2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Robot Plan Model Generation and Execution with Natural Language Interface*

Kyon-Mo Yang, Kap-Ho Seo, Sang Hoon Kang and Yoonseob Lim

Abstract— **Verbal interaction between a human and a robot may play a key role in conveying suitable directions for a robot to achieve the goal of a user's request. However, a robot may need to correct task plans or make new decisions with human help, which would make the interaction inconvenient and also increase the interaction time. In this paper, we propose a new verbal interaction-based method that can generate plan models and execute proper actions without human involvement in the middle of performing a task by a robot. To understand the verbal behaviors of humans when giving instructions to a robot, we first conducted a brief user study and found that a human user does not explicitly express the required task. To handle such unclear instructions by a human, we propose two different algorithms that can generate a component of new plan models based on intents and entities parsed from natural language and can resolve the unclear entities existed in human instructions. An experimental scenario with a robot, Cozmo, was tried in the lab environment to test whether or not the proposed method could generate an appropriate plan model. As a result, we found that the robot could successfully accomplish the task following human instructions and also found that the number of interactions and components in the plan model could be reduced as opposed to the general reactive plan model. In the future, we are going to improve the automated process of generating plan models and apply various scenarios under different service environments and robots.**

I. Introduction

Human-Robot Interaction (HRI) has received much attention in recent years due to its general-purpose, which offers user-friendly communication methods in human life with robots. Communication methods in HRI can be categorized into non-verbal communication and verbal communication methods. Among them, verbal communication has been shown to be the most effective method for collaboration with a robot because inexperienced users can easily instruct the robot without additional learning about the robot system [1]. Recently, several studies have suggested verbal based methods to collaborate with a robot in industrial field settings [8] - [12]. However, most of the robot asks users to repeat their requests in order to understand the exact intention of users given in natural language.

In this paper, we propose a new verbal interaction-based method that can generate new plan models that consist of actions and goals as well as execute actions to achieve the goal of the instructions given in natural language. In generating plan models, our system is equipped with several predefined rules to infer the meaning of a user's ambiguous instructions so that the system can select proper actions during planning. Therefore, a user does not give additional information while a robot is executing the user's instruction after generating a plan model. In addition, the robot selects an available action based on the current situation like a reactive planner so that a user does not need to consider sequences of actions while they give instructions to the robot. Based on a preliminary experiment, we found that both aspects of our system could result in a reduction in the number of interactions between humans and robots and the number of components created in plan model, making the interaction with the robot more convenient and efficient.

The paper is organized as follows. Section II presents the previous studies related to HRI with natural language understanding. Section III describes the detail of the proposed method to generate a reactive plan model and conduct it based on a planner. The simple experimental result of the proposed method is discussed in section IV, and the conclusion is presented in section V.

II. Backgrounds

A. Non-verbal Human-Robot Interaction

Several communication methods have been proposed to improve the performance of interaction in HRI. For example, a remote controller such as a smartphone or a haptic joystick is typically used to send a control signal to a robot directly [2] [3]. Although this method ensures that a user knows the status of the robot clearly, the user needs to master how to control a robot with a controller, which is not very user-friendly.

*This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No.10067169, 'Development of Disaster Response Robot System for Lifesaving and Supporting Fire Fighters at Complex Disaster Environment', by the Technology Innovation Program (10077553, Development of Social Robot Intelligence for Social Human-Robot Interaction of Service Robots) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea), and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2018R1D1A1B07049862).

Kyon-Mo Yang is with the Research Center for Human Robot Interaction, Korea Institute of Robotics and Technology Convergence, Pohang, 37553, South Korea and with the School of Mechanical, Aerospace and Nuclear Engineering, Ulsan National Institute of Science and Technology, Ulsan, 44919, South Korea (e-mail: kmyang@kiro.re.kr).

Kap-Ho Seo is with the Research Center for Human Robot Interaction, Korea Institute of Robotics and Technology Convergence, Pohang, 37553, South Korea (e-mail: neoworld@kiro.re.kr).

Sang Hoon Kang is with the School of Mechanical, Aerospace and Nuclear Engineering, UNIST, Ulsan, 44919, South Korea (co-corresponding author to provide phone: (+82)52-217-2729; e-mail: sanghkang@ unist.ac.kr).

