

Autonomous Task Execution of a Humanoid Robot using a Cognitive Model

KangGeon Kim, Ji-Yong Lee, Dongkyu Choi, Jung-Min Park and Bum-Jae You

Abstract—These days, there are many studies on cognitive architectures, which are developed based on human cognitive models. Some other studies are focused on applying these cognitive architectures to the autonomous task execution of humanoid robots. In this paper, we have shown that a real world robot, Mahru-Z can execute a task autonomously in the Blocks World domain, using a cognitive architecture, ICARUS. For this project, diverse techniques such as system integration, human-like manipulation based on vision, environmental information update techniques etc are used. Successful completions of these tasks imply that we can expect similar results for the more diverse and complicated tasks as well.

I. INTRODUCTION

In recent years, there have been several studies on cognitive architectures. There are various cognitive architectures such as Soar [1], ACT-R [2], and these architectures imitate the human intelligence such that it becomes possible to learn and provide solutions for problems through inferences. These architectures can learn through various ways such as problem solving, observations, failures etc [1], [3], [4], [5], [6].

We have developed a new system to materialize cognitive tasks in the real physical domain by applying ICARUS (a cognitive architecture developed by Stanford University) to the recently updated robot platform Mahru-Z. The ability of ICARUS is proved in many fields such as FreeCell Solitaire game [5], simulation of urban driving [7], and first-person shooter game, Urban Combat [8] etc. As we are working on developing a more intelligent human-like robot, therefore, applying a cognitive architecture developed in the field of cognitive science to a real robot would be very significant in our field.

However, there are many factors to be considered before applying a cognitive architecture to a real world humanoid robot. Usually, a robot performs its task through very active interaction with its environment; hence, it is very important to sense and recognize environment information and to reflect it back for task execution [9].

In this paper, experiments are conducted such that a robot performs the task of stacking the blocks by sorting them

according to colors in the Blocks World domain. Several pre-requisites and necessities are mentioned which would help to execute the autonomous tasks using the cognitive architectures. These necessities would also help to look through the system integration, human-like manipulation process based on vision, environment information update techniques etc. Further studies would be pointed out which could make the robot perform more diverse and complex tasks in the service space.

The remainder of this paper is organized as follows. ICARUS is introduced as a cognitive architecture in Section 2. The humanoid platform, Mahru-Z, is described in Section 3.1; Section 3.2 describes the communication interface between the cognitive architecture and the robot. Section 3.3 explains the recognition algorithm and the robot arm path planning method are discussed in Section 3.4. Section 4 presents the experiments of autonomous task execution.

II. ICARUS, COGNITIVE ARCHITECTURE

Cognitive architecture is a frame work for human intelligence based on the cognitive science theory. ICARUS distinguishes the conceptual and skill knowledge bases and the architecture has separate memories for them. It further differentiates long-term and short-term knowledge such that the long-term conceptual memory stores the definition of its conceptual knowledge or concepts while the short-term conceptual memory or belief memory, houses instances of these concepts that are true in the current state. Similarly, a long-term skill memory stores the definitions of ICARUS' procedural knowledge or skills, and the short-term skill memory records instantiated skills. These skill instances are, however, closely tied to the goals they achieve. The short-term skill memory also serves as a goal memory and it stores substantially more information than the skill instances themselves.

Table I shows some sample concepts used in the Blocks World domain. Concept definitions start with a head, which includes the name of the concept, and its arguments that are often variables marked with a preceding question mark as in ?gripper and ?block. The definitions also have several optional fields like :percepts that includes perceptual matching conditions, :tests that specifies conditions among matched variables, and :relations that has references to other concepts.

