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

Kartik Talamadupula† KRT@ASU.EDU
Gordon Briggs§ GORDON.BRIGGS@TUFTS.EDU
Matthias Scheutz§ MATTHIAS.SCHEUTZ@TUFTS.EDU
Subbarao Kambhampati† RAO@ASU.EDU
†Dept. of Computer Science & Engineering
Arizona State University, Tempe AZ USA
§HRI Laboratory
Tufts University, Medford MA USA

Abstract

Future envisioned mixed-initiative human-robot teams will require increasingly autonomous and capable robotic team members who can interact with their human teammates in natural ways. The challenge is to develop an integrated cognitive robotic architecture that will enable effective, natural human-robot interactions. In this paper, we present recent work on our integrated robotic DIARC architecture with its embedded Sapa Replan planner for open-world human-robot teaming tasks (i.e., tasks where not everything is known about the task objects and objectives ahead of time). We specifically focus on the required architectural mechanisms and capabilities for handling online instructions with unspecified objects and describe the novel mechanisms implemented in DIARC and Sapa Replan that enable online open-world tasking.

1. Introduction

Consider 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."[1]

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" (ibid.), among others. Since USAR is considered a "multi-hazard discipline", mixed-initiative human-robot teams could significantly improve the effectiveness of such missions (by being able to search spaces that are inaccessible to humans) while reducing the risk of human searchers getting trapped themselves in collapsing buildings.

1. See `http://www.fema.gov/urban-search-rescue`.

Currently employed tele-operated robots, however, are often not appropriate for such missions, e.g., because operating them is not possible due to wireless connectivity problems or lack of accurate sensory information, because the operation is too slow and ineffective given the urgency of the task, or because the tele-operation locks in human resources that could be better used in other ways. Hence, ultimately the goal is to use *autonomous robots* that complement human teammates and can 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 all the mechanical and control problems that need to be resolved for robots to be able to function properly in such environments, there is a critical feature of USAR missions, common to many other human team tasks as well, that presents a major challenge to the cognitive parts of robotic architectures: *the open-endedness of the mission* (i.e., the many aspects of the mission that are not known in advance including goals, tasks, and subtasks, locations of humans and objects, building layouts, etc.). While humans are able to handle such open-ended missions by negotiating novel or unknown aspect in natural language, current cognitive robotic architecture are not yet capable of dealing with the very same unknown and novel aspects in the same way as humans. In part, the problem is that natural language capabilities are significantly lacking in current robots; but equally important are novel architectural mechanisms that are required in other architectural components such as the task planners or belief models as well for robots to be able to cope with all that is entailed by the open-ended missions, as we will discuss in this paper. Specifically, we will focus on two critical aspects of open-ended missions that cognitive robotic architectures will have to handle: (A1) that *not all goals, tasks and subtasks are known ahead of time, but new goals may be assigned and new subtasks might get defined during task performance*; and (A2) that *not all information about task-relevant entities is available ahead of time, but new knowledge about unknown objects needs to be acquired during task performance (this includes knowledge about objects and their appearance, locations, people, activities, and others)*.

The rest of the paper is structured as follows. We start by introducing a motivating USAR example to illustrate both critical aspects (new goals and tasks, and new objects) in a human-robot natural language dialogue interaction as part of a larger USAR mission. We then first focus on the functional requirements of the natural language understanding system to be able to extract from the dialogue novel goals and tasks as well as information about unknown objects. We then describe the various additional architectural mechanisms and capabilities required for the robot to pursue the new goals, carry out the next tasks, and handle the new objects. This includes detailed descriptions of the requirements imposed on the task planner. In the discussion section, we address the challenges ahead and conclude with our accomplishments to date and a brief outline of next steps for future work.

