Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Preliminary Study on Fostering Computational Thinking by Constructing a Cognitive Model

Kazuaki KOJIMA[a*] & Kazuhisa MIWA[b]
[a]*Learning Technology Laboratory, Teikyo University, Japan*
[b]*Graduate School of Informatics, Nagoya University, Japan*
*kojima@lt-lab.teikyo-u.ac.jp

Abstract: Recent research on education has paid much attention to computational thinking. To foster computational thinking, this study proposes an approach that adopts cognitive modeling of problem solving processes for a production system. Cognitive models for a production system are useful to externalize knowledge used in problem solving processes; thus, these models are expected to make hidden assumptions explicit. We conducted lectures where undergraduate students created cognitive models of a toy problem. Tests to describe the rules necessary in solving this problem were provided to the students. The results indicated that the rules described in the pretests omitted many conditions in the pretests, whereas the presence of the conditions improved in the posttest. Therefore, the effectiveness of cognitive modeling in eliciting and clarifying assumptions in externalizing knowledge in problem solving processes was proved.

Keywords: Computational thinking, cognitive model, production system, problem solving

1. Introduction

The literature on education has paid considerable attention to computational thinking, the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent (Brennan, and Resnick, 2012; Yadav, Mayfield, Zhou, Hambrusch, and Korb, 2014). The acquisition of computational thinking skills has been observed as a side effect of learning to program, and the borders between computational thinking and coding or programming are considered unclear (Howland, and Nicholson, 2009). Therefore, popular systems used in fostering computational thinking are graphical programming environments and web-based simulation authoring tools (Grover, and Pea, 2013), which are easy for learners who are not information-engineering students.

To foster the computational thinking of novice learners, this study proposed an approach that adopts cognitive modeling of problem solving processes for a production system, one representative architecture that has been long used in cognitive science research. Instead, of constructs in general programming languages, such as repetition and selection, production systems use if-then rules comprising operations and their conditions. Cognitive models of a production system are useful in externalizing the declarative and procedural knowledge used in problem solving processes; therefore, they are expected to make hidden assumptions explicit and activate reflective thinking or metamonitoring in cognitive processes (Fum, Del Missier, & Stocco, 2007; Miwa, Morita, Nakaike, & Terai, 2014). These types of effects can enhance the essential abilities involved in computational thinking. As a first step in our approach, this study preliminary investigated the effects of creating a cognitive model on externalizing knowledge in problem solving processes.

2. Method

We conducted lectures where undergraduate students created cognitive models in a class of cognitive science. Although most students had experienced programming in other classes, they had not experienced training from experts in information engineering.

2.1.1 Tool and Procedures

To create cognitive models, the students used DoCoPro (Nakaike, Miwa, Morita, & Terai, 2009), a production system designed for learning by novice students. Figure 1 shows a screenshot of DoCoPro. Representations of the states observed in problem solving processes were shown in the working memory in the left frame. The students created their models by editing rules in the editor in the right frames and simulated and evaluated problem solving processes by executing the models with the controller in the upper frame.

