IEEK Transactions on Smart Processing and Computing

# Natural-Language-Based Robot Action Control Using a Hierarchical Behavior Model

**Hyunsik Ahn and Hyun-Bum Ko**

Department of Robot System Engineering, Tongmyong University / Busan, South Korea hsahn@tu.ac.kr

* Corresponding Author: Hyunsik Ahn

*Received November 18, 2012; Revised December 5, 2012; Accepted December 7, 2012; Published December 31, 2012*

***Abstract*:** In order for humans and robots to interact in daily life, robots need to understand human speech and link it to their actions. This paper proposes a hierarchical behavior model for robot action control using natural language commands. The model, which consists of episodes, primitive actions and atomic functions, uses a sentential cognitive system that includes multiple modules for perception, action, reasoning and memory. Human speech commands are translated to sentences with a natural language processor that are syntactically parsed. A semantic parsing procedure was applied to human speech by analyzing the verbs and phrases of the sentences and linking them to the cognitive information. The cognitive system performed according to the hierarchical behavior model, which consists of episodes, primitive actions and atomic functions, which are implemented in the system. In the experiments, a possible episode, "*Water the pot*," was tested and its feasibility was evaluated.

***Keywords*:** Natural language based robot control, Cognitive system, HRI, Conversational interaction, Robot intelligence

## 1. Introduction

As robotic technology advances, robots are expected to interact with humans in daily life to fulfill human demands, such as completing housework, helping people with disabilities, etc.. The primary method used to control robots should be natural language because it is natural for human cognition and is easy to learn and understand for those unfamiliar with computer languages. On the other hand, linking natural language to robotic actions and perceptions based on sensorimotor information is challenging in the field of robot intelligence [1, 2].

For a robot to follow natural language instruction, it needs to both understand speech and link it to its computational functions and parameters. This procedure requires gradual steps, including natural language processing (NLP), syntactic parsing, semantic analyzing of arguments, and linking of such arguments to cognitive information, which is connected to the computational functions of the robot controller. In addition, it is important to account for the range of human commands. People can order simple actions or complex tasks consisting of multiple actions. Occasionally, the tasks require that the action be repeated using the same or different parameters. Therefore, it is important to consider the reuse of commands and actions.

This paper proposes a methodology for robot action control using natural language based on a hierarchical behavior model. The proposed behavior model was implemented in a sentential cognitive system installed in a robot to link human commands to the actions of the robot. The model hierarchically connects the arguments of the command sentence to the atomic functions using the information stored in an object descriptor and a motion descriptor in the cognitive system. The sentence, which is parsed syntactically and analyzed semantically for verbs and phrases, is called an episode that calls the lower primitive actions and the lowest atomic functions necessary for performing the task hierarchically.

The main contributions of this paper are that 1) it proposes a hierarchical model for performing tasks by linking episodes, primitive actions and atomic functions as well as for producing various episodes by reusing and compounding primitive actions; 2) it uses syntactic parsing and semantic analysis to link a sentential command to a

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012-0003116).

robotic action; and 3) it proposes a methodology for connecting the verbs and phrases of a sentence to the cognitive information of the cognitive system, which is used to elicit robot actions.

The rest of this paper is organized as follow. Section 2 describes the work related to the area of connecting commands to robot actions. Section 3 presents a sentential cognitive system including multiple modules and overview of proposed robot action control. Section 4 details the hierarchical behavior model in relation to the cognitive system. Section 5 reports the experimental results, showing the implementation of the model and the performance of an episode. Section 6 concludes the paper.

## 2. Related Work

Natural-language-based robot control is related to multiple areas, ranging from robotic intelligence to cognitive linguistics, including grounding, natural language direction and semantic parsing. This study is related to grounding, which connects human language and cognitive information [3, 4]. To represent the meanings of the words grounded in reality, grounding language was used to describe the real world [5] and motor actions [6, 7], and to connect a graphics of a game to language [8]. Anchoring, which is a symbolic grounding that establishes and maintains a relationship between the symbol and sensory data, is also related to the present study in the same way as linguistic grounding [9, 10]. Within this context, this study focused specifically on the semantic parsing of action language that connects the phrases of the imperative sentences to the cognitive information of a robot.

One of the most prominent areas of linking language to robot actions is the field of natural language direction, which is used to navigate and move the robots according to human direction. [11, 12] presented grounding verbs in natural language commands to create a path to be used for robot navigation, and utilize a cost function that scores the matching of the language to path plans at each time step. These studies use a spatial description clause (SDC) to parse the instructions into a set of separable instruction clauses. In [13], statistical machine translation (SMT) was adopted to link natural language instruction to a map constructed by a robot. These approaches use translation and parsing tools, such as SDC and SMT to convert natural language commands to descriptions to move robots, but it is difficult to apply them to complex tasks, such as housework, which require the robot's hands to perform a range of primitive actions. To overcome this problem, this approach uses a hierarchical model, in which a human can order complex tasks that consist of multiple primitive actions. Primitive actions are natural language commands, which mean the elementary actions of humans that can be reused with the same verbs and different phrases.

