Architectural Mechanisms for Handling Human Instructions for Open-World Mixed-Initiative Team Tasks and Goals

Kartik Talamadupula KRTALAMAD@US.IBM.COM[1]
Deptartment of Computer Science and Engineering, Arizona State University, Tempe AZ USA

Gordon Briggs GORDON.BRIGGS.CTR@NRL.NAVY.MIL[2]
Matthias Scheutz MATTHIAS.SCHEUTZ@TUFTS.EDU
HRI Laboratory, Tufts University, Medford MA USA

Subbarao Kambhampati RAO@ASU.EDU
Deptartment of Computer Science and Engineering, Arizona State University, Tempe AZ USA

Abstract

One important challenge for cognitive systems research is to develop an integrated architecture that can enable effective, natural human-robot interactions in open worlds where new concepts, entities, and actions can be introduced through natural language during task performance. In this paper, we claim that in order to allow for such open-world tasking in natural language, all components in the robotic architecture that processes and executes human instructions require mechanisms for learning new information and applying it immediately. We focus on two aspects of open worlds – new goals and new objects – and describe the architectural machinery required to handle them: from representations and processing schemes for human utterances to open world quantified goals that involve novel objects introduced during task execution. We then present a proof-of-concept demonstration of these mechanisms implemented in the DIARC architecture on an autonomous robot and show in simulated scenarios the necessity of mechanisms for open-world tasking.

1. Introduction

Future mixed-initiative human-robot teams require increasingly autonomous and capable robotic teammates. Consider, for example, a typical *Urban Search and Rescue* (USAR) mission that involves

> "[...] the location, rescue (extrication), and initial medical stabilization of victims trapped in confined spaces [...] as it may be needed for a variety of emergencies or disasters, including earthquakes, hurricanes, typhoons, storms and tornadoes, floods, dam failures, technological accidents, terrorist activities, and hazardous materials releases."[3]

In such a scenario, a team of searchers and rescuers is dispatched to "conduct physical search and rescue in collapsed buildings" and "provide emergency medical care to trapped victims", among

1. Kartik Talamadupula is currently at IBM's T. J. Watson Research Laboratory in Yorktown Heights, NY.
2. Gordon Briggs is currently at the Naval Research Laboratory in Washington, D.C.
3. See `http://www.fema.gov/urban-search-rescue`.

others. Since USAR is considered a "multi-hazard discipline", mixed-initiative human-robot teams could substantially improve the effectiveness of such missions by searching spaces that are inaccessible to humans and reducing the risk of humans getting trapped in collapsing buildings.

Currently employed teleoperated robots, however, are often not appropriate for such missions. Operating them is not possible due to a number of reasons, including but not limited to: (a) wireless connectivity problems or lack of accurate sensory information; (b) slow and ineffective operation that is not suitable given the urgency of the task; or (c) tele-operation locks in human resources that could be better used in other ways. Ultimately, we want *autonomous robots* that complement human teammates and serve as genuine helpers in USAR missions (and beyond in other human-robot team tasks).

Building such autonomous robotic helpers, however, is a very difficult endeavor for many reasons. Aside from the mechanical and control problems that must be resolved for robots to function properly in such environments, there is a critical feature of USAR missions, shared with many other human team tasks, that presents a major challenge: *the open-endedness of the mission*. This includes the many aspects of the mission that are not known in advance including goals, tasks, and subtasks, locations of humans and objects, and building layouts. While humans can handle such open-ended missions by negotiating novel or unknown aspects in natural language, current robotic architectures are not yet capable of dealing with such unknown and novel aspects in the same way. In part, the problem is that natural language capabilities are lacking in current robots. However, equally important are architectural mechanisms that are required in other components – such as task planners or inference modules – to cope with open-ended missions.

The main claim of this paper is that for open-world tasking in natural language, *all components in a cognitive architecture involved in executing human instructions require mechanisms for learning new information and applying it immediately*. This claim implies that the natural language system must deal with all aspects of new words (e.g., type names for types of perceivable objects), the planning system must deal with new goals involving the concepts denoted by these words, and the perceptual and action systems must apply the new concepts to interpret perceptions and perform actions based on them. Without such mechanisms, robots will either be unable to learn new information or fail to apply that information dynamically during task execution.

We cannot cover all aspects of natural language instructions in a single paper, so we focus on two critical aspects of open-ended missions that cognitive architectures must handle: (A1) *not all goals, tasks, and subtasks are known ahead of time, and new goals may be assigned and new subtasks may be defined during task performance*; and (A2) *not all information about task-relevant entities is available ahead of time, and new knowledge about unknown objects must be acquired during task performance*, including knowledge about objects and their appearance, locations, people, and activities. To keep the architectural extensions tractable, we make three simplifying assumptions: (1) front-end and back-end natural language processing system (speech recognizer and parsers, text generator, and speech synthesizer) have all necessary knowledge for processing novel words; (2) the vision system can detect all parts of novel objects used in natural language definitions; and (3) the action system already knows how to manipulate unknown objects. In a fully open-world system, it would have to acquire this knowledge as well.

This paper's main contributions are the representations and processes required to go from spoken human utterances and instructions to formal representations of *open world quantified goals* that enable the handling of both aspects A1 and A2 (Talamadupula et al., 2010a). We have defined these structures and demonstrated their use in an integrated robotic system previously (Talamadupula et al., 2010b), but this paper is the first to present the detailed representations and processes required to gather necessary information to specify them in open-ended missions, as well as the detailed semantics of such goals. Specifically, we distinguish between existentally and universally quantified open-world goals and provide a formal semantics for them both. Moreover, we perform a broader evaluation of the integrated system, reporting both demonstrations on an autonomous robot and on additional case studies in simulation.

We begin the paper with background on human-robot teaming in open worlds and an overview of related work with respect to aspects A1 and A2 of open-ended missions. Next, we present motivating examples of human utterances and instructions to a robotic helper, followed by a description of the mechanisms needed for understanding such instructions. We then focus on the functional requirements on natural language that extracts novel goals and tasks, as well as information about task-relevant unknown objects, from dialogues with human instructors. Subsequently, we describe additional mechanisms and capabilities required for the robot to pursue new goals and tasks involving new objects. These include detailed descriptions of the requirements imposed on the task planner by open-world scenarios. We present an evaluation of our integrated system on a robotic platform and in simulation, and then summarize our accomplishments and discuss future work.

2. Background and Related Work

An important part of human-robot interaction is to let the robot receive and react to new information from a human commander (e.g., a new goal to perform a novel task). Thus, robots operating in teaming scenarios must be able to plan and revise a course of action in response to human instructions, possibly acquiring new information about novel objects and tasks.

