Advances in Cognitive Systems 5 (2017) 1-6 Submitted 3/2017; published 5/2017

A High Level Language for Human Robot Interaction

Chitta Baral CHITTA@ASU.EDU
Barry Lumpkin BTLUMP@GMAIL.COM
SCIDSE, Arizona State University, Tempe, AZ 85287

Matthias Scheutz MATTHIAS.SCHEUTZ@TUFTS.EDU
Department of Computer Science, Tufts University, Medford, MA 02155

Abstract

Language processing and high level execution and control are functional capabilities that arise in the context of cognitive systems. In the context of human-robot interaction, natural language is considered a good communication medium as it allows humans with less training about the robot's internal language to be able to command and interact with the robot. However, any natural language communication from the human needs to be translated to a formal language that the robot can understand. Similarly, before the robot can communicate with the human, it needs to formulate its communique in some formal language which then gets translated into natural language. In this paper, we present a high level language for communication between humans and robots and demonstrate various aspects through a robotics simulation. These language constructs borrow some ideas from action execution languages and are grounded with respect to simulated human-robot interaction transcripts.

1. Introduction and Motivation

Within the field of human-robot teamwork, there are highly varied implementations. On one end of the spectrum, the use of teleoperation allows robots to be used as tools in which a human operator has direct control over a robot's actions. Such systems are highly dependent on the operator's skill and are extremely hindered by situations with limited bandwidth. The other extreme contains highly autonomous robots that are simply given a high-level goal from a human supervisor who does not directly interfere in the robot's operation. This gives the operator the ability to handle many systems simultaneously but does not provide any flexibility in the event of unexpected events.

A more practical application would be an intelligent robot team in which humans and robots work together much in the same way a team of humans would. Each individual, human or robot, would be able to actively seek assistance from others when needed. For example, in an urban search and rescue scenario, it may be that the environment is too dangerous for humans, so a group of robots would be sent instead. These robots would be given tasks to autonomously complete, but since the environment is most likely unpredictable and possibly still changing, the robots would need to seek guidance throughout the operation.

For such human-robot interaction, natural language is considered to be a good communication medium as it allows humans with little training on the robot's internal language to be able to command and interact with the robot. Apart from requiring less training, utilizing natural language would allow for a faster dialogue as the human would not need to translate their thoughts into a structured format that the robot would understand. However, robots still require commands to be given in a structured format in order to be processed. As such, the natural language communication must be translated into a formal language which the robot can understand. Additionally, when the robot is forming a communique back to the human, it must first formulate the message in this formal language which is then converted into natural language the human can easily understand.

1.1 Related Work

There have been several works on the past that discussed language to communicate with and between robots. Following is a quick overview of some of the past works.

In the recent work (Ji et al., 2016) a Robot Communication Language (RCL) to share knowledge and instructions with one another has been proposed and used. In that, when a user said, "I am thirsty," RCL enabled the robot CoBot to inform the robot KeJia of the user's thirst, and KeJia was then able to instruct CoBot to get the user a bottle of water. The KeJia and CoBot experiments demonstrate how a communication language in robots is not only important for human-robot interaction but for robot-robot interaction as well.

The Human-Robot Interaction Operating System (HRI/OS) from the Peer-to-Peer Human-Robot Interaction project provides a focus on allowing agents to submit requests for help which is processed once the necessary resources become available, such as other agents (Scholtz, 2002; Fong et al., 2005, 2006). Another key aspect of the HRI/OS software is that it is designed to utilize spatial reasoning and perspective-taking to enable dialogue using relative locations.

The Jidowanki and Biron robots utilize a task negotiation dialogue in which the robots prompt the user with queries until a clear goal is assigned based on the current known environment (Clodic et al., 2007). Additionally, this system allows for a robot to submit a request for a plan modification in the event that it determines another, potentially better, plan is now available due to changes in the environment. The user is able to accept or reject this new plan, or even initiate their own plan modification.

A robotic wheelchair was used in (Fischer, 2011) to study the effect of interactive dialogue on how a user interacts with the system. The first study showed that an interactive dialogue allowed the user to better understand the capabilities of the system and become much more proficient in its use. The second study showed that slight changes in the wording used by the robot had a significant effect on users' engagement in human-robot interactions.

1.2 Our Approach

In this paper we go beyond most of the work mentioned above. We develop methodologies for natural language communication between a robot and its human controller in the context of the human controller directing the robot to perform certain tasks. This communication involves two parts:

The human communicating with the robot and the robot receiving that input and processing it: The human to robot communication that we consider can be categorized to four types.[1] (a) The human may direct the robot to do a certain task. (b) The human may provide knowledge for the robot to learn (or teach the robot) in the form of facts or new actions. (c) The human may ask a question to the robot. (d) The human may verbally respond to a query by the robot.

The robot communicating with the human: The robot to human communication involves the following types. (a) It answers the human's questions, often involving what it senses. (b) It reports what it has done and could not do. (c) It asks a question to the human regarding what it should do. This could be about stating multiple actions that it could do and asking which one the human would prefer. This could be about remembering some previously assigned goal and sensing an opportunity to achieve that goal, even though the human director has moved on with respect to the goals. This could be about making sure that the plan it has made is acceptable to the human. In that case, the robot can state what it is planning to do and ask for confirmation from the human.

To achieve the above communication, we need to develop formal languages linked with natural language. In particular, a Robot Input Language (RIL) and a Robot Output Language (ROL) are needed along with the ability to translate statements uttered by humans in a natural language to the RIL and then translate statements in ROL to a natural language to be communicated to the human.

2. Motivating Example

The following dialogue is a hypothetical conversation between a human director and a robotic searcher to simulate the types of communication that would be expected within human-robot interactions. The director can only see a map and communicate verbally with the searcher which is physically in the mapped environment.

