

# Coordination in Human-Robot Teams Using Mental Modeling and Plan Recognition

Kartik Talamadupula<sup>†</sup> Gordon Briggs<sup>‡</sup> Tathagata Chakraborti<sup>†</sup>  
Matthias Scheutz<sup>‡</sup> Subbarao Kambhampati<sup>†</sup>

<sup>†</sup>Dept. of Computer Science and Engineering  
Arizona State University  
Tempe, AZ 85281, USA  
{krt,tchakra2,rao}@asu.edu

<sup>‡</sup>HRI Laboratory  
Tufts University  
Medford, MA 02155, USA  
{gbriggs,mscheutz}@cs.tufts.edu

**Abstract**—Beliefs play an important role in human-robot teaming scenarios, where the robots must reason about other agents’ intentions and beliefs in order to inform their own plan generation process, and to successfully coordinate plans with the other agents. In this paper, we cast the evolving and complex structure of beliefs, and inference over them, as a planning and plan recognition problem. We use agent beliefs and intentions modeled in terms of predicates in order to create an automated planning problem instance, which is then used along with a known and complete domain model in order to predict the plan of the agent whose beliefs are being modeled. Information extracted from this predicted plan is used to inform the planning process of the modeling agent, to enable coordination. We also look at an extension of this problem to a plan recognition problem. We conclude by presenting an evaluation of our technique through a case study implemented on a real robot.

## I. INTRODUCTION

As robotic systems of increasing robustness and autonomy become a reality, the need for technologies to facilitate successful coordination of behavior in human-robot teams becomes more important. Specifically, robots that are designed to interact with humans in a manner that is as *natural* and *human-like* as possible will require a variety of sophisticated cognitive capabilities akin to those that human interaction partners possess [1]. Performing mental modeling, or the ability to reason about the mental states of another agent, is a key cognitive capability needed to enable natural human-robot interaction [2]. Human teammates constantly use knowledge of their interaction partners’ belief states in order to achieve successful joint behavior [3], and the process of ensuring that both interaction partners have achieved *common ground* with regard to mutually held beliefs and intentions is one that dominates much of task-based dialogue [4]. However, while establishing and maintaining common ground is essential for team coordination, the process by which such information is utilized by each agent to coordinate behavior is also important. A robot must be able to predict human behavior based on mutually understood beliefs and intentions. In particular, this capability will often require the ability to infer and predict plans of human interaction partners based on their understood goals. There has been a variety of prior work in developing coordination and prediction capabilities for

human-robot interaction in joint tasks involving physical interaction, such as assembly scenarios [5] and object handovers [6]. However, these scenarios assume the robot is in direct interaction with the human teammate and is able to observe the behavior of the human interactant throughout the task execution. Some forms of coordination may need the robot to be able to predict a teammate’s behavior from only a high-level goal and mental model.

Automated planning is a natural way of generating plans for an agent given that agent’s high-level model and goals. The plans thus generated can be thought of either as directives to be executed in the world, or as the culmination of the agent’s deliberative process. When an accurate representation of the agent’s beliefs about the world (the model and the state) as well as the agent’s goals are available, an automated planner can be used to *project* that information into a *prediction* of the agent’s future plan. This prediction process can be thought of as a simple plan recognition process; further in this paper, we will discuss the expansion of this process to include incomplete knowledge of the goals of the agent being modeled.

The main contribution of this work is to demonstrate how preexisting components within a robotic architecture – specifically the belief modeling and planning components – can be integrated to provide needed competencies for human-robot team coordination. First, we will present a simple human-robot interaction (HRI) scenario that will necessitate mental modeling and planning-based behavior prediction for successful human-robot team coordination. We will then present the formal representation of the beliefs in our system, and the mapping of these beliefs into a planning problem instance in order to *predict* the plan of the agent of interest. We will also discuss the expansion of this problem to accommodate state-of-the-art plan recognition approaches. Finally, we will describe the component integration within the DIARC [7] architecture that enables our theory on a real robot, and present the results of a case study.