Most prior work has focused on letting users specify and update goals for autonomous systems during task performance for goal types that are known in advance. For example, Bagchi et al. (1996) presented a system for controlling service robots equipped to handle the user's changing goals and advice at different levels of detail via a planner that can refine and modify them dynamically. There has also been research on streamlining the process of accepting human input to planners under the title of advisable planning (Myers, 1996, 1998), which lets humans specify partial plans, recommendations, or methods of evaluating plan quality in natural language. Of particular relevance to our project is the interactive TRAINS-95 system (Ferguson, Allen, & Miller, 1996), which uses natural language dialogue to elicit high-level advice from humans.

However, goals that involve novel entities not known beforehand are different from goals that only involve known entities: the new entity types must be acquired during task execution and the goal must be formulated in a way that does not make assumptions about specific properties of the objects, or the objects' relation to the rest of the world. Moreover, the natural language system must be able to handle novel linguistic expressions whose syntactic and semantic aspects are unknown. In this section, we briefly review past work on natural language understanding and planning relevant to such open-world human-robot interaction.

2.1 Natural Language Instructions in Open Worlds

Collaborative natural language interactions involve challenges of mutual understanding at multiple levels (Clark, 1996). These include lower-level concerns such as joint attention (Rich et al., 2010) and speech recognition (Gomez et al., 2012), which are necessary capabilities to have even a rudimentary interaction. In the context of this paper, however, we are primarily concerned with the challenges of *semantic* and *intentional* understanding in a collaborative setting (i.e., what was literally meant and what was proposed, respectively). One of the key challenges in semantic understanding is the ability to resolve references in utterances and ground them in physical and perceivable entities in the real world. Prior work on this challenge in dialogue-enabled architectures for human-robot interaction has utilized capabilities such as probabilistic-reasoning (Kruijff et al., 2010) and perspective taking (Lemaignan et al., 2012).

Most work, however, either focuses on closed worlds, or allows for learning of novel entities under the assumption that they are perceivable in the immediate environment. The latter case begins to addresses the A1 and A2 aspects of open-ended missions, but falls short of providing a full solution. The assumption that referrents are physically collocated obviates the need for explicit, high-level goals to gather information about these objects. Rather the perceptual subsystems of the robot are given implicit goals to immediately detect novel objects. Recent work has proposed an alternative way of specifying perceptual features of novel entities entirely in natural language, thus avoiding the need for entities to co-exist (e.g., Krause et al., 2014).

2.2 Planning in Open Worlds

The use of the closed-world assumption in conjunction with planning has been considered previously, notably in (Etzioni, Golden, & Weld, 1997) via the specification of *local closed-world* (LCW) statements. The representation used in that work, of closing a world that is open otherwise via the local closed-world statements, is complementary to the representation that is used here. The approach in this work is to provide support for open world quantified goals by *relaxing* the planner's assumption of a world closed with respect to object creation; that is, parts of a completely closed-world are being *opened*. This approach provides a method of specifying *conditional goals*, where goal existence hinges upon the truth value of facts. Semantics of goals involving sensing have received attention in Scherl and Levesque (1993) and Golden and Weld (1996). The latter work is particularly relevant as it considers representations that leads to tractable planning, proposing three annotations to specify goals involving sensing. Additional relevant work has been done on representations that can be used to specify more complex interactions between task-relevant entities and goals in an open world – specifically, temporally extended goals (Baral, Kreinovich, & Trejo, 2001; Bacchus & Kabanza, 1998) and goals with trajectory constraints (Gerevini et al., 2009). Finally, the goal reasoning community has also looked at the problem of goal generation, both in a general setting (Choi, 2011) as well as in the specific contexts of strategy simulation (Klenk, Molineaux, & Aha, 2013) and disaster relief tasks (Roberts et al., 2015).

3. Instructions from Human Utterances

Consider a robot that is carrying out its assigned tasks during a larger USAR mission when a human (H) contacts the robot (R) via a wireless audio transmitter:

```
H: Commander Z really needs a medical kit.
```

Now suppose that the robot did not know about `Commander Z` and consequently did not know about `Commander Z`'s needs. And further assume that the robot also does not know what a medical kit is or what it looks like, and consequently does not know whether and where to find one. The challenge then is to process the utterance in a way such that the robotic agent can:

1. assume that Z is the name of a commander and infer that `Commander Z` is a human person (e.g., because it knows that all commanders are human);
2. infer from the fact that the commander has a need to have something, that the commander *might also have a goal* to have something (this is often true, but not always, just think of "the need for a break", which does not imply "the goal to have a break");
3. further infer that it might have to have the goal for the commander to have something (based on the obligation that $\mathbf{O}\forall x.G(h, have(x)) \rightarrow G(R, G(h, have(x)))$ — here "**O**" denotes the standard deontic operator "obligatory" and "G(x,y)" indicates that x has goal y);
4. infer that Z's need (and thus Z's goal) is urgent (based on the use of "really" before "needs").

All the above inferences are possible without knowing what a medical kit is, based solely on the general knowledge the robot has about human commanders, the probabilistic rule that needs of people sometimes imply their goals, and the obligation that robots have to adopt goals of human commanders; and natural language semantics in the case of the modifier "really".[4] Fortunately for the robot, the human follows up right away with another sentence:

```
H:  There should be one in the room at the end of the hallway.
```

After resolving the anaphora (namely "one" referring to "medical kit"), the robot can now make an important inference about the medical kit: it is a *concrete physical object*. Because even though the robot might not know where the room at the end of the hallway is, the fact that it is a room and that a medical kit *should be* inside the room is sufficient for the inference (note that it also uses the principle that "should" implies "could", which is all that is needed to establish the precondition for being a physical object $\forall x \exists y.located(x, y) \wedge location(y) \rightarrow pobject(x)$.). Moreover, the robot can infer probabilistically that the object can be carried because it is located inside a room (based on the probabilistic common-sense knowledge that all things inside rooms can be carried). This lets it make the further inference that if it has the goal for Z to have it ($G(R, have(Z, medkit))$), it should then likely also have the goal to get it and deliver it to Z ($G(R, deliver(R, Z, medkit))$). This is based on a probabilistic "helping" principle that requires robots to bring items needed by humans:

$$\forall robots(r), humans(h), pobjects(x).G(r, have(h, x)) \wedge transportbl(x) \rightarrow G(r, deliver(r, h, x))$$

At this point, the robot, using a mixture of non-probabilistic and probabilistic principles, arrives at the conclusion that it might have to have a new delivery goal of a physical object of type "medical kit" to `Commander Z`. Since the robot cannot be sure that it should have this goal (as the probabilistic inference lowered its confidence in the validity of the conclusion), it is seeking clarification from the human:

```
R: OK, should I get it for him?
H: Yes.
```

4. Note that it is not possible to infer anything about the medical kit: such as that it is a physical object, that it can be picked up, or that it contains medical equipment (just substitute "vacation" for "medical kit" in the above sentence).