2. Contributions

In this work, we focus our attention on the problems posed by incomplete specifications (Kambhampati, 2007) and open worlds to the conduct of goal-directed, autonomous agents in the context of dealing with new goal and task instructions that may arrive as the agent is executing in the real world. It is important to note here that *all* human-robot teams are constituted in the service of spe-

Architectural Component	*Necessary Extensions*
Natural Language & Dialogue	The natural language understanding mechanisms require the ability to understand linguistic cues (such as modifiers that convey uncertainty) that allow for the autonomous agent to recognize situations where closed-world reasoning is insufficient. The natural language also needs to allow for NL generation requests from other components that may seek to obtain information through NL interaction with an interlocutor.
Belief Reasoner	The belief reasoner needs to contain rules that allow the autonomous agent to infer possible goals (for itself) from the goal and belief states of other agents. Additionally, the belief reasoning component needs to be able to recognize and differentiate between closed-world and open-world goals, such that the appropriate goal submission process can be undertaken with the Goal Management component.
Goal Manager	The ability to represent and provide information about goals in the open-world to the planner component.
Planner	A representation to denote open-world information – and goals associated with that information – needs to be devised. Subsequent to this, the planner needs to use this information to generate plans for both universally and existentially quantified open-world goals.

Table 1. A table showing the extensions necessary to each component of the architecture.

cific goals – either at a higher, abstract level; or a lower, more defined level. It makes little sense then to assume that these goals will remain static, or that they will all be specified up-front at the beginning of each scenario. Instead, flexible frameworks are indeed required that are expressive enough to denote most goals of interest, yet allow modifications (including addition and deletion) to goals with relative ease. Additionally, since these goals all stem from humans, the representation used by these goals must be on a level that humans are comfortable with – too high and no goals of relevance can be defined; too low and humans will fast lose track of what the team is trying to achieve. A similar constraint applies to the actions at the robotic agent's disposal, as well as its representation of the world. All of these must also be expressive enough representationally to be sufficient for any real-world domain that they might be required in. In particular, the planner should allow for actions with durations to handle goals with deadlines, given the reality that actions take time to execute in the physical world; and partial satisfaction of goals should be possible to allow the planner to "skip" seemingly unreachable goals.

Along with these, an important part of any online system is execution monitoring and replanning to allow the robot to receive and react to new information from a human commander (e.g., a change in goal deadline). Robots operating in teaming scenarios require the ability to plan (and revise) a course of action in response to human instructions. To accept information from a human commander, the robotic architecture parses and processes natural language (i.e., speech) into goals or new facts. If the architecture cannot handle a goal or fact by following a simple script located in its library, it calls the planner to find a method of achieving the goal.

On the planning side, while the state-of-the-art planning systems are very efficient, they focus mostly on closed worlds. Specifically, they expect full knowledge of the initial state, and expect up-front specification of the goals. Adapting them to handle open worlds presents many thorny challenges. Assuming a closed-world up-front will not only necessitate frequent replanning during execution, but can also lead to highly suboptimal plans in the presence of goals that are conditioned on information that can be known only by sensing during execution. Acquiring full knowledge up-front, on the other hand, would involve the robot doing a sensing sweep to learn everything about its world before commencing the planning – an infeasible task, since a robot cannot be simply commanded to "sense everything", but rather has to be directed to specific sensing tasks. Accounting for missing knowledge would involve making conditional plans to handle every type of contingency, and letting the robot follow the branches of the plan that are consistent with the outcomes of its sensing. Such full contingency planning is already known to be impractical in propositional worlds with bounded indeterminacy (Meuleau & Smith, 2002); it is clearly infeasible in open worlds with unknown numbers of objects, of (possibly) unknown types. In this paper, we demonstrate a first step towards integrating a planner that can function in an open world with a robotic architecture that can supply it with the information needed to make this problem solvable; and discuss the various architectural components that need to be extended (see Table 1) in order to enable this.

3. A Motivating Example from an USAR Scenario

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 that achieves the following results:

1. the robot assumes that Z is the name of a commander and infers that Commander Z is a human person (e.g., because it knows that all commanders are human)

2. it infers 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. it further infers 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. it infers 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".

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, that it contains medical equipment, etc. (just substitute "vacation" for "medical kit" in the above sentence).