Yoonseob Lim is with the Research Center for Diagnosis, Treatment and Care System of Dementia and the Center for Robotics Research, Korea Institute of Science and Technology, Seoul 136-791, South Korea (corresponding author to provide phone: (+82)2-958-6641; email: yslim@kist.re.kr).

TABLE 1. CHARACTERISTIC OF HRI METHODS

Method	Similarity of human interaction	Intuition of understandi ng	Necessity of learning
Remote controller	Low	High	High
Bio signal	Low	Low	Low
Direct physical interface	High	Low	Low
Nonverbal communication	High	Low	Low
Verbal communication	High	High	Low

Using a physiological signal like electroencephalography (EEG) or electromyography (EMG) to interact with a robot has also been proposed in HRI. Christopher Assad et al. proposed a method to control a robot arm and a manipulator based on EMG and inertial measurement unit (IMU) [4]. Shuyuan et al. decode the EEG signal triggered by a user's eye movements to control an exoskeleton robot under the restriction of dynamics and actuators [5].

Some methods for HRI is similar to how people communicate interactively with each other. A direct physical interface can be used to control a robot's movement or position through physical contact. For example, a robot can gently touch a person while guiding the human to the desired location [6]. Nonverbal communication such as gestures or facial expressions has also been used to instruct a robot. Quintero et al. used pointing gestures so that a robot can move an object to the desired position instructed by a human [7].

B. Verbal Human-Robot Interaction

Verbal communication has proliferated over recent years as one of the natural interaction methods with a robot. For example, a smart speaker, such as Google Home, Amazon Echo, or Apple HomePod, receives voice commands from a user and provides trivial services like "turning on a television". Jia et al. developed a manipulation system that can understand user's requests provided in natural language commands and execute appropriate actions to manipulate an object following commands like "pick up an object" [8]. Although those speakers and manipulation system can execute predefined actions with voice commands, modifying actions, or learning new actions by voice commands is not possible.

Several researchers have also focused on intuitive communication methods that users can program new capabilities of a robot through verbal communication. Tellex et al. developed a probabilistic graphical model that syntactically parses the natural language commands to operate a robotic forklift [9]. Forbes et al. presented an interactive interface system that can take new capabilities in various environments through the process of question and answering [10]. Kim and Yoon introduced an approach for aiding a service robot to make task-oriented interactions to get user decisions [11]. A robot learns from the history of the interaction between the robot and a human, and it tries to acquire essential human decisions to handle ambiguities or lack of information. Mohan et al. proposed an explanation-based task learning approach [12]. The human provides voice commands for sequences of each action to teach new tasks to a robot. In Table 1, we have summarized the characteristic of all the HRI methods reviewed in this section.

III. APPROACH

A. Verbal behavior of human instructions

We briefly conducted a user study to understand the verbal behavior of humans when a human gives instructions on a task to a young child. In the task, there are three different cubes, and the goal of the task is to match the top face of all the cubes. Four different people are recruited in the lab and are asked to give verbal instructions to a young child in Korean. The age of subjects is from 20 years old to 30 years old, and they do not have any background in the reactive robot plan model.

Table 2 shows the list of actual verbal instructions given by all the subjects. We found two different characteristics of verbal instructions from the experiment. First, subjects do not give details about the changes in the environment during the task. For instance, as in the instruction "*you move to the nearest cube*", the subject does not explain that the position of the child could be changed from "*on a start position*" to "*in front of the nearest cube*". This means that the robot needs to infer the possible actions from simple commands like "move".

- Behavior 1: A user does not consider and express an environmental change that can happen after executing a certain instruction.

TABLE 2. EXAMPLE OF VERBAL INSTRUCTIONS

Subject	Instruction
1	1. Move to the nearest cube 2. Roll the cube until you see the upper side of cube on top 3. Check the state, if it is right, move to the nearest cube. 4. Repeat instruction 2 and 3 until three cubes show an upper side.
2	1. Look for nearby three cubes 2. Roll the cube on the far left to make it to show an upper side on top 3. Roll other cubes to make them show an upper side on top
3	1. You have to move in a regular pattern 2. If you find a cube, approach the cube 3. Roll the cube until the cube shows an upper side on top 4. Check the state of the cube based on the marker on the cube 5. If upper side of the cube is appeared, do not roll the cube 6. If the cube does show an upper side, do not roll the cube 7. Repeat the instruction 4,5 and 6 until the cube shows an upper side on top 8. Follow instruction 2 to 7 repeatedly
4	1. Look for the nearest cube 2. Move to the cube 3. Roll the cube until that the upper side of cube is appeared on top 4. Stop to roll the cube when the upper side of cube is appeared on top 5. Follow instruction 1 to 4 two more times 6. Check the status of three cubes whether all top faces of cubes shows an upper side