In this scenario, the robot, carrying a yellow block, is situated in a long east to west hallway with a room to the north. Inside the room is a green box with the number 7 on the side. On the north end of the room is another hallway stretching east to west with a pink box on either end. The following is a possible conversation between the human director and the robot in order to find the green box and its number, then place a yellow block within one of the pink boxes.

Director: While going down the hallway, find the door on your right and go through it.
Searcher: done.
Director: Is there a green box in the room?
Searcher: yes.
Director: Go to the box.
Searcher: done.
Director: What is the number on the box.
Searcher: 7.
Director: Report the location of this box.
Searcher: Go down hallway. Enter first door on the right. Box is ahead on the left.
Director: Go through the door.
Searcher: door on left or on right?
Director: On right.
Searcher: done.
Director: Either go left or go right.
Searcher: choosing left. done.
Director: Go to the end of the hall. You should see a pink box.
Searcher: done. There is a pink box.
Director: Place a yellow block in the pink box.
Searcher: done.

1. More types of communications are needed when communicating with multiple robots at the same time. That is part of an ongoing work.

3. The Robot Input Language - RIL

As previously mentioned, RIL has four parts: Directives, Learning, Queries, and Answers. We propose an extension of Golog (Levesque et al., 1997) with temporal logic and "goal" statements as the language for Directives (RIL-D) and a database query language as the language for Queries (RIL-Q). The language for Learning (RIL-L) is composed of a logical syntax for learning about the world as well as language constructs for learning actions and goals for other agents. The syntax for the RIL Answer to a question (RIL-A) is determined by the specific question asked. This paper will be primarily scoped to RIL-D while providing a high level overview of the other languages.

3.1 RIL-D

The RIL-D language is for the human to specify a directive to the robot, and since they are in an interactive setting, the human expects not only the robot to act on that directive, but also to give a verbal response to the human. The directive given to the robot may specify exact actions that need to be executed, may have a sequence of steps to be taken, may have iterative statements, may have conditional actions, may specify non-deterministic choices, may specify certain goals that needs to be achieved, and may include observational commands.

The responses expected from the robot include: confirmation that a directed action was executed or a goal was achieved; refusal that an action could not be executed or a goal could not be achieved; a result of an observational command; a question back to the human, when the human interrupts with a contradictory or confusing request while the robot is executing a previous directive; and a question to the human, when the robot faces multiple choices and cannot decide which one to take.

As an illustration, given the scenario in which the robot is at the start of a hallway, the directive "Continue all the way to the end and then turn right" will result in the robot returning the confirmation "done" after reaching the end of the hallway and completing the action to turn to the right. Given a variant of the scenario above in which the robot is at the start of an impassable hallway, possibly filled with debris, the directive "Continue all the way to the end and then turn right" will result in the robot returning the refusal "failed" since it cannot complete the directive. If there is a green box numbered with a 7, the observational command "What is the number on the green box?" will return the value "7" after the robot has observed the number on the box. If the robot is given contradictory commands such as the goal "Go to the door ahead" followed by "Never go near a door" the robot would recognize that the second command prevents the completion of the first and would request the human to "clarify" and provide feedback on what commands to obey. For the situation with the robot in a hallway in which the path ahead of it forks to the left and right, the directive "Continue down the hallway" would result in the robot returning the question "left or right" then awaiting the human's reply.

3.2 The syntax of RIL-D

The syntax of RIL-D is specified in Table 1 along with brief descriptions. Syntactically, RIL-D takes Golog, removes test actions and adds temporal formulas and $goal(self, \phi)$ constructs. In Golog a test action forces one to chose the trajectory corresponding to the program before the test

	Syntax	Intuitive Meaning
1	Simple Action: An action a is an RIL-D program.	Execute action a, such as *turn_right*
2	Parameterized Action: If $f(X_1 \dots X_n)$ is a formula and $a(X_1 \dots X_n)$ is an action, $a(X_1 \dots X_n) : f(X_1 \dots X_n)$ is an RIL-D program.	Execute action $a(X_1 \dots X_n)$ where $X_1 \dots X_n$ satisfy the formula $f(X_1 \dots X_n)$
3	Parallel Action: If a and b are actions (simple, parameterized, or parallel), then $a \| \| b$ is an RIL-D program.	Execute actions a and b in parallel.
4	Sensing: If $X_1 \dots X_n$ are variables of sorts $s_1 \dots s_n$ and $f(X_1 \dots X_n)$ is a formula, then $sense(X_1 \dots X_n) : f(X_1 \dots X_n)$ is an RIL-D program.	Sense the values of $X_1 \dots X_n$ where $X_1 \dots X_n$ satisfy the formula $f(X_1 \dots X_n)$.
5	Observational Command: If ψ is a formula, then $sense() : \psi$ is an RIL-D program.	Sense whether or not the suggested observation ψ holds true in the current state of the world.
6	Self Goal: If $self$ is the robot agent, $\phi(X_1 \dots X_n)$ is a temporal formula, and $f(X_1 \dots X_n)$ is a formula, then $goal(self, \phi(X_1 \dots X_n)) : f(X_1 \dots X_n)$ is an RIL-D program.	Create and execute a plan for $self$ to satisfy $\phi(X_1 \dots X_n)$ where $X_1 \dots X_n$ satisfy the formula $f(X_1 \dots X_n)$.
7	Sequence: If a and b are RIL-D programs, then $a; b$ is an RIL-D program.	Execute the RIL-D program a immediately followed by the second program. b
8	Choice: If a and b are RIL-D programs, then $a \mid b$ is an RIL-D program.	Execute either RIL-D program a or b but not both.
9	Parametric Choice: If X_1 is a variable of sort s, $p(X_1 \dots X_n)$ is a program, and $f(X_1 \dots X_n)$ is a formula, then $pick(X_1, p(X_1 \dots X_n)) : f(X_1 \dots X_n)$ is an RIL-D program.	Choose one object X_1 matching the conditions specified in $f(X_1 \dots X_n)$ and then execute program $p(X_1 \dots X_n)$ given the chosen object.
10	Condition: If a and b are RIL-D programs and ϕ is a temporal formula, then $if\ \phi\ then\ a\ else\ b$ is an RIL-D program.	If the conditions specified in ϕ hold true, execute program a, otherwise execute program b.
11	While: If a is an RIL-D program and ϕ is a past time linear temporal logic formula, then $while\ \phi\ do\ a$ is an RIL-D program.	Check if the conditions specified in ϕ hold true, and if so, execute program a then repeat the process.