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 allows it to 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 to humans that they need:

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

So 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.
```

Now that the robot's new goal has been confirmed, three new questions arise and have to 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 (e.g., see (Williams et al., 2013) for a detailed description of the required capabilities).

3. *Where is Commander Z located?* This question is also not of immediate importance, but will have to be answered at some point (e.g., once the robot has found and retrieved a medical kit).

Hence, the robot both acknowledges the human confirmation and also follows up with a question about the medkit's appearance (note that appearance is all the robot requires to know about a medkit as further questions about its purpose, etc. 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 has to 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 demonstrating how both aspects (A1) and (A2) can come up naturally in the context of open-world tasks, which turns out to be quite complex and complicated in terms of the requirements it imposes on the robotic architecture. We will next describe how the above functionality of the robot can be accomplished in an integrated cognitive robotic architecture, for which we have used our DIARC architecture (Scheutz et al., 2013; Scheutz et al., 2007). We will specifically focus on what it takes in terms of representational and functional capabilities for the robot to be able to understand the above instructions and carry out the above dialogue, and how it can configure various of its components in a way that allows it to successfully pursue the goal (and eventually accomplish it if the environmental circumstances are right, i.e., there is indeed a medkit in the room at the end of the hallway, the robot can determine the whereabouts of Commander Z, etc.).

4. Open-World Instruction Understanding

The simple dialogue demonstrated several challenges that have to be addressed in a natural language understanding (NLU) system as part of a robotic architecture to be able to handle open-world instruction. Aside from the obvious challenges of having to cope with new words – out-of-vocabulary recognition in the speech recognizer and estimating the part-of-speech-tag in the tagger and/or parser – the NLU has to also deal with the lack of the 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, and so is their grammatical type as there are multiple possibilities (e.g., "Z" could be a proper name or an adverb like "now", and "medical" and "kit" could both be multiple grammatical types, nouns, adjectives, adverbs, etc.). Hence, based on the lexical ambiguities, it is impossible to guess a semantic type, left alone the

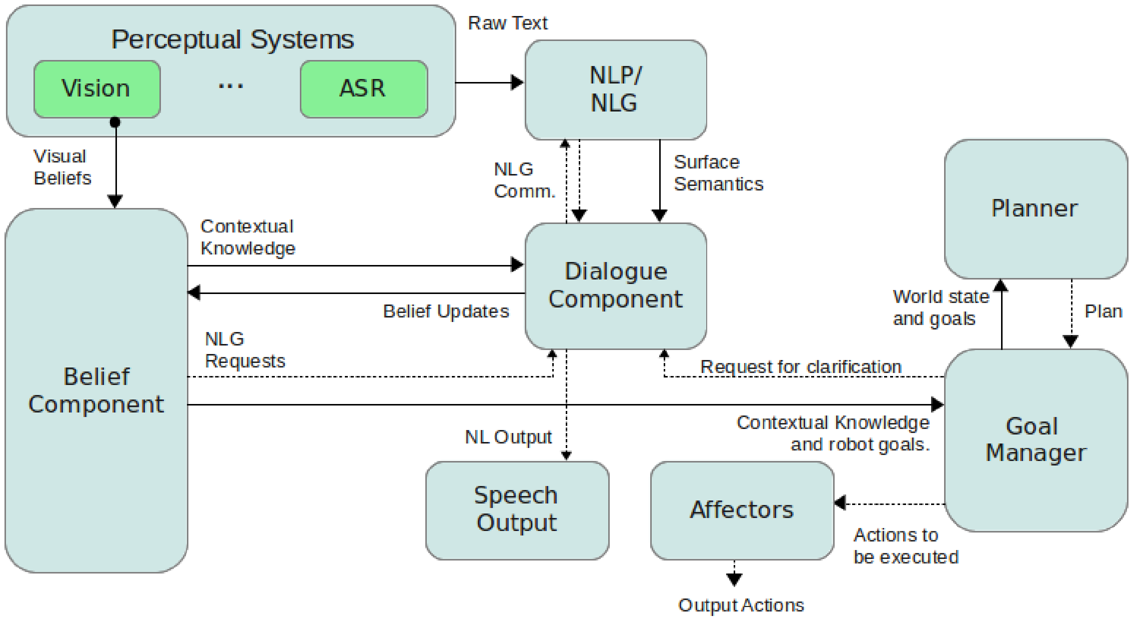

Figure 1. A schematic that shows the architecture of the `DIARC` integrated system.

meaning. As discussed before, the assumption that "Z" is a name and that the words "medical kit" denote an object, however, resolve only some of the problems, because the robot needs to be able to extract the implicit order expressed in the sentence "Get Commander Z a medical kit", it needs to turn this into goal expressions that the task planner can handle. The tricky part here is that the planner does not know about medical kits (as objects) and that even if it did know about medical kits, 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 because the goal is not that there "be a medkit such that Z has it", but that Z has one of that kind. It should be clear that a straightforward way of translating this into a two-place goal expression will not succeed. Moreover, understanding the next sentence and connecting it to the previous sentence is also 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 a fact "$\exists medkit(x) located-in(x, room-at-the-end-of-the-hallway)$". However, "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, which has a more conditional flavor and would in conjunction with the first sentence yield something like this: "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 what is meant by the two sentences as some sort of conditional goals, it seems more plausible that the planner could make sense of such a goal and generate a sequence of actions that would accomplish it: first go to the room at the end of the hallway, to then look for the medical kit in the room, and if one is found, to 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 component 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 component. 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 P denotes the urgency of the goal. Below we describe in greater detail how the process of obtaining this information with the natural language understanding system occurs, and how this information 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 a goal another agent (C_Z) has. The flow of information within the robotic architecture that originates from speech input is captured in the architecture diagram found in Figure 1. First the speech recognition component generates the text of the heard utterance, which is then forwarded to the natural language processing component. Parsing and initial semantic analysis is then performed on this received text data. The resulting surface semantics are then forwarded to the dialogue component for pragmatic analysis, which is described below.

