties of objects in its environment. Whenever the robot encounters a human, an object belief is formed to keep track of that human. The robot's simulation system keeps track of object beliefs representing humans, and spawns a copy of the robot's software systems for each human known to the robot (Not a fully identical copy - certain systems are omitted, see section 6.1.3). The life span of these simulated agents is linked to the associated object belief - as long as the robot knows of the existence of another human it keeps modeling that human's mental states.

Because the model the robot maintains for each human includes copies of the robot's own belief modeling and simulation systems, the model also has the mechanisms to simulate any known agents. Therefore, when the model (which gets its sensory input from its creator's beliefs, see section 6.1.2) forms an object belief about another human (or about the robot), it will internally create a new model to keep track of that agent's mental states, just as the robot did, leading to recursive modeling where at each level the robot/simulated agent creates a copy the robot architecture for any known agents, and then those copies in turn can create copies to track agents from their own perspective. To prevent infinite recursion the agents keep track of how many hops they are from the original robot, and they stop recursing at a preset level (in the examples presented here, the robot goes only as far as simulating a human which is simulating the robot).

### 6.1.2 Re-Imagining from Beliefs

Chapter 4 describes a system for using perspective taking to simulate the beliefs of an observed agent by forking the sensory input to the robot, applying perspective transformations and filters, then passing that altered sensory information to a copy of the robot's belief maintenance systems. This worked for the scenarios described there, but it does not take full advantage of the robot's object persistence capabilities. For example, if an object is temporarily occluded from the robot's perspective, it still knows it is there through its object belief about the object, and despite it currently being missing from the sensory stream, the robot should be able to imagine that a human arriving at that time might well be able to see it depending on their geometric relationship to the occlusion (see figure 6-2).

Another motivation to switching to the re-imagining method is the switch to recursive modeling. While the simpler method works in many cases, any errors will compound down through the recursive structure of agents imagining agents. For this more complicated situation, it is useful to be taking full advantage of the belief maintenance capabilities of each agent in the chain.

The new system leverages the fact that the robot's beliefs consist of a history of percept

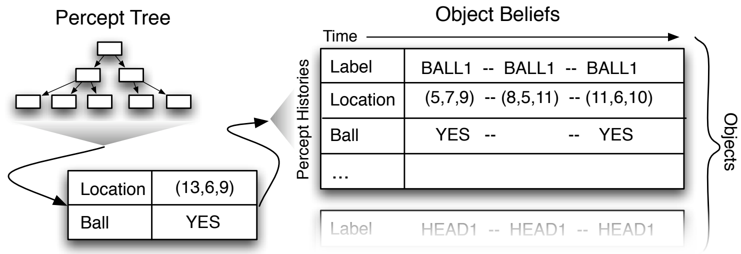


Figure 6-1: Object belief representation. The robot's belief about each object consists of a set of percept histories, where each history tracks the evaluations made by a single percept about sensor data related to that object. The belief system chooses which object to merge incoming percept evaluations into based on a distance metric. Percept histories can be used to look into an object's past, to filter noisy data, or to model object behavior in the short term future. For very simple data, a trivial history can be used that simply keeps the last data entered.

evaluations related to incoming sensor data (see figure 6-1). The robot, then, can create a list of current percept evaluations by looking at the most recent data in its object beliefs. These evaluations are the same format as the evaluations that come straight out of the percept tree, and can thus be passed to any simulated agents as incoming data, merging with the output of their own percept tree. Before any data is passed into the simulated agent, it is first filtered and transformed. Using its object beliefs to track occluding obstacles, the parent agent does a geometric line of sight calculation on each percept evaluation before passing it through to the simulated agent's system, as well as transforming the data to be from that agent's perspective. Additionally, only certain percepts that are marked as *visibly transferred* are candidates for this direct transfer process. These percepts represent a property that could be directly observable, such as location or color, and exclude more complicated evaluations that the robot has made about an object (such evaluations will be made instead by the simulated agent, if indeed they have enough information to do so).

For the simulated agents, the only sensory information they process directly is from their proprioceptive sensors (keeping track of the location of key body parts). The proprioceptive data goes through their percept tree, then is merged with the copied and transformed percept evaluations coming from their controlling agent (see figure 6-3). This process can continue recursively, with each simulated agent getting updated based on information known to the agent conducting the simulation.

### 6.1.3 Minimal Robots

As described above, each simulated robot is a copy of the mechanisms used to run the physical robot. However, some simplifications are made to streamline the process. The simulated robots do not have a full kinematic model. Instead, they have only a body, head and hands, lowering the computational cost to maintaining each agent. As described above, they also forgo most of the sensors, keeping only a proprioceptive sensor (which keeps track of their virtual body parts); they get most of their data directly from their parent's object beliefs. This means there is also less load on their percept tree.

The simulated agents retain the actions and action system of the physical robot, however the actions have been disabled so they cannot activate in the normal, self-motivated manner. It is important to disable these actions because it prevents the simulated robot from initiating an action based on its own motivations, when the real agent that is being simulated may not even be performing that action. Instead, the actions should be activated when the real agent is observed to be performing that action (the exception here is when making future predictions, section 6.2).

The critical part of the simulated agent's action system is the action contexts, which in standard behavior govern when an action is allowed to be performed. While the actions are disabled for these agent models, the context evaluations can still be accessed and it can thus be determined when a simulated agent is in a situation where they *could* perform that action.

## 6.2 Making predictions about future mental states

This section describes the techniques the robot uses to leverage the above recursive mental state calculations to make mental state predictions into the short term future. This capa-

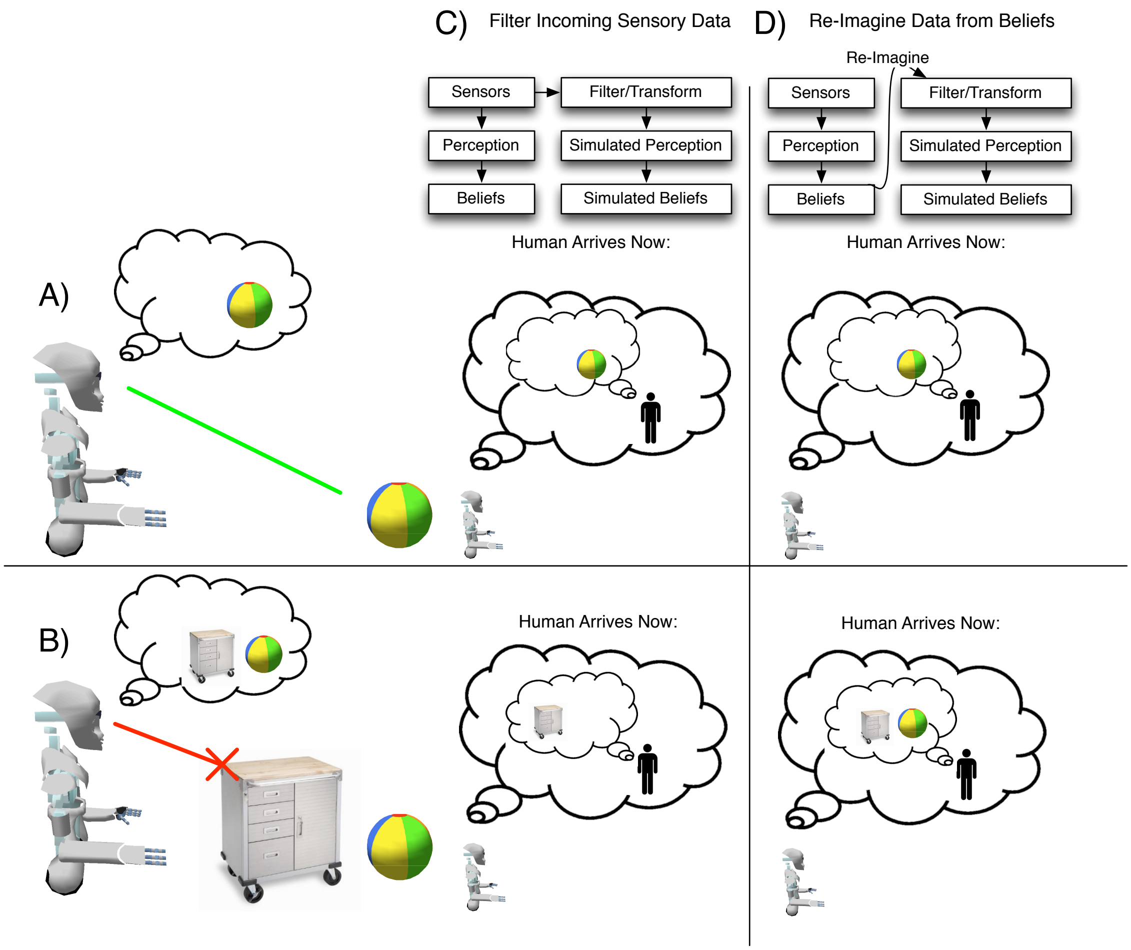


Figure 6-2: Motivating scenario for re-imagining sensor data. In row *(A)*, The robot has a clear line of sight to the beach ball. In row *B*, the robot has seen the beach ball and remembers where it is, but the cabinet has been rolled in the way and robot can not currently see it. Column *C* shows how the earlier system, which filters incoming sensory information, would respond to the two situations. Column *D* shows the performance of the system described in this chapter which re-imagines sensory data from the beliefs of the robot instead of providing a filtered version of raw incoming sensory data. As we can see, the simpler system performs correctly when the robot has clear line of sight (it would also perform correctly in situations where both robot and human have an occluded view, or only the human has an occluded view). However, it does not perform correctly when only the robot has an occluded view.

