A learning environment for externalizing procedural knowledge in problem solving:

A preliminary trial for tutoring problem posing skills

Kazuhisa Miwa[a*], Kazuya Shibayama[a] & Hitoshi Terai[b]
[a]*Graduate school of information science, Nagooya University, Japan*
[b]*Faculty of Humanity-Oriented Science and Engineering, Kinki University, Japan*
*miwa@is.nagoya-u.ac.jp

Abstract: Problem posing is a challenging task for naive students. For problem posing, students must identify knowledge and procedures for solving problems. We suppose that understanding of procedural knowledge for problem solving enhances problem-posing activities. We developed a learning environment wherein students find and formally describe various types of procedural knowledge while problem-solving. The system comprises two modules: the knowledge editor and the problem-solving simulator. Students externalize a procedural knowledge set for solving crypt-arithmetic tasks using the knowledge editor. With the problem-solving simulator, they also simulate the behavior of the model comprising the knowledge set. The practice in class confirmed that two-thirds of the participants constructed reasonable models with our system. They appeared to succeed in identifying and externalizing procedural knowledge for solving a relatively complex arithmetic task.

Keywords: Problem posing, Cognitive model, Procedural knowledge, Crypt-arithmetic task.

1. Introduction

Problem posing is a crucial educational method for facilitating the students understanding on the nature and structure of problems. However, students also encounter challenges while posing problems. Many studies have reported that when students are required to pose problems, they merely replicate a familiar example problem, not employing effective styles of problem posing (Kojima, Miwa, & Matsui, 2015). To enhance students' problem-posing activities, we must investigate the processes that underlie problem-posing activities.

Initially, we have to understand how to solve problems for posing problems. In other words, we need to identify the knowledge and mental procedures required to solve problems. For example, when we pose an arithmetic problem, we must possess the procedural knowledge required to solve that problem. One effective strategy for doing so is to focus and monitor our own problem-solving processes to understand how we, by ourselves, solve the problem. Such cognitive capability is known as meta-cognitive skills. Numerous studies have reported that meta-cognitive activities, such as self-explanation, improve students' learning processes and create positive learning effects (Chi et al., 1989; Aleven & Koedinger, 2002). These findings imply that meta-cognitive activities also provide beneficial advantages in problem posing. However, it is also challenging for naive students to engage in meta-cognitive activities because such activities' involve significant cognitive load.

The authors have developed the "learning by building cognitive models paradigm," wherein students construct computational cognitive models that solve cognitive tasks (Miwa et al., 2014a). We have confirmed three advantages of this learning paradigm: (1) theory-based thinking in which students learn to interpret and explain experimental results based on a theory (Saito et al., 2013); (2) mental simulations in which students learn to predict experimental results by performing mental simulations (Miwa et al., 2014b); and (3) externalization of cognitive processing in which students learn to identify the procedural knowledge required to perform a cognitive task (Miwa et al., 2015).

The final feature in the aforementioned advantages, drawn from learning by building cognitive models, may generate beneficial resources in teaching problem posing. Our preceding study included two class practices for undergraduates and graduates: Participants were required to construct a computational running model for solving subtraction problems and then develop a bug model that simulated students' arithmetic errors. Analyses indicated that by creating cognitive models, participants learned to identify buggy procedures that produce systematic errors and to predict expected erroneous answers by mentally simulating the mental model. Such learning skills for identifying procedural knowledge should provide students with the significant foundation for acquiring capabilities in sophisticated problem posing.

We believe that acquisition of this type of skill causes acquisition of problem-posing abilities. Our strategy is to promote students to learn skills that enable them to identify procedural knowledge necessary for problem-solving. Thus, we present a preliminary trial for developing a learning system for teaching problem-posing skills. We report here the construction of a learning environment and a preliminary evaluation performed with class practice that examines the extent to which participants could accurately externalize relatively complex procedural knowledge used in solving a crypt-arithmetic task.