## II. MOTIVATION

Consider a disaster response scenario inspired by an Urban Search and Rescue (USAR) task that occurs in a facility with a long hallway. Rooms 1 and 2 are at the extreme end of

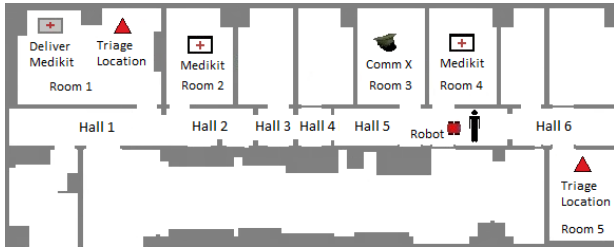


Fig. 1. A map of the human-robot teaming scenario discussed in this paper.

one side, whereas rooms 3-5 are on the opposite side (see Fig. 1). Consider the following dialogue exchange:

```
H: Comm. X is going to perform triage in room 5.
R: Okay.
H: I need you to bring a medical kit to room 1.
R: Okay.
```

The robot  $R$  has knowledge of two medical kits, one on each side of the hallway (in rooms 2 and 4). Which medical kit should the robot attempt to acquire? If commander  $X$  (CommX) does not already have a medical kit, then he or she will attempt to acquire one of those two kits. In order to avoid inefficiency caused by resource conflicts (e.g., wasted travel time), the robot ought to attempt to acquire the kit that is *not* sought by the human teammate.

The medical kit that CommX will select depends on a variety of factors, including – but not limited to – the *duration* of each activity and the *priority* given by CommX to each activity. If the commander had goals to perform triage in multiple locations, the medical kit he or she would acquire would be determined by what triage location he or she visits first. Additionally, the *beliefs* about the environment may differ between the robot and human teammates. Consider a variation of the previous dialogue / scenario (where previously there existed only one medical kit in room 2):

```
H: I just put a new medical kit in room 4.
H: Comm. X is going to perform triage in room 5.
R: Okay.
H: I need you to bring a medical kit to room 1.
R: Okay.
```

While the robot now knows there are two medical kits, CommX likely only knew of the original one, and will thus set out to acquire that one, despite it being at the opposite end of the hallway. Therefore, successful prediction of a human teammate’s behavior will require modeling that teammate, assuming he or she adopts a rational policy to achieve multiple goals given one’s best estimate of their belief state. One way of performing such modeling is by leveraging the *planning* system found within the robotic architecture. In the following, we will detail our process of modeling beliefs, casting them into a planning problem instance, predicting the plan of the agent of interest using this problem instance, and finally achieving coordination via that predicted plan.

### III. BELIEF MODELING

In our system, beliefs are represented in a special component that handles belief inference and interacts with various

other architectural components. We clarify at the outset that we use “belief” in the rest of this paper to denote the robot’s knowledge, and not in the sense of “belief space”. Beliefs about state are represented by predicates of the form  $bel(\alpha, \phi)$ , which denote that agent  $\alpha$  has a belief that  $\phi$  is true. Goals are represented by predicates of the form  $goal(\alpha, \phi, P)$ , which denote that agent  $\alpha$  has a goal to attain  $\phi$  with priority  $P$ .

Belief updates are primarily generated via the results of the semantic and pragmatic analyses performed by the natural language processing subsystem, which are submitted to the belief component (the details of this process are described in [8]). While the interpretation of natural language communication allows for the most direct inferences about an interlocutor’s belief state, our system does allow for belief updates to be generated from other input modalities as well (e.g., the vision system).

In order for a robot to adopt the perspective of another agent  $\alpha$ , we must consider the set of all beliefs that the robot ascribes to  $\alpha$ . This can be obtained by considering a belief model  $Bel_\alpha$  of another agent  $\alpha$ , defined as  $\{\phi \mid bel(\alpha, \phi) \in Bel_{self}\}$ , where  $Bel_{self}$  denotes the first-order beliefs of the robot (e.g.,  $bel(self, at(self, room1))$ ). Likewise, the set of goals ascribed to another agent can be obtained  $\{goal(\alpha, \phi, P) \mid goal(\alpha, \phi, P) \in Bel_{self}\}$ .

This belief model, in conjunction with beliefs about the goals / intentions of another agent, will allow the robot to instantiate a planning problem. Here, it is important to note that all agents share the same basic beliefs about the initial task goal and the initial environmental state (beliefs about subsequent goals and states can differ among agents, see Section IV-A for details).