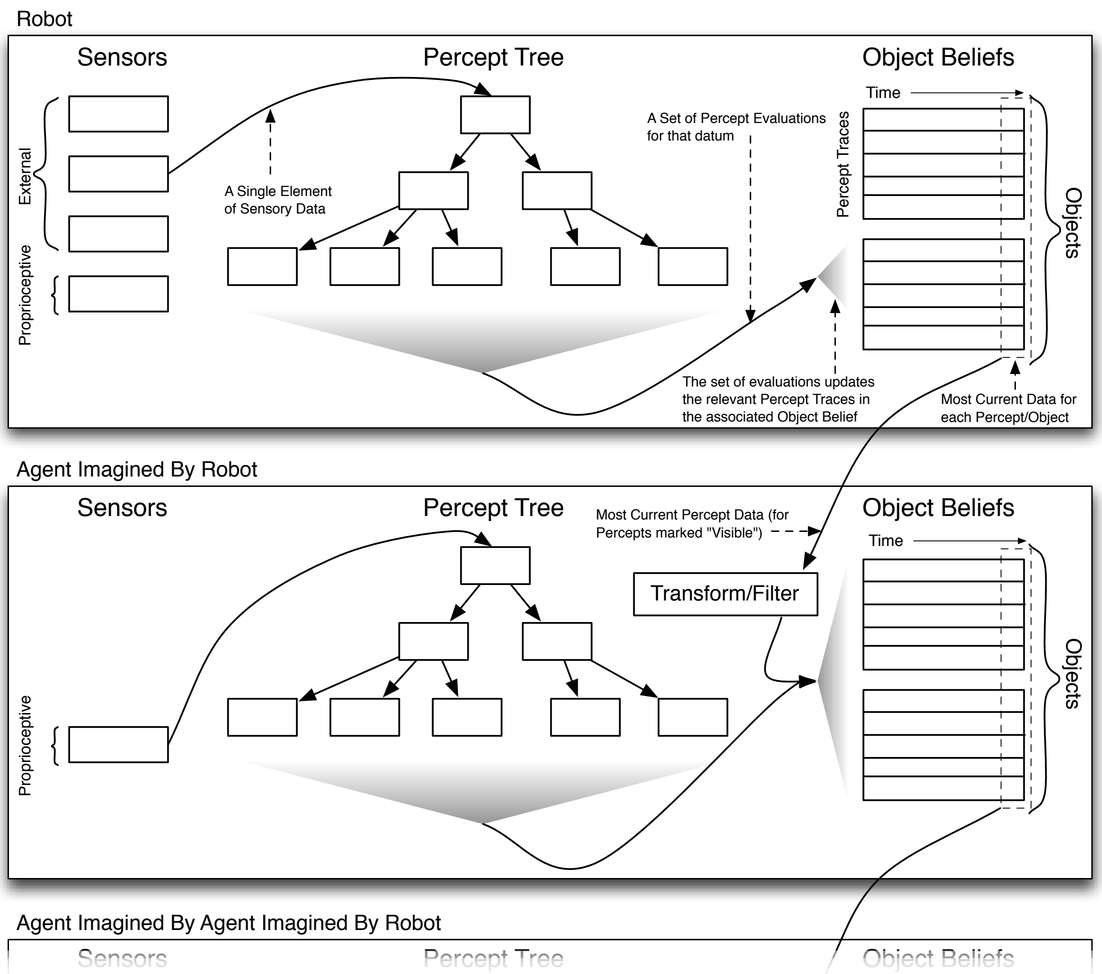


Figure 6-3: Recursive Sensory Data Propagation. In the robot itself, we see that the data follows the normal progression from the sensors, through the perception system, then is used to update the object beliefs. The simulated robots use their percept tree only for proprioceptive data - the bulk of their data comes from the most recent percept evaluations extracted from the controller's object beliefs. These are filtered for visibility and transformed, then merged in with the rest of the incoming percept evaluations that will used to update that simulated agent's object beliefs.

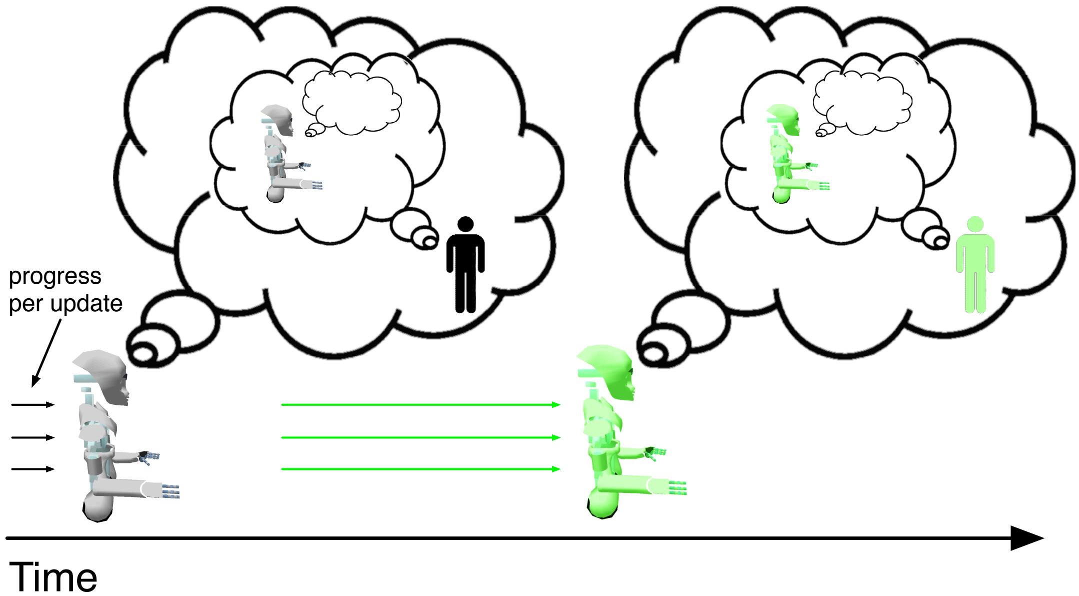


Figure 6-4: A *hypothetical* copy of the robot is used for mental state predictions. The robot has the capability to model mental states of agents around it (left). To make short term predictions, a copy of the robot (green, right) starts from the robot's current state and performs (in a virtual space) the actions the robot is about to perform, but performs them much faster. While doing this, it maintains the mental states of the surrounding agents as they participate in this accelerated timeline. This gives the robot the ability to predict the mental states of surrounding agents in the short term future.

bility will then be expanded upon in the next section to not only make predictions, but to also choose between several courses of action to find one that will bring about a particular mental state goal, eg. "cause the human to know X".

The robot utilizes the previously described theory of mind capabilities to keep track of a recursive structure of simulated agents and mental states, but here it utilizes this capability from a *hypothetical* version of the robot. This robot has the same systems as the actual robot, but is entirely virtual, and is allowed to move and complete its actions much faster than the actual robot. This hypothetical robot, then, can move ahead in time, performing the current robot's actions before the robot does. It can then monitor the effect this has on the observing simulated agents' mental states (see figure 6-4).

Because this hypothetical robot includes the full kinematic model of the robot and moves through its actions using the same processes as the real robot, these hypothetical predictions are as highly detailed as the the real robot's modeling will be. The robot can model who will see what, even taking into account the occlusions caused by its own body.

### 6.2.1 Differences From Realtime Robot

The hypothetical version of the robot includes all the software mechanisms in the real robot, including a full kinematic model that describes its body and motions. However, it exists only in the virtual world, and runs at a different time scale than the actual robot. Because of this, it must be disconnected from the physical world. It's joint positions are not transmitted to the physical robot. Also, the physical sensors that normally supply data to the inputs of the robot's systems are not connected to the inputs of the hypothetical robot - this data comes in in realtime, and would thus conflict with the robot's virtual future model of the world. Finally it is provided with a fake navigation system - the real robot travels around the space and localizes itself using sensory feedback, so instead of simulating those sensors the hypothetical robot is provided with a trivial navigation system that uses its exact knowledge of its virtual location.

To allow it to advance faster than the physical robot, the hypothetical robot's progress through motor actions is increased, so the joints move faster than on the physical robot and motor actions complete more quickly. The robot in general operates via an update loop which updates all of the robot's systems 30 times per second, advancing a virtual clock by 1/30th of a second each time. The hypothetical robot is provided with a different clock source, which also internally appears to advance 1/30th of a second per update - however, the hypothetical robot is actually getting updated much more frequently (approximately 30 times per update of the main system). In this way, the hypothetical robot advances from its virtual perspective at the normal 30hz, but actually its clock is running far faster than realtime. This allows it to start from the current state of the robot and move forward in time faster than realtime, checking on what will happen in the future.

### 6.2.2 Connecting Hypothetical Robot to Real robot

An important consideration is how to provide data to the hypothetical robot. Providing continuous access to sensors is not correct, because as the hypothetical robot moves forward in time the sensor data will start to be out of date. Additionally, we want to ensure that the hypothetical robot starts with the same knowledge of the world as the realtime robot, so its decisions and actions will most closely reflect what the physical robot would do in the upcoming situations. Since we are particularly concerned with tracking mental states in this hypothetical future, this knowledge must also include the information the physical robot has which is represented as the recursive structure of mental states of surrounding agents.

For these reasons, the hypothetical robot gets its information in two distinct ways (figure 6-5). First, as a hypothetical run is initiated, the hypothetical robot is "reset" to the current state of the physical robot. Then, as the hypothetical run begins to advance, this stream of information is cut off and the hypothetical robot gets no more live information. Instead, it relies on its belief maintenance mechanisms to preserve the information it has, and this information is propagated as appropriate through its recursive structure of simulated agents. New information comes from the proprioceptive sensors which operate based on the virtual body of the robot, as well as from updates to the robot's object beliefs that come about as a result of the actions it is performing (see section 6.2.3).

After the reset, and during the normal operation of the hypothetical robot, the mental state modeling mechanisms operate identically to those in the realtime robot (they are the same mechanisms). The only difference is that they are missing the constant stream of new sensor data; instead updates about world state come only from virtual proprioceptive sensing or from changes to the object beliefs that occur as the result of actions performed.

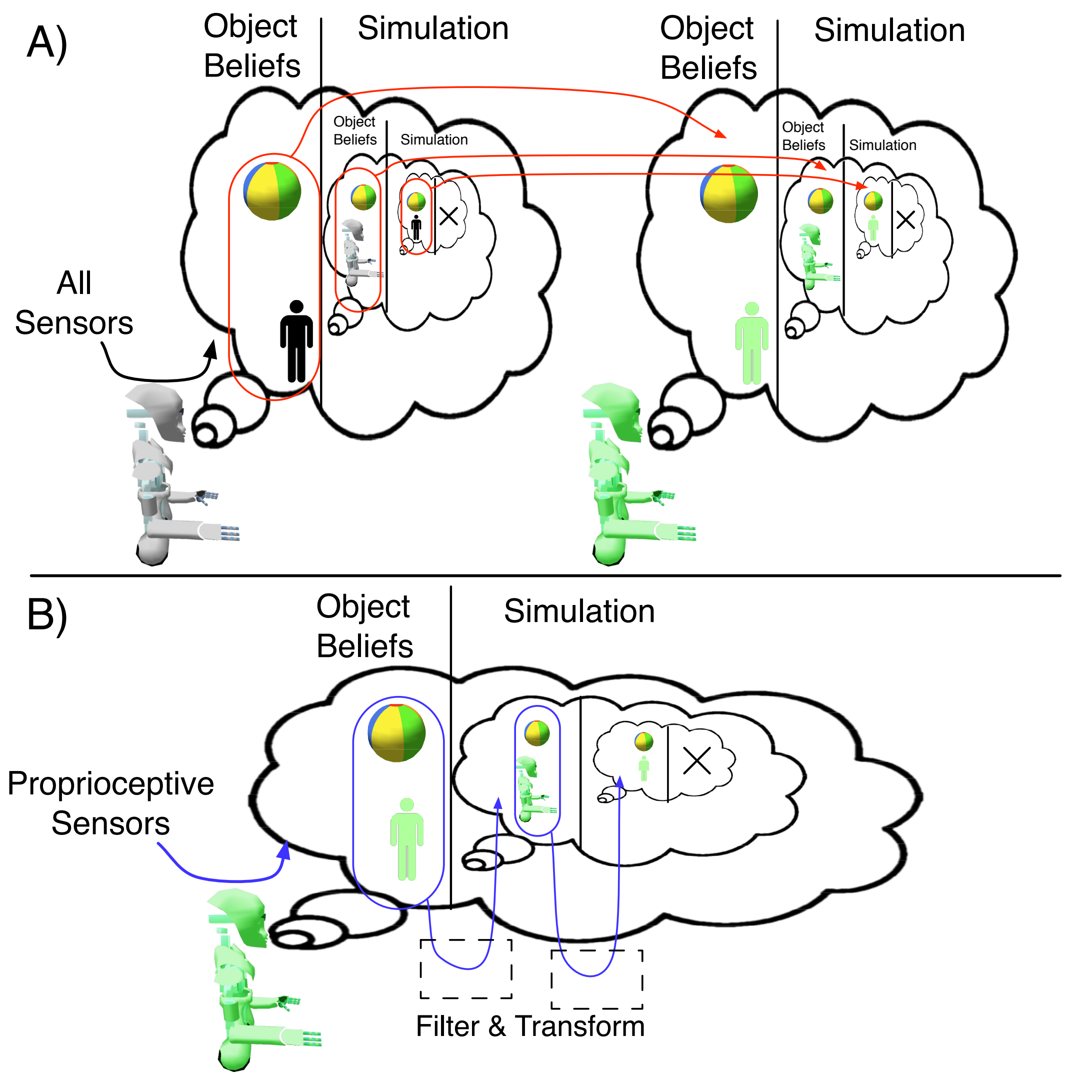


Figure 6-5: Dataflow During reset and operation of hypothetical robot. Part *A* shows the real robot on the right, and hypothetical robot on the left. The red arrows represent object beliefs being copied during the reset of the hypothetical robot. *B* shows the operation of the hypothetical robot. Blue arrows show how the data about objects propagates in this configuration (Except for the cutoff from the sensors, this is identical propagation as in the real robot)

### 6.2.3 Updating Object Beliefs

Section 6.2.2 describes why the hypothetical robot is cut off from live sensor data. Instead, the hypothetical robot only has access to its virtual proprioceptive sensors, since they are operating in its same time scale. As the robot performs an action, however, properties about the world are often expected to change as a result. For the real robot, this occurs because it actually modifies the world state, which is then reflected through its sensory input. Without sensor data or even the ability to change properties of real world objects, object beliefs of the hypothetical robot do not get updated reflecting the results of the action. This will cause it to consider that the action failed, and/or be unable to perform followup actions that depend on that change. At this point, the hypothetical robot will start making different choices than the physical robot would, because the physical robot with its live sensory streams will not be subject to this sensory blackout.