2. Task

The task used in our study is a crypt-arithmetic task. In this study, we propose an environment wherein students learn to understand their procedural knowledge to perform the task while building a computational model. The following is an example problem:

```
 DONALD
+GERALD
 ROBERT  D=5 is given
```

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

- Numeral processing
  If a column is x + y = z, and both x and y are known, then infer z by adding x and y. For example, in the rightmost column, we know D equals 5; therefore, 0 is assigned to letter T by applying this procedure.
- Specific numeral processing
  If a column is x + y = x, then infer that y equals 0 or 9. For example, in the fifth column, we obtain that E equals 0 or 9 independently, without any other information.
- Parity processing
  If a column is x + x = y, and we have a carry from the right column, then infer that y is an odd numeral. For example, in the second column, we obtained a carry by the inference in the first (i.e., right) column; therefore, we conclude that R is an odd numeral.
- Inequality processing
  If a column is x + y = z, and no carry is sent to the left column, then infer that z is greater than x (or y). For example, in the sixth column, we know that D equals 5, and no carry is sent to the left column; therefore, R is greater 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 solving the problem.

3. Learning System

We developed a learning environment to enable students to find and formally describe various types of procedural knowledge while solving problems. The system comprises two modules: the knowledge editor and the problem-solving simulator.

3. 1 Knowledge editor

Students externalize a set of procedural knowledge for solving crypt-arithmetic tasks with the knowledge editor.

Figure 1 demonstrates an example screenshot of the knowledge editor wherein the procedural knowledge of inequality processing such as, "If a column is x + y = z, and no carry is sent to the left column (b==0 in the figure), then infer that z is greater than x," is described.

A list of procedural knowledge installed for the problem solver is presented at the right side of the window. If we delete the check from each item of the list, we can simulate the behavior of the problem solver from whom the knowledge is excluded.

Figure 1: Knowledge editor for helping students externalize
a procedural knowledge set for solving crypt-arithmetic tasks

3. 2 Problem solving simulator

The problem-solving simulator is mounted on the learning system. The problem solver that simulates behavior has the potential to perform an exhaustive search for assignments of digits to letters. Specifically, it selects one of the letters that has not been determined and systematically assigns each digit to a letter. If a contradiction is found in the process of inference, 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 are required to give the problem solver adequate procedural knowledge with the knowledge editor.

(a) Problem-solving simulator (b) Example search tree

Figure 2: The problem-solving simulator, which simulates behavior,
can perform an exhaustive search for assignments of digits to letters.

Figure 2 (a) indicates an example screenshot of the problem-solving simulator, which presents the assignment status of digits to letters (left side) and further presents a series for information processing step by step (right window). The system also presents the problem solver's behavior, represented as a search tree of problem-solving processes (see Figure 2 (b)). Students can confirm inference steps one by one, forwarding the inference by clicking the inference button. At any point of the problem-solving process, students can install, delete, or revise knowledge using the editor and restart the inference from the problem-solving point.

The system can simulate a variety of problem-solving processes (Newell & Simon, 1972; Miwa, 2008). For example, the complete problem solver arrives at the solution within approximately 21 to 42 steps. However, if the specific numeral processing, such as: "If a column is x + y = x (see the fifth column), then infer that y equals 0 or 9," is excluded from the knowledge set, and the problem solver requires more than 100 steps for a solution using the trial-and-error method.

4. Preliminary evaluation

4. 1 Participants and Procedures

Participants in the practice included 45 undergraduates of Nagoya University. In the first class, they learned how to manage the knowledge editor and operate the problem-solving simulator. Specifically, participants were given an example problem: MEST + BADE = MASER; they installed seven pieces of procedural knowledge for solving the given problem with tutor's guidance, and they simulated behavior at each stage of the construction process.

In the second class (1 week after the first class), participants were given a problem: DONALD + GERALD = ROBERT, and they, by themselves, were required to find a procedural knowledge set for the solution, install it into the problem solver with the knowledge editor, and construct a model—these processes were to be completed within 70 minutes.