#### A. Case Analysis

First, we walk through our architecture’s handling of the motivating scenario. The simple case is where the robot has knowledge of the location of both medical kits and the location of CommX. The robot also believes that the commander’s belief space is equivalent (at least in terms of the relevant scenario details) to its own. This belief space is described below:

$$Bel_{self} = \{at(mk1, room2), at(mk2, room4), at(commX, room3), bel(commX, at(commX, room3)), bel(commX, at(mk1, room2)), bel(commX, at(mk2, room4))\}$$

For the sake of future brevity, we will express the predicates describing the robot’s beliefs about the beliefs of CommX using the notation  $Bel_{commX} \subseteq Bel_{self}$ , and the predicates describing the robot’s beliefs about the goals of CommX as  $G_{C_X} \subseteq Bel_{self}$ :

$$Bel_{commX} = \{at(mk1, room2), at(mk2, room4), at(commX, room3)\}$$

$$G_{C_X} = \{\}$$

A planning problem (as specified in Section IV-A) is submitted to the Sapa Replan planner. Since  $G_{C_X}$  is initially an empty set, no plan is computed by the planner.

However, the robot then receives the first piece of natural language input: ‘‘Comm. X is going to perform triage in room 5’’. As a result of the processing from the natural language subsystem, including applying pragmatics rules of the form described in [8], the robot’s belief model of CommX is updated:

$$\begin{aligned} Bel_{commX} &= \{at(mk1, room2), at(mk2, room4), \\ &at(commX, room3)\} \\ G_{C_X} &= \\ &\{goal(commX, triaged(commX, room1), normal)\} \end{aligned}$$

The new problem (with an updated  $G_{C_X}$ ) is submitted to the planner, which returns the following plan:

$$\begin{aligned} \Pi_{commX} &= \langle move(commX, room3, hall5), \\ &move(commX, hall5, hall6), \\ &move(commX, hall6, room4), \\ &pick\_up(commX, mk2, room4), \\ &move(commX, room4, hall6), \\ &move(commX, hall6, room5), \\ &conduct\_triage(commX, room5) \rangle \end{aligned}$$

This plan is used by the robot to denote the plan that CommX is likely utilizing. The robot is subsequently able to infer that the medical kit in room 4 has likely been taken by CommX, and can instead aim for the other available medkit, thus successfully achieving the desired coordination.

#### IV. AUTOMATED PLANNING

Automated planning representations are a natural way of encoding an agent’s beliefs such that a simulation of those beliefs may be produced to generate information that is useful to other agents in the scenario. These representations come with a notion of logical *predicates*, which can be used to denote the agent’s current belief: a collection of such predicates is used to denote a *state*. Additionally, *actions* can be used in order to model the various decisions that are available to an agent whose beliefs are being modeled; these actions will modify the agent’s beliefs, since they effect changes in the world (state). Finally, planning representations can also be used to specify *goals*, which can be used to denote the agent’s intentions and/or desires.

Together, these three features – predicates, actions, and goals – can be used to create an *instance* of a planning problem, which features a domain model and a specific problem instance. Formally, a planning problem  $\Pi = \langle D, \pi \rangle$  consists of the domain model  $D$  and the problem instance  $\pi$ . The domain model consists of  $D = \langle T, V, S, A \rangle$ , where  $T$  is a list of the object *types* in the model;  $V$  is a set of variables that denote objects that belong to the types  $t \in T$ ;  $S$  is a set of named first-order logical predicates over the variables  $V$  that together denote the state; and  $A$  is a set of actions or *operators* that stand for the decisions available to the agent, possibly with costs and/or durations.

Finally, a planning problem instance consists of  $\pi = \langle \mathbb{O}, \mathbb{I}, \mathbb{G} \rangle$ , where  $\mathbb{O}$  denotes a set of constants (objects), each with a type corresponding to one of the  $t \in T$ ;  $\mathbb{I}$  denotes the *initial state* of the world, which is a list of the predicates from  $S$  initialized with objects from  $\mathbb{O}$ ; and  $\mathbb{G}$  is

a set of *goals*, which are also predicates from  $S$  initialized with objects from  $\mathbb{O}$ .