This study is similar to that reported in [14] in that it adopts a hierarchical approach using primitive tasks to construct more complex tasks. On the other hand, although [14] did not consider the reusability of primitive commands, the current approach uses a hierarchical model to address the reusability of primitive actions as well as to construct complex tasks with primitive actions. [15] is also similar to the present work in that it uses natural language direction to manipulate the objects on a table and because it uses object schemas describing the characteristics of objects, such as the size, color, and weight, for use in the input information required to manipulate them. The present work, however, uses a more structural approach by adopting a cognitive system that includes a motion descriptor using a hierarchical behavior model as well as an object descriptor. In addition, this study also integrates sensory-motor, reasoning, and memory modules. A more systematic approach that employs semantic parsing to connect commands to cognitive information, including object schemas is also used.

## 3. Sentential Cognitive System and Robot Action Control

### 3.1 Sentential Cognitive System

Natural-language-based robot control requires that the robots receive perceptional and behavioral cognitive information. In the proposed approach, the robot performs actions based on a sentential cognitive system (SCS). The SCS has a multimodal scheme, as shown in Fig. 1.

In the lower part of the cognitive system, the perception and behavior modules input environmental information and output behavioral executions. The vision module is used to recognize visual events by capturing the scenes and recognizing objects. The sensory module covers all the robot's senses, except for visual and linguistic perception. The listening module transforms the speech of humans to sentence form and comprises speech recognition and language understanding. The speech module is for speech synthesis and the speaking of sentences. The utterance module takes charge of the robot's actions that are performed hierarchically with the combination of episodes, primitive actions and atomic functions.

The cognitive manager covers the interpretation and reasoning of events. The event interpreter is used to make a sentence for an event based on the perception and behavior. In the proposed approach, an event is defined as the basic unit of cognition and a sentence is adopted as a descriptor of the event. In particular, an event is the recognized change differentiating from the information stored previously in memory. The event interpreter translates the cognitive information obtained from the modules, generates sentences and revises the information of the memory. Spatial imagery is an imitation of the human mental model of spatial reasoning. If the SCS needs to determine the visual situation of a certain time, the spatial imagery constructs a scene virtually by placing the models of the objects in its image plane and derives the spatial context from the scene. In the reasoning module, the robot is endowed with innate reasoning rules.

A memory domain exists in the upper part of the system that consists of a sentential memory, which stores a series of sentences describing the events with auxiliary