Now that the robot's new goal has been confirmed, three new questions arise and must be answered before the robot is able to successfully complete the goal:

1. *What does a medical kit look like?* The answer to this question is essential for the robot to be able search for it.

2. *Where is the room at the end of the hallway?* This question does not necessarily have to be answered by the human, as long as the robot can devise a strategy to find it given it is currently located in a room that leads into the hallway. Williams et al. (2013) provide a detailed description of the required capabilities.

3. *Where is* `Commander Z` *located?* This question is also not of immediate importance, but it must be answered at some point, say once the robot has found and retrieved a medical kit.

Hence, the robot both acknowledges the human confirmation and follows up with a question about the medkit's appearance. Note that appearance is all the robot must know about a medkit, as further questions about its purpose are not relevant to achieve the goal:

```
R: OK, I'll get one for him.
R: What does it look like?
H: It is a white box with a red cross on it.
```

The robot must trust that the verbal description provided by the human is sufficient to search for medical kits. Hence, it configures its vision system based on that description, acknowledges again the new information, and provides additional confirmation that it has accepted the goal and is starting to pursue it right away.

This seemingly simple dialogue exchange demonstrates how both aspects (A1) and (A2) can arise naturally in the context of open-world tasks, which are quite complex and complicated in terms of the requirements they impose on the robotic architecture. We will next describe how the above functionality can be accomplished in an integrated cognitive robotic architecture, for which we have used DIARC (Scheutz et al., 2007, 2013). We will specifically focus on the representational and functional capabilities needed to understand the instructions, carry out the dialogue, and configure its components in ways that let it successfully pursue and accomplish the goal if the environmental circumstances are right. In other words, there is a medkit in the room at the end of the hallway, the robot can determine the whereabouts of `Commander Z` and so forth.

4. Open-World Instruction Understanding

The work presented in this paper is the first to successfully integrate approaches that address both the A1 and A2 (see Section 1) challenges within an integrated framework in the context of open-world mixed-initiative (i.e., that can involve initiatives from both a human as well as a robotic agent) teaming tasks. Table 1 presents a list of the extensions that must be made to DIARC's architectural components to enable this integration. In this section, we consider the examples from the previous section, which demonstrate several challenges that a natural language understanding system must address to handle open-world instruction. Aside from the obvious challenges of having to cope with new words – out-of-vocabulary speech recognition and estimating the part-of-speech tag – the

Table 1. A table showing the extensions that need to be made to each component in the DIACR architecture.

Architectural Component	*Necessary Extensions*
Natural Language and Dialogue	Processes linguistic cues that allow the autonomous agent to recognize situations where closed-world reasoning is insufficient, and generates requests to seek information from the human.
Belief Reasoner	Contains rules that let the autonomous agent infer possible goals from the goal and belief states of other agents. Additionally needs to recognize and differentiate between closed-world and open-world goals to undertake the appropriate goal submission process.
Goal Manager	Represents and provides information about goals in the open-world to the planner component.
Planner	Uses information on both universally and existentially quantified open-world goals to generate plans.

system must also deal with the lack of semantic and pragmatic knowledge when trying to make sense of utterances. In the first sentence "Commander Z really needs a medical kit" the lexical items "Z", "medical", and "kit" are unknown, as are their grammatical types; "Z" could be a proper name or an adverb like "now", and "medical" and "kit" could both be nouns, adjectives, or adverbs. Hence, based on the lexical ambiguities, it is impossible to guess a semantic type, let alone the meaning. As discussed before, the assumption that "Z" is a name and that the words "medical kit" denote an object, resolve only some of the problems, because the robot must extract the implicit order expressed in the sentence "Get `Commander Z` a medical kit", to turn this into goal expressions the task planner can handle. The tricky part is that the planner does not know about medical kits (as objects) and, even if it did know about them, it would not know where to find one. A simple "$G(have(Z, medkit)$" is not appropriate because "medkit" does not denote an object, but a type. Pulling out the type and quantifying it as in "$G(\exists x.have(Z,x) \wedge medkit(x))$." does not work either, because the type is unknown and the goal is not that there "be a medkit such that Z has it", but that Z has one of that kind. No straightforward way of translating this into a two-place goal expression will succeed.

Moreover, understanding the next sentence and connecting it to the previous sentence is critical: "There should be one in the room at the end of the hallway". The fact that "should be" is used instead of "is" is important for building the appropriate semantic representation: in the case of "is" it would be easy to assert the fact

$$\exists medkit(x)\, \text{locatedin}(x, \text{room-at-the-end-of-the-hallway}),$$

but "should" indicates that the assertion is not certain. Hence, forming a goal to get one from the room is not the right way to interpret this information, since it has a conditional flavor and would (in conjunction with the first sentence) yield something like "if there is a medkit in the room at the end of the hallway, then get it and bring it to commander Z". By viewing the two sentences as specifying some sort of conditional goals, it seems more plausible that the planner could make sense of them and generate a sequence of actions to accomplish them: first go to the room at the end of

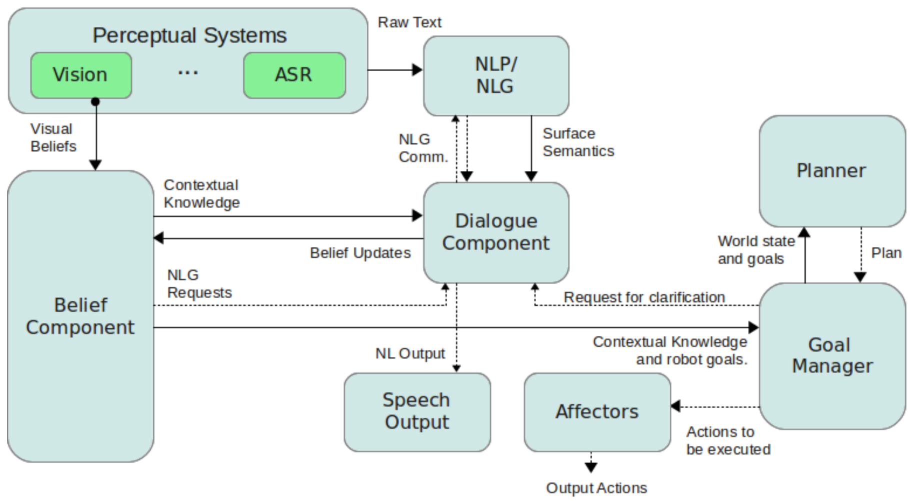

Figure 1. A schematic that shows the architecture of the `DIARC` integrated system.

the hallway, then look for the medical kit in the room, and if one is found, pick it up and bring it to `Commander Z`.