The first two concepts shown are primitive, in the sense that they consist solely of variable matchings against objects in the world like a gripper and a block and the conditions among these variables. The last concept, however, is a non-primitive one, which refers to another concept. In this case,

This work was supported by KIST [Cognitive Task Planning-based Whole Body Manipulation]

K. Kim is with Cognitive Robotics Center, Korea Institute of Science and Technology (KIST), Seoul, Korea. danny@kist.re.kr

J. Lee is with Cognitive Robotics Center, Korea Institute of Science and Technology (KIST), Seoul, Korea. wisrage@kist.re.kr

D. Choi is with Institute for the Study of Learning and Expertise, Palo Alto, California, USA. dongkyuc@stanford.edu

J. Park is with Cognitive Robotics Center, Korea Institute of Science and Technology (KIST), Seoul, Korea. pjmk@kist.re.kr

B. You is with Cognitive Robotics Center, Korea Institute of Science and Technology (KIST), Seoul, Korea. ybj@kist.re.kr

TABLE I

SOME SAMPLE CONCEPTS FOR OUR HUMANOID ROBOT, MAHRU-Z IN THE BLOCKS WORLD DOMAIN

((holding ?gripper ?block)
:percepts ((gripper ?gripper status ?block)
(block ?block)))
((right-gripper ?gripper)
:percepts ((gripper ?gripper))
:tests ((equal ?gripper 'gripper-r)))
((clear ?block)
:percepts ((block ?block))
:relations ((not (on ?other ?block))))

the reference is negated, and the concept (clear ?block) matches when the predicate (on ?other ?block) is not true in the current state.

In Table II, we provide some sample skills for Mahru-Z in the Blocks World domain. As with concepts, ICARUS' skills consist of a head and several optional fields. While the :percepts field serves the same purpose as that of concepts, the :start field specifies the preconditions of the skill, an :actions field has its implied actions in the world, and a :subgoals field includes a subgoal decomposition for the particular skill.

The first example shown is a primitive skill, which directly refers to the basic actions that an ICARUS agent can apply in the environment. However, the rest of the skills are non-primitive, and they include references to subgoals that, in turn, lead to other skills. In this way, both concepts and skills in the ICARUS architecture are hierarchically organized, and they provide rich vocabulary to describe complex states and procedures required by domains in the real world.

The ICARUS architecture runs in distinct cycles. In each cycle, it performs a series of cognitive processes including the perception of its surroundings, the inference of concept instances based on the perceived data, the evaluation of skills under the current state and the execution of actions implied by the chosen skill instance. Fig. 1 shows the ICARUS' operation in each cycle. At the beginning of each cycle, ICARUS receives information on the objects it can perceive from the environment. This information is deposited in the perceptual buffer. ICARUS then matches its concepts stored in the long-term conceptual memory against perceived objects, infers all the concept instances that are true in the current state, and stores them in its belief memory. Based on the current beliefs, the architecture evaluates its hierarchical skills and finds a skill path that is most relevant to the situation.

Once an executable skill path is found, the architecture executes actions implied by the skill path to change its environment. This, in turn, changes ICARUS' perception on the next cycle. Then the system continues to the next step of the procedure. It is notable that this cyclic operation gives reactivity to the ICARUS architecture while staying goal-

TABLE II

SOME SAMPLE SKILLS FOR OUR HUMANOID ROBOT, MAHRU-Z IN THE BLOCKS WORLD DOMAIN

((holding ?gripper ?block)
:percepts ((gripper ?gripper)
(block ?block shape ?shape)
(block ?from))
:start ((unstackable ?gripper ?block ?from)
(graspable-at-angle ?block))
:actions ((*grasp ?block ?gripper ?shape)))
((one-color-sorted ?color)
:percepts ((block ?block1 color ?color)
(block ?block2))
:start ((same-color ?block1 ?block2)
(clear ?block2))
:subgoals ((on ?block2 ?block1)
(one-color-sorted ?color)))
((one-color-sorted ?color)
:percepts ((block ?block1 color ?color)
(block ?block2)
(block ?block3))
:start ((same-color ?block1 ?block2)
(same-color ?block1 ?block3)
(on ?block2 ?block1))
:subgoals ((clear ?block3)
(on ?block3 ?block2)
(one-color-sorted ?color)))
((color-sorted)
:start ((not-color-sorted ?color))
:subgoals ((one-color-sorted ?color)
(color-sorted)))

oriented. Therefore, ICARUS agents can readily adapt to unexpected situations or outcomes. For example, if the robot fails to pick up a block or loses the grip of one, ICARUS will perceive the unexpected outcome in its subsequent cycle and correct the situation through retrials. We will discuss more on ICARUS' ability to recover from failures later in this paper. In the next section, we introduce our robot platform, MAHRU-Z in more detail.