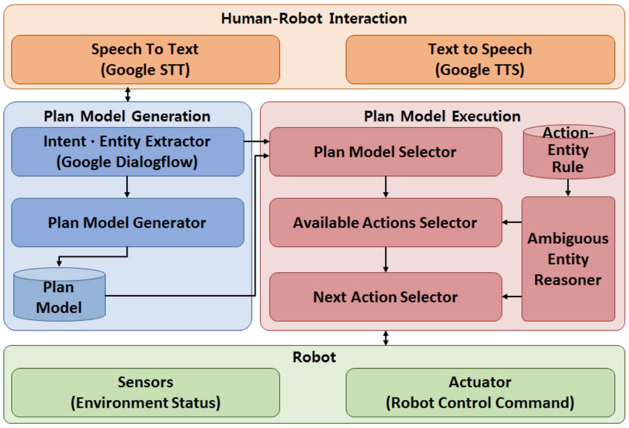

Figure 1. Software architecture

Such behavior indicates that a reactive plan model could be more effective than a deliberative plan model like STRIPS [13]. The deliberative planning system needs environmental change after executing actions because the purpose of the system is to make all sequences of actions based on the relationship between actions and environmental changes. However, the reactive plan model selects an available action based on the current status of an environment.

Secondly, according to the experiment, a human does not make explicit instructions on target objects of a verb. For example, in the instruction, "*you roll the cube,*" the target object "the cube" of verb "*roll*" is not clearly stated. In other words, "the cube" can be interpreted as anyone of three cubes that are in front of a child. Even if target objects are unclear, a human could semantically infer that the target object is the cube in front of a child after moving close to a cube. It means that the robot needs to understand the context of the verbal instructions.

- Behavior 2: The target object of verbs is often unclearly stated.

B. *The overall architecture of the proposed system*

Figure 1 shows the overall architecture of the system to generate a new plan and to execute actions based on a known plan model, which includes human-robot interaction, plan model generation, and execution, and robot hardware. A human can only make interaction with the robot through verbal commands (Google Dialogflow), and the robot is equipped with sensors that can receive visual and acoustical information (e.g., camera and microphone) as well as actuators that can be used to change the status of the physical environment where robot belongs. The plan model generation module creates a new plan model based on intents and entities extracted from verbal command. The plan model database stores all the generated plan models and transmits the model to the plan model execution module. The plan model execution module selects proper actions that meet the current interaction context and sends selected action commands to robot hardware. All software components operate on Robot Operating System (ROS) and are programmed in C++ [14].

C. *Plan Model Generation*

The plan model generation module consists of two types of components: Goal component and Action component. Goal components state final conditions about the environmental status so that the robot can evaluate whether or not the robot achieves the user's request at every action robot makes. Action components express possible conditions for each action that the robot could execute. Each component includes four factors referred from a reactive planning system: Type, Name, Precondition, and Action parameter.

- **Type *t***: denotes action component or goal component. If the type of component is "goal," the system always checks the final goal after executing action as a goal component. If the type of component is an action, the system selects the next action as an "action" component.
- **Name** *N*: a set of definitions to classify voice commands that a robot can execute. A robot is controlled based on the action name.
- **Precondition *P***: a set of environmental conditions to make a decision whether a robot could execute the action or not. A precondition consists of precondition key, p_{key}, and value, p_{value}.
- **Action parameter *M***: a set of additional information for a robot to execute a voice command. It includes object names or position names in a service environment.

For the generation of a new plan model, the proposed system extracts intents and entities from a user's verbal instruction (Table 3 shows example intents and entities used in this study). There are four different intents defined in the proposed system as below.

- **Goal command C_{goal}**: List of task requirements provided by users. It has clues about type as factors in goal component.
- **Action command C_{action}**: List of possible actions that a robot needs to execute. It has clues about the type, name, and action parameters as factors in the action component.