Within the dialogue component a series of pragmatic rules provide a means of translating between surface semantics understood by the natural language processing component and the belief 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 (e.g. instruction, statement, acknowledgment), α denotes the speaker, β denotes the listening agent, ϕ denotes the surface semantics of the utterance, M denotes a set of sentential modifiers, and the set of predicates $\psi_1 \wedge ... \wedge \psi_n$ denote 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 denotes the dialogue and belief context that must apply for this interpretation to be made.

For the the statement, "Commander Z needs a medical kit," the following pragmatic rule would apply:

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

Meaning that the robot would infer that agent α wants the listener β to believe agent γ has a goal with default urgency to have object X. We are omitting the $[[]]_C$ notation here as this rule will apply in general. The modifier notation introduced previous can also be utilized to infer information about goal urgency. For instance, what 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. These rules are described in the subsequent section.

4.2 Belief Reasoning

Having recently received the belief update $want(cmdrX, bel(self, goal(cmdrZ, have(cmdrZ, medkit), high)))$, various inferences are made in the belief component. To adopt beliefs based on communicated facts, have a simple, naive rule that encodes a credulous belief adoption policy:

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

At this point, the belief $bel(self, goal(cmdrZ, have(cmdrZ, medkit), high))$ is supported. The belief component also contains basic rules that reason about social roles and possible obligations. Some of these rules that are utilized in the scenario are described below:

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

This 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.

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

The above inference rule encodes the notion that if an agent β believes that a superior has a goal for ϕ to obtain, that 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. The intention-to-know predicate generates 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 is then submitted to the Goal Manager component. However, due to the uncertain state of a key object in this potential goal, specifically the fact that the medical kit "should be" at the current room ($should(at(medkit, current - room))$), this goal is not treated a a regular goal. Instead, the belief component submits this goal as a special type of goal known as an open world quantified goal, which is described in the following section.

5. Open-World Planning

In order to parse and act upon information and goals that are conditional in nature, as mentioned in the previous section ("if there is a medkit in the room at the end of the hallway, then get it and bring it to commander Z"), it will not do for the the planning system in use to simply assume a closed world (Etzioni, Golden, & Weld, 1997) with respect to unknown information (an assumption, unfortunately, that most state-of-the-art planners make). What is needed instead is both a framework for specifying conditional knowledge and rewards, and an approach for using that knowledge to direct the robot in such a way as to intelligently trade sensing costs and goal rewards. Accordingly, we use an approach for representing and handling a class of goals called *open world quantified goals* (OWQGs), which provide a compact way of specifying conditional reward opportunities over an "open" set of objects.

5.1 Open World Quantified Goals

Open world quantified goals (OWQGs) (Talamadupula et al., 2010a) combine information about objects that *may be* discovered during execution with goals that are contingent on the discovery of those objects. The human member of a human- robot team can use an OWQG to provide details about what new objects may be encountered through sensing, and include goals that relate directly to those sensed objects. Newly discovered objects may enable the achievement of goals, granting the opportunity to pursue reward. For example, detecting a medical kit (medkit) in a room will allow the robot to pick up that medkit and deliver it to another location (where picking it up accrues reward). Given that the reward in this case is for each medkit picked up, there exists a quantified goal that must be allowed partial satisfaction. In other words, the universal base, or total grounding of the quantified goal on the real world, may remain unsatisfied while its component terms may be satisfied. To handle this, we rely on the partial satisfaction capability (Van Den Briel et al., 2004) of the base planner, Sapa Replan (Cushing, Benton, & Kambhampati, 2008).

As an example, we present an illustration from the USAR scenario outlined in Section 3: the robot is directed 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* – however, the nature of this quantification exposes an interesting difference, which we now discuss.

5.1.1 Universally Quantified Goals

The first kind of OWQG is one where the goal is quantified universally (Golden, Etzioni, & Weld, 1994); that is, it is quantified over all possible instances of a particular object type. To illustrate, consider the following directive from the commander 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."
```