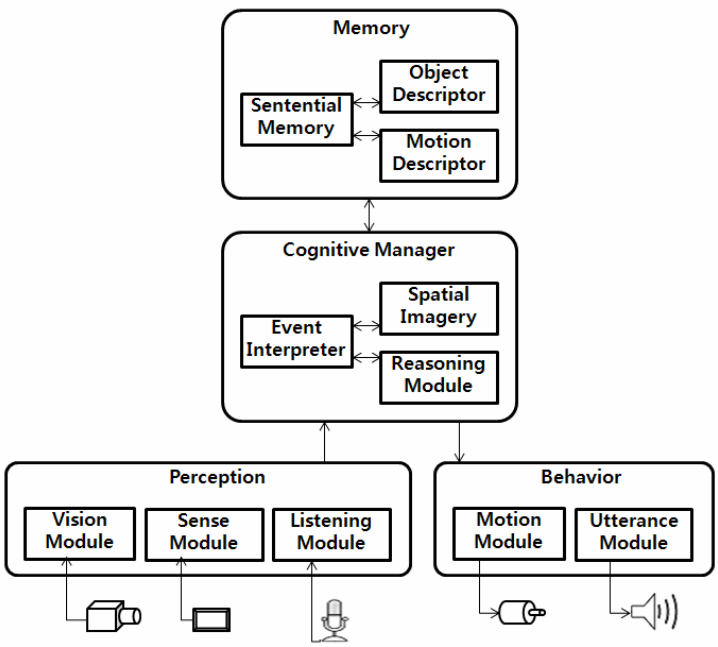



**Fig. 1. Sentential cognitive system (SCS).**

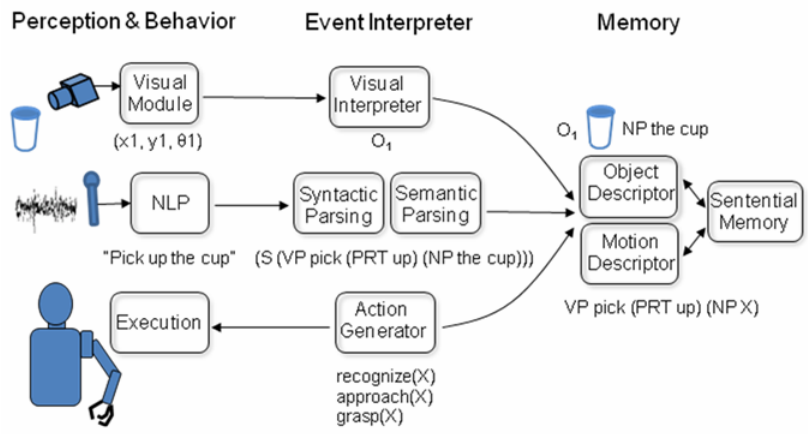



**Fig. 2. Block diagram of the natural language command "*Pick up the cup*" for robot motion and its execution.**

memories, i.e. an object descriptor and motion descriptor. The object descriptor stores information about objects, such as their shape, feature and current position. The object descriptor also provides object-related schematic information, such as a holding position and the history of the robot's action related to the object. The motion descriptor stores hierarchically the imperative verbs linked to episodes and primitive actions and is then connected to atomic functions, which are programmed as functions. This paper focuses on this module, and details the hierarchical behavior model in the next section.

If an event occurs internally or externally, the SCS generates a sentence describing the cognitive information of the event. An event should occur in a single module of the SCS and generate a simple English sentence clause. To interpret the sentence, semantic parsing is was adopted, where the sentences are parsed syntactically and analyzed using the sentential types, phrasal arguments are segmented according to the semantic structure, and the arguments are connected to the cognitive information.

## 3.2 Overview of Robot Action Control on the SCS

Fig. 2 gives an example of the execution of a natural language command, "*Pick up the cup*." Using the vision module of the cognitive system, the scene of a cup is captured and recognized by its features, such as its shape, position and color, which are stored in the object descriptor. When the listening module inputs a human voice, the module transfers it to a sentence using a NLP. The cognitive manager's event interpreter parses the sentence syntactically and semantically. The syntactic parsing of the sentence produces dividable units, including verbs and phrases, which are used in semantic parsing, which connects them to the cognitive information stored in the memory. The verb phrase (VP) is linked to the action of an episode stored in the motion descriptor (e.g., VP pick

(PRT[1] up)). The noun phrase (NP) is connected to an object stored in the object descriptor along with the cognitive features, such as shape, position and state (e.g. NP the cup). If the sentence is an imperative, the cognitive manager makes the event interpreter generate an action with the object as an episode by performing the atomic functions that are connected hierarchically with the verb. All the sentences interpreted from the events are stored in the sentential memory. On the other hand, in the case of complex commands, such as "*Clean the room*" and "*Water the pot*", the robot needs to know how to integrate a series of primitive actions to perform the command. In the next section, the methodology for performing complex tasks on the motion descriptor of the SCS will be detailed.

## 4. Hierarchical Robot Control by Natural Language Commands

### 4.1 Syntactic and Semantic Parsing

When human speech is input, the listening module translates it to a sentence. Subsequently, the event interpreter syntactically and semantically parses the sentence to determine the sentence type and to extract the verbs and phrases to be connected to cognitive information. The current approach depends more on the parsing processes than other natural language direction to move robots. For the syntactical parser, the Penn Treebank rule was adopted [16]. In the Penn Treebank rule, a sentence is segmented with phrases: verbs, nouns, adjectives, adverbs, prepositions, WH-adverbs, WH-nouns, and WH-prepositional phrases. For example, sentences are parsed and segmented into phrases, such as in the following examples:

(1) (S (NP Tom) (VP picks (PRT up) (NP the cup)))
(2) (S (VP pick (PRT up) (NP the cup)))
(3) (S When did (NP Tom) (VP pick (PRT up) (NP the cup)))

After syntactical parsing, the SCS performs semantic parsing that analyzes the parsed sentence and links the phrases in the sentence to the cognitive information, such as the computational motion functions and visual features of the recognized objects. In the case of sentence analysis, the sentence type can be determined from the order of the words: (1) is declarative for (S (NP VP)), (2) is an imperative for (S (VP)), and (3) is a WH question for (S WH Aux (NP VP)). If the sentence is imperative, it can be assumed that the human wants the robot to perform an action.

To link the phrases to the cognitive information, the cognitive manager links the extracted verb (VP pick) to an episode that is connected to a computational function in the motion descriptor. The noun phrase, (NP the cup), is connected to the object in the object descriptor. The object descriptor, which provides memory space for object schema, stores the cognitive information of objects, which includes object's features and the contextual information, such as the hold points for picking up an object.