Algorithm 1. Generating component of new plan model

procedure infer (intent $i \in I$)
  if (intent i is goal command c_{goal}) then type $t \leftarrow$ "goal"
  else (i is action command c_{action}) then type $t \leftarrow$ "action"
procedure extract (entities $E = \{e_1, e_2, \ldots, e_n\}$| c_{action} or c_{goal})
  if (type of entity $e_{verb} \in E$ = verb) then name $n \leftarrow$ e
  if (type of entity $e_{object} \in E$ = object) then action parameter $m \leftarrow$ e
procedure extract (entities E| condition command $c_{condition}$)
  if (type of entity $e_{subject} \in E$ = subject)
    then key of precondition $p_{key} \leftarrow e$
  if (type of entity $e_{pos} \in E$ = position) then value of $p_{value} \leftarrow e$
  if (type of entity $e_{object} \in E$ = object)
    then the of precondition $p_{key} \leftarrow e$
  if (type of entity $e_{status} \in E$ = status) then value of $p_{value} \leftarrow e$
procedure generate (goal or action component $c \in C$)

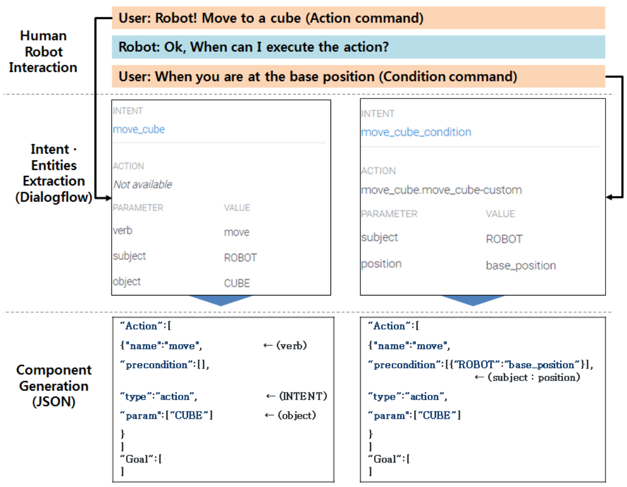

Figure 2. Example of generating component for new plan model

- **Condition command $C_{condition}$:** List of conditions for actions that a robot executes. It has clues about preconditions of action as factors in the action component.
- **Starting command C_{start}:** Intention to start generated plan model.

We used three different types of entities based on the structure of a sentence: subject, verb, and object. In addition, we add two more types of entities as below.

- **Position:** a set of words to express the current spatial definition of an object in a service environment.
- **Status:** a set of words to express the current state of the object in a service environment.

Algorithm 1 illustrates the flow of generating the new goal or action component. When the system receives the goal command (c_{goal}) or action command (c_{action}), it defines whether a component's type is a goal or action based on the type of intent (Step 1 ~ 3). And then system is able to define the name (n) and action parameters (m) using the entity extraction, $E = \{e_1, e_2, \ldots, e_n\}$. The verb in entities ($e_{verb}$) is defined as name ($e_{name}$), and object ($e_{object}$) in entities is defined as action parameter (step 4 ~ 6). Finally, system defines a set of preconditions (p) from subject ($e_{subject}$), object (e_{object}), position ($e_{position}$), or status (e_{status}) in entities (step 7 ~ 11). If the entities are not limited by a specific expression like "*nearest cube*," all characters of the entities are capitalized like "*CUBE*" in order to indicate that the entities need to be clarified by the action-entity reasoner. All the plans generated by Algorithm 1, is described as a JSON format (step 12).

Algorithm 2. Executing actions with known plan model
1: **procedure set** (current goal $G_{current}$ | plan model $p \in P$)
2: **if** (starting command C_{start} = the file name of p)
then current goal $G_{current} \leftarrow P_{goals}$, current actions $A_{current} \leftarrow P_{actions}$
3: **procedure select** (available actions $A_{available}$ | $A_{current}$, environment E)
4: **if** (p_{key} or p_{value} = capital expression)
then p_{key} or $p_{value} \leftarrow$ reasoning result r (action a)
5: **if** (p *of* $a_{current}$= current environment $E_{current}$)
then available action $A_{available} \leftarrow a$
6: **repeat 4 - 6 per the number of action components in plan model**
7: **procedure select** (next action a_{next} | $A_{available}$)
8: $a_{next} \leftarrow a \in A_{available}$ under Policy Φ_{random}(random selection)
9: **if** (action parameter $m \in M$ = capital expression)
then $m \leftarrow$ reasoning result r (action a)
10: **procedure execute** (next action a_{next})

Figure 2 shows an example of generating one action component of the new plan model. In this example, a user's verbal command, "***Robot! Move to a cube***" and "***When you are at the base position***" is used to create action component described in JSON format. Here, the base position is defined as the predefined location where the robot waits after the execution of an action.