With this limitation, the prediction time of the hypothetical robot would be limited by the length of time before the it performed an action that would make an important, non-proprioceptive change to the world (ie, changing properties about an object other than itself). Since the demonstration scenarios include largely object manipulation actions, this would mean the robot could usually look ahead no more than the length of the current action.

#### Expectation

A simple expectation mechanism helps alleviate this issue, as well as helping with sensing problems. For manipulation motor actions, the robot's object belief maintenance systems incorporate a simple mechanism which encodes the expectation that once an object is grasped, it will be in the robot's hand until it is released. On the physical robot, this helps with visual sensing problems, since it is hard to track the object when the robot is holding it. However, it also helps the hypothetical robot look forward further in the future. With a lack of live sensory information, the object belief will retain the location the robot last set it to: the position of the robot's hand when it released the object (See figure 6-6). This means that

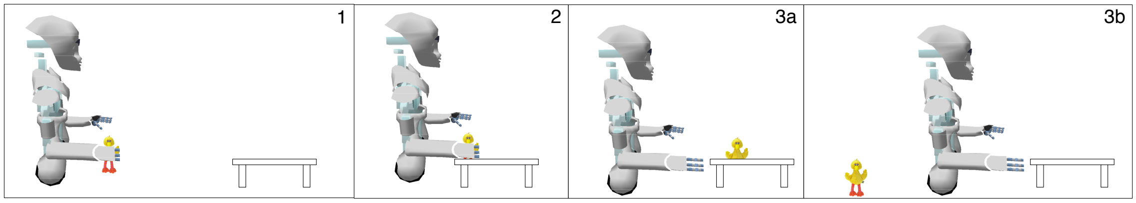


Figure 6-6: When the physical robot successfully carries an object, the result is frame *3a*, and the robot can model this result because it's sensors will detect the toy on the table. When the hypothetical robot carries a virtual object, the physical sensor feed will not be affected. In both the case of the physical and hypothetical robots, the belief maintenance mechanisms assume that an object, while grasped, stays in the hand, and thus they update the toy's location during the carry action. This allows both real and hypothetical robots to experience frame *2*, even though the hypothetical robot is not moving a real object, and the physical robot is unlikely to be able to track the toy visually during this process. It is important to cut off the physical sensor feed from reaching the hypothetical robot so the object belief of the toy will stay put once released (frame *3a*) and not snap back to it's real-world location (frame *3b*)

manipulation actions will update the robot's beliefs about the objects being manipulated, both in the real world (through expectation during the action, followed by sensor feedback afterward) and in the hypothetical world (through expectation during the action, followed by the persistence of that change). Algorithm 3 shows an example mechanism for the grasp actions. This technique allows the robot to predict how objects will move using the virtual world - however, a robust physics simulation in the virtual world could eliminate the need for this technique.

**Algorithm 3** Maintaining Object Property Expectations - Grasping Example

**UpdateHolding(ObjectBelief ob):**
**if** (ob.get(Type)≠HAND) OR (ob.get(Closed)=FALSE) **then**
    ob.set(Holding, NULL)
    **return** {Not a hand or not closed - release any held object}
**if** ob.get(Holding)≠NULL **then**
    ob.get(Holding).set(Position, ob.get(Position))
    **return** {Already holding and hand still closed, just update position of held object}
**if** ob.get_last(Closed)=FALSE **then**
    {Hand just closed - it was recently open}
    **for all** $o$ ∈Object Beliefs **do**
        **if** ($|o$.get(Position)−ob.get(Position)$|$)<Grasp_Threshold **then**
            ob.set(Holding, o) {During hand closing, close objects become held}

### Generative Goals

Another mechanism to allow object beliefs to progress properly for the hypothetical robot is based on a change in the way action context conditions and goal conditions are handled. In previous work, an action became eligible for activation based on the action's *context*. The *context* was represented as a set of conditions that must be true in the world, as represented by the robot's object beliefs. For a goal directed action where the robot could monitor the success of the action, the goal condition would be similarly represented.

These conditions are quite general, in that they could represent any arbitrary function that could be applied to the robot's object beliefs. However, because of that, they cannot be reversed to ask the question, "what would the object beliefs be like if this condition were true?" The mechanism described in this chapter is an attempt to model preconditions and goals in that way - to have the robot "visualize" what the world would be like if its action completed successfully. This switch is useful for looking forward in time - if we can visualize the world after an action, the hypothetical robot can skip straight to that world state, without worrying that the rules and physics of the virtual world the hypothetical robot lives in will fully model the world's reaction to the robot's movements.

To this end, we switch to a different mechanism to represent the conditions necessary to implement an action's *context* or *goal*. Instead of applying an evaluation function to the object beliefs which will determine if they satisfy some condition, we instead switch to a generative strategy. This generative strategy means that we generate the set of beliefs that indicates the successful completion of the action, then check those generated object beliefs against the current set and see if there exist matching beliefs.

As described in previous sections, an object belief is a structure intended to organize information known about a particular physical object. This data takes the form of percept evaluations, which occur when a particular percept makes an evaluation on a unit of incoming sensor data. The object belief, then is a collection of these percept evaluations on incoming data, grouped so that all the evaluations in a particular object belief relate to the same real world object. The generative condition system, then, must generate a hypothetical object belief, or set of beliefs, that describe a particular desired world state.

These object beliefs may be very detailed, or very sparse - the generator need only specify the properties of the objects that are necessary. In this way the generator is describing a class of world states - any world state can match the generated beliefs, as long as it contains object beliefs that can be matched to the generated object beliefs.

The generators need not be fixed, they can be parameterized based on the conditions under which they are used. Commonly, actions take some kind of parameter. For example, a grasping action operates on a particular object. The identity of this target object, then, will be an important aspect to include in the generated object beliefs for the robot. For the robot to begin the action, the robot will "visualize" a situation where the robot is close to the object, there is nothing it its hand, etc. And, at the end of the action, the robot can "visualize" a set of object beliefs that describe the robot with its hand closed, and with the specific object inside it. If that visualization matches the world when the action is done, the action succeeded. See figure 6-7.

These generative beliefs can serve as a shortcut when trying to maintain correct object beliefs while supporting the chaining of multiple hypothetical actions in the virtual world used by the hypothetical robot. In this way, instead of having the hypothetical robot run through an entire action to its conclusion, the robot can simply determine if the context is valid, then immediately generate the goal condition beliefs and merge their data into its current object beliefs.

## 6.3 Changing the future

The previous section describes a mechanism for operating a hypothetical copy of the robot faster than realtime in order to make predictions about how the robot's actions will affect the mental states of the agents around it. In this section, a simple planning mechanism is used to explore multiple possible futures. By trying out different possible action combinations,

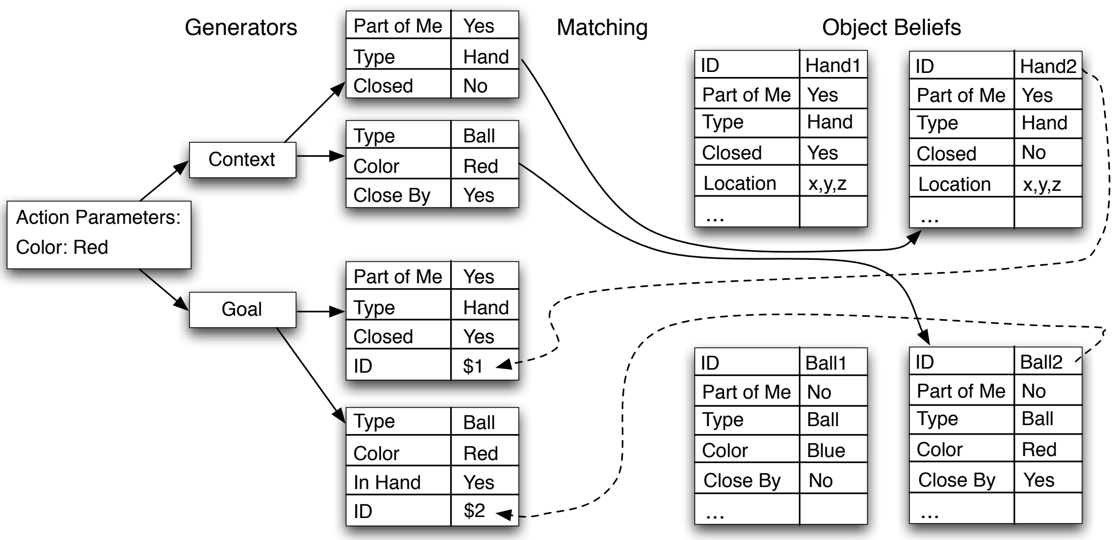


Figure 6-7: Generative Context/Goal Conditions. One technique to maintain correct object beliefs while supporting chaining of multiple hypothetical actions in the virtual world is the use of generative context and goal conditions (as opposed to using evaluation functions to detect these conditions.) Here is an example of the operation of generative conditions for an example grasping action. The context generator creates a set of beliefs whose presence is required for the action to fire. If the matcher can match those against the existing object beliefs, the action is eligible for activation. Often the goal generator depends on the outcome of the matching in the context generation phase - here we see that the goal generator doesn't just predict a goal state where *an* object is in the robot's hand - it requires *the same* object that the action originally activated with to be in the robot's hand.

the robot can explore different futures and determine which actions lead to a future where the human's mental states are configured as the robot desires.

Though the structure of the planning is quite simple, the domain in which this planning occurs is quite rich because it includes the embodied, geometrically correct theory of mind abilities of the robot. These plans, therefore, take into account not only the direct effects of the robot's actions, but also the mental state changes these actions cause, and even the possible behaviors of the human in response to these changes.

### 6.3.1 Search the Future

In order determine a course of action that produces the desired mental states in a nearby human, the robot attempts many different action sequences using the hypothetical robot running in its virtual simulator. The planning described here is structurally quite simple - an exhaustive search of possible action sequences within a certain time window. This strategy can and should be expanded upon to increase performance in agents with many actions - however, it serves to demonstrate the power of the domain in which the search is taking place. This domain allows futures being examined to include both the effect of the actions taking place on the recursive mental states of the agents being modeled, as well as the effect of the changes in mental states on the possible actions of the agents present.