4. 2 Result

The following was a representative model construction process. During the simulation process, participants encountered a crucial stage of problem solving and hypothesized a part of procedural knowledge required for processing the specific stage of problem solving. They tried to provide the problem solver with the procedural knowledge, but usually, they initially failed in the installation. They noticed the failure by forwarding the problem solving one step and confirming that the expected result was not obtained. Through the trial-and-error processes, once they accurately installed the knowledge set and passed through the crucial problem-solving stage, then they forwarded the inference process and faced another specific stage of problem solving. They again tried to identify knowledge for the stage.

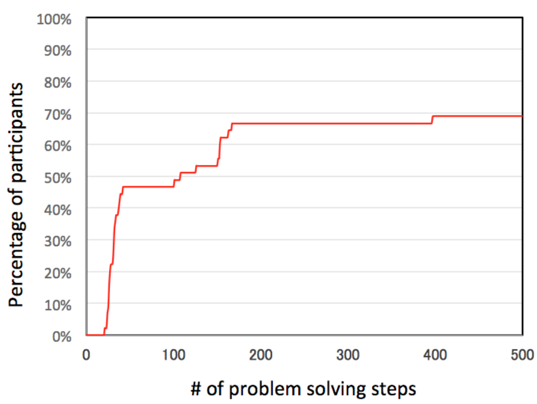

Figure 3: Percentage of undergraduate participants who constructed successful models for problem solving.

Our primary concern is to what rate and to what degree the participants accurately identified the procedural knowledge set for problem solving and successfully had the problem solver achieve the solution. Figure 3 indicates the class practice's results: The horizontal axis indicates problem-solving steps, and the vertical axis indicates the rate of participants who constructed the model that reached the solution by the problem-solving steps indicated in the horizontal axis.

Figure 3 indicates that 46.7% of participants constructed models that solved the problem within 42 steps. The second group comprised 20% of participants who solved it within 167 steps. The other 34.3% of participants failed to successfully construct the model.

5. Conclusions

We developed a learning environment to enable students to find and formally describe various types of procedural knowledge applied when solving crypt-arithmetic tasks. Our class practice confirmed that two-thirds of participants constructed reasonable models with our system. They appeared to succeed in identifying and externalizing procedural knowledge for solving such a relatively complex arithmetic task.

Our report here is limited to the first half of the project. The next crucial step is to examine whether developing such externalization of procedural knowledge actually enhances problem-posing activities. Our preceding study confirmed that following generative steps of problem posing positively impacts students' ability to pose a variety of problems (Kojima, Miwa, & Matsui, 2013). Similarly, we expect to improve problem posing through the development of such monitoring activities for mental processing.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers 25560110, 15H02927.

References

Aleven, V. & Koedinger, K.R. (2002). An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor, Cognitive Science, 26, 147-179.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems, Cognitive Science, 13, 145-182.

Kojima, K., Miwa, K., & Matsui, T. (2015). Experimental Study of Learning Support through Examples in Mathematical Problem Posing. Research and Practice in Technology Enhanced Learning. 10:1 doi:10.1007/s41039-015-0001-5

Kojima, K., Miwa, K., & Matsui, T. (2013). Supporting Mathematical Problem Posing with a System for Learning Generation Processes through Examples. International Journal of Artificial Intelligence in Education, 22, 161-190.

Miwa, K. (2008). A cognitive simulator for learning the nature of human problem solving. Journal of Japanese Society for Artificial Intelligence, 23, 374-383.

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

Miwa, K., Morita, J., Terai, H., Kanzaki, N., Kojima, K., Nakaike, R., & Saito, H. (2014b). Use of a Cognitive Simulator to Enhance Students' Mental Simulation Activities. LNCS, 8474 (ITS 2014), 398-403.

Miwa, k., Kanzaki, N., Terai, H., Kojima, K., Nakaike, R., Morita, J., and Saito, H. (2015). Learning mental models on human cognitive processing by creating cognitive models. LNCS, 9112 (AIED 2015), 287-296.

Newell, A. & Simon, H. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice-Hall.

Saito, H., Miwa, K., Kanzaki, N., Terai, H., Kojima, K., Nakaike, R., & Morita, J. (2013). Educational Practice for Interpretation of Experimental Data Based on a Theory. Proceedings of the 21th International Conference on Computers in Education (ICCE 2013). 234-239.