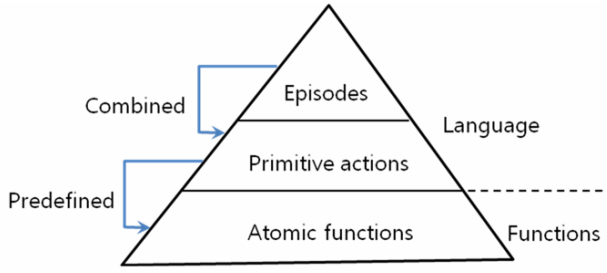



**Fig. 3. Hierarchical behavior model.**

### 4.2 Hierarchical Behavior Model

This approach adopts a hierarchical behavior model to provide the effective usability of predefined primitive actions. Fig. 3 shows the hierarchical behavior model, which consists of three levels: atomic functions, primitive actions and episodes. Episodes are the commands asking the robot to perform a task. The episodes are a series of primitive actions that become the basic unit for teaching the robot actions by combining the primitive actions. This primitive action calls the atomic functions, which are predefined with the atomic functions in the motion descriptor of the SCS, and performs them physically in the motion module.

The episodes are the general tasks that a human wants to make the robot perform and are linked to the verbs of the sentences that the user speaks, as shown in the left of Table 1. For example, if a user orders the robot to water the pot, the robot performs an action according to the hierarchical structure with a series of primitive actions: move to the cup, pick up the cup, move to the pot, approach the pot, pour the cup, and retract hands. A primitive action, such as "*Pick up the cup*," calls the atomic functions: *recognize(X)*, *extend(X)*, *grasp(X)*, *and retract()*. The motion descriptor of the SCS stores the components of each level of the hierarchical model, preserves their linkage, and performs the human linguistic commands hierarchically.

A primitive action is a basic unit of action in the view of human speech. A user can order the robot to perform an action with a primitive action without any programming to control the robot. The sequence of primitive actions constructs an episode that is the robot's task. For example, in the case of the command, "*Pick up the cup*," a user would not say, "Move your hand above the cup, approach the cup, grasp the cup and retract your hand". A user would simply say, "*Pick up the cup*." Generally, an episode consists of multiple primitive actions that combine to form a complex task. Sometimes, an episode is just a single primitive action, such as "*Pick up the cup*." Primitive actions are the smallest unit of human speech used to order the robot to perform a task.

A primitive action is the lowest language level in the conversational communication. On the other hand, the atomic functions are the unit of a program in the controller of the robot. If an action is difficult to describe with a

[1] PRT is the abbreviation of a particle in the Link Grammar Parser

**Table 1. Definition of the episodes, primitive actions and atomic functions.**

Episodes

| # | Episodes |
|---|---|
| 1 | Water the pot |
| 2 | Give me water |
| 3 | Clean the room |
| 4 | Close the door |
| 5 | Take care of the puppy |
| 6 | Turn off the gas valve |

Primitive Actions

| # | Primitive actions |
|---|---|
| 1 | Pick up X |
| 2 | Move to X |
| 3 | Pour X |
| 4 | Retract |
| 5 | Approach to X |
| 6 | Rotate X |

Atomic Functions

| # | Functions | Parameter |
|---|---|---|
| 1 | extend | X |
| 2 | recognize | X |
| 3 | retract | |
| 4 | grasp | X |
| 5 | approach | X |
| 6 | rotate | X |
| 7 | moveto | X |
| 8 | push | X |

verbal sentence or if it is ineffective to express it with a verb because of the complexity of the action, it can be implemented with atomic functions. The atomic functions are sensorimotor functions based on the unit of a simple motion.

Table 2 lists the execution of the command "*Water the pot*." The syntactically parsed phrase of (NP the pot) is searched in the object descriptor with its features. The phrase calls primitive actions by replacing X with (NP, the pot) and (NP the cup). These primitives call the atomic functions: *recognize(O1)*, *extend(O1)*, *grasp(O1)*, and *retract()*. The value of the parameters of the primitive actions originates from the features of the object.

**Table 2. The example of performing an episode, "*Water the pot*".**

| Episodes | Primitive Actions | Atomic Functions |
|---|---|---|
| VP Water (NP the pot) | VP Move (PP to (NP the cup)) | recognize(O1)<br>moveto(O1) |
| | VP Pick (PRT up) (NP the cup) | recognize(O1)<br>extend(O1)<br>grasp(O1)<br>retract() |
| | VP Move (PP to(NP the pot)) | recognize(O2)<br>moveto(O2) |
| | VP Approach (PP to (NP the pot)) | recognize(O2)<br>extend(O2) |
| | VP Pour (NP the cup) | rotate(wrist,a1,s1,t1)<br>rotate(wrist a2,s2,t2) |
| | VP Retract (NP hands) | retract() |

## 4.3 Reusability of Behavior

Primitive actions are simple actions that can appear repeatedly in common tasks in daily life. Ordinary people want to construct multiple episodes using natural language without any knowledge of the actual programming. This means that a user should be able to teach a robot using natural language and construct an episode by ordering a series of primitive actions. In Table 1, "*Pick up the cup*" can be used for both "*Water the pot*" and "*Give me water*."