4.1 Dialogue Reasoning

Within DIARC, goals are generated from various components and submitted to the Goal Manager to be executed (Schermerhorn & Scheutz, 2010). Goals prompted by natural language interactions are first generated within the Belief component based on belief updates received from the natural language system, and then forwarded to the Goal Manager. Goals are represented within the Belief component as predicates of the form $goal(\alpha, \phi, P)$, where α denotes the agent, ϕ represents the state α wants to obtain or the action α wants performed, and where P denotes the urgency of the goal. Below we describe in greater detail how natural language understanding obtains this information and how it is then utilized to generate a goal for the robotic system.

In the scenario presented in Section 3, the robot receives information, via natural language input from C_X, regarding the goal of another agent (C_Z). Figure 1 captures the flow of information within the robotic architecture that originates from speech input. First, the speech recognition component generates the text of the heard utterance, which is then forwarded to natural language processing. Parsing and initial semantic analysis are then performed on this received text data. The resulting surface semantics are then forwarded to the dialogue component for pragmatic analysis.

Within the dialogue component, a series of pragmatic rulestranslate between surface semantics understood by natural language processing and the updates sent to the belief component. A pragmatic rule in the dialogue component takes the form

$$[[UtteranceType(\alpha, \beta, \phi, M)]]_C := \psi_1 \wedge ... \wedge \psi_n,$$

where the $UtteranceType$ denotes a category of speech act (e.g., instruction, statement, acknowledgment), α refers to the speaker, β denotes the listening agent, ϕ specifies the surface semantics of the utterance, M specifies a set of sentential modifiers, and the set of predicates $\psi_1 \wedge ... \wedge \psi_n$ describe the inferred meaning of the utterance. The ability to model sentential modifiers has been used previously to understand the belief model implications of certain adverbial modifiers such as "still" and "now" (Briggs & Scheutz, 2011). Finally, the $[[\,]]_C$ notation indicates the dialogue and belief context that must apply for this interpretation.

For the the statement, "`Commander Z` needs a medical kit," the system would apply the pragmatic rule

$$Stmt(\alpha, \beta, needs(\gamma, X), \{\}) := want(\alpha, bel(\beta, goal(\gamma, have(\gamma, X), normal))),$$

which leads the robot to infer that agent α wants the listener β to believe agent γ has a goal with default urgency to have object X. We omit the $[[\,]]_C$ notation here as this rule will apply in general. The modifier notation introduced earlier also lets one infer information about goal urgency. For instance, if the robot heard "`Commander Z` **really** needs the medical kit", this should indicate increased urgency, which can be represented in a new pragmatic rule

$$Stmt(\alpha, \beta, needs(\gamma, X), \{really\}) := want(\alpha, bel(\beta, goal(\gamma, have(\gamma, X), high))).$$

While the belief update generated by this rule helps the robot maintain a mental model of C_X, a few reasoning steps must occur within the belief component before this information generates a goal for the robot. We describe these rules in the subsequent section.

4.2 Belief Reasoning

Having recently received the belief update

$$want(cmdrX, bel(self, goal(cmdrZ, have(cmdrZ, medkit), high))),$$

the belief component makes various inferences. To adopt beliefs based on communicated facts, a naive rule encodes a credulous belief adoption policy

$$want(\alpha, bel(self, X)) \Rightarrow bel(self, X).$$

This supports the belief $bel(self, goal(cmdrZ, have(cmdrZ, medkit), high))$. The belief component also contains basic rules that reason about social roles and possible obligations. For example, different team roles and ranks can be specified for a hierarchical team

$$commander(\alpha) \wedge crewman(\beta) \Rightarrow outranks(\alpha, \beta).$$

The above rule represents the superior / subordinate relationship between a commander (such as C_X and C_Z) and someone of rank "crewman," such as our robot. Moreover, knowledge about obligations connected to differences in rank can be captured. The rule

$$outranks(\alpha, \beta) \wedge bel(\beta, goal(\alpha, \phi, P)) \Rightarrow itk(\beta, goal(\beta, \phi, P))$$

encodes the notion that if an agent β believes that a superior has a goal for ϕ to obtain, it should have an intention-to-know (itk) whether it should also adopt that goal. Because the robot has the necessary rank knowledge (i.e., $commander(cmdrZ) \wedge crewman(self)$), this rule fires, generating the intention-to-know whether or not it should help `Commander Z` get a medical kit. This results in a clarification question toward C_X, "Ok, should I get it for `Commander Z`?"

The response by C_X, "Yes." triggers a contextually dependent dialogue rule that confirms the content of the itk predicate. As such, $goal(self, have(cmdrZ, medkit), high)$, is supported. Ordinarily, this newly asserted goal predicate would be submitted to the Goal Manager. However, due to the uncertain state of a key object in this potential goal, specifically that the medical kit "should be" at the current room ($should(at(medkit, \text{current-room}))$), it is not treated a a regular goal. Instead, the belief component submits it as a special type of goal described in the next section.

5. Open-World Planning

In order to parse and act upon goals that are conditional in nature, the planning system cannot simply assume a closed world (Etzioni et al. (1997)) with respect to unknown information. Instead we need a framework for specifying conditional knowledge and an approach for using that knowledge to trade sensing costs and goal utilities. Accordingly, we use an approach that provides a compact way to specify conditional opportunities over an "open" set of objects.

5.1 Representing Open-World Quantified Goals

We now introduce a construct that can be used to specify open-world planning knowledge as specified above. Open-world quantified goals (Talamadupula et al., 2010a) combine information about objects that may arise during execution with goals that are contingent on their discovery. The human member of a human-robot team can specify new objects that sensing may reveal, and state goals that can be achieved using those objects. For example, detecting a medical kit (medkit) in a room will let the robot pick it up and deliver it to another location. Quantification is introduced because the goal is on each medkit picked up and delivered, but not all medkits have to be picked up. As an example, recall the USAR scenario introduced in Section 3: a human directs the robot to pick up a medkit from the *room at the end of the hallway* and deliver it to `Commander Z`. This goal can be classified as open world, since it references objects that do not exist yet in the planner's object database. It is also *quantified*, but the nature of this quantification is such that the robot does not have to find and pick up *all* medkits. To handle cases like these, we rely on the partial-satisfaction capability (van den Briel et al., 2004) of the base planner, Sapa Replan (Cushing, Benton, & Kambhampati, 2008), which uses the algorithm defined in (Benton, Do, & Kambhampati, 2009) for finding the best plan. By using partial-satisfaction planning, we enable the planner to pursue only a subset of the assigned goals, rather than forcing it to achieve all of them. We discuss the issues of quantified goals in detail in the following paragraphs. For a full description of the syntax and semantics of open world quantified goals, please see Talamadupula (2014).