This goal can be written formally as an OWQG in the following syntax:

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

The above goal can be interpreted as follows: line 2 denotes the variable that the goal is quantified over – 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* predicate that informs that planner that a sensing action has been performed, thus stopping repeated sensing. Lines 5 and 6 list the properties that will 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 stands for the goal over such an object, while line 8 indicates that there is a reward of 100 units associated with fulfilling that goal, and that the goal is *soft* (that is, it must be seen as an opportunity, and need not 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 of which there exists only a single instance. Consider the following utterance by a commander:

`"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 goal presented in Section 5.1.1; in this instance, the commander 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 runtime – the planner does not need to look into all rooms to find the medical kit. This observations leads to a simple idea that can be used to model 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 the following `OWQG`:

```
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 one specified previously, 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` sub-type. To be sure, this is but an approximation that enables the planner to handle existentially quantified goals in the same manner as universally quantified ones, and depends on the assumption that there will only be one object (constant) of type `endroom`; we are currently working on a more principled approach to handle this instead.

5.2 Implementation

While walking through the human-robot interaction 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 as an `OWQG`. In this section we will first describe how `OWQG`s are submitted to the Goal Manager and then communicated to the planner, then we discuss how the planner works with `OWQG`s.

5.2.1 Goal Submission and Management

As mentioned previously, if in the usual case a goal predicate of form $goal(self, X, P)$ is supported in the belief component, it would be submitted as a regular goal to the goal manager. However, certain rules within belief trigger treating goals as `OWQG`s. For instance, we have an inference rule:

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

meaning that if the location of the object X is uncertain, the goal should be treated as an `OWQG`. The goal submission mechanism checks whether or not the `OWQG` predicate is supported. If not, a goal is submitted as a normal goal. Otherwise, the `OWQG` submission process begins. This involves supplying the Goal Manager with information about what variables are to be considered open and what sensory information (and hence what sensory actions) can be taken to make inferences about the state of these open variables.

In the case of our scenario, the location of the medical kit is not known for certain. Therefore, the following rule is used to denote L as the open variable associated with the `OWQG` to obtain the medical kit for Commander Z:

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

Likewise, similar rules specifying the object to be sensed for is the object X. The information inferred by these rules is then submitted with the `OWQG` to the Goal Manager, which attempts to submit the `OWQG` to the planner. However, in our scenario, there is still a problem left: how can the robot act on a goal to look out 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 what the medkit looks like needs to be included in the sensing action that the robot has to perform. The Goal Manager detects that it lacks a visual descriptor for the sense variable type, and formulates a request for clarification (specifically a request for a visual description). This request is submitted to the dialogue component, which initiates the natural language generation process, resulting in the query, "What does it look like?" This produces feedback from the human interactant, who supplies the visual description. With the requisite information available, the Goal Manager then submits the `OWQG` to the planner.

5.2.2 Planner

To handle open world quantified goals, the planner *grounds* the problem into the closed-world using a process similar to Skolemization. More specifically, we generate *runtime objects* (from the sensed variable – in this case, `medkit`) that explicitly represent the potential existence of an object to be sensed. These objects are marked as system generated runtime objects. The runtime objects are then added to the problem and ground into the closure condition. The facts generated by following this process are included in the set of facts in the problem through the problem update process. The goals generated on runtime objects are similarly added. This process is repeated for every new object that the process may instantiate in the case of universally quantified `OWQG`s.

We treat the closure condition *optimistically*, meaning a particular state of the world is considered closed once the ground closure condition is true. On every update coming in to the planner from the world (via the architecture), the ground closure conditions are checked, and if true the runtime objects and goals instantiated on them are removed from the problem. By planning over this representation, we provide a plan that is executable given the planning system's current representation of the world until new information can be discovered (via a sensing action returning the closure condition). The idea is that the system is interleaving planning and execution in a manner that moves the robot towards rewarding goals by generating an optimistic view of the true state of the world.

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`, as it were), the planner generates a runtime object `medkit!1`. Subsequently, the facts `(hascolor medkit!1 white)` and `(hassymbol medkit!1 redcross)`, along with the goal `(found medkit!1 er1)` (with accompanying reward 100) would be generated and added to the problem[2]. A closure condition `(lookedfor medkit!1 er1)` would also be created. When the planner receives an update from the world that includes this condition as one of the true facts, it will update its representation of 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 (a sensing action) – once the condition becomes true, the runtime object (and facts) are no longer needed, since the truth values in the real world are known. This process, of resolving the uncertainty in a dynamic world with a combination of sensing and replanning when there are updates, is very reminiscent of the planner *FF-Replan* (Yoon, Fern, & Givan, 2007).

6. Evaluation

In order to evaluate the representational and architectural extensions implemented on the integrated `DIARC` architecture in this work, we ran the robot controlled by this architecture on an urban search and rescue (USAR) task. Specifically, our robot was put in a scenario where it had to listen for and understand natural language instructions from a human teammate. In this case, the human-robot dialog was as follows:

```
"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."
```

2. The exclamation mark indicates a runtime object.

```
"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."
```

From the above exchange, the robot needs to understand that since `Commander Z` *needs* a medical kit, there is a goal on delivering that `medkit` to the commander. The robot also engages the human in further conversation, in order to ascertain the details of both where the commander is, as well as where the `medkit` may be, and what the `medkit` looks like. This information is used to inform the `OWQG` that is eventually used in the planning process ("white box with a red cross"). Finally, the robot issues a verbal confirmation to the teammate that it is now engaged in the task.

For a full video of this interaction in real time, please see the following: `http://www.youtube.com/watch?v=RJ1VSIi1CM4`.

7. Future Work

While the cognitive mechanisms described in this paper have enabled us to perform a useful human-robot team interaction, there is ample opportunity to extend and improve these mechanisms to facilitating even more complex and natural human-robot teaming behavior. Two of these potential extensions are described below.

7.1 Mental Modeling and Planning Extensions

Teaming scenarios will also require the prediction of teammate behavior based on knowledge of their beliefs and goals. This requires the ability to maintain accurate mental models of interaction partners (Scheutz, 2013). Some of the mechanisms utilized to achieve this capability have been described previously in this paper, though more are necessary to achieve human-like performance. We are currently working toward integrating the belief component and planner directly, in order to use the planner as a mechanism by which to predict the behavior of a human teammate. This will be accomplished by instantiating a planner problem from the perspective of the *teammate* and deriving an expected plan.

7.2 Reasoning With Uncertainty

Belief and dialogue reasoning is presently performed with classical logical representations. While this is sufficient to enable the intelligent behaviors described in this paper, more robust behavior from the robot may require probabilistic belief representation and reasoning. We have begun to analyze a similar interaction scenario in terms of a Dempster-Shafer (DS) theory inference system (Nunez et al., 2013). DS based reasoning can also be used to reason about the the credibility of information sources (Wickramarathne, Premaratne, & Murthi, 2012). This ability could be useful in developing a more sophisticated belief adoption policy that factors in the *trust* the robot has in a speaker. Instead of a naive (highly credulous) adoption rule, a rule that factors in basic model of trust or reliability can be obtained:

$$want(\alpha, bel(\beta, \phi))_{[a_1,b_1]} \wedge trusts(\beta, \alpha)_{[a_2,b_2]} \Rightarrow_{[a_3,b_3]} bel(\beta, \phi)_{[a_4,b_4]}$$

That is to say, the degree to which an agent β believes a communicated proposition ϕ is dependent on the degree to which β believes that agent α wants him or her to believe ϕ, the degree to which β trusts α, and the degree to which an inter-agent communication can be used to make inferences. Currently, we are working on integrating the DS reasoning mechanisms as described in (Nunez et al., 2013) with the natural language and belief modeling components to enable such capabilities.

8. Conclusion