The search takes place by running many trials of the hypothetical robot operating in the virtual world. However, instead of allowing it to take the highest valued allowable action at every opportunity, a new action selection policy is put into place that forces the robot to perform a time bounded depth first search of the action space. In between each trial, the hypothetical robot is reset back to the state of the realtime robot. Because the actions take an unknown amount of time and the state of the world after the action is not explicitly known, the actions and targets available in the future are not initially known. Thus the search tree must be built up as the robot explores; the robot can determine which actions and targets are possible after a given initial sequence by examining the situation once it arrives there.

At any particular point in time, the context mechanisms for the actions will indicated which actions are possible to perform. When a search begins, the robot notes which actions are possible and those become the children of the root node of the action tree. Whenever a node in the tree is visited for the first time, the robot must check which actions are relevant in this new context to create the children for that node. Different actions may well be available to the child than to the parent, because the object beliefs of the hypothetical robot may have changed based on the actions it performs (figure 6-8).

For the purposes of the recursive mental state modeling, this search process is the same as the normal operation of the hypothetical robot. The simulated humans are watching the world and robot through the object beliefs, keeping track of their mental states as before. When the time bound expires and the next trial begins the mental states of the simulated humans are returned back to the original state they received from the realtime robot's model of that human.

Additionally, just as before, the humans are themselves free to act. As the robot changes the world, the changing world state may change the behavior of one of the simulated humans, and this will be factored in to the robot's conclusions about that action sequence.

### 6.3.2 Representation of Mental-State-Goals

The robot has some set of scenario specific task goals. When the robot is searching through the potential results of future sequences of actions, it can find sequences that allow it to achieve its main task goals. Because of the mental state modeling during this process, the sequence the robot finds to be successful may well take into account and even depend on causing mental state changes in the humans around the robot, thereby altering their behavior in a way that works with the chosen actions and the robot's task goal.

In these cases the human's specific mental states need not be directly accessed by the robot - instead, the robot need only track its progress towards its task goals. The mental states being maintained about the humans are still important to this process, however their role

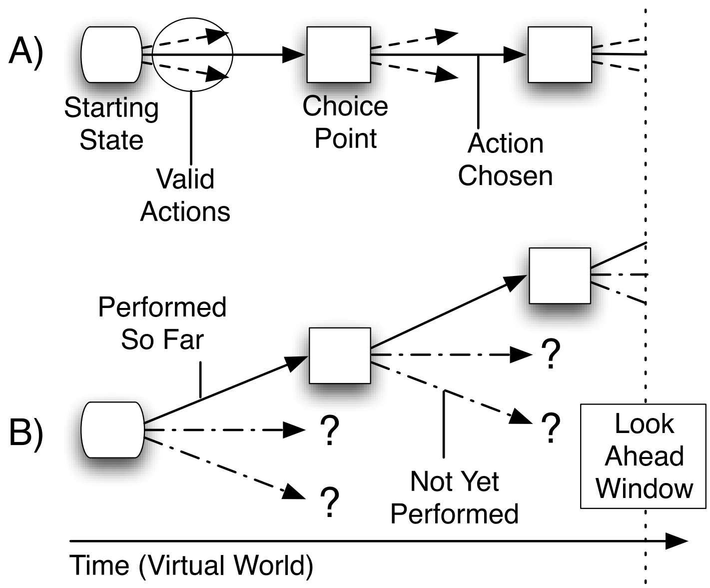


Figure 6-8: *A)*, represents the path through action space taken by the hypothetical robot in section 6.2. In this case, the hypothetical robot greedily chooses the highest value action out of the actions available at each choice point. This path is the same that the physical robot would follow next without any more specific input, so can be used to predict mental states that occur along that trajectory. *B)* represents the search that occurs in section 6.3.1. The normal greedy selection is blocked; instead whenever a new choice point is reached the robot notes all the available actions from that point and enumerates through them, choosing a new one each time it reaches that point. The robot continues to run hypothetical trials until every path through the tree is traversed (paths are bounded by the time window). In this way, the robot can simulate various possible action sequences, modeling the mental states of the surrounding agents as it does so.

is to affect the actions that the simulated human will take in the virtual world simulation being traversed by the hypothetical robot. The mental states of the humans will inform their action selection, and their actions will impact the virtual world in a way that the hypothetical robot will note. In this case the mental states act as a hidden variable in the planning process. However, there are also times when it is beneficial to examine these states directly.

There are multiple reasons why the robot might want to examine the human mental states directly. This system is by its nature designed for high resolution but short term planning. If these short term plans are taking place in the context of a lower resolution but long term plan, that outer planning mechanism may have restrictions about what the humans should or shouldn't know that are important to the long term plan but wouldn't be discovered in the short term look-ahead. Another reason is that certain mental state configurations might be specified as requirements of the scenario the robot is engaged in by whatever higher level process or external programmer configured the goals of the robot. In this case, instead of the mental states of the humans being used exclusively to help the robot better predict the future and achieve its task goals, the configuration of the human mental states is in itself one of the robot's main goals.

For these reasons, the robot uses *mental state goals* to represent a goal that describes some configuration of mental states in the recursive mental state modeling structure. The *mental state goals* consist of two parts. The first part describes a particular configuration of object beliefs. This part operates in the same way as the action's goals and contexts from section 6.2.3, by generating one or more partial object beliefs. These object beliefs can be matched against an agent's existing object beliefs to determine if any of the existing object beliefs match the generated world description. Two different types of *mental state goals* are defined - those that succeed if the described conditions is met, and those for whom that means failure.

The second part of the *mental state goal* defines the relationship between the agent that has the goal and the agent in which the desired mental state configuration should occur.

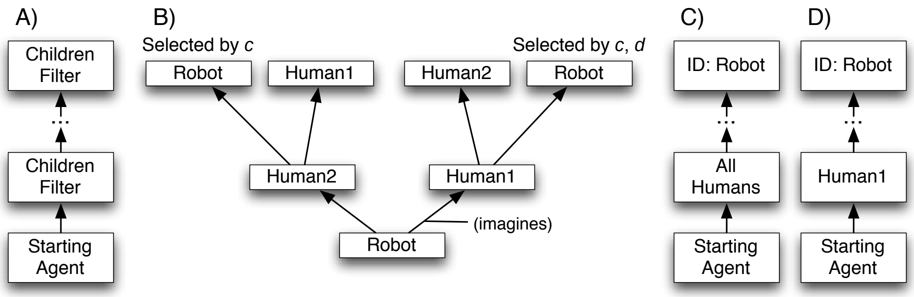


Figure 6-9: *A)* shows the structure of an *agent specifier* used to indicate which recursively modeled agent a particular mental state configuration goal refers to. *B)* is an example structure, created by a robot who is modeling mental states of agents Human1 and Human2. *D)* is an example *agent specifier* that specifies, combined with a target mental state X, "the robot knows that Human1 knows that the robot knows X". *C)* is an example which includes multiple agents, and could be stated as "the robot knows that the humans know that the robot knows X". More technically, if this is a positive goal (true when the target agents have the specified mental configuration) this goal will be true if every model of known humans contains a model of the robot's mental states, and that model contains configuration X. If this is a negative goal (true when the target agents do not have the specified mental configuration), it will be true when every known human's model of the robot's mental states do not contain X.

This is a relative relationship, to allow the goal to mean the same thing to various agents in the recursive modeling structure. For example, a goal might be "don't let Human1 know that I know X". "I know X" is represented by a belief generator, as described above. The relationship from the agent with the goal to the agent whose mental states the goal concerns is described by an *agent specifier*. The *agent specifier* is designed to navigate from the current agent to a particular agent in the recursive structure of agents that the original agent is imagining. In this case it looks to see if the current agent is currently modeling mental states for Human1. If so, it checks to see if that simulation of Human1 is currently modeling mental states of the original agent. If that is also happening, it can go ahead and check the generated belief against the original agent's model of Human1's model of the original agent. These relationship can also include wildcards - see figure 6-9.

### 6.3.3 Mapping between Real/Hypothetical robot for planning, goals

The *mental state goals* are most frequently evaluated using the hypothetical robot, as it searches the possible consequences of its actions. However, these same goals should be applicable to the actual robot, where the real actions will be performed. Similarly, the search through action space takes place within the hypothetical robot, but the chosen sequence will eventually play out on the real robot. For these and other implementation reasons, it is useful to have mechanisms be applicable equally well to a number of different agents all existing at once in the system.

The *Context Tree* is a mechanism used throughout the c6 architecture as a mechanism for different modules to communicate. It is a hierarchical, runtime-scoped blackboard, which can store and return information for arbitrary keys. Writing to this tree will write to the data to the current location in the hierarchy, and reading will return information from the first located use of that key, where the search begins in the current position in the hierarchy and traverses up towards the tree's root. Therefore, children inherit non-overriden values from their parents. The "current location" is managed through runtime scoping, meaning that the place in the execution flow that a call is made determines its current context. Normally this means that an agent has a top level context, each of its major subsystems exist in subcontexts, and systems within those systems exist in deeper subcontexts.

In this work, the robot is "imagining" other agents through re-use of its own mechanisms, including much of its software mechanisms. Because the imagined humans exist entirely within the robot's modeling, they create their contexts within the robot's own context. Those agents themselves are also imagining other agents, so they, in term, instantiate agents which will exist within their own context. This process happens another time, in parallel, for the hypothetical robot (see figure 6-10).

The context tree provides a useful structure which organizes these agents into a tree, and allows systems to navigate between agents as simple tree operations. Additionally, software can be written that, instead of relying on pre-specified agent properties, gets the necessary agent properties from the current context. This means that that software mechanism can

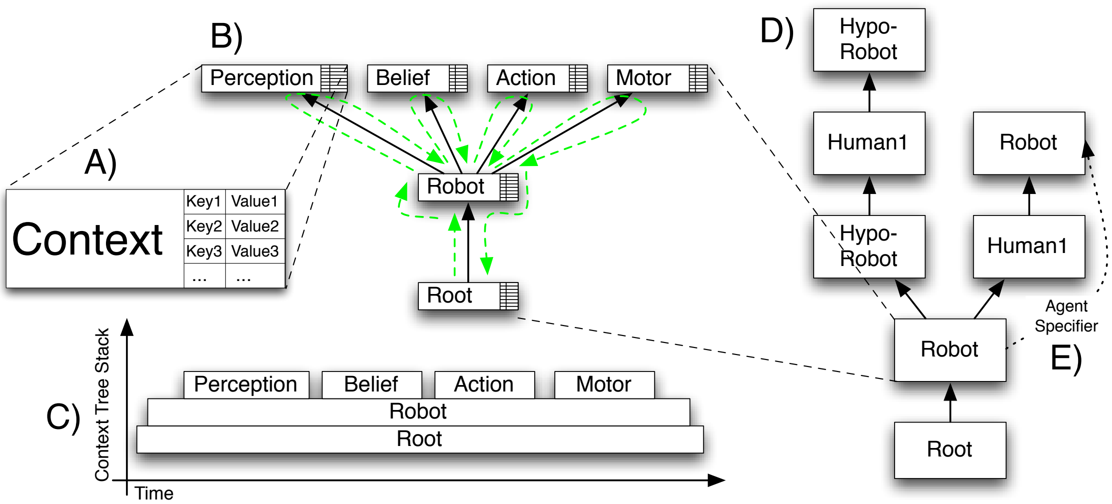


Figure 6-10: Breakdown of context tree use. The context tree provides a hierarchical runtime scoped blackboard that is used to communicate between modules and keep track of values and agents. Each context contains a simple blackboard that maps from keys to values $(A)$. A robot normally uses this mechanism to keep track of parameters between its various subsystems $(B)$. The green arrows indicate how the flow of program execution passes between the system as the robot updates itself at each execution step. The robot itself is allowed to update, and as part of its update it updates each of its subsystems. $(C)$ shows this from another angle, showing how the active (top) context tree context changes over time, with the parent contexts pictured below providing default values whenever a specific requested value is not present in a child. $(D)$ shows how this tree is extended once we start tracking a nearby human to model their mental states (each of the agent boxes in $(D)$ also contain the structure shown in $(B)$ but it is omitted for clarity). $(E)$ shows an example of a system making use of this structure. The *agent specifier* is coded to know how to navigate from a given current position in the context tree to one or more target positions that define the specified agents. In this way the same specifier can be applied to the Robot, or to the Hypo-Robot, and yield the expected, relative results.

easily be applied/transferred to different agents, by simply executing it during the desired agent's update time.