Table 1. The Syntax of RIL-D

action in such a way that the test action holds true. This language does not allow such planning via test actions. Planning is directly specified via $goal(self, \phi)$. However, the language does have observational commands which are similar to test actions but the purpose is that the human director may command the robot to make an observation which is then returned as the value observed or a failure to make the requested observation. The value to be returned by the observation is not known until the observation action is actually performed.

We now illustrate each of these constructs through the following examples and translations, with the example numbers corresponding to Table 1:

(1) "Turn 90 degrees to the right": $turn\_right$.

(2) "Proceed through the doorway": $go\_through(X) : is(X, door)$.

(3) "Push the door while turning to the left": $push(X) : is(X, door) || turn\_left$.

(4) "What is the number on the green box": $sense(X) : has(Y, number\_on, X) \wedge$
$has(Y, color, green) \wedge is(Y, box)$.

(5) "Is there a chair in front of you": $sense() : is(X, chair) \wedge has(X, location, front)$.

(6) "Go out of the room": $goal(self, \Diamond \neg has(self, at, X)) : has(self, at, X) \wedge is(X, room)$.

(7) "Go through the door and then turn right": $go\_through(X) : is(X, door); turn\_right$.

(8) "You can either turn right and pick up the blue box or turn left and pick up the pink box":
$(turn\_right; (pick\_up(X) : has(X, color, blue) \wedge is(X, box))) \mid$
$(turn\_left; (pick\_up(Y) : has(Y, color, pink) \wedge is(Y, box)))$.

(9) "Select one yellow block and place it in a box":
$pick(Y, (put\_in(Y, X) : is(X, box))) : is(Y, block) \wedge has(Y, color, yellow)$.

(10) "If there is a door on your right, go through it, otherwise turn around":
$if\ sense() : has(X, location, right) \wedge is(X, door)$
$then\ go\_through(X)$.
$else\ turn\_around$.

(11) "Continue all the way to the end of the hallway":
$while\ \neg has(self, at\_end, X) \wedge is(X, hall)$
$do\ go\_straight\_one\_step$.

Returning to the more complex example from the motivation, a close translation of the Director's commands can be written as:

- $while\ \neg sense() : has(X, location, right) \wedge is(X, door)$ • $do\ go\_straight\_one\_step$.
- $go\_through(X) : is(X, door) \wedge has(X, location, right)$.
- $sense() : has(self, at, Y) \wedge is(Y, room) \wedge is(X, box) \wedge has(X, color, green) \wedge has(X, at, Y)$.
- $goal(self, \Diamond has(self, at, X)) : is(X, box) \wedge has(X, color, green)$.
- $sense(X) : has(Y, number\_on, X) \wedge has(Y, color, green) \wedge is(Y, box)$.
- $go\_through(X) : is(X, door) \wedge has(X, location, right)$.
- $turn\_left \mid turn\_right$.
- $while\ \neg has(self, at\_end, X) \wedge is(X, hall)$ • $do\ go\_straight\_one\_step$.
- $sense() : is(X, box) \wedge has(X, color, pink) \wedge has(X, location, front)$.
- $pick(Y, (put\_in(Y, X) : is(X, box) \wedge has(X, color, pink))) : is(Y, block) \wedge has(Y, color, yellow)$.

3.3 The semantics of RIL-D

The semantics of an RIL-D program in essence say what are the valid ways in which the world will progress. It provides the information about the action execution and the responses given by the robot. The semantics would consider an initial state s_0[2] and generate a set of possible trajectories for a given RIL-D program, each consisting of $t_1, \ldots t_n$, where t_i is an action or a response of the robot.

For example, suppose that in s_0 the fluents $is(h1, hall)$, $has(self, at\_end, h1)$, $has(self, at, h1)$, $is(d1, door)$ hold true, however $has(d1, location, right)$ has not yet been sensed and the following RIL-D program is given:

$if\ sense() : has(X, location, right) \wedge is(X, door)$
$then\ go\_through(X).$
$else\ turn\_around.$

The two possible trajectories for $t_1, \ldots, t_n$, both with $n = 2$ and ROL-R response $R(X)$, are: $t_1 = go\_through(X)$, $t_2 = R(done)$ OR $t_1 = turn\_around$, $t_2 = R(done)$.

We now give a more formal definition where within the set of possible trajectories, each trajectory $t_1, \ldots t_n$, denoted by α, is a trace of a program p (which may have ψ, and the temporal formula ϕ), and contains the ROL-R response $R(X)$:

1. for $p = a$ where a is an action, if the executability conditions for a are satisfied in s_0, then $n = 2$, $t_1 = a$, and $t_2 = R(done)$, otherwise $n = 1$ and $t_1 = R(failed)$.

2. for $p = a(X_1 \ldots X_m) : f(X_1 \ldots X_m)$ where $a(X_1 \ldots X_m)$ is an action, if both $f(X_1 \ldots X_m)$ and the executability conditions for $a(X_1 \ldots X_m)$ are satisfied in s_0, then $n = 2$, $t_1 = a$, and $t_2 = R(done)$, otherwise $n = 1$ and $t_1 = R(failed)$.

