Learning by Building Cognitive Models that Reflect Cognitive Information Processing: A Preliminary Class Exercise

Kazuhisa Miwa

Graduate School of Information Science
Nagoya University
Nagoya, Japan 464-8601
Email: miwa@is.nagoya-ua.jp

Hitoshi Terai

Faculty of Human-oriented Science
Kindai University
IIzuka, Japan 820-8555
Email: terai@fuk.kindai.ac.jp

***Abstract*—It has been confirmed that it is important for students to monitor their own cognitive activities during learning. We challenge this idea based on our "learning by creating cognitive models" paradigm. Within a learning environment developed for cognitive modeling, we let students construct cognitive models that reflect their own cognitive information processing. A cryptarithmetic task was used. We conducted a cognitive science class in which participants were required to build computational models that behaved in a manner similar to how the students themselves behaved. As a result, 9 of the 22 (41%) models accurately reflected the participants' problem-solving paths.**

Keywords - Cognitive Models; Procedural Knowledge; Production System

I. OBJECTIVES

The model-based approach is a primary methodology in cognitive science. Cognitive scientists have used computational models as research tools to understand the human mind [1]. The authors have examined the functions of cognitive modeling as a learning tool and proposed the "learning by creating cognitive models" [2][3].

Previous studies have confirmed that creating cognitive models improves active construction of rule-based mental models [4]. Through such activities, participants learn to cultivate meaningful insights into the kinds of procedural knowledge that underlie the observed behaviors of problem solvers.

It appears important for students to monitor their own cognitive activities while learning [5][6][7]. This monitoring activity may correct learners' incorrect knowledge, and improve their learning processes. However, it is generally not easy for naive students to correctly understand their own mental activities. This effort relates to meta cognitive activities. Previous research indicated that performing such a meta cognitive activity is difficult for naive learners, and should be trained.

We investigated this issue based on our "learning by creating cognitive models" paradigm. Fig. 1 shows a diagram of our approach. It is difficult for students to grasp their inner mental activities, but easy to understand their problem solving behaviors that can be observed in the external world. In our approach, students first assume a set of procedural knowledge used for problem solving, and externalize the set as a cognitive model. The model runs as a computer program, and derives the behavior deducted from the assumed knowledge. The model functions as a hypothesis-deduction machine. The participants detected differences between derived behaviors deducted by the hypothesized knowledge and how their own problem solving actually behaved. These detected differences revealed significant information about their own internal thought processes. This assume-execute-observe cycle thus improved students' understandings of their own cognitive processing.

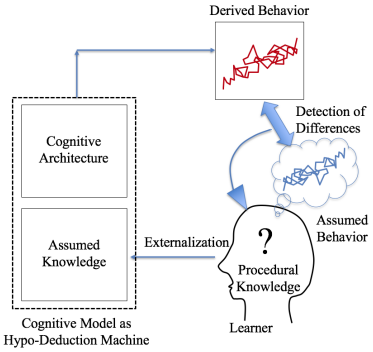

Figure 1. Conceptual diagram of our approach.

Our research question in this study is whether students can construct cognitive models that reflect their own cognitive information processing. We conducted a cognitive science class in which participants were required to build computational models that behaved in a manner similar to the students themselves. This paper provides a preliminary analysis of the model and its findings.

In Section 2, we explain an arithmetic task used in the current study. In Sections 3 and 4, we introduce our learning system, and explain an overall structure of the class practice. Finally, in Section 5, we report results of the class practice. Section 6 summarizes our conclusions.

II. TASK

We used a cryptarithmetic task [8][9] in our study. The following is an example problem used in our class practice.

 ISBN: 978-1-61208-531-9

```
 IGEAF     F = 6
+DBJAD
------
CIHEGH
```

This problem is simple; however, the cognitive information processing for its solution is relatively complex. In fact, multiple types of procedural knowledge are used during the solution processes. The following discussion describes some examples.

- **Numeral processing**: If a column is x + y = z, and both x and y are known, then we can infer z by summing x and y. For example, in the rightmost column, when we know F = 6 and D = 9, we can assign H a value of 5.
- **Specific numeral processing**: If a column is x + y = x, then we can infer that y = 0 or 9. For example, in the fifth column, we can obtain D = 0 or 9 independently without any other information. In this case, we can determine that D = 9 because C should be 1, meaning that a carry is sent to the left-side column.
- **Parity processing**: If a column is x + x = y and we have a carry from the right column, then we can infer that y is an odd number. For example, in the second column, we obtained a carry by the inference from the first (i.e., rightmost) column; therefore, we conclude that G is an odd number.
- **Inequality processing**: If a column is x + x = y, and no carry is sent to the left column, then we can infer that x is less than 5. For example, in the second column, when we know there is no carry to the left column; A is less than 5.