5.1.1 Universally Quantified Goals

The first case involves a goal that is quantified universally (Golden, Etzioni, & Weld, 1994) – that is, quantified over all possible instances of a particular object type. Consider a directive from a person to the robot:

```
Medical kits can be found inside rooms. They are white in color
with a red cross on them. Find all medical kits if possible.
```

This goal can be written formally as:

```
1 (:open
2   (forall ?r - room
3     (sense ?mk - medkit
4      (lookedfor ?mk ?r)
5       (and (hascolor ?mk white)
6            (hassymbol ?mk redcross))
7   (:goal (found ?mk ?r)
8            [100] - soft))))
```

Line 1 indicates that this is an open world-quantified goal, whereas Line 2 introduces the variable over which the goal is quantified – in this case, over all objects of type room. Line 3 contains the object type that the robot must sense for; this is the run-time discovery that gives the world its open nature. Line 4 is a *closure condition* that informs the planner that a sensing action occurred, thus stopping repeated sensing. Lines 5 and 6 list the properties that hold for the object that is sensed, where these properties are generated from information provided via the dialogue rules in Section 4. Finally, Line 7 describes the goal over such an object, while line 8 indicates that there is a utility value of 100 units associated with fulfilling that goal. It additionally specifies that the goal is *soft*, in that it is an opportunity, and not something that must necessarily be fulfilled.

5.1.2 Existentially Quantified Goals

In contrast to universally quantified goals, there may exist goals that depend on objects that are not yet known, but for which there is only one instance. Consider an utterance by a different person:

> `Commander Z needs a medical kit. There is one in the room at the end of the hallway.`

This goal is fundamentally different from the one presented in Section 5.1.1; in this instance, the human is specifying that there is exactly *one* medical kit for the robot to locate and transport to `Commander Z`. Even though this is still an open-world goal – given that the planner does not know about this object until it discovers it at run time – it need not look into all rooms to find the medical kit. We can model such existentially quantified open-world goals using the same construct as in the previous section; we merely restrict the type of the variable that the goal is quantified over. We are then left with:

```
1 (:open
2   (forall ?r - endroom
3     (sense ?mk - medkit
4      (lookedfor ?mk ?r)
5       (and (hascolor ?mk white)
6            (hassymbol ?mk redcross))
7   (:goal (found ?mk ?r)
8            [100] - soft))))
```

The only difference between this goal and the previous one occurs in line 2; the existentially quantified goal is compiled into a universally quantified one by restricting the variable that quantification occurs over from type `room` to the narrower `endroom` subtype. We should emphasize that this is an approximation that lets the planner handle existentially quantified goals in the same manner as universally quantified ones, and depends on the assumption that there will only be one object of type `endroom`. We are currently working on a more principled approach to handle such situations.

5.2 Processing Open-World Quantified Goals

In this section, we describe the processing and management of the goals in the architecture. In the human-robot interaction scenario in Section 4.2, we left off after the human operator instructed the robot to get a medical kit for `Commander Z`, but before the goal had been submitted from the Belief component to the Goal Manager component. In this section, we describe how open-world quantified goals are submitted to the Goal Manager, how they are communicated to the planner, and how the planner uses them.

5.2.1 Goal Submission and Management

As mentioned previously, if a goal predicate of the form $goal(self, X, P)$ can be handled by the belief component, it would be submitted as a regular goal to the goal manager. However, certain rules lead it to treat goals as open world. For instance, the inference rule

$$goal(\alpha, have(\beta, X), P) \wedge should(at(X, L)) \Rightarrow \texttt{OWQG}(\alpha, have(\beta, X), P)$$

(where `OWQG` stands for an open-world quantified goal) means that if the location of the object X is uncertain, then an open-world goal should be generated. The goal submission mechanism checks whether checks whether this holds, in which case it supplies the Goal Manager with information about what variables are open and what sensory information, and sensory actions, should support inferences about this state.

In our scenario, the location of the medical kit is not known. Therefore, we use a rule to specify that L is the open variable associated with the open-world goal to obtain the medical kit for `Commander Z`:

$$goal(\alpha, have(\beta, X), P) \wedge should(at(X, L)) \ \Rightarrow \texttt{OWQG}\_openvar(\texttt{OWQG}(\alpha, have(\beta, X), P), L)$$

Similar rules specify that the object to be sensed for is X. The information inferred by these rules is submitted with the goal to the Goal Manager, which attempts to submit it to the planner. However, there is still a problem: how can the robot act on a goal to look for the medkit without knowing what it looks like? The robot begins the scenario without a visual description of objects of the medical kit type. Somehow the knowledge of the medkit's appearance must be included in the sensing action the robot performs. The Goal Manager detects that it lacks a visual descriptor for the sense variable type and formulates a request for clarification (specifically for a visual description). It submits this request to the dialogue component, which initiates natural language generation, resulting in the query, "What does it look like?". This produces feedback from the human interactant, who supplies the visual description. With this information available, the Goal Manager then submits the open-world goal to the planner.

5.2.2 Plan Generation

To handle open world quantified goals, the planner *grounds* the problem into the closed-world using a process similar to Skolemization. More specifically, the planner generates *runtime objects* (from the sensed variable, which in this case is the `medkit`) that represent explicitly the potential existence of an object to be sensed, which it marks explicitly as system-generated runtime objects. The planner adds these objects to the problem and to the closure condition. The new facts generated as a result of adding the object are appended to the problem instance via the planner's state update process. The goals generated on runtime objects are similarly added. This process repeats for every new object that may instantiate the universally quantified goals.

The planner then treats the closure condition *optimistically*, meaning it considers a particular state of the world closed once the ground closure condition is true. On every update from the world, it checks the ground closure condition and, if true, it removes the runtime objects and goals instantiated on those objects. By operating over this representation, the planner provides a plan that is executable given its current representation of the world until new information is discovered via a sensing action returning the closure condition. The system interleaves planning and execution in a manner that moves the robot towards goals by generating an optimistic view of the true world state.