3. for $p = a||b$ where a and b are actions of the types a, $a(X_1 \ldots X_m)$, or $a||b$, if the executability conditions for a and b are satisfied in s_0, then $n = 2$, $t_1 = \{a, b\}$, and $t_2 = R(done)$, otherwise $n = 1$ and $t_1 = R(failed)$.

4. for $p = sense(X_1 \ldots X_m) : f(X_1 \ldots X_m)$, $n = 1$ and if there exists values $v_1 \ldots v_m$ of the sorts $X_1 \ldots X_m$ such that $f(v_1 \ldots v_m)$ holds in s_0, then $t_1 = R(v_1 \ldots v_m)$, otherwise $t_1 = R(failed)$.

5. for $p = sense() : \psi$, $n = 1$ and if ψ holds in s_0, then $t_1 = R(yes)$, otherwise $t_1 = R(no)$.

6. for $p = goal(self, \phi(X_1 \ldots X_m)) : f(X_1 \ldots X_m)$, α is a trace of p such that $f(X_1 \ldots X_m)$ holds in s_0, α satisfies $\phi(X_1 \ldots X_m)$, $t_1 = R(acknowledged)$, and $t_n = R(done)$. If no α exists that satisfies ϕ then $t_1 = R(failed)$.

7. for $p = p_1; p_2$, if there exists an i such that $s_0, t_1, \ldots, t_i$ is a trace of p_1, and $t_i, \ldots, t_{n-1}$ is a trace of p_2, then $t_n = R(done)$, otherwise $t_1 = R(failed)$.

2. This can be generalized to a history of states and actions of the form $s_0, a_1, s_1, a_2, \ldots, s_m$, if future commands may need to look back to history.

8. for $p = p_1 \mid p_2$, if α is a trace of p_1 or if α is a trace of p_2 then $t_n = R(done)$, otherwise $t_1 = R(failed)$.

9. for $p = pick(X_1, q(X_1 \dots X_m)) : f(X_1 \dots X_m)$, if there exists values $v_1 \dots v_m$ of the sorts $X_1 \dots X_m$ such that $f(v_1 \dots v_m)$ holds in s_0 and $t_1, \dots, t_{n-1}$ is a trace of $q(v_1 \dots v_m)$, then $t_n = R(done)$, otherwise if there does not exist such $v_1 \dots v_m$, then $n = 1$ and $t_1 = R(failed)$.

10. for $p = if\ \phi\ then\ p_1\ else\ p_2$, either α is a trace of p_1 with $t_n = R(done)$ if ϕ is satisfied by the history $s_0, a_1, \dots, s_m$ or α is a trace of p_2 with $t_n = R(done)$ if ϕ is not satisfied by the history $s_0, a_1, \dots, s_m$.

11. for $p = while\ \phi\ do\ p_1$, either $n = 1$ with $t_1 = R(done)$ and ϕ is not satisfied by the history trace $s_0, a_1, \dots, s_m$ or ϕ is satisfied by the trace of the history $s_0, a_1, \dots, s_m$ and there exists some $i <= n$ such that $s_m, t_1, \dots, t_i$ is a trace of p_1 and $s_0, a_1, \dots, s_m$, $seq\_act\_state(t_1, \dots, t_i)$ is a trace of the new history of p and α is a trace of p with $t_n = R(done)$.

In the last item, $seq\_act\_state(t_1, \dots, t_n)$ results in a sequence of alternating actions and states a_{m+1}, $s_{m+1}, \dots, s_i$ that corresponds to the trace, $t_1, \dots, t_n$ such that after completing $t_1, \dots, t_n$, the history would be of the form $s_0, a_1, \dots, s_m, a_{m+1}, s_{m+1}, \dots, s_i$ where s_i is the present state. When computing $seq\_act\_state(t_1, \dots, t_n)$ only the actions within $t_1, \dots, t_n$ are considered as the response (of the robot) do not alter the state of the robot.

The following extend some of the syntax examples to demonstrate the progression of the world in which $R(X)$ is an ROL-R response:

1. Suppose that in s_0 the fluents $is(b1, box)$, $has(b1, number\_on, 7)$, $has(b1, color, green)$, $has(self, at, r1)$, $is(r1, room)$ hold true and the following RIL-D program is given:
   $sense(X) : has(Y, number\_on, X) \wedge has(Y, color, green) \wedge is(Y, box)$.
   The trajectory $t_1, \dots, t_n$ will be $t_1 = R(7)$ with $n = 1$.

2. Suppose that in s_0 the fluents $has(self, at, r1)$, $is(r1, room)$, $is(b1, box)$, $has(b1, location, right)$, $has(b1, color, blue)$, $is(b2, box)$, $has(b2, color, pink)$, $has(b2, location, left)$ hold true and the following RIL-D program is given:
   $(turn\_right; (pick\_up(X) : has(X, color, blue) \wedge is(X, box))) \mid$
   $(turn\_left; (pick\_up(Y) : has(Y, color, pink) \wedge is(Y, box)))$.
   There are two possible trajectories for $t_1, \dots, t_n$ both with $n = 3$: $t_1 = turn\_right$, $t_2 = pick\_up(b1)$, $t_3 = R(done)$ OR $t_1 = turn\_left$, $t_2 = pick\_up(b2)$, $t_3 = R(done)$.

3. Suppose that in s_0 the fluents $has(self, at, r1)$, $is(r1, room)$, $is(b1, box)$, $is(bl1, block)$, $has(bl1, color, yellow)$, $is(bl2, block)$, $has(bl2, color, yellow)$, $has(self, picked\_up, bl1)$, $has(self, picked\_up, bl2)$ hold true and the following RIL-D program is given:
   $pick(Y, (put\_in(Y, X) : is(X, box))) : (is(Y, block) \wedge has(Y, color, yellow))$.
   There are two possible trajectories for $t_1, \dots, t_n$ both with $n = 2$: $t_1 = put\_in(bl1, b1)$, $t_2 = R(done)$ OR $t_1 = put\_in(bl2, b1)$, $t_2 = R(done)$.