The *mental state goals* and the planning mechanisms are examples of this technique. The *agent specifier* of a particular mental state goal specifies an agent by starting at the "current" agent, then navigating down through the children, selecting children that match the first filter. The next filter can be applied to the children of those matching children, etc., until the specified agents are located. Similarly the action exploration system is applied to the agent in which it is executed, thus it can be run in the planning mode inside the hypothetical robot, and then, once a desired sequence is found, in execution mode in the context of the real robot.

The hierarchical feature also eases the implementation of these recursive structures of agents. Any values that are not explicitly overridden are inherited from the parent. This means that each agent can, by default, share the values of its parent, but also be provided different values where appropriate. A good example is the "world time". This value tracks time within the system, used by many subsystems within an agent. This value can be set at the root, and is rarely overriden because the same time normally applies to each subsystem. However, the hypothetical robot runs faster than realtime, and so the time key gets overridden with a faster changing value in its context. This automatically affects the agents imagined by the hypothetical robot, because they exist in subcontexts within the hypothetical robot's context.

## 6.4 Evaluation

In order to demonstrate these mechanisms, a scenario is designed which allows the robot to display different behaviors based on changing only its mental state goals. These different behaviors are recorded, and shown to subjects. The reaction of the subjects shows that the manipulation of their mental states was successful. Moreover, when presented with evidence showing how the manipulation occurred, they recognize it as such.

### 6.4.1 Demonstration Scenario

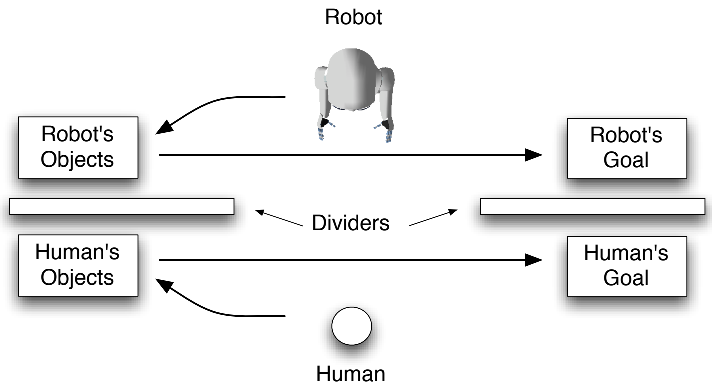


Figure 6-11: Top down view of the demonstration scenario, a competitive game between the human and the robot. The robot stays on the upper part of the diagram pictured, and the human on the lower part. Each player has access to a matching set of objects on the left side, and each has their own goal area on the right side. Occlusions block the view of each player from the opposing player's object and goal areas, however they can see each other as they travel between the object repository and goal.

The demonstration revolves around a simple competitive game played between the human and the robot. The layout of the game area is pictured in figure 6-11. The object of the game is to win by placing the correct object into the goal area. The object choice depends on the choice of the opponent. The robot wins if the two placed objects are different. The human wins if the objects are the same.

At the beginning of the game, both players have access to the same set of objects, and face each other across a symmetric game area. The game ends when both players have placed an object in their goal area, and at that point winner can be determined.

The rules of this game create a situation where the player who goes second has the advantage of potentially seeing the item played by their opponent. If the human goes second and sees the item played by the robot, it is straightforward for them to win by playing the object they saw the robot play.

For this demonstration, the robot takes its turn first. It is thus to the robot's advantage to manage the information that can be observed from its behavior. If it proceeds in a straightforward manner, the human will be able to watch and observe the object the robot plays, then play the same object and win. To win, the robot must instead hide this information from the human.

| Condition | Robot's Goals | Robot's Behavior |
|---|---|---|
| 1 | •Cylinder in goal<br>•Human doesn't see me carry cylinder<br>•Human sees me carry football | •Transports cylinder behind back<br>•Carries decoy football |
| 2 | •Cylinder in goal<br>•Human doesn't see me carry cylinder | •Transports cylinder behind back |
| 3 | •Cylinder in goal | •Transports cylinder openly |

Figure 6-12: Set of robot's goals and resulting behavior for each of three demonstration conditions.

For the demonstration, the game was played three different times, each time with a different set of mental state goals for the robot (see table 6-12). These different mental state goals change the behavior of the robot as it plays the game. In each case, the robot has the same overall task goal - transport the cylinder to the goal location. However, the way it accomplishes this task varies in the three conditions based on the mental state goals.

In the first case, the robot has two mental state goals. While transporting the cylinder to the goal, it should attempt to prevent the human from seeing that it is transporting the cylinder. Additionally, it has the goal of having the human see that the robot is carrying the football. The robot finds an action sequence that results in carrying the cylinder hidden behind its back, with the football carried out in the open. In this way it may fool the human into thinking that the robot is playing the football, causing the human to lose by playing the football in response. In the second case, the robot has only the mental state goal to keep the human from seeing that the robot is transporting the cylinder. In this case, it tries a number of strategies until finding the action sequence that has it carrying the cylinder with its left hand, hidden behind its back from the human. The human can't see what the robot played, so is likely to choose randomly and win half the time. In the final case, the robot has no mental state goals, its only goal is to transport the cylinder. It simply carries the cylinder over to the goal. This would likely cause the robot to lose the game, as the human can easily see the object the robot is carrying. Figure 6-13 shows the robot's behaviors as it plays the games under these three different sets of mental state goals.

In order to find these action sequences, the robot is using the action planning mechanisms described previously in this chapter. Figure 6-14 shows stills from a visualizer of these mental state planning mechanisms during the planning session for condition 1. This visualizer overlays the hypothetical simulation the robot is performing over a video image of the actual environment the robot is operating in. The overlay shows how the planning mechanism creates a geometrically correct prediction of the mental states being formed by performing a full kinematic simulation of the robot performing the actions, situated in the environment in which the actions will be performed.

### 6.4.2 Human Subjects Study Design

In order to evaluate the performance of the robot in these demonstration scenarios, a human subjects study was performed. This study was designed to evaluate if the robot's behavior could successfully manipulate the mental states of the subjects, as measured through their choices in the game. It was also used to measure if these behaviors had any effect on the subjects' perception of the robot's competencies and their evaluation of the robot as a potential partner.

Subjects participated in the study online, by accessing a website. The subjects were broken into three different groups. All subjects were instructed that they would be playing a simulated game with the robot. After the game and rules were described, they were shown a video of the robot performing its turn. This video was recorded from the perspective of the human player, with the robot programmed to treat the camera as if it were the opposing player (thus any actions which would hide an object from the competing human would hide that object from the camera).

Each of the three groups were shown different videos; each group saw a video of the robot

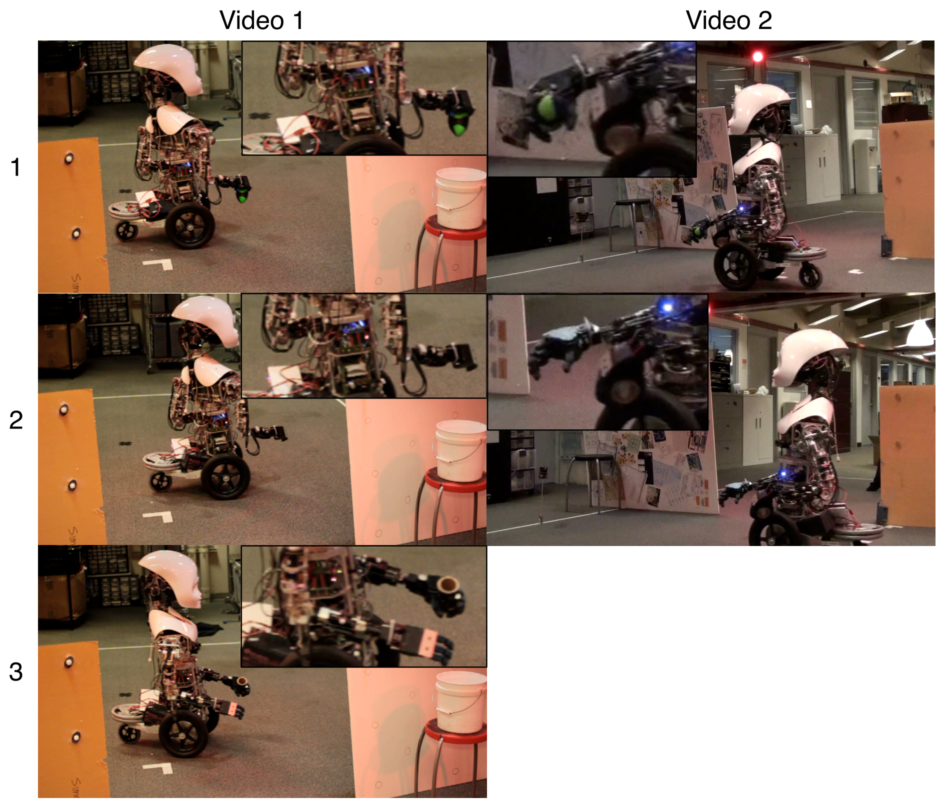


Figure 6-13: Photos taken during the robotic performance of the demonstration scenario under three different sets of mental state goals (see table 6-12). The left column shows a video still from the perspective of the human opponent during the three conditions. The right column shows the same scene from a different angle, revealing what the robot is hiding from the human. In all row's the robot must transport the cylinder to the goal. In row 1, the robot's mental state goals are to reveal that it carries the football, and hide that it carries the cylinder. It accomplishes this by carrying the football in the right hand, and carrying the cylinder in the left hand which it hides behind its body. In row 2, the robot's mental state goal is to hide that it carries the cylinder, but with no goal about carrying a decoy object. It does this by carrying the cylinder in its left hand, again hidden behind the body. In row 3, the robot has no mental state goals. In this case it simply carries the cylinder in its left hand openly.