As an example, consider the scenario at hand and its open-world quantified goal. When the robot finds a room at the end of the hallway (an object with name `er1` of type `endroom`) the planner generates a runtime object `medkit!1`. Subsequently, it generates the facts `(hascolor medkit!1 white)` and `(hassymbol medkit!1 redcross)`, along with the goal `(found medkit!1 er1)` (with accompanying utility of 100 units) and adds them to the problem,[5] also creating a closure condition `(lookedfor medkit!1 er1)`. When the planner receives an update from the world that includes this condition as one of the facts, it updates the problem by deleting the two facts related to the runtime object and the goal. The planner only outputs a plan up to and including an action that will make the closure condition true; once it becomes true, the runtime object and facts are no longer needed since truth values in the real world are known. This process of addressing uncertainty in a dynamic world with a combination of sensing and replanning whenever there are state updates is reminiscent of *FF-Replan* (Yoon, Fern, & Givan, 2007).

6. Evaluation

We now describe the comprehensive evaluation of our work, which consists of two parts. First, we provide a proof-of-concept demonstration of the integrated architecture on a physical robot showing that it carries out instructions involving open-world quantified goals as intended. This shows both that the integrated system is operational and that it can be used in real-time human-robot interaction contexts. Second, we provide results from case studies that examine different ways of formulating goals. Each case considers the absence of crucial pieces of information needed to construct an open-world quantified goal. This is to show that the planner cannot produce desirable plans in the absence of all the required components as set forth in this paper. Overall, the two-fold evaluation scheme serves to demonstrate the complete behavior of the system in scenarios with open-world tasking, first by showing the whole system at work and then by examining component functionality.

6.1 Proof-of-Concept Robot Demonstration

The architectural extensions described earlier (in Section 4 and Section 5) were implemented in the respective components of our `DIARC` architecture (see Figure 1): the NLP/NlG and the Dialogue components, the Belief component, the Goal Manager component, and the Planner component, thus substantiating our main claim that all components involved in processing and executing human instructions require mechanisms for learning new information and applying it immediately (Section 1). We then used DIARC to control an autonomous MDS robot in an urban search and rescue

5. The exclamation mark indicates a runtime object.

task. We made five knowledge-based assumptions about the initial architectural configuration: The vision component had box and cross object and texture detectors, respectively, as well as an "on" relation detector, so that it could be configured to detect objects like the medkit decomposable into primitive objects and their mereological relations (Krause et al., 2014). The speech recognizer and synthesizer contained the word "medkit". The action system contained a "pick-up" action for objects with handles and, when instructed to pick up the medkit, it used the vision system to find a handle-like part to plan a trajectory for the robot's hand through the handle's opening using gradient-descent search. The Dialogue and Belief components contained pragmatic and inference rules as described in Sections 4.1 and 4.2, respectively. The Planner component included an action model based on the robot's available percepts obtained from the vision system and available actions in the Goal Manager.

We then put the MDS robot in the scenario described in Section 3, where it had to detect and understand instructions from its human teammate:

```
H: Cindy, Commander Z really needs a medical kit.
   There should be one in the room you are in.
R: Okay. Understood. The commander really needs a medkit.
   Should I get one for him?
H: Yes. He is in the room with the green door.
R: Okay. I will get one for him.
R: What does it look like?
H: It is a white box with a red cross on it.
R: Okay.
```

Note that the robot first must infer that it should have a goal to deliver a `medkit` to the commander from the information that `Commander Z` *needs* one. After arriving at this conclusion based on inferences described in Section 4, it requests information necessary to carry out the task. Specifically, the robot engages the human in a task-based dialogue about the details of (1) where the commander is, (2) where the `medkit` may be, and (3) what the `medkit` looks like. This information is used to assemble all parts of the open-world goal representation used in planning as described in Section 5. Throughout the dialogue interaction, the robot verbally acknowledeges instructions and ultimately issues confirmation to the teammate that it is now engaged in the task.[6] The demonstration worked seven out of ten times.[7] If we restrict the performance evalution to properly carrying out the above dialogue and generating the appropriate goals in cases with no speech recognition errors, the system has a 100% success rate.

6.2 Case Studies

To further evaluate the main claim of this paper (see Section 1), we present a planner-centric simulated evaluation that examines whether the system can produce plans in the absence of key compo-

6. A full video of this interaction in real time is available at `http://www.youtube.com/watch?v=RJ1VSIi1CM4`.
7. This is a very high task success rate given the uncertainties on the speech recognition and perceptual side, as well as the challenges on the actuation side). The "pick-up" action was especially tricky since moving the robot's hand in front of its Swiss ranger distance and eye-based camera sensors partially occluded the medkit; hence open-loop control had to be used for picking it up, which is error prone given that the robot's torso is mounted on a self-balancing Segway platform.

nents of open-world quantified goals and the nature of those plans. The aim was to demonstrate the importance of each component in allowing for open-world tasking through natural language, and the failure of the system to carry out such instructions in their absence.

6.2.1 Urban Search and Rescue

We again use an Urban Search and Rescue domain and consider the open-world quantified goal representation introduced in Section 5.1.1 through the rest of this section:

```
(:open
  (forall ?r - room
    (sense ?mk - medkit
     (lookedfor ?mk ?r)
      (and (hascolor ?mk white)
           (hassymbol ?mk redcross))
  (:goal (found ?mk ?r)
           [100] - soft))))
```

We examine different types of instructions that can be formally captured in the goal formalism and have different effects. For instance, consider the instruction:
Case 1: *"Medical kits can be found inside rooms."*

```
(:open
  (forall ?r - room
    (sense ?mk - medkit)))
```

In this case, the planner has information regarding the location of medical kits, but no goal on obtaining them. Since the medical kits are not mentioned in any of the planner's assigned goals, the information about where medkits can be found does not change the plan produced. Such a representation can, for example, be used for making predictions of what the robot would expect to see if it were to look around inside of rooms.

Case 2: *"Commander Z really needs a medical kit. Medical kits can be found inside rooms."*

```
(:open
  (forall ?r - room
    (sense ?mk - medkit
     (lookedfor ?mk ?r)
       (:goal (found ?mk ?r)))))
```

In this case, in addition to the information about medical kits possibly being in rooms, we gave the planner a goal for finding those medical kits in order to get them to Commander Z. However, the system can only make a plan if it already knows what medical kits look like, or what properties they possess. Otherwise, it is unable to produce *any* satisfactory plan, since the goal cannot be fulfilled without further perceptual information about medkits. The failure to produce a plan could then be used by the system to inquire about the missing information explicitly (e.g., "What does a medkit look like?").