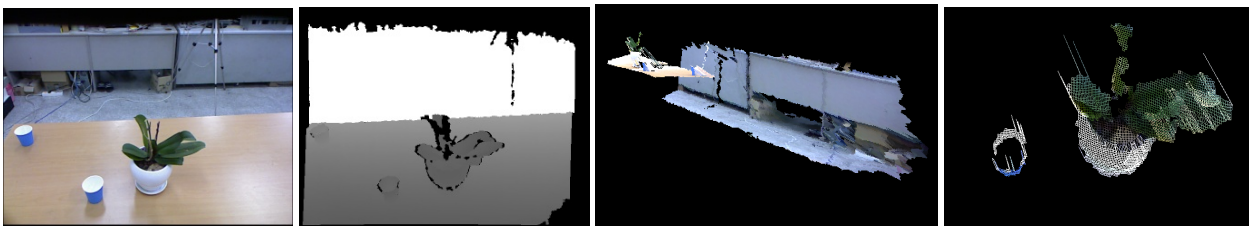
In addition, the same verbs need to be used in a sentence for an action, even if multiple objects are used. In this case, the robot executes different actions according to the shape of the objects. This means that the robot executes the actions slightly differently, depending on the shape and position of those objects. For example, the primitive action "*Pick up X*" could be applied to many objects, such as bottles, cups and dishes. With the primitive action, the robot can perform multiple actions, such as "*Pick up the cup*," "*Pick up the bottle*," and "*Pick up the can*." The statements describing the objects of the sentences are then compared with the objects stored in the object descriptor. The SCS finds the shape and position of the objects and calculates the proper point to grasp them to execute a command.

## 4.4 Teaching Episodes

New episodes are composed of a series of primitive actions predefined with atomic functions. When a user wishes to teach a new episode with natural language, the SCS of the robot should notice the start and end of the series. To give notice of the start and end of the episode, the users can inform them with predefined sentences, as shown in Table 3. The users can tell the robot what is the

**Table 3. Example of natural-language-based teaching of an episode, "*Water the pot*".**

| | Sentences | State |
|---|---|---|
| User | The episode of watering the pot begins | |
| Robot | The episode of watering the pot begins | Start an episode |
| User | Move to the cup | |
| Robot | (action) I moved to the cup | Success |
| User | Pick up the cup | |
| Robot | (action) I picked up the cup | Fail |
| User | You failed | |
| Robot | I failed | Recognize fail |
| User | Pick up the cup | |
| Robot | (action) I picked up the cup | Success |
| User | Move to the pot | |
| Robot | (action) I moved to the pot | Success |
| User | Approach to the pot | |
| Robot | (action) I approached to the pot | Success |
| User | Pour the cup to the pot | |
| Robot | (action) I poured the cup to the pot | Success |
| User | Retract hands | |
| Robot | (action) I retracted hands | Success |
| User | The episode of watering the pot ends | |
| Robot | The episode of watering the pot ends | End of the episode |



Color image | Depth image | 3D representation | 3D detection of cup

**Fig. 4. 3D object recognition using Kinect.**

(a) Pick up the cup-extend(x) | (b) Pick up the cup-grasp(x) | (c) Retract

(d) Approach the pot | (e) Pour the cup | (f) Retract

**Fig. 5. An episode, "*Water the pot*," using multiple primitive actions.**

start and end of an episode by saying "*The episode begins*" and "*The episode ends*." In the procedure of learning, the robot might fail to perform the primitive actions. If the robot fails to perform the action, the user can ask the robot to repeat it until the performance is successful.

## 5. Experimental Results

The proposed robot control methodology was tested with the hierarchical behavior model using the scenario, in which the user orders the robot with a natural language command and the robot follows the instruction. The testbed of the robot is TUBO (Tongmyong University roBOt), which has 16 degrees of freedom. The robot uses a range imaging sensors, Microsoft Kinect [17], for object recognition. The sensor captures the images of the objects located on a table, and recognizes the objects using a 3D recognition algorithm, including segmentation and feature extraction of the objects using the depth and color images in a 3D environment, and detects the position of the objects. Fig. 4 presents the results of object recognition. By generalizing both the color and depth images, the vision module produces 3D images and calculates the real position of the cup and pot, and stored them in an object descriptor.