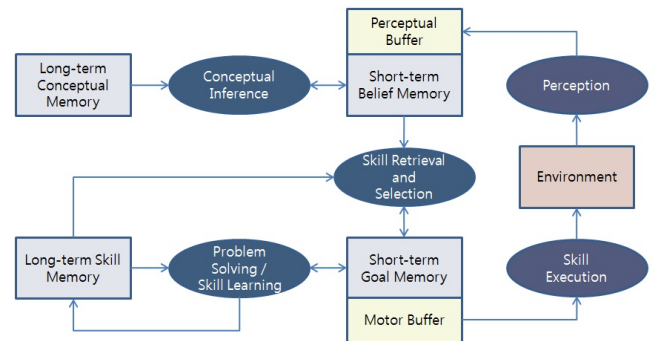


Fig. 1. An execution cycles in ICARUS architecture

III. APPLYING TO THE HUMANOID ROBOT

In this section, we will introduce 4 main streams: humanoid robot platform used in the experiment, interface for the interaction between Cognitive Architecture and the robot, vision algorithm, and robot arm path planning.

A. Humanoid Platform, Mahru-Z

We performed a demonstration of the network-based humanoid operation using humanoid robots. In our system, MAHRU-Z is used as the humanoid robot (Fig. 2). MAHRU-Z was developed by the Korea Institute of Science and Technology (KIST) and is equipped with 7DOF for each arm, 2DOF for the neck and 3DOF for each hand. A bumblebee stereo camera is mounted on the robot's head to obtain the depth image. All experiments were performed on two different types of PCs: Windows XP, Pentium(R) Core 2 Duo CPU T7700 2.4GHz, 1GB RAM for the perception system and Linux, Pentium(R) M CPU 1.73GHz, 1GB RAM for the control system.

The robot is specifically designed as a network-based humanoid, which can send various sensor data to external servers over the wireless network to perform high-level recognition, inference, and decision-making. Its system structure also facilitates easy updating of various application services, through the network, so the same robot can be used to provide improved services depending on its environment and commands.

B. Communication Interface between the Cognitive Architecture and the Robot

For the communication between the cognitive architecture and the robot, we built a TCP based communication interface. External environment information recognized by the robot or the robot action command generated from the architecture is transmitted through this interface. The serial process is as follows:

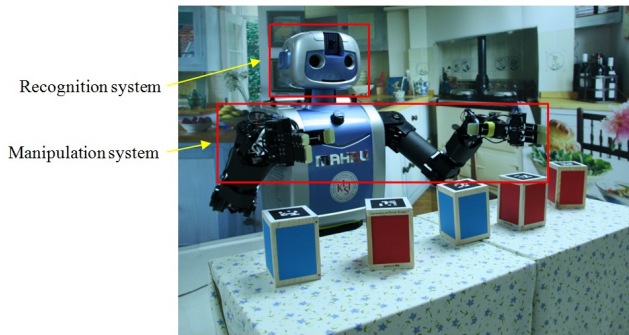


Fig. 2. Mahru-Z a humanoid platform

- 1 **Request of recognition** The architecture requests recognition of external environment information from the robot using its sensor system.
- 2 **Result of recognition** The robot sends the results of recognition to the architecture.
- 3 **Action command** The action command generated from the architecture is transmitted to the robot.
- 4 **Result of performance** The robot sends the results of performance to the architecture to judge whether the command is executed correctly or not.

Cognitive architecture goes through the process of learning or inferring on the basis of environment information updated every time and corresponds to the external environmental changes. The updated items are object information, table information, robot status information etc. However, there are some important issues for the robot task performance which need to be addressed; VIZ, how to recognize the diverse environmental information, how to update the information, how often to update etc. If the whole information is updated whenever the command is executed, it will help us to increase the task capacity but it is very time consuming. So, we need to set the best update frequency.