Case 3: *"Commander Z really needs a medical kit. Medical kits can be found inside rooms. They are white in color and have a red cross on them."*

```
(:open
  (forall ?r - room
    (sense ?mk - medkit
     (lookedfor ?mk ?r)
      (and (hascolor ?mk white)
           (hassymbol ?mk redcross))
  (:goal (found ?mk ?r)))))
```

In this case, the planner has all the information required: the specification of the goal, a description of the object, and where to find it. However, the goal assigned to the planner in this case is a *hard* goal, meaning that it must be fulfilled under all circumstances. As a result, the planner is obligated to produce a plan that duly sends the robot inside each room that is discovered. However, this may not be feasible for resource-related reasons, and it also increases the overall cost of the plan without an assurance of increased benefit.

Case 4: *"Commander Z really needs a medical kit. Medical kits can be found inside rooms. They are white in color and have a red cross on them. Try to get one for him if possible."*

```
(:open
  (forall ?r - room
    (sense ?mk - medkit
     (lookedfor ?mk ?r)
      (and (hascolor ?mk white)
           (hassymbol ?mk redcross))
  (:goal (found ?mk ?r)
           [100] - soft))))
```

The difference between the previous case and this one is merely the goal specification; here we have a *soft* goal with an associated utility. In this example, the plan includes actions for checking inside rooms for medical kits, but only if resource and cost constraints are satisfied; otherwise, the planner can skip the goal entirely. This choice is not available to the planner in the previous cases.

The four cases thus show how difference in the goal formulation and the provided information will result in different system behavior and thus allow for very different types of goal instructions in open worlds.

6.2.2 The Warehouses Domain

In addition to testing on the USAR domain, we also evaluated the paper's main claim in a simulated case study in a warehouse-inspired domain (Talamadupula et al., 2013) to show that the proposed construct is general and domain independent. Consider the following dialogue between a human instructor and the robotic agent:

```
H: The dispatcher really needs a headphone item.
H: There should be one in the shelf at the end of the aisle.
R: Okay, should I get it for him?
H: Yes.
R: Okay, I'll get one for him.
R: What does it look like?
H: It is a brown shape with barcode bc1 on it.
```

The open-world quantified goal assembled at the end of this exchange is:

```
(:open
  (forall ?s - shelf
    (sense ?i - item
     (lookedfor ?i ?s)
      (and (has_barcode ?i bc1)
           (in ?i ?s))
  (:goal (and (inventoried ?i bc1 ?s)
           [100] - soft)))))
```

This case plays out exactly like the previous one in that the generated plan includes actions to look out for the item on a shelf, but only if resource and cost constraints are satisfied.

Together, these five cases demonstrate the breath and versatility of the open-world goal formalism and the ways syntactic variations can generate different system behavior. In other work, we have reported additional evaluations that vary utilities and penalties, as well as the nature, of the goals (hard or soft) (e.g., see Talamadupula et al. (2010a)).

7. Conclusions and Future Work

In this paper, we discussed the important problem of open-endedness in tasks like Urban Search and Rescue, which poses problems for cognitive architectures. We discussed the challenges of coping with new goal and task instructions that make reference to unknown objects when neither object type nor the location of the object is known. We also examined how various components in the DIARC architecture work together in assembling all the necessary information for formulating and achieving open-world quantified goals. This included the generation of inferences based on available information that, in turn, drive dialogue interactions with the human instructor to obtain missing information.

The cognitive mechanisms described in this paper have already enabled useful novel human-robot team interaction, but there remain opportunities to extend them to facilitate even more complex behavior. For example, teaming scenarios also require prediction of teammate behavior based on knowledge of the teammates' beliefs and goals, which means maintaining accurate mental models of interaction partners (Scheutz, 2013). Some of the mechanisms we utilized to achieve this capability have been described previously in this paper, though more are necessary to achieve more human-like performance. We are working toward integrating the belief component and planner directly, in order to use the planner as a predictive model of human behavior. We will initialize a planner with knowledge about a particular human teammate and then plan from that teammate's perspective.

Another direction is to extend the current belief and dialogue reasoning to include explicit representations of uncertainty. While the current classical logical representationss and inference schemes are sufficient to enable the intelligent behaviors described in this paper, more robust robot behavior will benefit from probabilistic representation and reasoning. We have begun to analyze similar interaction scenarios in terms of inference using Dempster-Shafer theory (Nunez et al., 2013).

Finally, there is more work to be done in order to make architectures like DIARC even more open-world taskable. Our current assumptions about the availability of speech recognition and synthesis of novel words should be dropped, as well as assumptions that actions be known ahead of time. Rather, the speech recognizer, syntactic and semantic parsers, and the components for utterance generation and speech synthesis should handle novel words of any grammtical kind and action components, in particular ones for manipulation, should deal with unknown objects without making assumptions about pre-defined actions (cp. to Gualtieri et al., 2016 for open-world grasping). With these and other additions, such as one-shot learning of objects and actions (Scheutz et al., 2017) and planning for human-robot teaming (e.g., Talamadupula et al., 2014), it will be possible for cognitive systems to successfully negotiate open worlds in a way that makes them practically useful for team tasks.

Acknowledgements
This research is supported in part by the ARO grant W911NF-13-1-0023, the ONR grants N00014-13-1-0176, N00014-09-1-0017 and N00014-07-1-1049, and the NSF grant IIS201330813.

References

Bacchus, F., & Kabanza, F. (1998). Planning for temporally extended goals. In *Annals of Mathematics and Artificial Intelligence*, Vol. 22, 5–27. Springer.

Bagchi, S., Biswas, G., & Kawamura, K. (1996). Interactive task planning under uncertainty and goal changes. *Robotics and Autonomous Systems*, *18*, 157–167.

Baral, C., Kreinovich, V., & Trejo, R. (2001). Computation complexity of planning with temporal goals. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (pp. 509–514). Seattle, WA: AAAI Press.

Benton, J., Do, M., & Kambhampati, S. (2009). Anytime heuristic search for partial satisfaction planning. *Artificial Intelligence Journal*, *173*, 562–592.

Briggs, G., & Scheutz, M. (2011). Facilitating mental modeling in collaborative human-robot interaction through adverbial cues. *Proceedings of the Twelfth Annual SIGdial Conference on Discourse and Dialogue* (pp. 239–247). Portland, OR: Association for Computational Linguistics.

Choi, D. (2011). Reactive goal management in a cognitive architecture. *Cognitive Systems Research*, *12*, 293–308.

Clark, H. H. (1996). *Using language*. Cambridge, UK: Cambridge University Press.

Cushing, W., Benton, J., & Kambhampati, S. (2008). *Replanning as a deliberative re-selection of objectives* (Technical Report). Arizona State University.

Etzioni, O., Golden, K., & Weld, D. S. (1997). Sound and efficient closed-world reasoning for planning. *Artificial Intelligence*, *89*, 113–148.