3.4 RIL-L

The RIL-L language is for the human to impart knowledge to the robot. The knowledge given to the robot may be in the form of a description of the environment, new actions the robot can perform, or the goals of other agents in the world.

To give a new description, if $h(X_1, \ldots, X_n)$ is a predicate and $b(X_1, \ldots, X_n)$ is a formula then $h(X_1, \ldots, X_n)$: $-b(X_1, \ldots, X_n)$ is an RIL-L statement in which if $b(X_1, \ldots, X_n)$ is satisfied, then $h(X_1, \ldots, X_n)$ holds true.

To teach a new action, if $u(X_1, \ldots, X_n)$ is an unknown action, $a_1, \ldots, a_m$ is an ordered list of existing actions (simple, parameterized from the variables $X_1, \ldots, X_n$, or parallel), and $f(X_1, \ldots, X_n)$ is a formula then $u(X_1, \ldots, X_n) \leftarrow a_1, \ldots, a_m : f(X_1, \ldots, X_n)$ is an RIL-L statement. This can also be given without parameters for a simple action that is simply a sequence of previously known actions.

To give knowledge of other agents' goals, if a is a non-$self$ agent, $\phi(X_1, \ldots, X_n)$ is a temporal formula and $f(X_1, \ldots, X_n)$ is a formula, then $goal(a, \phi(X_1, \ldots, X_n)) : f(X_1, \ldots, X_n)$ is an RIL-L statement which states that a is executing a plan to satisfy $\phi(X_1, \ldots, X_n)$ where $X_1, \ldots, X_n$ satisfy the formula $f(X_1, \ldots, X_n)$.

3.5 RIL-Q

The syntax of RIL-Q is as follows: $query(\lambda X_1. \ldots \lambda X_n.\Phi(X_1, \ldots, X_n))$ where $\Phi(X_1, \ldots, X_n)$ is a first-order logic formula with free variables $X_1, \ldots X_n$.

Intuitively, $query(\lambda X_1. \ldots \lambda X_n.\Phi(X_1, \ldots, X_n))$ expresses the query of $\{(X_1, \ldots, X_n) \mid \Phi(X_1, \ldots, X_n)\}$ is true with respect to the current state. In the future, we will explore the need of generalizing $\Phi(X_1, \ldots, X_n)$ to have temporal constructs.

For example, consider the query "In what room was box number 7?" This can be expressed as $query(\lambda R.is(R, room) \wedge is(X, box) \wedge has(X, number\_on, 7) \wedge has(X, at, R))$.

3.6 RIL-A

The RIL-A language is the formal representation of the answer given by the human to a previous question from the robot in the language ROL-Q.

The response given in RIL-A will depend on the specific type of query. A "$select$" query would take a response of the type $(a_1, \ldots, a_n)$ to state which of the suggested sets of actions is desired. If none of the plans is desired, a new plan of the same form, $(a_1, \ldots, a_n)$, can be provided instead. A "$Should\ I\ do$" query would be answered simply with yes or no.

4. The Robot Output Language - ROL

The ROL has three parts: Reports, Answers, and Questions. As previously discussed, this paper is primarily scoped to the RIL-D language, so here we give a high level overview of the output languages.

4.1 ROL-R

When the robot receives a command or a directive in the language RIL-D it processes that command and may respond to that directive. For example, it may say that the given command is not doable and why (there is no door on the right to take); or that it has executed that command.

Presently, the semantics of RIL-D only consist of simple replies such as yes, no, $done$, and $failed$. Future work will extend responses to indicate why commands may have failed or whether something unexpected has occurred.

4.2 ROL-A

When the robot receives a query in the language RIL-Q it may answer it by Yes, No, or when the RIL-Q question is $\lambda X_1, \ldots \lambda X_n \Phi(X_1, \ldots, X_n)$, with $n \geq 1$ then it may give the value of $X_1, \ldots, X_n$.

To continue the example from section RIL-Q: Given that the robot has previously seen the box with the number 7 inside of room r3, the result to the query "$query(\lambda R.is(R, room) \wedge is(X, box) \wedge has(X, number\_on, 7) \wedge has(X, at, R))$" would simply be $r3$.

4.3 ROL-Q

The syntax of ROL-Q is as follows where a is an action, ϕ is a goal, and $\Phi(X_1, \ldots, X_n)$ is a first-order logic formula with free variables $X_1, \ldots X_n$:

- $select((a_{11}; \ldots; a_{1n_1}), ..., (a_{k1}; \ldots; a_{kn_k}))$ can request which plan of action to execute.
- $should(a_1; \ldots; a_n)$ OR $should(a_1; \ldots; a_n, \phi)$ can request if sequences of actions should be taken.
- $clarify(\phi)$ for an in-completable goal OR $clarify(\phi_2, \phi_1)$ for a conflict between goals.
- $\lambda X_1 \ldots . \lambda X_n . \Phi(X_1, \ldots, X_n)$ can request knowledge.

For example: When given the choice between picking up a box on the right and a box on the left, the robot may query $select((turn\_right; pick\_up(b1)), (turn\_left; pick\_up(b2)))$

Similarly, the robot could return the query: $should(turn\_right; pick\_up(b1), (pick\_up(X) : is(X, box) \wedge (has(X, location, right) \vee has(X, location, left))))$

5. Implementation and Experimental Validation

Earlier in Section 2 we gave an hypothetical motivational example of a dialogue. We implemented and experimentally validated our approach using the multi-modal CReST corpus consisting of human-human dialogues in an "instruction-following" task (Eberhard et al., 2010) and an Urban Search and Rescue (USAR) scenario.