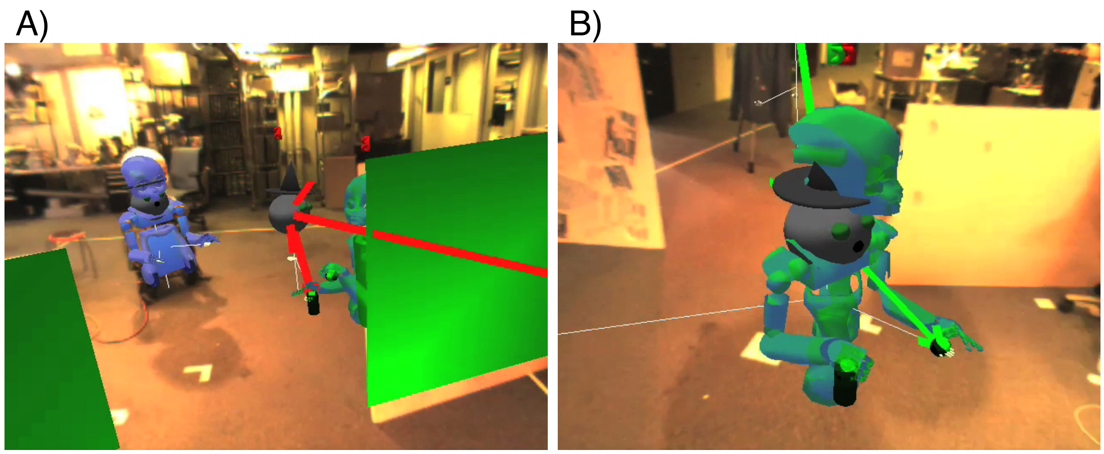


Figure 6-14: Visualizer showing robot's planning mechanism during plan construction for condition 1. The hypothetical robot (green) simulates different action sequences, searching for one which satisfies its goals. Here the goals are to deliver the cylinder to the target, make sure the human sees the robot carry the football (pictured here as a round ball), and make sure the human doesn't see the robot carry the cylinder. *A)* shows a trial run where the robot has failed because the human sees the robot carrying the cylinder. The red arrow shows the trace of the mental state goal which failed. In this case the robot is modeling a human (offscreen back and right) which is in turn modeling the robot, and that human's model of the robot has now gained the mental state "holding cylinder", which is a failure condition. The round head with pointed hat near the hypothetical robot visualizes the human's model of the robot's location - here it is slightly different than the actual hypothetical robot position because the human has lost track of the robot's position behind the barrier (only it's hand has re-emerged in this still), and thus it maintains the last known location. In *B)*, the robot is shown performing what will be a successful action sequence. The green arrows show the trace for the mental state "human sees robot carry ball", which is successfully being created. The cylinder is behind the body of the robot, hidden from the human.

performing the actions motivated by the goals in the corresponding condition from table 6-12). After watching this video, the subjects were instructed to fill out their answers to several questions. The first question asked them to indicate which item they would place in their goal area in response to the robot's actions. Next they were asked if, in future games, they would prefer to team with the robot or play against the robot. Finally a set of questions asked them to rank the robot on several criteria.

In condition three, the robot openly carries the cylinder to the goal. For subjects in group 3 the study is over at this point. In conditions one and two, however, the subjects were shown an additional video. This video depicts the robot from a second angle, revealing the hidden object that the robot kept occluded from the original camera. After seeing this second video, the subjects are then asked the same questions again to evaluate how their answers change in response to this new information.

### 6.4.3 Study Results and Discussion

The answers provided by the subjects were analyzed to address three hypotheses.

- **Hypothesis 1:** The mental state manipulation is successful, as measured by the subjects choice of object. If the robot is successful, people will be fooled by the robot's decoy object in condition one, they will be unsure what to play in condition two, and they will correctly see the robot's actions in condition three and thus be able to win. After seeing the second video, revealing the robot's hidden hand, people will choose the object the robot was hiding.

- **Hypothesis 2:** Subjects will choose the robot as a teammate more frequently when they observe its mental state manipulation capabilities. People will be more willing to team with the robot that hides objects behind its back than the robot that openly carries objects, and will change their mind about teaming with the condition one robot once they realize it had been manipulating mental states.

- **Hypothesis 3:** People are more willing to attribute mental states to the robot once they see that it is pursuing a strategy of mental state manipulation, rather than simply transporting an object to the goal. This hypothesis is evaluated by the subjects' change in rating of several statements after the robot's deception is revealed.

Across the three conditions, 113 subjects completed the entire questionnaire. 41 subjects were in the condition one group, 37 in condition two, and 35 in condition three.

#### Hypothesis 1: Success of Mental State Manipulation

Figure 6-15 shows the subject's object choices across the three conditions. The subjects are instructed to choose the winning object; winning occurs when the robot and the human both play the same object. Subjects in condition 3 watch a video of the robot openly carrying the cylinder, and as expected, they prefer to play the cylinder. In condition 2, the subject watch a video where the robot hides an object behind its back as it approaches the

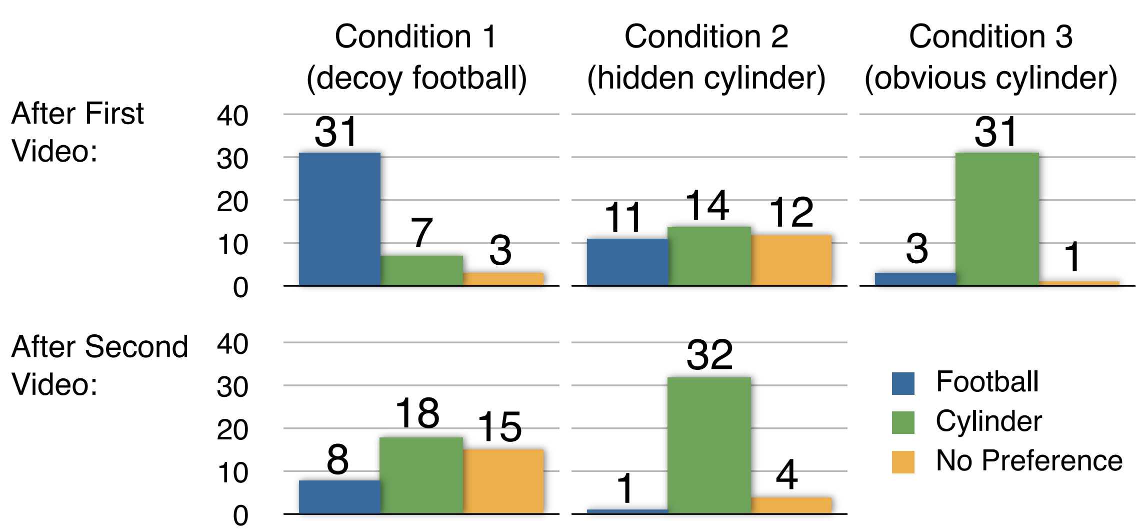


Figure 6-15: Data showing object chosen by human player across each condition, before and after having seen the second video. Human is instructed to choose the winning object, which is defined to be the same object the robot placed in its goal.

goal. As expected, subjects exhibit little preference for a particular object. While some explicitly indicate that they have no preference, the rest are fairly even distributed between the cylinder and the football. In condition three, the subjects see a video of the robot openly transporting the football, with the robot hiding the cylinder behind its back. This strategy appears to work - 76 percent of the subjects choose to play the football, indicating that they expect the robot to play the football. The subjects' choice between these items differs at the .001 significance level ($p < .001$) between the three conditions based on the chi-square test.

Additionally, conditions one through three can be thought of as a spectrum, where the robot in each case tries strategies of decreasing complexity to win the game by getting the human to choose the football. Indeed, if we code the responses numerically, with 1=football, 2=no-preference, and 3=cylinder, we can look at the relationship between the chosen outcome and the conditions one through three. The football is the best choice (from the robot's perspective), indicating that the robot will win. No-preference indicates that the robot has an even chance of winning, and if the human chooses cylinder, the robot has lost. Kendall's tau-b trend test for this relationship is 0.574 with $p < .001$, indicating a statistically significant moderately positive relationship between condition number and object choice. As expected, the human makes better choices (worse for the robot) as the complexity of the robot's strategies decrease.

In condition one and two subjects change their choice of object after seeing the second video feed that shows the robot from another angle, revealing both hands. In condition two this change happens as expected. After seeing the first video, where the robot's hand carries a hidden object, the subjects have little preference for any object. After the second video reveals that the hidden object is the cylinder, the subjects switch to the cylinder. From the McNemar-Bowker Test, this change is significant at the .01 level ($p = .001$).

Condition one is somewhat different. After watching the first video, subjects had a strong preference for the football, indicating that they didn't notice the robot carrying a second object behind its back. When this deception is revealed, and they learn that the robot had the cylinder behind its back, many switch to choosing the cylinder (cylinder becomes the most chosen item). This switch is as expected in the hypothesis; though the robot is now seen to be carrying both objects, so in theory could play either one, subjects tend to attribute the deceptive intention to the robot and predict that it will play the object it had kept hidden rather than the one it had held out in the open. As in condition two, the McNemar-Bowker Test indicates that this change is significant at the .001 level ($p < .001$).

An informal analysis of the written comments about this question sheds some light on this difference from condition two. Of the 12 subjects that indicated no-preference, six give a mechanistic description of the robot's behavior. They do not attribute any deceptive motive to the robot, instead they simply note that they now see that both objects are being transported, so they have no information about what the robot will do. Three of the 12 react in the opposite direction - now that they have seen that the robot has tricked them, they attribute such high levels of deceptive capability that they are unsure which object to choose, because they feel the deception that has been revealed is itself some kind of deception. The remaining six indicate a level of confusion with the task, or seem to have missed some parts of the video. Of the eight subjects that chose the football, six of them wrote responses that revealed they had missed seeing the cylinder in the robot's hand even in the second video, or that they felt they had seen the robot start to place the football in the goal (an unfortunate artifact of the timing of the end of the video). In contrast, eleven of the 18 who chose the cylinder after seeing the second video explicitly mention information in their comments that indicates they were aware of the robot's mental state manipulation - that it is likely to play the cylinder because it was "hiding" it from the subject.

### Hypothesis 2: Willingness of Subjects to Team with Robot

After each subject watched a video and chose a particular object to play, they were also asked whether, in future games, they would prefer to have the robot on their team, or on their competitor's team (See figure 6-16). This subjects in condition one and two, who watch two different videos showing different perspectives of the robot's turn, answer this question twice, once after each video.

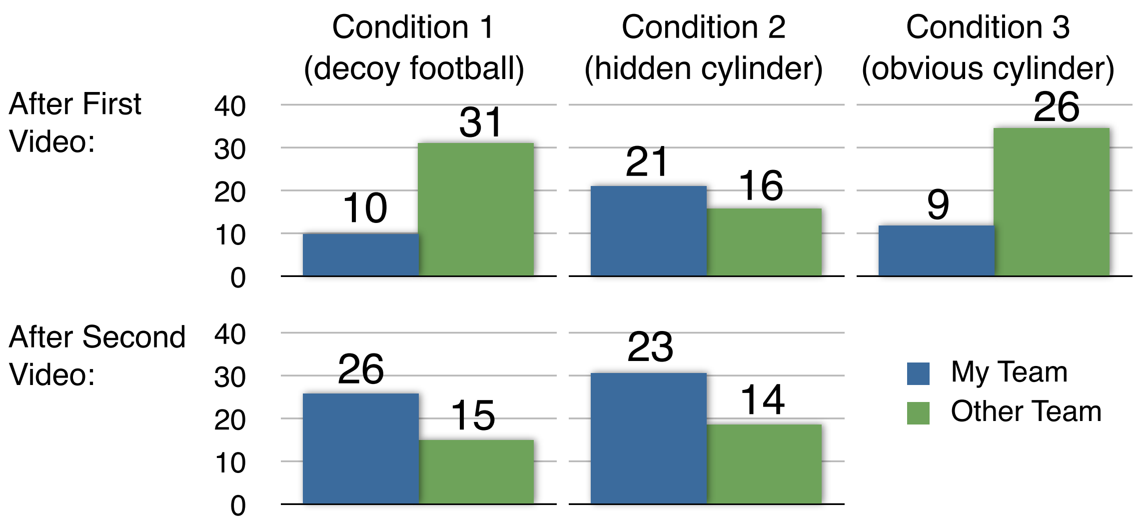


Figure 6-16: Data showing subjects' teaming preference across the three conditions, before and after having seen the second video. Human is asked, if they were to play another game, whether they prefer to have the robot on their team or the other team.

Hypothesis two predicts that subjects will be more willing to team with a robot that is able to perform mental state manipulations. From the analysis of Hypothesis one, we know that subjects largely were fooled by the robot's deception in condition one, choosing the football. Thus, after watching only the first video, we expect subjects in conditions one and three to be less likely to want to team with the robot as compared with condition two, where they witness the robot hiding an object. This expectation was fulfilled. Using a logistic regression model with teaming preference as the dependent variable and condition as the independent variable, the odds of wanting the robot as a teammate in condition one is only 24.6 percent of the odds of wanting the robot as a teammate in condition two ($p < .01$). The odds of wanting the robot as a teammate in condition three is only 26.4 percept of the odds of wanting the robot as a teammate in condition two ($p < .01$).