University students easily understand such procedural knowledge sets if they are given; however, they may face challenges finding the knowledge by themselves and externalizing it while working on the problem.

III. The Learning System

We have developed a learning environment that enables students to construct rule-based cognitive models. The system consists of two modules, i.e., a knowledge editor and a problem-solving simulator.

A. Knowledge editor

First, students externalize a set of procedural knowledge (such as describing rules to solve cryptarithmetic tasks) using the knowledge editor.

Fig. 2 shows an example screenshot of the knowledge editor in which the rule of inequality processing is described, i.e., if a column is x + y = z and no carry is sent to the left column (b == 0 in the figure), then we can infer that z is greater than x.

B. Problem-solving simulator

The problem-solving simulator is mounted on the learning system. The problem solver that simulates the behavior has the potential to perform an exhaustive search for the assignment of digits to letters. Specifically, it selects one of the letters whose numeral value has not been determined and systematically assigns each digit to a letter. If a contradiction is found in the inference process, another assignment is tested. If the problem solver has no procedural knowledge, it is impossible to derive the solution because the problem space spreads exhaustively. Students must give the problem solver adequate procedural knowledge through the knowledge editor.

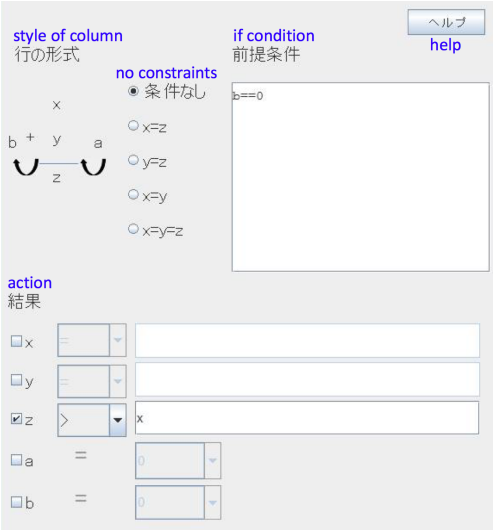

Figure 2. Example screenshots of Knowledge Editor of the learning system.

Fig. 3 shows an example screenshot of the problem-solving simulator, which presents a problem status (the assignment status of digits to letters) and an inference status (a step-by-step series for information processing). A list of rules installed for the problem solver is presented on the right-hand side of the window. Rules that can fire at a specific problem-solving step are marked by bold red lines. In this case, three rules are available. The conflict resolution mechanism is simple, and the most specific rule that provides the most specific inference result has priority for firing. Students can test any rule by forcibly firing it and confirming the resulting inferences. Moreover, students can modify the model very easily. For example, if we uncheck items in the list, students can simulate the behavior of the problem solver with that knowledge excluded.

The system also presents the problem solver's behavior, represented as a search tree of problem-solving processes. Students can confirm inference steps one at a time by clicking the inference button to apply the inference. At any point in the problem-solving process, students can install, delete, or revise knowledge using the editor and restart the inference from a given point in their problem-solving.

IV. Class Practice

Class practice was performed as part of a cognitive science class at the first author's university. Participants included 25 undergraduates from Nagoya University. In the initial week, the participants spent one hour learning how to manage the knowledge editor and operate the problem-solving simulator. Specifically, participants were given an example problem: MEST + BADE = MASER. They then installed seven pieces of procedural knowledge to solve the given problem with a tutor's guidance, and they then simulated behavior at each stage of the construction process.

 ISBN: 978-1-61208-531-9

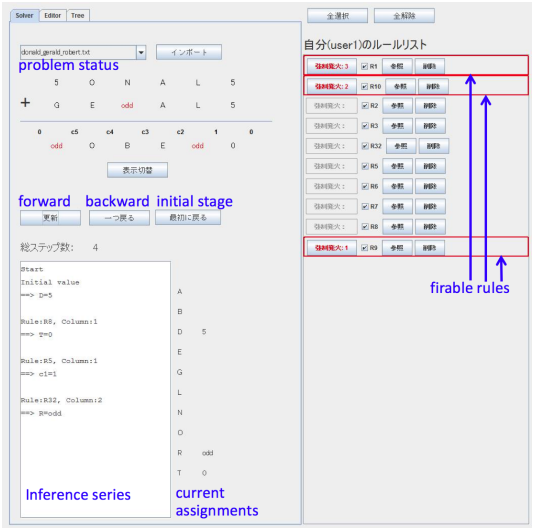

Figure 3. Example screenshots of Problem-solving simulator of the learning system.