When a user orders the robot to perform an action using natural language commands, the listening module of the SCS inputs the speech and translates it to a sentence with an NLP program, Dragon NaturallySpeaking (Nuance) [18]. In the event interpreter, the sentence is parsed syntactically using the Link Grammar Parser using the Penn Treebank rule [19], and the sentence type is determined. If the sentence is an imperative, the verb of the sentence is linked to a verb of episodes in the motion descriptor, and the noun phrases of the sentence are linked to the objects stored in the object descriptor. The verbs of the episode are then called the primitive action, and the atomic functions hierarchically to perform the actions.

Fig. 5 shows the behavior of the robot acting according to the primitive actions. When a user commands the robot to "*Water the pot,*" the listening module translates the speech to a sentence and parses it syntactically in an event interpreter. The extracted verb, "*water*", is connected to the episode, "*Water the X*." The phrase "*the pot*" is then searched for in the object descriptor using its cognitive information, such as its position, shape and color. In the motion descriptor, the SCS finds the lower level primitive actions, and the actions called the atomic actions located at the lowest level. Fig. 5 (a–f) shows each of the robot's actions.

Fig. 6 shows the learning and execution of an episode stored in the sentential memory implemented with MS

| | SNO | ETIME | MODULE | TYPE | VERB | AVERB | A1 | A2 | SPACE | TIME | ADJ1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 50 | 2012-09-10 17:19:57.143 | L | D | begins | NULL | (NP (NP the episode) (PP of (S (VP watering (NP the pot))))) | NULL | NULL | NULL | NULL |
| 2 | 51 | 2012-09-10 17:19:59.253 | U | D | begins | NULL | (NP (NP the episode) (PP of (S (VP watering (NP the pot))))) | NULL | NULL | NULL | NULL |
| 3 | 52 | 2012-09-10 17:20:12.207 | L | I | Move | NULL | NULL | NULL | (PP to (NP the cup)) | NULL | NULL |
| 4 | 53 | 2012-09-10 17:20:14.363 | M | D | move | NULL | (NP I) | NULL | (PP to (NP the cup)) | NULL | NULL |
| 5 | 54 | 2012-09-10 17:20:45.300 | L | I | Pick (PRT up) | NULL | NULL | (NP the cup) | NULL | NULL | NULL |
| 6 | 55 | 2012-09-10 17:20:47.707 | M | D | pick (PRT up) | NULL | (NP I) | (NP the cup) | NULL | NULL | NULL |
| 7 | 56 | 2012-09-10 17:21:18.830 | L | D | failed | NULL | (NP You) | NULL | NULL | NULL | NULL |
| 8 | 57 | 2012-09-10 17:21:22.550 | U | D | failed | NULL | (NP I) | NULL | NULL | NULL | NULL |
| 9 | 58 | 2012-09-10 17:21:38.177 | L | I | Pick (PRT up) | NULL | NULL | (NP the cup) | NULL | NULL | NULL |
| 10 | 59 | 2012-09-10 17:21:40.723 | M | D | pick (PRT up) | NULL | (NP I) | (NP the cup) | NULL | NULL | NULL |
| 11 | 60 | 2012-09-10 17:22:04.113 | L | I | Move | NULL | NULL | NULL | (PP to (NP the pot)) | NULL | NULL |
| 12 | 61 | 2012-09-10 17:22:06.270 | M | D | move | NULL | (NP I) | NULL | (PP to (NP the pot)) | NULL | NULL |
| 13 | 62 | 2012-09-10 17:22:32.753 | L | I | Approach | NULL | NULL | NULL | (PP to (NP the cup)) | NULL | NULL |
| 14 | 63 | 2012-09-10 17:22:36.410 | M | D | approach | NULL | (NP I) | NULL | (PP to (NP the cup)) | NULL | NULL |
| 15 | 64 | 2012-09-10 17:22:59.533 | L | I | Pour | NULL | NULL | (NP the cup) | (PP to (NP the pot)) | NULL | NULL |
| 16 | 65 | 2012-09-10 17:23:02.643 | M | D | pour | NULL | (NP I) | (NP the cup) | (PP to (NP the pot)) | NULL | NULL |
| 17 | 66 | 2012-09-10 17:23:19.770 | L | I | Retract | NULL | NULL | (NP hands) | NULL | NULL | NULL |
| 18 | 67 | 2012-09-10 17:23:22.067 | M | D | retract | NULL | (NP I) | (NP hands) | NULL | NULL | NULL |
| 19 | 68 | 2012-09-10 17:24:01.723 | L | D | ends | NULL | (NP (NP the episode) (PP of (S (VP watering (NP the pot))))) | NULL | NULL | NULL | NULL |
| 20 | 69 | 2012-09-10 17:24:04.177 | U | D | ends | NULL | (NP (NP the episode) (PP of (S (VP watering (NP the pot))))) | NULL | NULL | NULL | NULL |