The dialogue shown in Table 2 is an example of a USAR scenario. In this scenario shown in Figure 1, Cindy is the robot which will interact with Commanders X, Y, and Z. Initially, Commander X and Cindy are together, Commander Y is in the hallway, and Commander Z is with an injured

civilian. Commander X wants Cindy to find a medical kit and bring it to Commander Z, but while doing this, Cindy must avoid being seen by the enemy. Upon entering the hallway, Commander Y will order Cindy to follow him. Cindy will recognize the conflict and request clarification on which goal to follow. She will then find the medical kit and bring it to Commander Z, but will be detected and damaged on the way. After requesting help from Commander Z, her goal to remain undetected is overridden so that she can return to Commanders X and Y.

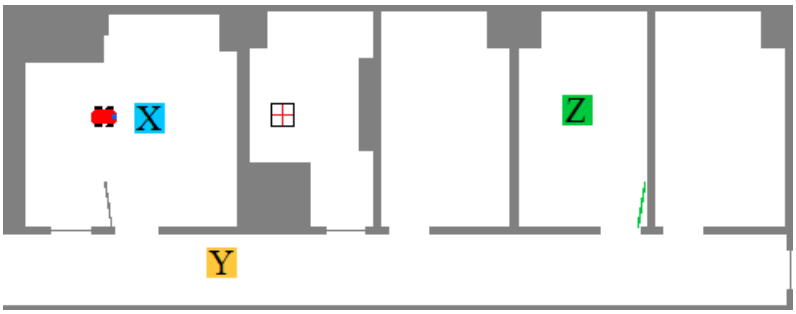

Figure 1. USAR Environment

We used a Lambda calculus based approach to translate English to the formal language of RIL-D. In this approach the meanings of words are given as Lambda calculus formulas and from that the meaning (or formal representation of) phrases and sentences are computed. We first used such an approach in (Dzifcak et al., 2009), but later ran into the difficulty of coming up with Lambda calculus representation of words. This led us to come up with an Inverse Lambda algorithm (Baral et al., 2011) using which Lambda calculus representation of words could be inferred in an inverse manner from examples of sentences and their formal representation. More recently we have developed the NL2KR (Natural Language to Knowledge Representation) framework (Nguyen et al., 2015) that can be used to develop translation systems from natural language to specific formal languages.

To implement RIL-D we used Answer Set Programming (ASP) (Gelfond & Lifschitz, 1988; Baral, 2003). The ASP rules were designed to generate a series of instructions formatted for the Agent Development Environment (ADE) robot simulator (Kramer & Scheutz, 2006). The ASP rules were designed to support the majority of RIL-D, namely the following syntactic constructs: Simple Action, Parameterized Action, Self Goal, Sequence, Choice, Parametric Choice, Condition, and While. Sensing was partially implemented by requiring the simulated robot to be at the same location as the object it was trying to sense with the state of the object pre-programmed. Parallel actions were excluded to simplify the generated plans for ease of validation but could easily be enabled by removing the ASP rule that prevented simultaneous actions.

Following is a small subset of ASP rules that was used to express the RIL-D semantics. Based on the work in (Son et al., 2001) a predicate $trans(p, t_1, t_n)$ is defined, which holds in an answer set S iff $s(t_1), a_{t_1}, \ldots, s(t_n)$ is a trace of p where $s(i) = \{holds(f, i) \in S \mid f \text{ is a fluent}\}$ and a_i is either an action or response-set such that $occ(a_i, i) \in S$ indicates that action a_i occurs at time interval i. The other predicates used in the ASP rules are defined in Table 3

$trans(null, T, T) \quad \leftarrow time(T).$

	Dialogue	Translation
X	Cindy, CmdrZ really needs a medical kit.	$goal(cmdrZ, \Diamond has(cmdrZ, have, M)) : is(M, med\_kit)$
X	There should be one in the first room in the hallway down to the left	$is(m, med\_kit) \wedge has(m, at, r1) \wedge is(r1, room) \wedge is(z, room) \wedge has(z, past, r1) \wedge is(h, hallway) \wedge has(r1, connected, h) \wedge has(z, connected, h)$
C	OK.	$acknowledged$
C	Should I get it for CmdrZ?	$goal(self, \Diamond has(cmdrZ, have, m))$
X	Yes.	$goal(self, \Diamond has(cmdrZ, have, m))$
X	He is in the room with the green door.	$has(cmdrZ, at, r2) \wedge is(r2, room) \wedge is(d, door) \wedge has(d, doorconnected, r2) \wedge has(d, color, green)$
C	OK.	$acknowledged$
X	But remain undetected.	$goal(self, \Box(has(self, state, undetected)))$
C	OK.	$acknowledged$
C	What is a medical kit?	$query(\lambda X.has(med\_kit, appearance, X))$
X	It's a white box with a red cross.	$has(m, color, white) \wedge is(m, box) \wedge has(m, visual\_feature, z) \wedge has(z, color, red) \wedge is(z, cross)$
C	Alright, I'm on my way.	$acknowledged$
Y	Cindy, follow me.	$goal(self, \Box has(self, follow, cmdrY))$
C	I really need to deliver the medical kit to CmdrZ.	$clarify(goal(self, \Box has(self, follow, cmdrY)), goal(self, \Diamond has(cmdrZ, have, M)) : is(M, med\_kit))$
Y	OK.	$acknowledged$
Y	I'm going to see X now.	$goal(cmdrY, \Diamond has(cmdrY, at, cmdrX))$
C	OK.	$acknowledged$
Y	Meet me when you are done.	$goal(self, \Diamond(has(self, state, available) \wedge \Diamond has(self, at, cmdrY)))$
C	OK.	$acknowledged$
C	CmdrX, the door is closed, what shall I do?	$clarify(goal(self, \Diamond has(cmdrZ, have, M)) : is(M, med\_kit) \wedge is(d, door) \wedge has(d, state, closed))$
X	Use your hand to push it.	$goal(self, \Diamond push\_with(d, selfs\_hand))$
C	Got it.	$done$
C	CmdrZ, take the medical kit, my arm motors are not working.	$\neg(functional(self\_arm\_motors)) \rightarrow take(CmdrZ, x) \wedge is(x, med\_kit)$
Z	Thank you Cindy.	$done$