Ferguson, G., Allen, J., & Miller, B. (1996). TRAINS-95: Towards a mixed-initiative planning assistant. *Proceedings of the Third Conference on Artificial Intelligence Planning Systems* (pp. 70–77). Edinburgh, UK: AAAI Press.

Gerevini, A. E., Haslum, P., Long, D., Saetti, A., & Dimopoulos, Y. (2009). Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence Journal*, *173*, 619–668.

Golden, K., Etzioni, O., & Weld, D. (1994). Xii: Planning with universal quantification and incomplete information. *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*. Bonn, Germany: Morgan Kaufmann.

Golden, K., & Weld, D. (1996). Representing sensing actions: The middle ground revisited. *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning*. Cambridge, MA: Morgan Kaufmann.

Gomez, R., Kawahara, T., Nakamura, K., & Nakadai, K. (2012). Multi-party human-robot interaction with distant-talking speech recognition. *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction* (pp. 439–446). New York, NY: ACM.

Gualtieri, M., ten Pas, A., Saenko, K., & Platt, R. (2016). High precision grasp pose detection in dense clutter. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 598–605).

Klenk, M., Molineaux, M., & Aha, D. W. (2013). Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence*, *29*, 187–206.

Krause, E., Zillich, M., Williams, T., & Scheutz, M. (2014). Learning to recognize novel objects in one shot through human-robot interactions in natural language dialogues. *Proceedings of Twenty-Eighth AAAI Conference on Artificial Intelligence* (pp. 2796–2802). Québec City, QC: AAAI Press.

Kruijff, G.-J. M., Lison, P., Benjamin, T., Jacobsson, H., Zender, H., Kruijff-Korbayová, I., & Hawes, N. (2010). Situated dialogue processing for human-robot interaction. In H. I. Christensen, G.-J. M. Kruijff, & J. L. Wyatt (Eds.), *Cognitive systems*, Vol. 8, 311–364. Springer.

Lemaignan, S., Ros, R., Sisbot, E. A., Alami, R., & Beetz, M. (2012). Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *International Journal of Social Robotics*, *4*, 181–199.

Myers, K. (1996). Advisable planning systems. In A. Tate (Ed.), *Advanced planning technology: Technological achievements of the ARPA/rome laboratory planning initiative*, 206–209. AAAI Press.

Myers, K. (1998). Towards a framework for continuous planning and execution. *Proceedings of the AAAI Fall Symposium on Distributed Continual Planning*. Orlando, FL: AAAI Press.

Nunez, R. C., Dabarera, R., Scheutz, M., Briggs, G., Bueno, O., Premaratne, K., & Murthi, M. N. (2013). Ds-based uncertain implication rules for inference and fusion applications. *Proceedings of the Sixteenth International Conference on Information FUSION* (pp. 1934–1941). Istanbul, Turkey: IEEE.

Rich, C., Ponsler, B., Holroyd, A., & Sidner, C. L. (2010). Recognizing engagement in human-robot interaction. *Proceedings of the Fifth ACM/IEEE International Conference on Human-Robot Interaction* (pp. 375–382). Osaka, Japan: ACM.

Roberts, M., Vattam, S., Alford, R., Auslander, B., Apker, T., Johnson, B., & Aha, D. W. (2015). Goal reasoning to coordinate robotic teams for disaster relief. *Proceedings of ICAPS-15 PlanRob Workshop* (pp. 127–138). Jerusalem, Israel: AAAI Press.

Scherl, R. B., & Levesque, H. J. (1993). The frame problem and knowledge-producing actions. *Proceedings of the Eleventh National Conference on Artificial Intelligence* (pp. 689–695). Washington D.C.: AAAI Press.

Schermerhorn, P., & Scheutz, M. (2010). Using logic to handle conflicts between system, component, and infrastructure goals in complex robotic architectures. *Proceedings of the IEEE Conference on Robotics and Automation* (pp. 392–397). Anchorage, AK: IEEE.

Scheutz, M. (2013). Computational mechanisms for mental models in human-robot interaction. In *Virtual, augmented, and mixed reality. designing and developing augmented and virtual environments*, 304–312. Las Vegas, NV: Springer.

Scheutz, M., Briggs, G., Cantrell, R., Krause, E., Williams, T., & Veale, R. (2013). Novel mechanisms for natural human-robotic interactions in the DIARC architecture. *Proceedings of AAAI Workshop on Intelligent Robotic Systems* (pp. 66–72). Bellevue, WA: AAAI Press.

Scheutz, M., Krause, E., Oosterveld, B., Frasa, T., & Platt, R. (2017). Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. *Proceedings of the Sixteenth Conference on Autonomous Agents and Multi-Agent Systems* (pp. 1378–1386). São Paulo, Brazil: ACM.

Scheutz, M., Schermerhorn, P., Kramer, J., & Anderson, D. (2007). First Steps toward Natural Human-Like HRI. *Autonomous Robots*, *22*, 411–423.

Talamadupula, K. (2014). *Planning challenges in human-robot teaming*. Doctoral dissertation, Computer Science Department, Arizona State University, Tempe, AZ.

Talamadupula, K., Benton, J., Kambhampati, S., Schermerhorn, P., & Scheutz, M. (2010a). Planning for human-robot teaming in open worlds. *ACM Transactions on Intelligent Systems and Technology*, *1*, 14.

Talamadupula, K., Benton, J., Schermerhorn, P., Kambhampati, S., & Scheutz, M. (2010b). Integrating a closed world planner with an open world robot: A case study. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 1561–1566). Atlanta, GA: AAAI Press.

Talamadupula, K., Briggs, G., Chakraborti, T., Scheutz, M., & Kambhampati, S. (2014). Coordination in human-robot teams using mental modeling and plan recognition. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2957–2962). Chicago, IL: IEEE.

Talamadupula, K., Smith, D. E., Cushing, W., & Kambhampati, S. (2013). A theory of intra-agent replanning. *ICAPS 2013 Workshop on Distributed and Multi-Agent Planning*. Rome, Italy: AAAI Press.

van den Briel, M., Sanchez, R., Do, M., & Kambhampati, S. (2004). Effective approaches for partial satisfaction (over-subscription) planning. *Proceedings of the National Conference on Artificial Intelligence* (pp. 562–569). San Jose, CA: AAAI Press.

Williams, T., Cantrell, R., Briggs, G., Schermerhorn, P., & Scheutz, M. (2013). Grounding natural language references to unvisited and hypothetical locations. *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence* (pp. 947–953). Bellevue, WA: AAAI Press.

Yoon, S., Fern, A., & Givan, R. (2007). FF-replan: A baseline for probabilistic planning. *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling* (pp. 352–359). Providence, RI: AAAI Press.