(a)

| | SNO | ETIME | MODULE | TYPE | VERB | AVERB | A1 | A2 | SPACE | TIME | ADJ1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 21 | 70 | 2012-09-10 17:27:54.910 | L | I | Water | NULL | NULL | (NP the cup) | NULL | NULL | NULL |
| 22 | 71 | 2012-09-10 17:27:57.630 | M | D | move | NULL | (NP I) | NULL | (PP to (NP the cup)) | NULL | NULL |
| 23 | 72 | 2012-09-10 17:28:14.910 | M | D | pick (PRT up) | NULL | (NP I) | (NP the cup) | NULL | NULL | NULL |
| 24 | 73 | 2012-09-10 17:28:34.130 | M | D | move | NULL | (NP I) | NULL | (PP to (NP the pot)) | NULL | NULL |
| 25 | 74 | 2012-09-10 17:28:47.503 | M | D | approach | NULL | (NP I) | NULL | (PP to (NP the cup)) | NULL | NULL |
| 26 | 75 | 2012-09-10 17:28:59.880 | M | D | pour | NULL | (NP I) | (NP the cup) | (PP to (NP the pot)) | NULL | NULL |
| 27 | 76 | 2012-09-10 17:29:15.300 | M | D | retract | NULL | (NP I) | (NP hands) | NULL | NULL | NULL |

(b)

**Fig. 6. Results of the sentential memory of the SCS after learning and executing "W*ater the pot*", (a) is the procedure of learning the new episode and (b) is executing it. MODULE is the module that the event occurred, L for listening, U for utterance, and M for motion modules. TYPE is the type of a sentence, D for declarative and I for imperative sentences.**

(a) Pick up the cup (b) Pick up the bottle (c) Pick up the can

**Fig. 7. Reusing the primitive action, "*Pick up* X", with other objects.**

SQL. In the table, ETIME indicates the time that the event happened, and MODULE and TYPE are the modules of the SCS and the type of the sentence, respectively. VERB is the verb of the sentence and AVERB are auxiliary verbs. A1 and A2 are the first and second arguments of the sentence. SPACE and TIME are the prepositional phrases indicating the space and time in which the event occurs. Fig. 6(a) shows how the robot learns a new episode with natural language. When a user indicates the start and end of an episode with the sentences, "*The episode of watering the pot begins*" and "*The episode of watering the pot end,*" the motion descriptor of SCS stores the sequence of the primitive actions. Fig. 6(b) shows the performance of the episode. The episode calls primitive actions hierarchically stored in the motion descriptor.

Fig. 7 shows the reuse of the primitive actions when the primitive action "*Pick up X*" was applied to a cup, can, and bottle. Table 4 lists the success rate of the execution of

**Table 4. Results of ordering the episode, '*Water the pot*'.**

| # | position (x, y, z) (mm) | success rates (10 times) |
|---|---|---|
| 1 | 200, 400, 750 | 8 |
| 2 | 350, 400, 750 | 8 |
| 3 | 300, 300, 750 | 7 |
| 4 | 300, 400, 750 | 10 |
| 5 | 400, 400, 750 | 9 |
| 6 | 400, 450, 750 | 9 |
| 7 | 450, 450, 750 | 10 |
| 8 | 400, 350, 750 | 10 |
| 9 | 450, 300, 750 | 8 |
| 10 | 500, 450, 750 | 9 |

the episode, "*Water the pot*." The success rate of the execution of the episode was checked when the cup was located in various positions. The results revealed an average success rate of 88%. Table 5 lists the error of reusability of "*Pick up X*" when the action was tested using a cup, bottle and can. The average success rate was approximately 83%. The failure was attributed to various sensorimotor errors, such as visual sensing errors, kinematics errors and mechanical errors of the robot's movement.

**Table 5. Result of reusing the primitive action, "*Pick up X*", with other objects.**

| objects | success rates (10 times) |
|---|---|
| Cup | 9 |
| Bottle | 8 |
| Spray can | 8 |

## 6. Conclusion

This paper proposed a methodology of natural-language-based robot control using a hierarchical behavior model based on a sentential cognitive system. The model consists of three hierarchical levels of behavior: episodes, primitive actions, and atomic functions. The natural language command of a user is translated to a sentence and parsed syntactically. Based on semantic parsing, the verbs of the sentences are connected to the episodes, and the phrases are linked to the cognitive information stored in the object descriptor. The primitive actions were made reusable by applying them to multiple episodes and by substituting the noun phrases, i.e. objects, while using the same verbs in the sentences for primitive actions. This approach can be applied to a service robot that provides help with housework or that assists people with disabilities. On the other hand, this study has some limitations in that the performance of the commands depends on the accuracy of the sensory motor modules of the SCS. Therefore, future work will focus on increasing the accuracy of the modules and applying other natural language commands that are used in daily life to the proposed model.