D. Plan model execution

Algorithm 2 shows the overall flow of executing actions using generated plans from algorithm 1. The plan model selector receives starting command (C_{start}) from a user and then loads an appropriate plan model (P) based on the command from the plan model database (step 1 ~ 2). The system sets the current goals ($G_{current}$) from goal components (P_{goal}) in the plan model (P), and it sets current actions

TABLE 3. DEFINITION OF INTENTS AND ENTITIES

Intent (Type)	**Description**	
Goal (Goal command)	The purpose of plan model.	
move_base_position (Action command)	User wants a robot to move to a base position	
move_cube (Action command)	User wants a robot to move to a cube	
roll_cube (Action command)	User wants a robot to roll a cube	
condition (Action command)	User wants to set conditions that a robot execute an action .It is a condition command	
start (Start command)	User wants a robot to start plan	
Entities (Type)	**Representative word**	**Involved words**
Robot (Subject)	ROBOT	cozmo, robot, robot 1
Action (Verb)	finish	achieve, finish
	move	move, moves
	roll	roll, rolls
Cube (Object)	CUBE	CUBE, cube, cubes
	cube1	cube1, cube 1, the cube 1
	cube2	cube2, cube 2, the cube 2
	cube3	cube3, cube 3, the cube 3
Robot position (Position)	IFO[CUBE]	IFO[CUBE], in front of a cube
	base_position	base position, base_position
Cube status (Status)	Side	Side, side
	Top	Top, top

TABLE 4. DEFINITION OF ACTION-ENTITY RULE

Number	Type	Action	Entity Rule
1	Precondition	move	ROBOT = robot 1
2			CUBE = cube that in front of a robot
3		roll	ROBOT = robot 1
4			CUBE = cube that in front of a robot
5	Action parameter	move	ROBOT = robot 1
6			CUBE = one of cubes that are in a service environment. It is selected randomly
7		roll	ROBOT = robot 1
8			CUBE = cube that in front of a robot

($A_{current}$) from action components (P_{action}). At step 3, the available action selector acquires the status of a current environment (E) from sensors and compares the status with preconditions of each action. If the precondition key (p_{key}) or value (p_{value}) is expressed at the capital letter, it is replaced with the reasoning result (r) through the ambiguous word reasoner (Step 4). In the reasoner, there is a predefined description of the relationship between actions and ambiguous entities (Table 4). For instance, if the precondition is "*CUBE*" and action is "*move*," precondition "*CUBE*" is replaced to "*cube 1*". After then, if precondition key (p_{key}) or value (p_{value}) of a current action ($a_{current}$) is in the same status, the action ($a_{current}$) is a candidate for next actions as available action ($a_{available} \in A_{available}$) (Step 5). In Step 6, the plan model selector repeats the comparison process until it finishes checking about all possible actions ($A_{current}$) defined in the plan model. Finally, the system selects one action as a next action (a_{next}) by a policy of selection (Step 7 ~ 10).

IV. EXPERIMENTAL RESULTS

To evaluate the performance of generating a new plan model and executing the generated model, we have selected