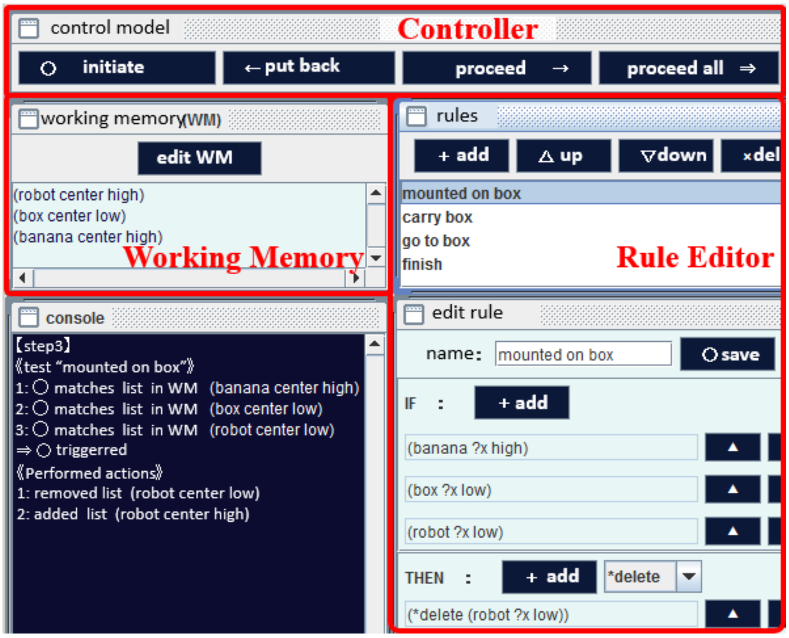

Figure 1. Part of Screenshot of DoCoPro

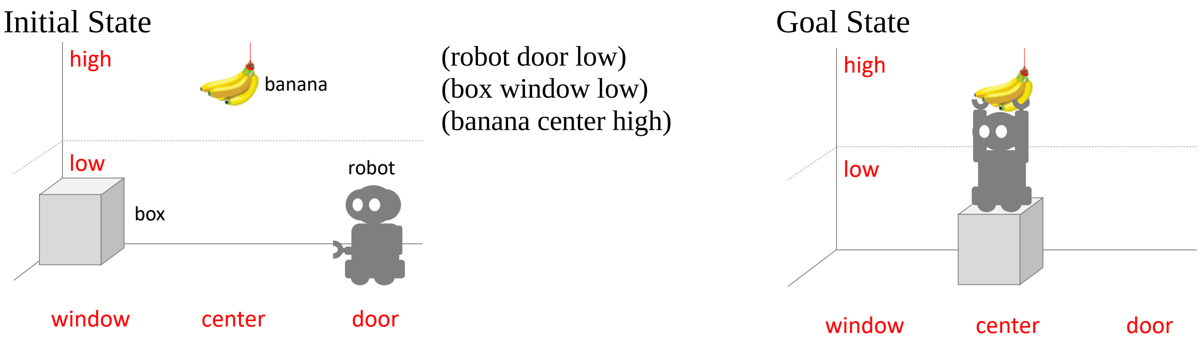

Figure 2. Initial and Goal States in Robot and Banana Problem

Three lectures for cognitive modeling were conducted. In the first lecture, the students learned about a production system with DoCoPro. DoCoPro has instructional texts that enable a learner to learn how to construct production system models with an example of creating a block-stacking model. The students were asked to complete the instructional texts by themselves before the start of the second lecture, which was 2 weeks following the first lecture.

In the second lecture, the students were given a *robot and banana* problem, an altered version of the famous toy problem *monkey and banana*. Figure 2 illustrates the initial and goal states of the presented problem. The students were also provided with a representation of the initial state

(Figure 2), asked to design knowledge that enables the robot to obtain the banana and model it with DoCoPro, and told that the robot can carry the box and be mounted on the box to ensure the banana is within the robot's reach.

Next, the students were asked to describe the if-then rules necessary to solve this problem with natural sentences when the problem was first presented. We refer to this task as *pretest 1*. They were instructed to design general rules adaptable to various initial states and told that the number of rules was four. Each student then engaged in creating a model of the robot and banana problem with DoCoPro in 60 min. After the model creation, they again described rules of the problem by using sentences (*pretest 2*).

In the third lecture, which was 1 week after the second lecture, the students were presented four different initial states of the robot and banana problem, in which the box is placed in the same location as the robot, the box is under the banana, the box and robot are under the banana, and the banana is put on the place where the robot stands. The students' task was to enhance models they created in the second lecture in a manner where the models could solve the problem from any of the initial states in 60 min. This task was expected to facilitate the sophistication of the conditions in the rules of the student models. At the end of the third lecture, the students were provided the task to describe rules in the same manner (*posttest*).

2.1.2 Data Analysis

We assessed whether the rules described by each student in pretest 1, pretest 2, and the posttest included the information required for a complete model that can solve the robot and banana problem from the five initial states. For each piece of information, the students' descriptions were categorized as *present* when including corresponding information, *incomplete* when including corresponding information whose conditions and operations were specialized or insufficient, or *absent* when including no relative information. The pieces of information for the four rules were as follows.