## References


[1] S. Lauria, G. Bugmanna, T. Kyriacou, and E. Klein, "Mobile Robot Programming Using Natural Language", *Robotics and Autonomous Systems*, Vol. 38, pp. 171–181, 2002. Article (CrossRef Link)

[2] D. Roy, K. Hsiao, and N. Mavridis, "Mental Imagery for a Conversational Robot", *IEEE Trans. on SMC, Part B*, Vol. 34, No. 3, pp. 1374-1383, 2004. Article (CrossRef Link)

[3] Deb Roy, "Semiotic Schemas: A Framework for Grounding Language in Action and Perception," *Artificial Intelligence*, Vol. 167, pp. 170–205, 2005. Article (CrossRef Link)

[4] Michael Levit and Deb Roy, "Interpretation of Spatial Language in a Map Navigation Task," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 37, No. 3, pp. 667-679, 2007. Article (CrossRef Link)

[5] K. Hsiao, N. Mavridis, and D. Roy, "Coupling Perception and Simulation: Steps toward Conversational Robotics," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, 2003. Article (CrossRef Link)

[6] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial Language for Human–Robot Dialogs," *IEEE Trans. on SMC Part C*, pp. 154-167, 2004. Article (CrossRef Link)

[7] S. Narayanan, "Karma: Knowledge-based Active Representations for Metaphor and Aspect," *Ph.D. dissertation*, Univ. California at BerkTomy, 1997. Article (CrossRef Link)

[8] Raymond J. Mooney, "Learning to Connect Language and Perception," *Proceedings of the 23th AAI Conference on Artificial Intelligence(AAAI)*, Chicago, 2008. Article (CrossRef Link)

[9] S. Coradeschi and A. Saffiotti, "An Introduction to the Anchoring Problem," *Robotics and Autonomous Systems*, vol. 43, pp. 85–96, 2003. Article (CrossRef Link)

[10] A. Chella, M. Frixione, and S. Gaglio, "Anchoring Symbols to Conceptual Spaces: the Case of Dynamic Scenarios," *Robotics and Autonomous Systems*, Vol. 43, pp. 175–188, 2003. Article (CrossRef Link)

[11] Y. Wei, E. Brunskill, T.s Kollar, and N. Roy, "Where to Go: Interpreting Natural Directions Using Global Inference," *in Proc. Of ICRA*, 2009. Article (CrossRef Link)

[12] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Grounding Verbs of Motion in Natural Language Commands to Robots," *in Proc. of International Symposium on Experimental Robotics*, 2010.

[13] Article (CrossRef Link)

[14] C. Matuszek, D. Fox, K. Koscher, "Following Directions Using Statistical Machine Translation," *in Proc. of HRI 2010*, pp. 251–258, Mar. 2010. Article (CrossRef Link)

[15] P. E. Rybski, K. Yoon, J. Stolarz, and M. M. Veloso, "Interactive Robot Task Training through Dialog and Demonstration," *in Proc. of HRI*, pp. 56-63, ACM, 2007. Article (CrossRef Link)

[16] K. Hsiao, S. Vosoughi, S. Tellex, R. Kubat, D. Roy, "Object Schemas for Responsive Robotic Language Use," *in Proc. of HRI* pp. 233-240 ACM, 2008. Article (CrossRef Link)

[17] M. P. Marcus, B. Santorini, M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: the Penn Treebank," *Computational Linguistics*, Vol. 19, 1993. Article (CrossRef Link)

[18] http://www.microsoft.com/en-us/kinectforwindows/

[19] http://dragonmobile.nuancemobiledeveloper.com/

[20] Article (CrossRef Link)

**Hyunsik Ahn** is an associate professor of Department of Robot System Engineering of Tongmyong University, Busan, Korea. He received his B.S. and M.S. degrees in Department of Electronics Engineering from Kyungpook National University, Korea, in 1986 and 1989 respectively, and his Ph.D. in 1998. He received M.S. degree in sociology from Pusan National University in 2007. He worked for Research Institute of Industrial Science and Technology (RIST) as a senior researcher from 1992 to 1998. Dr. Ahn was a visiting researcher at Computational Perception Lab. of Georgia Tech, USA, in 2007. He serves as a vice president of the computer society of The Institute of Electronics Engineers of Korea (IEEK). He currently serves as a program chair of International Conference on Green and Human Information Technology (ICGHIT) 2013. He is a member of the IEEE. His research interests include robot intelligence, robot vision, and cognitive robotics. He is also interested in ethics of technology and sociology of robot society.

**Hyun-Bum Ko** received his B.S. degree in Department of Robot System Engineering of Tongmyong University, Busan, Korea, in 2011. Currently, he is a graduate student of M.S. in the University. His research interests are robot intelligence, robot vision, and ontology for robotics.