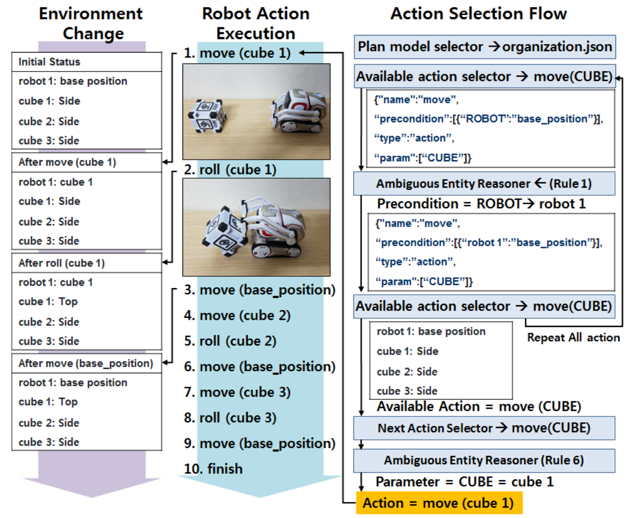

Figure 4. Action execution based on generated plan model

one simple scenario. Environment configuration of the scenario is that there is one robot, "Cozmo" and three different cubes, "*cube 1*", "*cube 2*", and "*cube 3*". The goal of the scenario is that a user wants Cozmo to match the top face of all the cubes. The robot can move around the environment and roll the cubes using their forks. From the camera in the Cozmo and the markers on the cubes, the system receives the positions of the robot and the status of each cube. Status of cubes is classified as "Top" and "Side." The robot can be either "at the base position" and "in front of a cube." Table 3 shows the definition of intents and entities to extract from natural language instruction. Table 4 describes the rules between actions and entities for executing a new plan model.

Figure 3 shows the generated component, while the user verbally interacts with the robot. In this example scenario, the user interacted with the robot eight times, and the proposed system generated three action components and one goal component. Figure 4 shows the flow where a robot selects actions based on the status of the service environment. On the

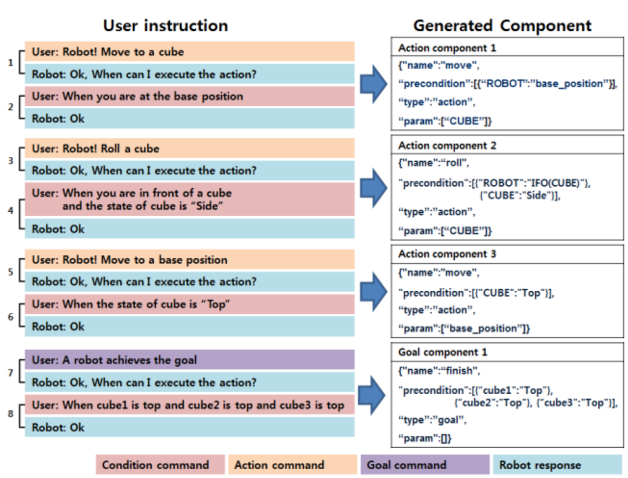

Figure 3. New plan generation using user instruction

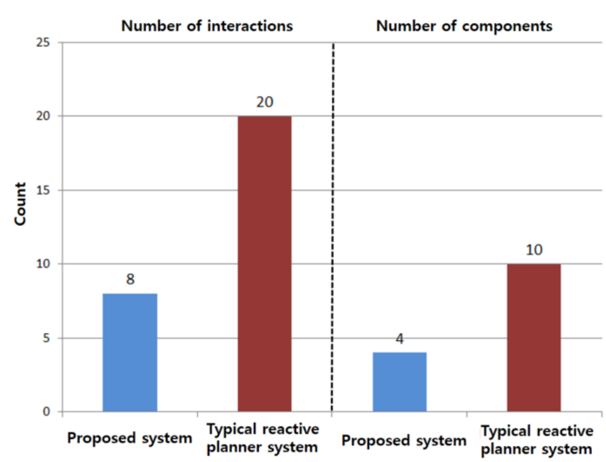

Figure 5. Comparison result of proposed system and a representative reactive planner system

right side, the example execution process of the plan model is displayed. First, in the plan model selector, system loads the organization of generated plan model. Second, available action selector selects action component "move(CUBE)" in the plan model. Third, ambiguous entity reasoner replaces the unspecified word "*ROBOT*" with "*robot 1*" based on predefined rule 1 in table 4. Fourth, the available action selector assesses whether the precondition of the action component "move(CUBE)" is the same as the current status of the environment. If it is not the same, the selector assesses another action component in the plan model. Action selector repeats the same process from 2 to 4 about all action components in the plan model and decides one action "move (CUBE)" in the candidate of the set of available action. Finally, it infers the word "*CUBE*" in the action parameter set to "*cube 1*" based on rules 6 in table 4.

If the current environmental status is identical to the precondition of goal component, the robot executes action "finish." Otherwise, it conducts the action selection process until the environmental status becomes the same as the goal.