Rule 1 (the robot goes to the box)

R1C1 (Condition 1 in Rule 1): the vertical position of the banana is high
R1C2: the horizontal positions of the robot and box are different
R1O (Operation in Rule 1): alter the horizontal position of the robot to the same as the box

Rule 2 (the robot carries the box)

R2C1: the vertical position of the banana is high
R2C2: the horizontal positions of the banana and box (and/or robot) are different
R2C3: the horizontal positions of the robot and box are identical
R2O1: alter the horizontal position of the robot to be the same as the banana
R2O2: alter the horizontal position of the box to be the same as the banana

Rule 3 (the robot mounts on the box)

R3C1: the vertical position of the banana is high
R3C2: the vertical position of the robot is low
R3C3: the horizontal positions of the banana and box (or robot) are identical
R3C4: the horizontal positions of the robot and box are identical
R3O: alter the vertical position of the robot to high

Rule 4 (the robot obtains the banana)

R4C1: the vertical positions of the banana and robot are identical
R4C2: the horizontal positions of the banana and robot are identical
R4O: halt (a statement representing the end of the problem solving)

The following is an example of Rule 1: If the descriptions by a student included the sentence "the positions of the robot and box are different," the R1C2 of the student was categorized as present. The descriptions by another student, "the robot is at the door" and "the box is at the window," were categorized as incomplete R1C2, and absent R1C1 because no other conditions regarding Rule 1 were included. Sentences indicating R1O, such as "the robot goes to the window" and "the robot moves around," were categorized as incomplete, and "the robot moves to the box" as present.

3. Results

We analyzed the data of 60 students who participated in all three lectures. Figures 3, 4, 5, and 6 indicate the categories for each condition and operation of the four rules in the three tests. In pretests 1 and 2, many conditions were absent, whereas operations were rarely absent. Notably, many of the three operations in Rules 1 and 2 were incomplete. Present conditions and operations increased in the posttest. We compared the three tests by using the chi-square test; the result indicated significant differences in R1C1 ($\chi^2(4) = 35.42$, $p < .01$), R1C2 ($\chi^2(4) = 14.58$, $p < .01$), R1O ($\chi^2(4) = 20.89$, $p < .01$), R2C1 ($\chi^2(4) = 27.51$, $p < .01$), R2C2 ($\chi^2(4) = 28.20$, $p < .01$), R2O1 ($\chi^2(4) = 29.36$, $p < .01$), R2O2 ($\chi^2(4) = 28.75$, $p < .01$), R3C1 ($\chi^2(2) = 30.67$, $p < .01$), R3C2 ($\chi^2(2) = 10.18$, $p < .01$), R3C3 ($\chi^2(4) = 23.97$, $p < .01$), R3C4 ($\chi^2(4) = 20.91$, $p < .01$), R4C1 ($\chi^2(4) = 31.43$, $p < .01$) and R4C2 ($\chi^2(4) = 30.78$, $p < .01$), and a marginally significant difference in R4O ($\chi^2(2) = 5.35$, $p < .10$). No significant differences were found in R2C3 ($\chi^2(4) = 2.43$, *n.s.*) and R3O ($\chi^2(4) = 4.08$, *n.s.*). Furthermore, the results of residual analysis indicated the following.

R1C1: absent was high and present was low ($p < .01$) in pretests 1 and 2
absent was low and present was high ($p < .01$) in the posttest

R1C2: absent was high ($p<.01$), incomplete was high ($p < .10$), and present was low ($p<.01$) in pretest 1
absent was low and present was high ($p < .01$) in the posttest

R1O : incomplete was high and present was low ($p < .01$) in pretest 1
incomplete was low and present was high ($p < .01$) in the posttest

R2C1: absent was high and present was low ($p < .01$) in pretest 1
absent was high and present was low ($p < .05$) in pretest 2
absent was low and present was high ($p < .01$) in the posttest

R2C2: absent was high and present was low ($p < .01$) in pretests 1
absent was low and present was high ($p < .01$) in the posttest

R2O1: incomplete was high and present was low ($p < .01$) in pretest 1
incomplete was low and present was high ($p < .01$) in the posttest