This planning problem  $\Pi = \langle D, \pi \rangle$  can be input to an automated planning system, and the output is in the form of a *plan*  $\Upsilon = \langle \hat{a}_1 \dots \hat{a}_n \rangle$  – which is just a sequence of actions such that  $\forall i, a_i \in A$ , and  $\langle \hat{a}_1 \dots \hat{a}_n \rangle$  are each copies of the respective  $a_i$ s initialized with objects from  $\mathbb{O}$ .

#### A. Mapping Beliefs into a Planning Problem

In this section, we formally describe the process of mapping the robot’s beliefs about other agents into a planning problem instance. First, the initial state  $\mathbb{I}$  is populated by all of the robot’s initial beliefs about the agent  $\alpha$ . Formally,  $\mathbb{I} = \{\phi \mid bel(\alpha, \phi) \in Bel_{robot}\}$ , where  $\alpha$  is the agent whose beliefs the robot is modeling. Similarly, the goal set  $\mathbb{G}$  is populated by the robot’s beliefs of agent  $\alpha$ ’s goals; that is,  $\mathbb{G} = \{\phi \mid goal(\alpha, \phi, P) \in Bel_{robot}\}$ , where  $P$  is the *priority* assigned by agent  $\alpha$  to a given goal. This priority can be converted into a numeric quantity as the reward or penalty that accompanies a goal. Finally, the set of objects  $\mathbb{O}$  consists of all the objects that are mentioned in either the initial state, or the goal description:  $\mathbb{O} = \{o \mid o \in (\phi \mid \phi \in (\mathbb{I} \cup \mathbb{G}))\}$ .

Next, we turn our attention to the domain model  $D$  that is used in the planning process. For this work, we assume that the actions available to an agent are known to all the other agents in the scenario; that is, we rule out the possibility of beliefs on the models of other agents (of course, rolling back this assumption would result in a host of interesting possibilities – we allude to this in Section IV-C). However, even with full knowledge of an agent  $\alpha$ ’s domain model  $D_\alpha$ , the planning process must be carried out in order to extract information that is relevant to the robot’s future plans.

#### B. Coordination Using Plans

In order to facilitate coordination between agents using the robot’s knowledge of the other agent  $\alpha$ ’s beliefs, we utilize two separate planning problems,  $\Pi_R$  (robot) and  $\Pi_\alpha$  (agent  $\alpha$ ) respectively. The robot’s problem consists of its domain model  $D_R = \langle T_R, V_R, S_R, A_R \rangle$  and the initial planning instance  $\pi_R$ , which houses the initial state that the robot begins execution from as well as the initial goals assigned to it. The robot also has some beliefs about agent  $\alpha$ ; these beliefs are used to construct  $\alpha$ ’s problem  $\Pi_\alpha = \langle D_\alpha, \pi_\alpha \rangle$  following the procedure outlined previously (note that currently, we use the same domain model for the robot and agent  $\alpha$ ; i.e.,  $D_R$  and  $D_\alpha$  are the same).

Both of these planning problems are given to separate instances of the planning system, and respective plans  $\Upsilon_R$  and  $\Upsilon_\alpha$  are generated. A key difference between the two plans must be pointed out here: although  $\Upsilon_R$  is a *prescriptive* plan – that is, the robot must follow the actions given to it by that plan,  $\Upsilon_\alpha$  is merely a *prediction* of agent  $\alpha$ ’s plan based on the robot’s knowledge of  $\alpha$ ’s beliefs.

In the case of coordination with agent  $\alpha$  that needs to happen in the future, the robot can turn to the simulated plan  $\Upsilon_\alpha$  generated from that agent’s beliefs. The crux of this approach involves the robot creating a new goal for itself

(which represents the coordination commitment made to the other agent) by using information that is extracted from the *predicted* (or simulated) plan  $\Upsilon_\alpha$  of that agent. Formally, the robot adds a new goal  $g_c$  to its set of goals  $\mathbb{G}_R \in \pi_R$ , where  $g_c$  is a first-order predicate from  $S_R$  instantiated with objects extracted from the relevant actions of agent  $\alpha$  in  $\Upsilon_\alpha$ .