In the second week, in a 70-minute training phase, the participants were given a training problem: DONALD + GERALD = ROBERT. They were then required to find a procedural knowledge set for the solution independently, install it in the problem solver with the knowledge editor, and then construct a model. In the third week, the participants were given the target problem: IGEAF+DBJAD = CIHEGH. They were required to construct a model for its solution. After model construction, they were required to solve the same problem by hand while writing their solution processes on an experimentation sheet. Both the model and participant solution processes were analyzed.

V. RESULTS

One participant could not construct a complete model that reached the solution within 100 problem-solving steps. Two other participants' problem-solving paths were not clearly identified due to insufficient written descriptions on the examination sheets. We excluded these three participants from our analysis.

The average number of problem-solving steps for the other 22 participants was 20.9 steps. Fig. 4 shows the detailed result of the models' performances. The horizontal axis indicates problem-solving steps, and the vertical axis indicates the number of participants who constructed the model that reached the solution using that step.

We analyzed the participants' problem-solving paths. All participants initially processed the fifth column (I + D = I) and derived the decisive information D = 9. Then, the participants processed the other columns, coordinated multiple pieces of information obtained through the preceding problem-solving processes, and focused on a specific letter with a limited range of possible numerical values for the following trial-and-error search. Specifically, for the example in Fig. 5, first, G = odd was determined by processing the second column in which the same letters (A and A) were summed and a carry was received from the right-side column. Then, based on the information that G = odd, a limited range of possible assignments (i.e., G = 3 or 7) was obtained because other odd numbers (1, 5, and 9) had already been assigned to other letters (C, H, and D, respectively). Next, the participant began to examine G = 3 in a trial-and-error search.

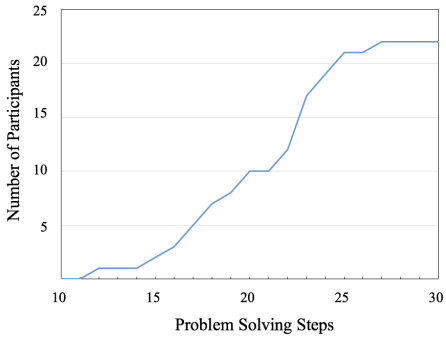

Figure 4. Number of participants who constructed successful models for problem solving.

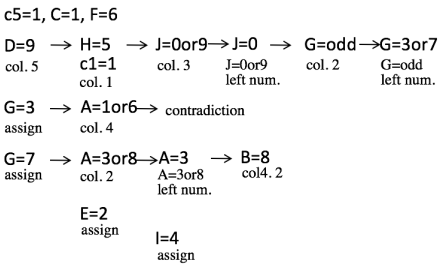

Figure 5. Human problem solving behavior; example case of similar processes.

Fig. 6 shows the problem-solving path of this participant's model. The path is similar to that of the participant shown in Fig. 5. Initially, the model drew G = 3 or 7, found that G = 3 was impossible, and reached the solution by examining another assignment, G = 7.

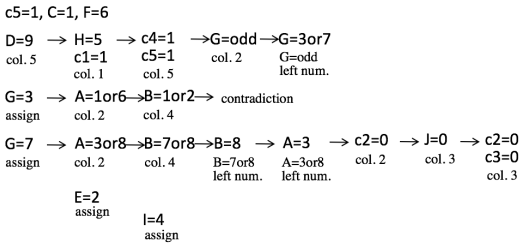

Figure 6. Model problem solving behavior; example case of similar processes.

We focused on the overall patterns of the problem-solving paths determined by trial-and-error search driven by an examined letter, such as G in Fig. 5 and Fig. 6. Table I shows the result of analysis. In 9 of the 22 cases, the patterns of the participants' behaviors were similar to those of the models; however, the 13 other cases were not similar.

Fig. 7 and Fig. 8 show an example case in which the

 ISBN: 978-1-61208-531-9

TABLE I. RESULTS OF REVIEW TEST

IGEAF+DBJAD		DONALD+GERALD	
Match	9 (.41)	Solved	7 (.78)
		Unsolved	2 (.22)
UnMatch	13 (.59)	Solved	3 (.23)
		Unsolved	10 (.77)

participant and model behaviors did not match. In Fig. 7, the participant inferred that A = 2, 3, or 4 by combining A < 5, which had been determined by processing the second column with the information that no carry was sent to the left-side column and the information that 1 was already assigned to C. Based on this information, the participant examined each assignment to the letter A. Fig. 8 shows the problem-solving path through which the model that the participant constructed had run. The model initially inferred that G = 3 or 7, guiding a subsequent trial-and-error search that differed from the participant's path. The model did not infer information related to the letter A and did not focus on the letter A for the initial trial-and-error search, thereby changing the problem-solving path.