R2O2: incomplete was high and present was low ($p < .01$) in pretest 1
incomplete was low and present was high ($p < .01$) in the posttest

R3C1: absent was high and present was low ($p < .01$) in pretest 1
absent was low and present was high ($p < .01$) in the posttest

R3C2: absent was high and present was low ($p < .01$) in pretests 1

R3C3: absent was high ($p < .01$), incomplete was high ($p < .10$), and present was low ($p < .01$) in pretest 1
incomplete was high ($p < .10$) in pretest 2
absent was low and present was high ($p < .01$) in the posttest

R3C4: absent was high and present was low ($p < .01$) in pretests 1
absent was low ($p < .01$), incomplete was low ($p < .05$), and present was high ($p < .01$) in the posttest

R4C1: absent was high and present was low ($p < .01$) in pretests 1
absent was low ($p < .01$), incomplete was low ($p < .05$), and present was high ($p < .01$) in the posttest

R4C2: absent was high and present was low ($p < .01$) in pretests 1
absent was low and present was high ($p < .01$) in posttest

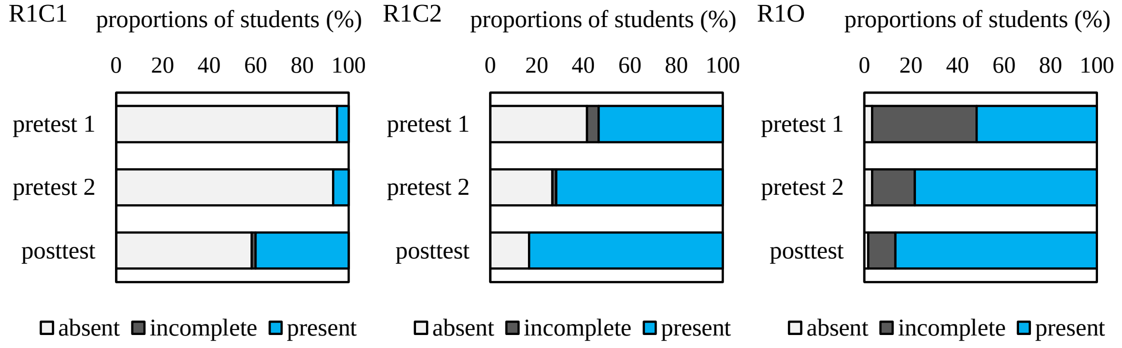

Figure 3. Categories for Rule 1 in Three Tests

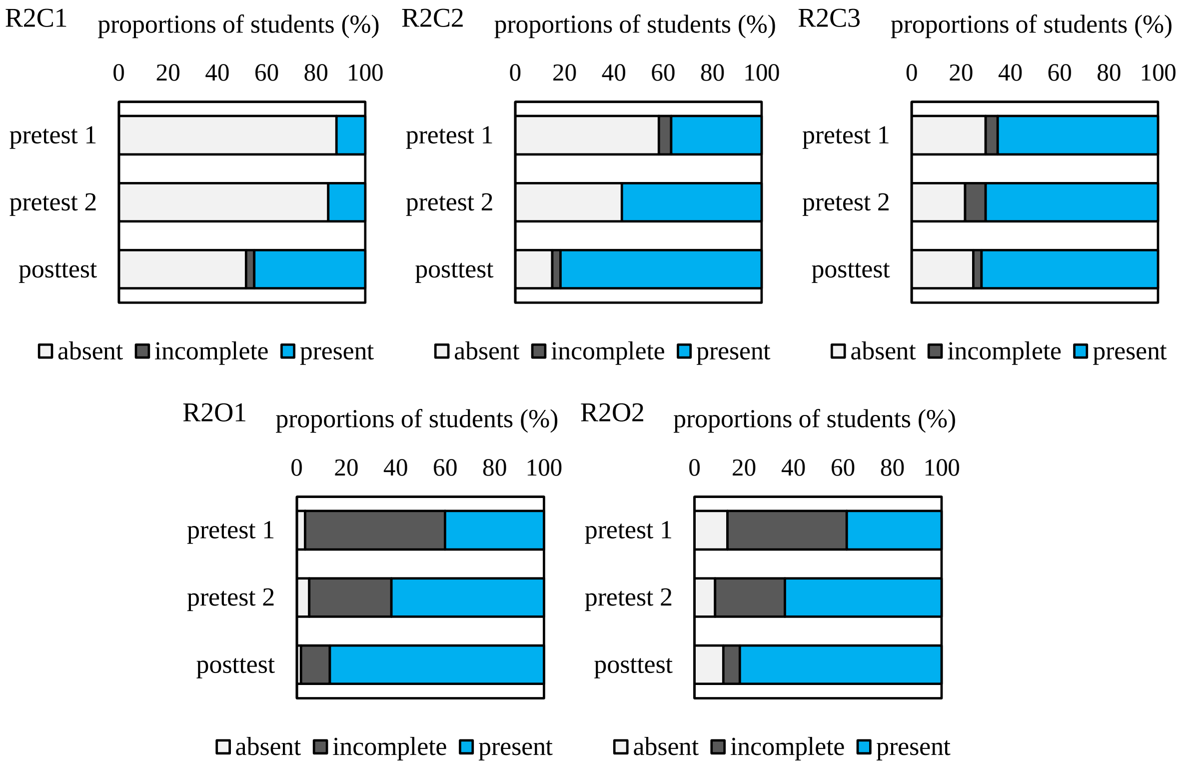

Figure 4. Categories for Rule 2 in Three Tests

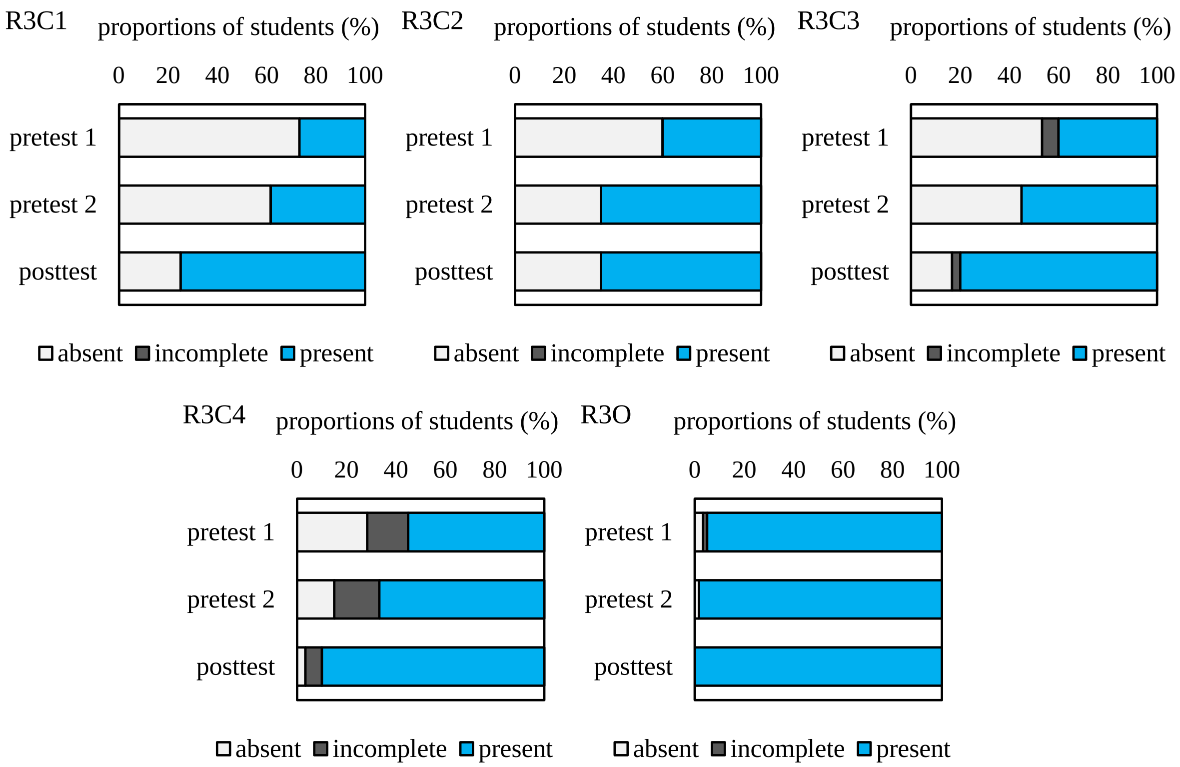

Figure 5. Categories for Rule 3 in Three Tests

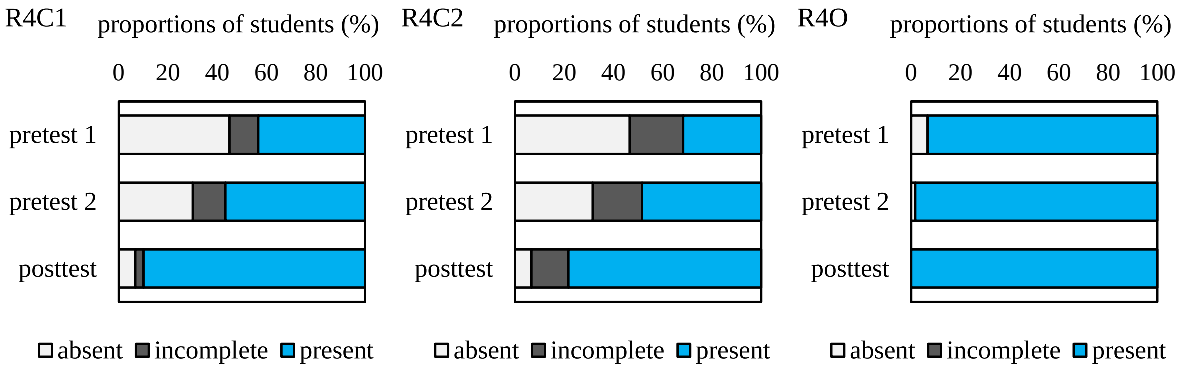