### C. Plan Recognition

So far, we have assumed that the goals of  $\text{CommX}$  are known completely, and that the plan computed by the planner is exactly the plan that the commander will follow. However, this is unlikely to hold for many real world scenarios, given that we are only equipped with a *belief* of the likely goal of  $\text{CommX}$  based on updates from  $\text{CommY}$ ; this may not be a full description of the actual goal. Further, in the case of an incompletely specified goal, there might be a set of *likely* plans that the commander can execute, which brings into consideration the issue of plan or goal recognition given a stream of observations and a possible goal set. This also raises the need for an online re-recognition of plans, based on incremental inputs or observations. In this section, we propose a plan recognition approach that takes these eventualities into account.

1) *Goal Extension and Multiple Plans:* To begin with, it is worth noting that there can be multiple plans even in the presence of completely specified goals (even if  $\text{CommX}$  is fully rational). For example, there may be multiple optimal ways of achieving the same goal, and it is not obvious beforehand which one  $\text{CommX}$  is going to follow. In the case of *incompletely* specified goals, the presence of multiple likely plans become more obvious. We thus consider the more general case where  $\text{CommX}$  may be following one of several possible plans, given a set of observations.

To accommodate this, we extend the robot's current belief of  $\text{CommX}$ 's goal,  $\mathbb{G}$ , to a hypothesis goal *set*  $\Psi$  containing the original goal  $\mathbb{G}$  along with other possible goals obtained by adding feasible combinations of other possible predicate instances not included in  $\mathbb{G}$ . To understand this procedure, let's first look at the set  $\hat{S}$ , defined as the subset of the predicates from  $S$  which cannot have different grounded instances present in any single goal. The existence of  $\hat{S}$  is indeed quite common for most scenarios, including our running example where the commander cannot be in two different rooms at the same time; hence for example, we need not include both  $\text{at}(\text{commX}, \text{room3})$  and  $\text{at}(\text{commX}, \text{room4})$  in the same goal. Hence  $\text{at}(\text{?comm}, \text{?room})$  is one of the (lifted) predicates included in  $\hat{S}$ .

Now, let us define  $Q = \{q \mid q_0 \in \mathbb{G}\} \cap \hat{S}$  as the set of such lifted *unrepeatable* predicates that are already present in  $\mathbb{G}$ , where  $q_0$  refers to a lifted domain predicate  $q \in S$  grounded with an object from the set of constants  $\mathbb{O}$ , and similarly,  $q$  is the lifted counterpart of the grounded domain predicate  $q_0$ . Following this representation, the set difference  $\hat{S} \setminus Q$  gives the unrepeatable predicates in the domain that are absent in the original goal, and its power set gives all possible combinations of such predicates. Then, let  $B_1 = (\mathcal{P}(\hat{S} \setminus Q))_{\mathbb{O}}^*$  denote all possible instantiations of these

predicates grounded with constants from  $\mathbb{O}$ . Similarly,  $B_2 = \mathcal{P}((S \setminus \hat{S})_{\mathbb{O}}^*)$  denotes all possible grounded combinations of the repeatable predicates (note in the case of  $B_1$  we were doing the power operation before grounding to avoid repetitions). Then we can compute the hypothesis set of all feasible goals as  $\Psi = \{G \mid G \in B_1 \cup B_2\}$ .

Identifying the set  $\hat{S}$  is an important step in this procedure and can reduce the number of possible hypotheses exponentially. However, to make this computation, we assume some domain knowledge that allows us to determine which predicates cannot in fact co-occur. In the absence of any such domain knowledge, the set  $\hat{S}$  becomes empty, and we can compute a more general  $\Psi = \{G \mid G \in \mathcal{P}(S_{\mathbb{O}}^*)\}$  that includes all possible combinations of all possible grounded instances of the domain predicates. Note that this way of computing possible goals may result in many unachievable goals, but there is no obvious domain-independent way to resolve such conflicting predicates. However, it turns out that since achieving such goals will incur infinite costs, their probabilities of occurrence will reduce to zero, and such goals will eventually be pruned out of the hypothesis goal set under consideration.