The subjects change their answer to the teaming question after watching the second video in condition one. Using McNemars Test, this change in teaming preference is significant at the .001 level ($p < .001$). Combined with the differences in teaming preference between the multiple conditions, these differences indicate that when people are aware of mental state manipulation capabilities in a robot, they are more willing to team.

From the analysis of Hypothesis one, it is clear that the manipulation has an effect on subject's initial object choice, and there is strong evidence that, after seeing the second video, subjects change their choice of object. In condition two, it is expected that the subjects were largely aware that the robot was hiding an object, so the change in object choice after seeing the object indicates simply that they now know the object it was hiding behind its back. The evidence from the teaming question provides additional evidence for this assertion - the subjects do not significantly change their preference for the robot as a teammate after seeing the second video (unlike in condition one), indicating that though they are learning what object the robot is carrying, they are not learning anything knew about its skill level.

### Hypothesis 3: Attribution of Mental States to Robot

In addition to the above questions, subjects were asked to rate their agreement with four statements on a five point scale. They were asked to perform these ratings after watching the first video, then again after watching the second video. For condition one, the differences in these ratings indicate their change in evaluation of the robot once they are aware it possesses and uses mental state manipulation capabilities. The questions are displayed in figure 6-17. For all 4 judgments, the Wilcoxon Matched-Pairs Signed-Rank Test indicates with $p < .01$ that the rating changed after the second video .

The question, "The Robot takes my presence into account," was designed to verify that the subjects were able to perceive, in a technical sense, that the physical actions the robot was taking were performed in such a way as to change the perceptual experience of the subject. The question "The robot cares about winning the game" was designed determine if they attributed intent to the robot's actions, rather than simply perceiving them mechanistically. The change in these values that occurred after watching the second video indicates that people were able to perceive that the robot's actions were directed at their perceptual experience, and that that discovery changed their attribution of intent to the robot.

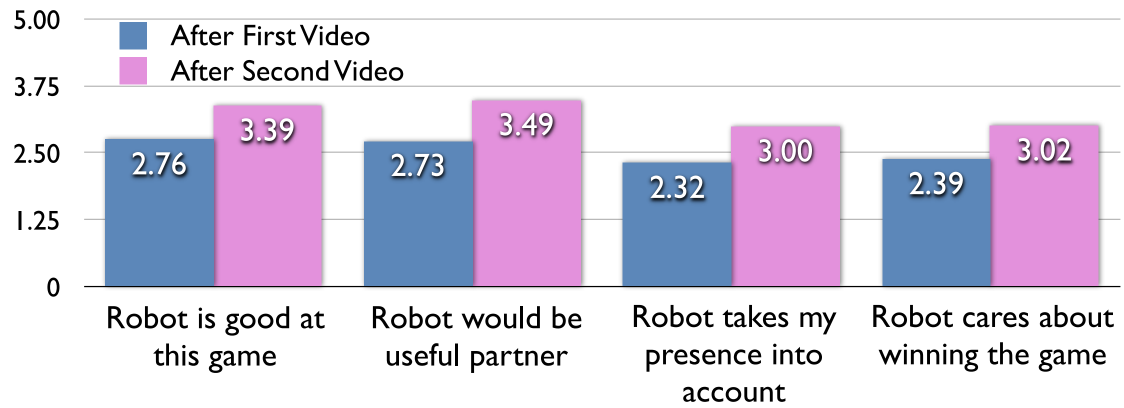


Figure 6-17: Data showing subject's rating of the robot on four questions (using a five point scale) in condition one. Subjects in condition one were asked these questions once after watching the first video of the robot's turn. They are asked to rate the robot again after the robot's hidden behavior is revealed through the second video.

### Conclusion

Through the subject's object choices in the three conditions, the study showed that the mental state manipulation performed by the robot was successful. The mental state manipulation goals the robot pursued did indeed change the behavior of the subjects.

The study also showed that these behaviors were readable to the subjects. After watching the manipulation behavior from a second angle, the subjects were able to better predict the robot's actions based on a correct understanding of its motivation for hiding its actions.

Finally, these capabilities had a positive effect on subjects' willingness to work with the robot, and raised their rating of the robot's capabilities. Subjects' discovery of the mental state manipulation changed both their mechanistic description of the robot's behavior, as well as their description of its behavior in terms of intentions.

# Chapter 7

# Conclusion

## 7.1 Contributions

This thesis presents a novel approach for producing cooperative and competitive robotic behavior which allows the robot to account for and manipulate mental states. This approach combines theory of mind mechanisms with embodied future simulations to allow the robot to take action to achieve its mental state goals.

This thesis presents a self-as-simulator architecture which implements this approach. It describes the modules created to achieve several necessary theory of mind skills, then shows how these are integrated together and used to construct a detailed future simulation which allows the robot to predict and change future mental states of people.

Finally this architecture is demonstrated and evaluated through human subject studies.

## 7.2 Insights and Future Work

This section describes key insights learned from the research described here, as will as limitations and future directions for this work, broken up into several subject areas.

### 7.2.1 Embodiment

#### Connection to World

The concept of embodiment can be used to talk about several aspects of this work. The humans are embodied, and therefore connections exist between their hidden mental states and observable world. The world affects their mental states, and their mental states cause physical action which affects the world. The original architecture I have built on top of is designed to control the behavior of an embodied agent, and therefore it also contains structures and metaphors designed to connect the internal mental states of that agent with the world around it. The re-use of these embodied mechanisms is what allows the agent to make sense of the observable human behavior. This is an observational use of its capacities, even though it is using mechanisms that were created for embodied action; however, to make changes to the mental states and thus behavior of the human, it must physically act. It must use its embodiment to realize the changes in the world which is has determined will effect the human's experience in the way it desires.

One focus of this work has been to leverage this connection between hidden mental states and the observable world to help better understand behavior. Mental states such as goals influence the physical behavior of a human or agent. This relationship between mental states and physical action can be used to predict future behavior from a known goal, or instead to infer an unknown goal from current behavior. Modeling the knowledge of another agent, again by leveraging how physically observable information such as geometric perspective can affect their mental states, can help further refine these inferences.

Other systems which work with modeling mental states tend to focus on the inferences that can be made in the space of mental states, leveraging some mental state information to make inferences about other states or future states, e.g. using information about a current action to determine a current goal. These systems can often make complicated inferences in this space. In contrast, while the work described here can make some inferences in this space, the focus is on the connections between these states and the observable world that are created by being an embodied agent, and how to leverage the rich information contained in these connections to go back and forth between mental state information and the world state that produces or is produced by those states. This relationship takes several forms, such as world state that is producing mental states (as in perception), world state that is revealing mental states (as in action/goal recognition), or mental states that are predicting world state (as when looking forward from inferred goals). The agent powered by the architecture described here uses these connections, which are implemented through the human's and robot's embodiment, to better cooperate and compete with humans. By using its knowledge of how the human's mental states connect to the world, the robot not only models the human's mental states but plans its behavior to specifically alter the human's mental states through changing that world.

**Robotic Embodiment**

The human exists in the physical world, their perceptual and motor systems tying their mental states to that world. The eventual goal of the research described here is to help a robot cooperate and compete with humans in the real world, modeling and manipulating their mental states by leveraging those connections.

Certain aspects of this work also have relevance to agents interacting with humans through a virtual environment such as a game or simulation. Some of the mechanisms described here, such as visual perspective taking, can be applied directly; the human is experiencing information about the virtual world through their avatar, so monitoring the avatar's visual perspective will yield insights about the human's knowledge of the virtual world. Other mechanisms, like watching the motions of another agent to try to determine the physical action they are engaged in, may also be possible to apply in a simulated world depending on the detail of the simulation.

However, as the goal is to create a system that functions in the world, working with and manipulating the beliefs of real humans, it is important to also test the system in this setting. It is fundamentally different experientially for a human to interact with a robot in their own world rather than through controlling an avatar in a simulation. There is evidence that subjects' react differently to similar behaviors depending on whether they are performed by a virtual agent or a physical one [Kidd and Breazeal, 2004], and humans may well react differently to an experience where a robot manipulates their knowledge of the real world around them than to the slightly removed situation where an agent's actions toward their avatar change their beliefs about virtual objects. Finally the space of actions available to a human are much larger than those they can express through a simulation, and it will be interesting to see how they react to the robot's behavior, and if the robot can successfully work with these reactions when the humans are allowed their full range of behavior.

The study presented here is a first step toward these comprehensive explorations. By watching a video of the robot performing these actions from the correct perspective, the subjects get to observe the robot manipulating these real-world beliefs, but because it is a video the subjects do not get to fully explore the interaction, and the actions they can choose to perform in response are quite constrained. To expand upon this evaluation, interesting future work would be to run a similar study to that described here but to do so live, making it possible to observe how the subjects behave if given complete freedom, and how the architecture handles such behavior. Finally the reactions of the subjects to the behavior of the robot will be quite interesting - if the robot succeeds at a manipulation task, how much more powerfully will it affect the human's evaluation of the robot's capabilities if it is happening to them directly, with the robot interactively responding to their actions, rather than being observed by them through video demonstration?

### 7.2.2 Simulation and Re-use

The techniques described here re-use the mechanisms of the robot's behavior architecture where possible to help model the behavior and mental states of the human. The key benefit to this strategy is that it provides a grammar and vocabulary to allow the robot to work with human actions and mental states. Situating detected human actions and mental states within the structures the robot uses for its own actions gives the robot two important advantages. First, it allows the robot to utilize those structures to leverage low level mental state inferences it has made (such as through observing physical actions and performing perspective taking) into higher level mental state information (such as goals). Second, the robot must relate observed information to its own current knowledge and goals, so it can determine how to act in such a way as to achieve its goals taking into account what it knows about the human. This relating of the mental states from other to self is facilitated by the two being in the same representation.

Additionally, by using the robot's own systems to model the behavior of the humans, certain interesting meta-congnitive skills can begin to present themselves "for free". An example here is that because each human the robot is tracking is modeled using a copy of the robot's own systems, that model includes the mechanism which makes a model of all surrounding agents. Thus, the robot's ability to model "what the human thinks the robot thinks" comes automatically from its ability to model "what the human thinks".

These techniques do limit what the robot can model to its own behavioral repertoire. In some ways this is not necessarily a disadvantage, because the purpose of the modeling it to affect its behavior. Thus, modeling aspects of the human which it cannot relate to its own systems and act upon may not be helpful.

There are, however, certain areas where a deviation from this kind of re-use could benefit the robot. In particular, depending on the robot, a human may have very different perceptual and motor capabilities than the robot does. In these cases the techniques described here for re-using the robot's physical motion planning to detect the motions of the human may be inappropriate. Likewise, using the perceptual model of the robot as a filter for what the human can and cannot see may also be incorrect. Many of the same techniques described here could still be used, but the robot would require new mechanisms to take the perceptual perspective of the human and also to map observed physical actions to the robot's own action space.

### 7.2.3 Uncertain World

The architecture described here is intended to operate in the real world, and thus must face certain sources of uncertainty, including noise which interferes with how the world is perceived, uncertainty in predicting future outcomes, and even uncertainty in completing actions.

In the interest of simplicity, throughout this work uncertainty has not been directly addressed at a system level, instead at various points in the architecture a winner take all strategy has been employed to choose the best of available options. This collapses any extra information present about uncertainty and the probabilities of competing options, making the next stage simpler but also discarding other options that could turn out to be correct. In many of these cases it is straightforward to see how certain kinds of uncertainty could be accounted for by making simple additions to the architecture. In others cases, more significant changes would be needed.