C. Recognize the External Environmental Information

The robot vision sensor has the most important role in recognizing the external environment information. In order to obtain the external environmental information to transmit to the cognitive architecture, the robot starts to recognize by its vision sensor. The recognition tasks include the following: capturing the type of the target object, the present position, the pose and the status. To obtain the real time environment information, it is necessary to apply the optimal algorithm and to use the robust method for the light condition in the external environment. In the case of multi object, we need to verify the relations between the objects and to execute multi object recognition algorithm by avoiding the occlusion such as recognizing two different objects as one.

In this paper, we made the patterns on the top of the block and the robot could recognize the block position and the pose using the pattern (Fig. 3). The stereo camera let us know the exact position in 3D and the direction of the pattern gives us the right direction information of the block. We used ARToolKit which is an open source for the pattern recognition [10]. The same color block has the same pattern ID which enables us to distinguish among colors.

This algorithm is also used to evaluate the position right after the block has moved on. As we know the destination, we need to check the color of the block so that we can know whether it has moved to the right position or not.

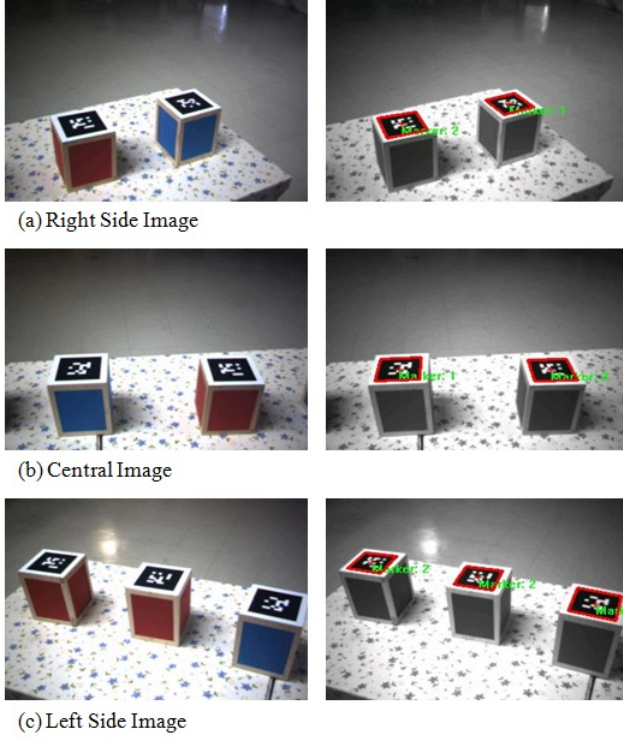


Fig. 3. **Pattern on the block** The robot recognizes the block position and the pose using the pattern. Original color image (left) and recognition results (right).

D. Robot Arm Path Planning

Cognitive architecture infers the skills to achieve the goal and translates them into an action list which enables the robot to execute the task. This action command is transmitted to the robot control part by the communication interface. The action command made in the architecture indicates only the final destination of the block. So the robot control system has to interpret the command message and transform it to make the robot move. Depending on the position and the direction of the block, the approaching path should be different. Of course there are many things to consider such as the via point, the approaching direction of the robot arm, the height of the lifting-up, the alternative path to avoid the collision between the blocks etc. Also we need to consider the collision problem between the block and the robot arm, and between both the arms of the robot itself.

As there are many limitations in the work space for the robot arms, we need to generate the path considering the work space of both the arms of the robot. In this paper, we simply designated the relative coordination on the basis of the object to make an approaching path [11], [12]. We are

still working on the issue of generating the path automatically by avoiding the obstacles.

IV. EXPERIMENT

The tasks that we want to perform with the real Mahru-Z are as follows. The cognitive architecture must go through a process of inference in order to stack the different colored blocks on the table and to sort into the same colors. This task is a part of the autonomous arrangement and organization of the robot and can be extended to various tasks also. Our purpose in this stacking task is to prove the capacity of the cognitive architecture. The robot will start to recognize the 5 different colored blocks on the table set at random and will stack the same colored blocks together. First of all, the robot will recognize the working environment. The external working environment in this experiment consists of a block, a table, the robot arm etc. Our final goal in this research is to stack the same colored blocks together and to line them up. In the cognitive architecture, the concept and the skill must be defined to achieve this goal. Once the task is started, the architecture will infer the best performance order to achieve this final goal step by step. The following image shows Mahru-Z sorting the blocks according to their colors based on the inferred command from the architecture (Fig. 4).