2) *Goal / Plan Recognition:* In the present scenario, we thus have a set  $\Psi$  of goals that  $\text{CommX}$  may be trying to achieve, and observations of the actions  $\text{CommX}$  is currently executing (as relayed to the robot by  $\text{CommY}$ ). At this point we refer to the work of Ramirez and Geffner [9] who provided a technique to compile the problem of plan recognition into a classical planning problem. Given a sequence of observations  $\theta$ , we recompute the probability distribution over  $G \in \Psi$  by using a Bayesian update  $P(G|\theta) \propto P(\theta|G)$ , where the prior is approximated by the function  $P(\theta|G) = 1/(1 + e^{-\beta\Delta(G,\theta)})$  where  $\Delta(G, \theta) = C_p(G - \theta) - C_p(G + \theta)$ .

Here  $\Delta(G, \theta)$  gives an estimate of the difference in cost  $C_p$  of achieving the goal  $G$  without and with the observations, thus increasing  $P(\theta|G)$  for goals that explain the given observations. Note that this also accounts for agents which are not perfectly rational, as long as they have an inclination to follow cheaper (and not necessarily the cheapest) plans, which is a more realistic model of humans. Thus, solving two planning problems, with goals  $G - \theta$  and  $G + \theta$ , gives us the required probability update for the distribution over possible goals of  $\text{CommX}$ . Given this new distribution, the robot can compute the future actions that  $\text{CommX}$  may execute based on the most likely goal.

3) *Incremental Plan Recognition:* It is also possible that the input will be in the form of a stream of observations, and that the robot may need to update its belief as and when new observations are reported. The method outlined in the previous section would require the planner to solve two planning problems from scratch for each possible goal, after every new observation. Clearly, this is not feasible, and some sort of incremental re-recognition is required. Here we begin to realize the advantage of adopting the plan recognition technique described above: by compiling the plan recognition problem into a planning problem, the

task of updating a recognized plan becomes a replanning problem with updates to the goal state [10]. Further, every new observation does not produce an update, since in the event that the agent being observed is actually following the plan that has been recognized, the goal state remains unchanged; while in the case of an observation that does not agree with the current plan, the goal state gets extended by an extra predicate. Determining the new cost measures thus does not require planning from scratch, and can be computed by using efficient replanning techniques.

## V. IMPLEMENTATION

For our proof-of-concept validation, we used the Willow Garage PR2 robot. The PR2 platform allows for the integration of ROS localization and navigation capabilities with the DIARC architecture. Components in the system architecture were developed in the Agent Development Environment (ADE) (see <http://ade.sourceforge.net>) which is a framework for implementing distributed cognitive robotic architectures. Speech recognition was simulated using the standard simulated speech recognition in ADE (which allows input of text from a GUI), and speech output was provided by the MaryTTS text-to-speech system.

### A. Belief Component

The belief component in DIARC utilizes SWI-Prolog in order to represent and reason about the beliefs of the robotic agent (and beliefs about beliefs). In addition to acting as a wrapper layer around SWI-Prolog, the belief component contains methods that extract the relevant belief model sets described in Section III and handling the interaction with the planner component. Specifically, this involves sending the set of beliefs and goals of a particular agent that needs to be modeled to the planner. Conversion of these sets of predicates into a planner problem is handled in the planner component.

### B. Planner

In order to generate plans that are predicated on the beliefs of other agents, we employ the Sapa Replan [11] planner, an extension of the metric temporal planner Sapa [12]. Sapa Replan is a state-of-the-art planner that can handle: (i) actions with costs and durations; (ii) partial satisfaction [13] of goals; and (iii) changes to the world and model via replanning [14]. Sapa Replan additionally handles temporal planning, building on the capabilities of the Sapa planner. To facilitate replanning, the system contains an *execution monitor* that oversees the execution of the current plan in the world; the monitor interrupts the planning process whenever there is an external change to the world that the planner may need to consider. The monitor additionally focuses the planner’s attention by performing objective (goal) selection, while the planner, in turn, generates a plan using heuristics that are extracted by supporting some subset of those objectives. The full integration of Sapa Replan with the DIARC architecture is described in our earlier work [15].

### C. Plan Recognition

For the plan recognition component, we used the probabilistic plan recognition algorithm developed by Ramirez and Geffner [9]. The base planner used in the algorithm is the version of greedy-LAMA [16] used in the sixth edition of the International Planning Competition in 2008. To make the domain under consideration suitable for the base planner, the durations of the actions were ignored while solving the planning problems during the recognition phase. We report initial observations from using the plan recognition component (implemented using LAMA) in Section VI-B.