Table 2. Urban Search and Rescue Dialogue

	Predicate	Intuitive Meaning
1	$time(X)$.	X represents a single point in time.
2	$action(X)$.	X represents a valid action.
3	$leq(X_1, X_2)$.	X_1 and X_2 are two time points in which X_1 is smaller than X_2.
4	$goal(X_1, X_2)$.	X_1 is a program that is satisfied in time X_2.
5	$htf(X_1, X_2)$.	The temporal formula X_1 holds in time X_2.
6	$proc(X)$.	X is a procedure consisting of a head and a tail.
7	$head(X_1, X_2)$.	X_1 is a procedure with the head: program X_2.
8	$tail(X_1, X_2)$.	X_1 is a procedure with the tail: program X_2.
9	$choiceAction(X)$.	X is a choice action consisting of possible programs represented by the $in(X_1, X_2)$ predicate.
10	$in(X_1, X_2)$.	X_1 is a program within the list of possible programs X_2.
11	$choiceArgs(X_1, X_2, X_3)$.	X_1 is a program in which formula X_2 holds at the current time and program X_3 is executed.
12	$hf(X_1, X_2)$.	Formula X_1 holds at time X_2.
13	$if(X_1, X_2, X_3, X_4)$.	X_1 is a program in which either program X_3 is executed if formula X_2 holds at the current time otherwise program X_4 is executed.
14	$while(X_1, X_2, X_3)$.	X_1 is a program in which so long as formula X_2 holds, program X_3 will be executed.

Table 3. The ASP Predicates

$$
\begin{aligned}
trans(A,T,T+1) &\leftarrow time(T), action(A), A \neq null, occ(A,T).\\
trans(A,T1,T2) &\leftarrow time(T1), time(T2), leq(T1,T2),\\
&\quad goal(A,TF), htf(TF,T2).\\
trans(A,T,T) &\leftarrow time(T), goal(A,TF), htf(TF,T).\\
trans(P,T1,T2) &\leftarrow time(T1), time(T2), leq(T1,T2),\\
&\quad time(T3), leq(T1,T3),\\
&\quad leq(T3,T2), proc(P),\\
&\quad head(P,P1), tail(P,P2),\\
&\quad trans(P1,T1,T3),\\
&\quad trans(P2,T3,T2).\\
trans(N,T1,T2) &\leftarrow time(T1), time(T2), leq(T1,T2),\\
&\quad choiceAction(N), in(P1,N),\\
&\quad trans(P1,T1,T2).\\
trans(S,T1,T2) &\leftarrow time(T1), time(T2), leq(T1,T2),\\
&\quad choiceArgs(S,F,P), hf(F,T1),\\
&\quad trans(P,T1,T2).
\end{aligned}
$$

$$
\begin{aligned}
trans(I, T1, T2) \quad &\leftarrow time(T1), time(T2), leq(T1, T2), \\
&\quad if(I, F, P1, P2), hf(F, T1), \\
&\quad trans(P1, T1, T2). \\
trans(I, T1, T2) \quad &\leftarrow time(T1), time(T2), leq(T1, T2), \\
&\quad if(I, F, P1, P2), not\ hf(F, T1), \\
&\quad trans(P2, T1, T2). \\
trans(W, T1, T2) \quad &\leftarrow time(T1), time(T2), leq(T1, T2), \\
&\quad while(W, F, P), hf(F, T1), \\
&\quad time(T3), leq(T1, T3), \\
&\quad leq(T3, T2), trans(P, T1, T3), \\
&\quad trans(W, T3, T2). \\
trans(W, T, T) \quad &\leftarrow time(T), while(W, F, P), not\ hf(F, T).
\end{aligned}
$$

ASP rules were used as a planning system that was connected to the ADE in the following manner: the ADE simulator generates the initial appearance of the world and waits for a command from the user to be sent to the ASP system for plan generation. Once the ASP implementation of RIL-D returns a plan, the simulator executes the plan with the corresponding ROL output. Following execution, the ADE system continues waiting for further commands from the user to repeat the process starting from the current state of the simulated world. This allowed us to not only confirm the completeness of the RIL and ROL languages in supporting the CReST and USAR corpus, but also to ensure successful task execution. Here we were only able to give small glimpses of our implementation and validation. Additional details of both are available at (Lumpkin, 2012).

5.1 Integrating RIL and ROL in a Robot Architecture: Ongoing and future work

We have begun to address the challenges of natural language dialogues in the context of the integrated robotic DIARC architecture which has been used successfully in a variety of human subject HRI experiments (Brick & Scheutz, 2007; Scheutz et al., 2006). DIARC integrates cognitive tasks such as natural language understanding and complex action planning and sequencing with lower level activities such as multi-modal perceptual processing. The natural language processing components include algorithms for human-like incremental reference resolution (Scheutz et al., 2004) and dialog-like human-robot interactions with simple forms of backchannel feedback such as nodding or saying "okay" (Brick & Scheutz, 2007). Natural language understanding is tightly coupled with action execution (Brick et al., 2007), a pre-requisite for the robot's ability to start actions quickly (e.g., nodding). It also includes algorithms for handling disfluencies, in particular, lexical disfluencies, abandoned utterances, repetitions, as well as some repairs and corrections, in the context of spoken instruction understanding (Cantrell et al., 2010).