A definition of the concept and the skill is needed to cover the various cases that can occur during the experiment process. For example, when the object block is hidden by another block, the robot cannot grab the object block. Then there must be a concept and a skill defined in order to

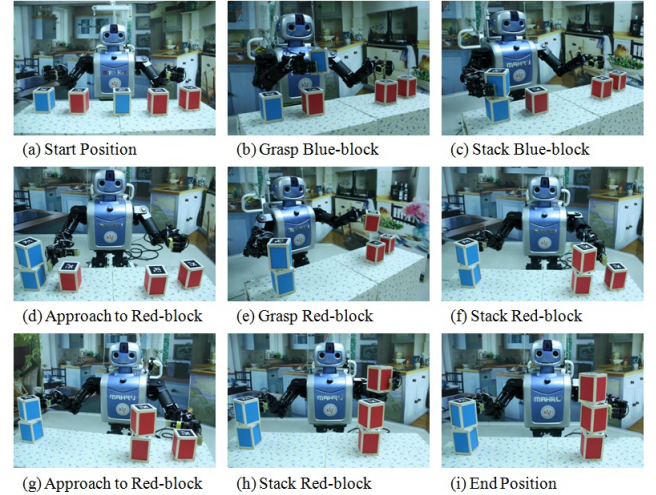


Fig. 4. Mahru-Z performing the color sorting of blocks

shift the obstacle block and move the object block.

As noted earlier, ICARUS sometimes faced unexpected outcomes such as failing to grab a block or dropping a block elsewhere than its intended location. In such cases, the architecture reacted to the unexpected situation, and persistently made attempts to fix the failure or proceeded with a different skill that is known to work in the new situation. Of course, these failures caused an inevitable increase in ICARUS' cognitive cycles to complete its tasks. For instance, at 10% probability of action failures, it took 17.1 cycles on average for ICARUS to complete the color sorting task, which is an increase of approximately 30%.

A. Environment Information Update Issue

The architecture and the robot can exchange the information periodically and can easily adapt to the environmental changes. When the environment information is exchanged many times, the cognitive architecture can perceive the present situation accurately and can solve the problems wisely. However, as it is time consuming, it is important to set the optimal environment information update cycle and its range. In this experiment, we are not updating all environmental information but we only check the recently moved block to see if it has arrived to the right position (Fig. 5). Once the movement is checked, then it is recorded in the environment information. As a result, we can have the optimal update rate with high efficiency.

The process of recognizing the object block is as follows. First of all, as the robot knows the position of the destination,

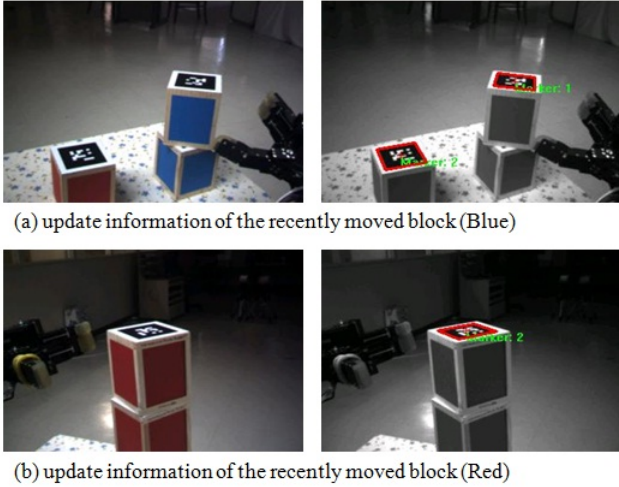


Fig. 5. **Environmental Information Update** Robot is able to judge whether the block is in the proper position or not. Original color image (left) and recognition results (right).

it has to judge whether the block is inside the proper position range which is limited by the range of estimated movement. When the position of the object block is in the estimated position or within the probable error range, we regard it as a regular and normal movement, and then update the position of the block. If the block is not in the estimated range, we regard it as an error situation. Then the whole procedure of the recognition for all blocks starts again. However, there are some risks for this method. When the moving block touches another block which leads to a change in the position, this slight movement of another block is not checked. To avoid this kind of error situation, we need to update not only the position of the block that had regular and periodic movements but also the location information of all the blocks.