## VI. EVALUATION

In this section, we present a demonstration of the plan prediction capabilities described in Section IV through a set of proof-of-concept validation cases. These cases include an implementation with the full robotic architecture on an actual robotic platform (Willow Garage PR2), as well as a more extensive set of cases that were run with a limited subset of the cognitive architecture in simulation. These validation cases are not intended to be a comprehensive account of the functionality that our belief modeling and planning integration affords us, but rather indicative of the success of our architectural integration (which also seeks to highlight some interesting and plausible scenarios in a human-robot teaming task). First, we present a video of an instance similar to the case described in Section III-A evaluated on a PR2 robot and annotated with the robot’s knowledge of CommX’s beliefs, as well as its prediction of the commander’s plan: <http://tinyurl.com/beliefs-anno>.

### A. Simulation Runs

We also utilized that same scenario to perform a more extensive set of simulations. We varied the number of medical kits the robot believes CommX knows about (1 vs. 2), the believed location of each medical kit (rooms 1-5), and the believed goals of CommX (triage in room 1, room 5, or both). The commander is believed to always start in room 3. This yields 90 distinct cases to analyze. The resulting prediction of CommX’s plan is then compared with what we would expect a rational individual to do. However, in some scenarios there are multiple optimal plans that can be produced by different strategies. The first strategy,  $Opt_1$ , is where the individual favors picking up medkits towards the beginning of their plan (e.g. at their starting location), and the second,  $Opt_2$ , is where the individual favors picking up medkits toward the end of the plan (e.g. in the same room as the triage location).

The results of these simulation runs show that the robot successfully predicts which medical kit CommX will choose in 90 out of 90 cases (100.0% accuracy) if  $Opt_1$  is assumed. If  $Opt_2$  is assumed, the robot is successful in predicting 80 out of 90 cases correctly (88.9% accuracy). This demonstrates (for unestablished reasons) a bias in the planner for plans that comport with  $Opt_1$  behavior. Nonetheless, these results confirm that the mental modeling architecture can be successful in predicting the behavior of rational agents.

Robot Condition	Cases with no conflict: $Opt_1$	Cases with no conflict: $Opt_2$
Robot at <i>room2</i>	55.83%	47.50%
Robot at <i>room3</i>	25.0%	33.33%
Robot at <i>room3</i> w/ mental modeling	<b>100.0%</b>	<b>91.67%</b>

TABLE I

PERFORMANCE OF THE ROBOT WITH, AND WITHOUT, MENTAL MODELING CAPABILITIES.

Next, we evaluated the following question: *what does this mental modeling ability give us performance-wise?* We compared the medical kit selection task between a robot with and without mental modeling capabilities. The robot without the mental modeling capabilities still looks for a medkit but can no longer reason about the goals of CommX. We considered 120 cases: 20 combinations of medical kit locations where the two kits were in different locations (as this would be a trivial case)  $\times$  3 possible goal sets of CommX (as described above)  $\times$  2 sets of beliefs about medkit existence (as described above). To demonstrate the efficacy of the belief models, we also consider two different starting locations of the robot - we now include room 3 in addition to room 2 - as there would naturally be more selection conflicts to resolve if both the robot and CommX started in the same location. We calculated the number of cases in which the robot would successfully attempt to pick the medical kit not already taken by the human teammate first. The results are tabulated in Table I. As shown, the mental modeling capability leads to significant improvements over the baseline for avoiding potential resource conflicts.

### B. Plan Recognition

We considered two proof of concept scenarios to illustrate the usefulness of plan recognition: reactive, and proactive. In the *reactive* case, the robot only knows CommX's goal partially: it gets information about CommX having a new triage goal, but does not know that there already existed a triage goal on another location. In this case, by looking at the relative probabilities of all triage related goals, the robot is quickly able to identify which of the goals are likely based on incoming observations; and it reacts by deconflicting the medkit that it is going to pick up. In the *proactive* case, the robot knows CommX's initial state and goals exactly, but CommX now assumes that the robot will bring him a medkit without being explicitly asked to do so. In such cases, the robot can *adopt* the goal to pick up and take a medkit to CommX by recognizing that none of CommX's observed actions seem to be achieving that goal.