The natural language understanding systems in DIARC are being updated to automatically convert natural language instructions from a human operator into a subset of RIL-D, RIL-L, RIL-Q, and RIL-A. The conversion into logical forms is effected by combining lexical items with syntactic annotations from a combinatorial categorial grammar (CCG) and logical semantic annotations. Repeated λ-conversions then lead to λ-free logical formulas that represent meanings (e.g., the goals and actions specified in the natural language instruction). Once a goal is recognized, DI-

ARC searches for known action sequences that achieve the goal which it then executes (otherwise, it sends the goal description to the planner which produces a new plan to achieve it). In addition to executing the actions, DIARC supports producing output that can be in the format of ROL-R, ROL-A and ROL-Q.

6. Conclusion

In a human-robot interaction scenario one of the important modes of communication is via natural language. To facilitate this communication, in this paper, we proposed a formal high level language with multiple components. Our proposed language has two main parts: RIL and ROL which refer to the Robot Input Language and the Robot Output Language. The RIL has four parts RIL-D, RIL-L, RIL-Q and RIL-A, which express directives, learning, queries and answers, respectively. The ROL has three parts ROL-R, ROL-A and ROL-Q, which express responses, answers (to queries), and queries, respectively. The syntax and semantics of each of these seven sub-languages are based on their needs, and for some of them we borrow constructs from the literature and make appropriate modifications. For example, the RIL-D language borrows several constructs from GOLOG, but at the same time avoids features from GOLOG that we considered to be inappropriate from an HRI viewpoint. We validated the usefulness and expressive-completeness of our language (in terms of the features it has) by going over a corpus of human-human dialogues that simulated HRI involving remote collaboration and showing that the conversation in that corpus can be expressed in our language then executed in a simulated robot environment. We have also begun embedding our language into the integrated robot architecture DIARC where natural language instructions to the robot will be translated to a subset of our language that is understood by DIARC, and the output of DIARC can be mapped to constructs in our language which then get translated to natural language.

References

Baral, C. (2003). *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press.

Baral, C., Gonzalez, M., Dzifcak, J., & Zhou, J. (2011). Using inverse λ and generalization to translate english to formal languages. *Proceedings of the International Conference on Computational Semantics, Oxford, England, January*.

Brick, T., Schermerhorn, P., & Scheutz, M. (2007). Speech and action: Integration of action and language for mobile robots. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1423–1428). San Diego, CA.

Brick, T., & Scheutz, M. (2007). Incremental natural language processing for hri. *HRI* (pp. 263–270).

Cantrell, R., Scheutz, M., Schermerhorn, P., & Wu, X. (2010). Robust spoken instruction understanding for hri. *Proceedings of the 2010 Human-Robot Interaction Conference*.

Clodic, A., Alami, R., Montreuil, V., Li, S., Wrede, B., & Swadzba, A. (2007). A study of interaction between dialog and decision for human-robot collaborative task achievement. *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*

(pp. 913–918). IEEE.

Dzifcak, J., Scheutz, M., Baral, C., & Schermerhorn, P. W. (2009). What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. *ICRA* (pp. 4163–4168).

Eberhard, K., Nicholson, H., Kuebler, S., Gundersen, S., & Scheutz, M. (2010). The indiana cooperative remote search task (crest) corpus. *Proceedings of LREC 2010: Language Resources and Evaluation Conference*. Malta.

Fischer, K. (2011). How people talk with robots: Designing dialog to reduce user uncertainty. *AI Magazine*, *32*, 31–38.

Fong, T., Kunz, C., Hiatt, L., & Bugajska, M. (2006). The human-robot interaction operating system. *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (pp. 41–48). ACM.

Fong, T., et al. (2005). The peer-to-peer human-robot interaction project. *Space*, *6750*.

Gelfond, M., & Lifschitz, V. (1988). The stable model semantics for logic programming. *Logic Programming: Proc. of the Fifth Int'l Conf. and Symp.* (pp. 1070–1080). MIT Press.

Ji, J., Fazli, P., Liu, S., Pereira, T., Lu, D., Liu, J., Veloso, M., & Chen, X. (2016). Help me! sharing of instructions between remote and heterogeneous robots. *Social Robotics: 8th International Conference, ICSR 2016, Kansas City, MO, USA, November 1-3, 2016 Proceedings* (pp. 786–795). Springer.

Kramer, J., & Scheutz, M. (2006). ADE: A framework for robust complex robotic architectures. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4576–4581). Bejing, China.

Levesque, H., Reiter, R., Lespérance, Y., Lin, F., & Scherl, R. (1997). Golog: A logic programming language for dynamic domains. *The Journal of Logic Programming*, *31*, 59–83.

Lumpkin, B. (2012). *A high level language for human robot interaction*. Master's thesis, Arizona State University.

Nguyen, V., Mitra, A., & Baral, C. (2015). The NL2KR Platform for building Natural Language Translation Systems. *Proc. of ACL 2015*.

Scheutz, M., Eberhard, K., & Andronache, V. (2004). A real-time robotic model of human reference resolution using visual constraints. *Connection Science Journal*, *16*, 145–167.

Scheutz, M., Schermerhorn, P., Kramer, J., & Middendorff, C. (2006). The utility of affect expression in natural language interactions in joint human-robot tasks. *Proceedings of the 1st ACM International Conference on Human-Robot Interaction* (pp. 226–233).

Scholtz, J. (2002). Human-robot interactions: Creating synergistic cyber forces. *Multi-robot systems: from swarms to intelligent automata. Kluwer*.

Son, T., Baral, C., & McIlraith, S. (2001). Planning with different forms of domain dependent control knowledge – an answer set programming approach. *Proc. of LPNMR'01* (pp. 226–239).