We verified that the ICARUS architecture is able to make high-level decisions in the robotic Blocks World domain even under the possibility of action failures, and that the humanoid, MAHRU-Z can execute the commands from ICARUS correctly. The physical restrictions in the robot platform have added complexities in various dimensions, but the architecture successfully manages to produce a valid procedure to achieve the given goals.

V. CONCLUSION AND FUTURE WORK

A. Conclusion

In this paper, we could confirm and prove that cognitive architecture performs a very important role in robot intelligence. The robots must be more intelligent to live together with human beings and to satisfy the necessities of human beings. It is very important that the humanoid robots, which are similar to the human beings, perform the tasks by imitating the human intelligence. This could overcome the limitations of the present robots which perform the simple tasks in a regular way repeatedly. Now, the robots could take a new step forward to co-exist with human beings at home and to work with human beings at their work places and help them with the diverse tasks.

B. Future Work

The main purpose of this paper was to see whether the architecture applied to a real robot could make good responses to the external environment efficiently and could function properly. Further studies will follow constantly regarding the recognition ability to adapt to the various environmental changes and positive reactions. We could verify the capacity of the architecture by a simple task called the "block stacking problem"; so now this architecture might be used in more diverse and complex tasks in near future.

In this experiment, the robot was performing the tasks at its own place; however, a real humanoid robot is able to move to the task place and the moving task will be added in further studies. The robot's capability to move around with its biped will open up new possibilities in its applications and it certainly will challenge the ICARUS architecture's ability to control more complicated behaviors.

Until now, the tasks were performed by generating simple arm paths. But, the automatic arm path generating algorithm, while avoiding the real collision, is a very interesting subject of research and we are discussing and working on it with our research team.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Cognitive Robotics Center and reviewers' comments.

REFERENCES

- [1] J. Laird, P. Rosenbloom, and A. Newell, Chunking in Soar: The anatomy of a general learning mechanism, *Machine Learning*, vol. 1, 1986, pp. 11-46.
- [2] J. Anderson, *Rules of the Mind*, Lawrence Erlbaum, Hillsdale, NJ; 1993.
- [3] R. Jones, and P. Langley, A Constrained Architecture for Learning and Problem Solving, *Computational Intelligence*, vol. 21, 2005, pp. 480-502.
- [4] N. Nejati, P. Langley, and T. Konik, "Learning Hierarchical Task Networks by Observation", in *Proceedings of the Twenty-Third International Conference on Machine Learning*, Pittsburgh, PA, 2006.
- [5] P. Langley, D. Choi, and S. Rogers, Acquisition of hierarchical reactive skills in a unified cognitive architecture, *Cognitive Systems Research*, vol. 10, 2009, pp. 316-332.
- [6] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, *Journal of Robotics and Autonomous Systems*, vol. 47, no. 2-3, 2004, pp. 109-116.
- [7] D. Choi, M. Morgan, C. Park, and P. Langley, "A testbed for evaluation of architectures for physical agents", in *Proceedings of the AAAI-2007 Workshop on Evaluating Architectures for Intelligence*, Vancouver, BC, 2007.
- [8] D. Choi, T. Konik, N. Nejati, C. Park, and P. Langley, "Structural transfer of cognitive skills", in *Proceedings of the Eighth International Conference on Cognitive Modeling*, Ann Arbor, MI, 2007.
- [9] K. Kim, J. Lee, S. Kim, J. Lee, M. Jeong, C. Kim, and B. You, "Coordinated Task Execution by Humanoid Robot", *ICROS-SICE International Joint Conference*, 2009, pp. 3496-3503.
- [10] <http://www.hitl.washington.edu/artoolkit/>
- [11] S. Kim, C. Kim and J. Park, "Human-like arm motion generation for humanoid robots using motion capture database", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3486-3491.
- [12] A. Morales, T. Asfour, D. Osswald, S. Schulz and R. Dillmann, "Towards an anthropomorphic manipulator for an assistant humanoid robot", *Robotics: Science and Systems-Workshop on Humanoid Manipulation*, MIT, USA, 2006.