## VII. CONCLUSION

In this paper, we described a means of achieving coordination among different agents in a human-robot teaming scenario by integrating the belief modeling and automated planning components within a cognitive robotic architecture. Specifically, we used the planning component to predict teammate behavior by instantiating planning problems from a teammate's perspective. We described the formal representation of the beliefs and the planning models, and the

mapping of the former into the latter. We further discussed extensions to our current approach that utilize state-of-the-art plan recognition approaches. An evaluation of our integrated architecture's predictive capabilities was conducted using a PR2 robot, which showed that appropriate plans were produced for different sets of beliefs held by the robot. We also presented collated results from a simulation study that ranged over a wide variety of possible scenarios - these results confirmed that the mental modeling capabilities led to significant improvements in coordination behavior.

## VIII. ACKNOWLEDGEMENTS

This work was supported in part by the ARO grant W911NF-13-1-0023, the ONR grants N00014-13-1-0176 and N0014-13-1-0519, and the NSF grant #111323.

## REFERENCES

- [1] M. Scheutz, P. Schermerhorn, J. Kramer, and D. Anderson, "First Steps toward Natural Human-Like HRI," *Autonomous Robots*, vol. 22, no. 4, pp. 411–423, May 2007.
- [2] M. Scheutz, "Computational mechanisms for mental models in human-robot interaction," in *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments*. Springer, 2013, pp. 304–312.
- [3] G. Klein, P. J. Feltovich, J. M. Bradshaw, and D. D. Woods, "Common ground and coordination in joint activity," *Organizational simulation*, vol. 53, 2005.
- [4] H. H. Clark and S. E. Brennan, "Grounding in communication," *Perspectives on socially shared cognition*, vol. 13, no. 1991, pp. 127–149, 1991.
- [5] W. Y. Kwon and I. H. Suh, "A temporal bayesian network with application to design of a proactive robotic assistant," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3685–3690.
- [6] K. W. Strabala, M. K. Lee, A. D. Dragan, J. L. Forlizzi, S. Srinivasa, M. Cakmak, and V. Micelli, "Towards seamless human-robot handovers," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 112–132, 2013.
- [7] M. Scheutz, G. Briggs, R. Cantrell, E. Krause, T. Williams, and R. Veale, "Novel mechanisms for natural human-robot interactions in the diarc architecture," in *Proceedings of AAAI Workshop on Intelligent Robotic Systems*, 2013.
- [8] G. Briggs and M. Scheutz, "Facilitating mental modeling in collaborative human-robot interaction through adverbial cues," in *Proceedings of the SIGDIAL 2011 Conference*. Association for Computational Linguistics, 2011, pp. 239–247.
- [9] M. Ramirez and H. Geffner, "Probabilistic plan recognition using off-the-shelf classical planners," in *Proceedings of the 24th Conference on Artificial Intelligence*, 2010, pp. 1121–1126.
- [10] K. Talamadupula, D. E. Smith, and S. Kambhampati, "The Metrics Matter! On the Incompatibility of Different Flavors of Replanning," *arXiv preprint arXiv:1405.2883*, 2014.
- [11] K. Talamadupula, J. Benton, S. Kambhampati, P. Schermerhorn, and M. Scheutz, "Planning for human-robot teaming in open worlds," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 1, no. 2, p. 14, 2010.
- [12] M. B. Do and S. Kambhampati, "Sapa: A multi-objective metric temporal planner," *Journal of Artificial Intelligence Research*, vol. 20, no. 1, pp. 155–194, 2003.
- [13] M. Van Den Briel, R. Sanchez, M. B. Do, and S. Kambhampati, "Effective approaches for partial satisfaction (over-subscription) planning," in *AAAI*, 2004, pp. 562–569.
- [14] R. J. Firby, "An investigation into reactive planning in complex domains," in *AAAI*, vol. 87, 1987, pp. 202–206.
- [15] P. Schermerhorn, J. Benton, M. Scheutz, K. Talamadupula, and S. Kambhampati, "Finding and exploiting goal opportunities in real-time during plan execution," in *Intelligent Robots and Systems (IROS), 2009*. IEEE, 2009, pp. 3912–3917.
- [16] S. Richter, M. Helmert, and M. Westphal, "Landmarks revisited," in *AAAI*, vol. 8, 2008, pp. 975–982.