Figure 5 shows the comparison result of the required number of interactions and the number of components of the proposed and those of the representative reactive model. For the number of interactions, the representative reactive planner based system requires the user to state-specific expressions three times more than the proposed system under a simple example scenario. It is because the previous system needs all individual action components for all the cubes

V. Conclusion

In this study, we proposed a method to generate a new reactive plan based on natural language instructions. In order to achieve the goal of the reactive plan without the user's intervention, the proposed system uses the action-entity reasoner in order to infer ambiguous expressions in the user's verbal command. Thanks to the action-entity reasoner, the proposed system can handle the instructions with ambiguous expressions, and a user does not need to give additional information to a robot in the middle of executing the plan, unlike previous works. An experimental simple scenario test showed that the proposed method requires less interaction and fewer components in the plan model than the representative reactive planning system.

The promising results warrants improvement of the automated process of generating action-entity rules that are used in the action-entity reasoned, and more experimental tests with various scenarios under different service environments and robots.

Acknowledgment

This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No.10067169, 'Development of Disaster Response Robot System for Lifesaving and Supporting Fire Fighters at Complex Disaster Environment', by the Technology Innovation Program (10077553, Development of Social Robot Intelligence for Social Human-Robot Interaction of Service Robots) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea), and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2018R1D1A1B07049862).

References

[1] J. E. Laird, K. Gluck, J. Anderson, K. Forbus, O. Jenkins, C. Lebiere, D. D. Salvucci, M. Scheutz, A. Thomaz, G. Trafton, R. E. Wray, S. Mohan, and J. R. Kirk. "Interactive task learning," *IEEE Intelligent System*, vol. 32, pp. 6–21, August 2017.

[2] A. M. Walker and D. P. Miller, "Tele-operated robot control using attitude aware smartphones," in *2012 The 7th ACM/IEEE International Conference on Humanrobot Interaction (HRI' 12)*, pp. 269–270.

[3] D. J. Brooks and H. A. Yanco, "Design of a haptic joystick for shared robot control," in *2012 The 7th ACM/IEEE International Conference on Humanrobot Interaction (HRI' 12)*, pp.113–114.

[4] C. Assad, M. Wolf, T. Theodoridis, Glette K., and A. Stoica, "BioSleeve: a natural EMG-based interface for HRI," in *2013 The 8th ACM/IEEE International Conference on Human-robot Interaction (HRI' 13)*, pp. 69–70.

[5] S. Qiu, Z. Li, W. He, L. Zhang, C. Yang, and C.-Y. Su, "Brain–machine interface and visual compressive sensing-based teleoperation control of an exoskeleton robot," *IEEE Trans. Fuzzy Systems*, vol. 25, no. 1, pp. 58–69, February, 2017.

[6] T. L. Chen and C. C. Kemp, "Lead me by the hand: Evaluation of a direct physical interface for nursing assistant robots," in *2010, 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI' 10)*, pp. 367–374.

[7] C. P. Quintero, R. Tatsambon, M. Gridseth and M. Jagersand, "Visual pointing gestures for bi-directional human robot interaction in a pick-and-place task", in *2015 The 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp.349–354.

[8] Y. Jia, N. Xi, J. Y. Chai, Y. Cheng, R. Fang, and L. She, "Perceptive feedback for natural language control of robotic operations" in *2014, In Proc.of the IEEE International Conference on Robotics and Automation (ICRA),* pp. 6673–6678

[9] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," In *2011, Twenty-Fifth AAAI Conference on Artificial* Intelligence, pp. 1507–1514.

[10] M. Forbes, R. P. Rao, L. Zettlemoyer and M. Cakmak, "Robot programming by demonstration with situated spatial language understanding," In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2014–2020.

[11] Y. Kim and W. C. Yoon, "Generating task-oriented interactions of service robots" *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 44, no. 8, pp. 981–994, August, 2014.

[12] S. Mohan and J. Laird, "Learning goal-oriented hierarchical tasks from situated interactive instruction, in *2014 Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 387–394

[13] B. C. Williams and P. P. Nayak, "A reactive planner for a model-based executive," in *1997 International Joint Conferences on Artificial Intelligence(IJCAI)*.

[14] Willow Garage, and tanford Artificial Intelligence Laboratory, "ROS," Internet: http://www.ros.org/.