One place where this winner take all strategy is used is in combining the different sources of inference information to determine a human's goal. The mechanism described here shows how these multiple sources of inference can be more powerful together than separately, however the way these data sources are combined could certainly be improved. Currently the robot finds a single best-fit physical action and combines that information with the most likely current world state from the model of the human's knowledge to choose a single goal directed action that fits the context and the physical motion. It would be fairly easy to imagine how it could instead track several possible physical actions that match the human's motions, compare those with the space of possible world states it's modeling from the human's perspective, then produce a set of possible goals with different likelihoods.

Another area where an improvement like this could increase robustness would be the search through potential future actions. The search described here is quite simple. Though it accomplishes its goal of demonstrating this mental state planning mechanism, there is room to improve its ability to work with uncertain futures. The current strategy searches for a way for the agent to perform a series of actions which achieve its mental state goals, which themselves evaluate to a boolean "yes" or "no" by examining the specified mental state model for the mental state in question. By modeling perceptual ambiguities (simulated ambiguities, in the case of modeling the mental states of other creatures), the mental states in question could include probabilities updated to reflect the certainty the agent has for a particular mental state. If this information were present, the mental state goals could provide a level of certainty, instead of a simple boolean value. In this way, the robot could search its action space for sequences that provided the highest chance of success, instead of simply choosing the first path that could in theory succeed.

Representing the outcome of actions is another place where mechanisms could be adapted to better handle the uncertainty of the real world; instead of predicting the outcome of actions based on a deterministic calculation, actions with probabilistic predictions could be more robust for a robot's planning mechanisms. In the current system, the robot is simulating the future by using its actions which are designed to be performed in realtime, and each action, as it runs, modifies the world state producing a new condition where new actions are relevant. If the robot were to switch to a new kind of action, which instead produced a distribution of potential world states, it would make the action search process more complicated.

One strategy would be to break the distribution of potential world states into several discrete options, search each of them and combine the probability of that branch occurring with the probability of achieving the mental state goals using those actions. Another option would be to switch to a more complicated action representation, where actions could themselves map from an input distribution over world states to a resulting distribution over world states. Whatever strategy is used, adding mechanisms to support modeling of actions with uncertain results would require changes but could help this architecture function in real world noisy scenarios.

### 7.2.4 Scalability

The architecture described here provides mechanisms that re-use certain aspects of a robot's behavioral mechanisms to work with the mental states humans. This means that certain kinds of additions are simple yet powerful. Adding a new physical motion, goal directed action, or mechanism to pull new information out of the sensory stream gives the robot that new ability, but also allows the robot to re-use that ability when modeling the behavior of the human. Thus, switching to a new task domain with new requirements for actions and sensing is fairly easy, because all that is required is to make the first-person mechanisms the robot will use in that domain.

The simple planning system described here uses an exhaustive search, and thus requires a discrete space of actions and action targets. In many domains it may be advantageous to allow a continuous space of targets. To allow for continuous spaces, it would be necessary to create a more sophisticated planning strategy, which would likely increase the running time required for planning. One possibility would be to try to redefine mental state goals so that they have a scalar value indicating how close they are to being satisfied, so the planning mechanism could try to search through these continuous spaces, iterating towards a good value to satisfy the goals.

Scaling in complexity is difficult for these techniques; this system is very detailed in its planning, but is also computationally expensive. In its current state it works as a real-time planning mechanism for a robot with a limited action space interacting with a handful of agents; certain gains could be made in efficiency, but it is not intended to make long term plans in a large action space. Instead, it is designed to be used in conjunction with other mechanisms that would handle longer term, lower resolution planning. Very interesting future work involves utilizing the information available through these introspective mechanisms to form longer term, lower resolution plans which allow the robot to plan out longer strategies while still being able to make these very detailed simulations to guide the specifics of its immediate actions. One possible avenue would be to have the robot use a hierarchical representation of its actions (perhaps learned) to represent higher level skills; these skills achieve certain goals through their lower level actions, but in long term planning the specifics of how they would achieve that goal could be ignored.

# Bibliography


Janet W. Astington and Alison Gopnik. Developing understanding of desire and intention. In Andrew Whiten, editor, *Natural Theories of Mind: Evolution, Development and Simulation of Everyday Mindreading*, chapter 3, pages 39–50. Basil Blackwell, 1991.

Bruce Blumberg, Marc Downie, Yuri Ivanov, Matt Berlin, Michael Patrick Johnson, and Bill Tomlinson. Integrated learning for interactive synthetic characters. *ACM Transactions on Graphics*, 21(3: Proceedings of ACM SIGGRAPH 2002), 2002.

C. Breazeal, D. Buchsbaum, J. Gray, and B. Blumberg. Learning from and about others: Toward using imitation to bootstrap the social competence of robots. *Artificial Life*, 11, 2005.

Brooks, A.G., Berlin, M., Gray, J., and Breazeal, C. Untethered robotic play for repetitive physical tasks. In *ACM International Conference on Advances in Computer Entertainment*, 2005.

D. Buchsbaum. Imitation and social learning for synthetic characters. Master's thesis, MIT, 2004.

R. Burke, D. Isla, M. Downie, Y. Ivanov, and B. Blumberg. CreatureSmarts: The art and architecture of a virtual brain. In *Proceedings of the Game Developers Conference*, pages 147–166, San Jose, CA, 2001.

B. Calvo-Merino, D. E. Glaser, J. Grèzes, R. E. Passingham, and P. Haggard. Action observation and acquired motor skills: An fmri study with expert dancers. *Cerebral Cortex*, 15(8):1243–1249, 2005.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

Rawichote Chalodhorn, David B. Grimes, Keith Grochow, and Rajesh P. N. Rao. Learning to walk through imitation. In *IJCAI*, pages 2084–2090, 2007.

P.S. Churchland and P.M. Churchland. *Folk Psychology.* MIT Press, 1998.

J. Demiris and G. Hayes. Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in animals and artifacts*, pages 321–361. MIT Press, Cambridge, MA, 2002.

Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems*, 54(5):361–369, 2006.

W. Dong, B. Lepri, A. Cappelletti, A.S. Pentland, F. Pianesi, and M. Zancanaro. Using the influence model to recognize functional roles in meetings. In *Proceedings of the ninth international conference on Multimodal interfaces*, pages 271–278. ACM New York, NY, USA, 2007.

M. Downie. Behavior, animation, and music: The music and movemnt of synthetic characters. Master's thesis, MIT, 2000.

V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, 1998.

Noah D. Goodman, Vikash K. Mansinghka, and Joshua B. Tenenbaum. Learning grounded causal models. *Proceedings of the Twenty-Ninth Annual Conference of the Cognitive Science Society*, 2007.

A. Gopnik and H.M. Wellman. The theory theory. In L. A. Hirschfeld and S. A. Gelman, editors, *Mapping the mind: Domain specificity in cognition and culture*, chapter 10, pages 257–293. Cambridge University Press, 1994.

R. Gordon. Folk psychology as simulation. *Mind and Language*, 1:158–171, 1986.

Jesse Gray, Guy Hoffman, Sigurdur Orn Adalgeirsson, Matt Berlin, and Cynthia Breazeal. Expressive, interactive robots: Tools, techniques, and insights based on collaborations. In *HRI 2010 Workshop: What do collaborations with the arts have to say about HRI?*, 2010.

Gray, J., Breazeal, C., Berlin, M., Brooks, A., and Lieberman, J. Action parsing and goal inference using self as simulator. In *14th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*, Nashville, Tennessee, 2005. IEEE.

Kai-yuh Hsiao. *Embodied Object Schemas for Grounding Language Use.* PhD thesis, MIT Media Lab, 2007.

Damian A. Isla and Bruce M. Blumberg. Object persistence for synthetic creatures. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1356–1363, New York, NY, USA, 2002. ACM. ISBN 1-58113-480-0. doi: http://doi.acm.org/10.1145/545056.545132.

Y. Ivanov and R. Hamid. Weighted Ensemble Boosting for Robust Activity Recognition in Video. *MACHINE GRAPHICS AND VISION*, 15(3/4):415, 2006.

O. C. Jenkins and M. J. Matarić. Deriving action and behavior primitives from human motion data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2551–2556, 2002.

O.C. Jenkins, G.G. Serrano, and M.M. Loper. Interactive Human Pose and Action Recognition Using Dynamical Motion Primitives. *INTERNATIONAL JOURNAL OF HUMANOID ROBOTICS*, 4(2):365, 2007.

M. Johnson and Y. Demiris. Perceptual perspective taking and action recognition. *International Journal of Advanced Robotic Systems*, 2(4):301–308, December 2005.

R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

R. Kelley, A. Tavakkoli, C. King, M. Nicolescu, M. Nicolescu, and G. Bebis. Understanding human intentions via hidden markov models in autonomous mobile robots. *Proceedings of the 3rd international conference on Human robot interaction*, pages 367–374, 2008.

C. D. Kidd and C. Breazeal. Effect of a robot on user perceptions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

T. Kushnir, A. Gopnik, L. Schulz, and D. Danks. Inferring hidden causes. *Proceedings of the 25th Annual Meeting of the Cognitive Science Society*, pages 699–703, 2003.

John E. Laird. It knows what you're going to do: adding anticipation to a quakebot. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 385–392, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-326-X. doi: http://doi.acm.org/10.1145/375735.376343.

S.W. Lee and K. Mase. Activity and location recognition using wearable sensors. *Pervasive Computing, IEEE*, 1(3):24–32, 2002.

Stacy C. Marsella and David V. Pynadath. Modeling influence and theory of mind. *Artificial Intelligence and the Simulation of Behavior*, 2005.

Maja J. Matarić. Sensory-motor primitives as a basis for imitation: linking perception to action and biology to robotics. In *Imitation in animals and artifacts*, pages 391–422. MIT Press, Cambridge, MA, USA, 2002. ISBN 0-262-04203-7.

A. N. Meltzoff. Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental Psychology*, 31:838–850, 1995.

A. N. Meltzoff and M. K. Moore. Explaining facial imitation: a theoretical model. *Early Development and Parenting*, 6:179–192, 1997.

A.N. Meltzoff and R. Brooks. Self-Experience as a Mechanism for Learning About Others: A Training Study in Social Cognition. *Developmental psychology*, 44(5):1257, 2008.

Andrew N. Meltzoff. Imitation and other minds: The "like me" hypothesis. In S. Hurley and N. Chater, editors, *Perspectives on Imitaion: From Neuroscience to Social Science*, volume 2, pages 55–77. MIT Press, Cambridge, MA, 2005.

S. Nichols and S.P. Stich. *Mindreading: An Integrated Account of Pretence, Self-Awareness, and Understanding Other Minds.* Oxford University Press, USA, 2003.

G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.

Deb Roy. Semiotic schemas: A framework for grounding language in the action and perception. *Artificial Intelligence*, 167(1–2):170–205, 2005.

Paul Scerri, D. Pynadath, N. Schurr, A. Farinelli, S. Gandhe, and M. Tambe. Team oriented programming and proxy agents: The next generation. In *Proceedings of 1st international workshop on Programming Multiagent Systems*, 2004.

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*, 35(4):460–470, 2005.

H.M. Wellman, D. Cross, and J. Watson. Meta-Analysis of Theory-of-Mind Development: The Truth about False Belief. *Child Development*, 72(3):655–684, 2001.

H. Wimmer and J. Perner. Beliefs about beliefs: Representation and constraining function on wrong beliefs in young children's understanding of deception. *Cognition*, 13:103–128, 1983.