Figure 6. Categories for Rule 4 in Three Tests

4. Discussion

As described in section 3, descriptions of the four rules in the robot and banana problem omitted many conditions in pretest 1. In pretest 2, the descriptions of conditions did not differ overall from pretest 1, although the students had experienced creating models in DoCoPro. In contrast with the pretests, presence of conditions was improved in the posttest after the students had experienced the enhancement of the models. That phenomenon may be because the students carefully examined the conditions in the rules of their models by adapting the rules to various situations. The state when the banana is in the high position (R1C1, R2C1, and R3C1) is the prerequisite when the operations in Rules 1, 2, and 3 are necessary, and the relation of the positions of the banana, box, and robot must be clarified to appropriately use the three rules. Perhaps those can be hidden assumptions in this problem, and people can easily solve these problems without an awareness of such assumptions. To foster computational thinking, however, support enabling a learner to elicit and clarify such assumptions must be indispensable. In conclusion, the results proved the effectiveness of cognitive modeling for factors such as support.

The next task of our study is to further analyze the data collected from the three lectures. Although the descriptions to externalize knowledge in problem solving were improved through modeling, some of the students could not successfully create models. Another task is to enhance support for learning by constructing cognitive models. Although the present conditions increased in the posttest, their numbers remained low for R1C1 and R2C2; we regard this as a critical problem to be addressed.

References

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of 2012 Annual Meeting of the American Educational Research Association*.

Fum, D., Del Missier, F., & Stocco, A. (2007). The Cognitive Modeling of Human Behavior: Why a Model is (Sometimes) Better than 10,000 Words. *Cognitive Systems Research*, *8*(3), 135-142.

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38-43.

Howland, K., Good, J., & Nicholson, K. (2009). Language-Based Support for Computational Thinking. *Proceedings of IEEE Symposium on VL/HCC 2009* (pp. 147-150).

Miwa, K., Morita, J., Nakaike, R., & Terai, H. (2014). Learning through Intermediate Problems in Creating Cognitive Models. *Interactive Learning Environments*, *22*(3), 326-350.

Nakaike, R., Miwa, K., Morita J., & Terai, H. (2009). Development and evaluation of a web-based production system for learning anywhere. *Proceedings of ICCE2009* (pp. 127-131).

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1).