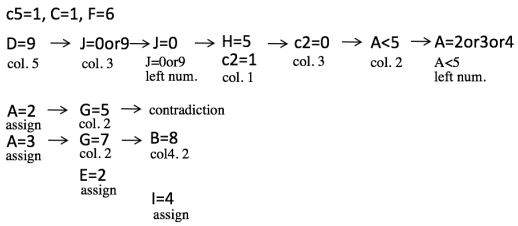

Figure 7. Human problem solving behavior; example case of different processes.

We also investigated the general capacities of the models. The participants intended to construct the models to solve the target problem: IGEAF+DBJAD=CIHEGH. However, if the models were constructed by general rules for problem solving, those could also solve other problems. We examined if each model was able to solve the training problem: DONALD + GERALD = ROBERT. Table I shows that models that successfully trace the participants' problem solving paths are more likely to solve the training problem.

VI. CONCLUSIONS

We analyzed 22 participants who successfully constructed sophisticated models that could solve the given task in approximately 21 steps. However, only 9 of the 22 (41%) models traced the participants' problem-solving paths. This implies that it was relatively difficult for the participants to construct a model that reflected their own cognitive information processing. First, this problem comes from the participants' programming abilities. Some participants appeared unable to implement appropriate rules even though they noticed their own procedural knowledge. This implies that our next step is to improve the learning environment developed in the current study. Another reason is that model construction that reflects each participant's cognitive processing was not emphasized in this class practice. Some participants attempted to construct high-performance models that solved the task as quickly as possible or general models that could perform a variety of tasks. We believe this can be improved based on instructor suggestion.

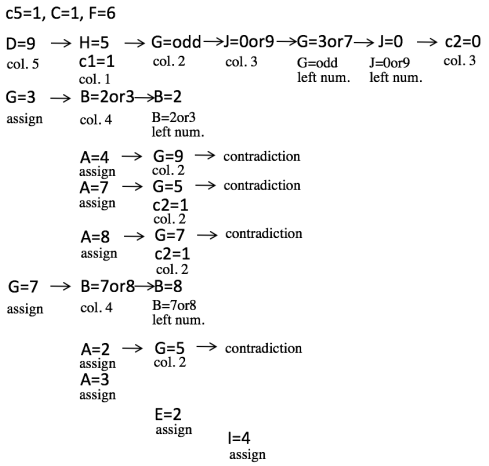

Figure 8. Model problem solving behavior; example case of different processes.

ACKNOWLEDGMENTS

This research was partially supported by HAYAO NAKAYAMA Foundation for Science & Technology and Culture, and JSPS KAKENHI Grant Numbers 15H02927, 15H02717.

REFERENCES

[1] D. Fum, F. D. Missier, and A. Stocco, "The cognitive modeling of human behavior: Why a model is (sometimes) better than 10,000 words." Cognitive Systems Research, vol. 8, 2007, pp. 135–142.

[2] K. Miwa, R. Nakaike, J. Morita, and H. Terai, "Development of production system for anywhere and class practice." in Proceedings of the 14th International Conference of Artificial Intelligence in Education, 2009, pp. 91–99.

[3] K. Miwa, J. Morita, R. Nakaike, and H. Terai, "Learning through intermediate problems in creating cognitive models." Interactive Learning Environments, vol. 22, 2014, pp. 326–350.

[4] K. Miwa, N. Kanzaki, H. Terai, K. Kojima, R. Nakaike, J. Morita, and H. Saito, "Learning mental models on human cognitive processing by creating cognitive models." Lecture Notes in Computer Science (AIED 2015), vol. 9112, 2015, pp. 287–296.

[5] C. Conati and K. Vanlehn, "Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation," International Journal of Artificial Intelligence in Education (IJAIED), vol. 11, 2000, pp. 389–415.

[6] R. Azevedo and A. F. Hadwin, "Scaffolding self-regulated learning and metacognition–implications for the design of computer-based scaffolds," Instructional Science, vol. 33, no. 5, 2005, pp. 367–379.

[7] R. K. Atkinson, A. Renkl, and M. M. Merrill, "Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps." Journal of Educational Psychology, vol. 95, no. 4, 2003, p. 774.

[8] A. Newell and H. A. Simon, Human problem solving. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[9] K. Miwa, "A cognitive simulator for learning the nature of human problem solving." Journal of Japanese Society for Artificial Intelligence, vol. 23, 2008, pp. 374–383.

 ISBN: 978-1-61208-531-9