In this paper, we have discussed the important problem of open-endedness in tasks like Urban Search and Rescue, which poses significant problems for cognitive-robotic architectures. We discussed specifically the challenges of coping with new goal and task instructions that make reference to unknown objects where neither object type nor the location of the object is known. We then discussed how various components in the DIARC architecture work together to generate the inferences that drive the dialogue to gather more information where necessary and to generate goal representations, in particular, "open world quantified goals", that overcome the past difficulties of task planners to express and plan with goals that make reference to unknown objects.

In a next step of the architecture development – to improve the architecture's open-world capabilities – we are currently working on making the probabilistic nature of some of the representations, as well as the confidence the robot has in its interpretations, more explicit. This will allow the robot to make better, more robust inferences in cases of ambiguity and severe lack of knowledge. We are also adding mental modeling capabilities that will allow the robot to keep track of the mental states of its teammates, which should allow it to better understand human commands and infer unknown facts and activities that humans might presume the robot already knows.

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

Briggs, G., & Scheutz, M. (2011). Facilitating mental modeling in collaborative human-robot interaction through adverbial cues. *Proceedings of the SIGDIAL 2011 Conference* (pp. 239–247).

Cushing, W., Benton, J., & Kambhampati, S. (2008). Replanning as a Deliberative Re-selection of Objectives. *Arizona State University CSE Department TR*.

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

Golden, K., Etzioni, O., & Weld, D. (1994). Xii: Planning with universal quantification and incomplete information. *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning, KR*.

Kambhampati, S. (2007). Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain models. *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE* (p. 1601).

Meuleau, N., & Smith, D. E. (2002). Optimal limited contingency planning. *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence* (pp. 417–426).

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 16th International Conference on Information FUSION*.

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

Scheutz, M. (2013). Computational mechanisms for mental models in human-robot interaction. *HCI International* (pp. 304–312).

Scheutz, M., Briggs, G., Cantrell, R., Krause, E., Williams, T., & Veale, R. (2013). Novel mechanisms for natural human-robot interactions in the diarc architecture. *Proceedings of AAAI Workshop on Intelligent Robotic Systems*.

Scheutz, M., Schermerhorn, P., Kramer, J., & Anderson, D. (2007). First steps toward natural human-like HRI. *Autonomous Robots*, *22*, 411–423.

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*.

Talamadupula, K., Benton, J., Schermerhorn, P. W., Kambhampati, S., & Scheutz, M. (2010b). Integrating a closed world planner with an open world robot: A case study. *AAAI*.

Van Den Briel, M., Sanchez, R., Do, M. B., & Kambhampati, S. (2004). Effective approaches for partial satisfaction (over-subscription) planning. *AAAI* (pp. 562–569).

Wickramarathne, T. L., Premaratne, K., & Murthi, M. N. (2012). Consensus-based credibility estimation of soft evidence for robust data fusion. *Proceedings of the 2nd International Conference on Belief Functions* (pp. 301–309). Springer.

Williams, T., Cantrell, R., Briggs, G., Schermerhorn, P., & Scheutz, M. (2013). Grounding natural language references to unvisited and hypothetical locations. *Proceedings of AAAI*.

Yoon, S. W., Fern, A., & Givan, R. (2007). Ff-replan: A baseline for probabilistic planning. *ICAPS* (pp. 352–359).