Hayashi, Y., et al. (Eds.) (2017). Workshop Proceedings of the 25th International Conference on Computers in Education. New Zealand: Asia-Pacific Society for Computers in Education

Preliminary Study on Learning by Constructing a Cognitive Model Based on Problem-Solving Processes

Kazuaki KOJIMA[a*], Kazuhisa MIWA[b], Ryuichi NAKAIKE[c], Nana KANZAKI[d], Hitoshi TERAI[e], Jun'ya MORITA[f], Hitomi SAITO[g], & Miki MATSUMURO[b]

[a]*Learning Technology Laboratory, Teikyo University, Japan*
[b]*Graduate School of Information Science, Nagoya University, Japan*
[c]*Department of International Tourism, Heian Jogakuin Universiry, Japan*
[d]*College of Nagoya Women's University, Japan*
[e]*Faculty of Humanity-Oriented Science and Engineering, Kindai University, Japan*
[f]*Faculty of Informatics, Shizuoka University, Japan*
[g]*Faculty of Education, Aichi University of Education, Japan*
*kojima@lt-lab.teikyo-u.ac.jp

Abstract: Construction of models is promising as a learning activity, and computational environments are useful for that. However, it can be a heavy task for novice learners to construct computational models because it requires considerable instruction and practice of programming languages. We designed a basic framework for learning by experiencing construction of models on a production system in the domain of cognitive science. In this framework, a model abstractly describing human problem-solving processes and its computer model implemented on the production system is prepared by an instructor in advance. A learner is given the abstract model and processes of problem solving produced by executing the implementation model, and then engaged in instantiating the abstract model into an implementation model. This activity is expected to deepen learner understanding of mental processes embedded in the abstract model. We preliminary studied the effect of learning a model which simulates subtraction requiring regrouping in the framework. The results confirm the possibility that such experience can improve learner understanding of mental processes behind the model, and necessity to expand learning activities in the framework.

Keywords: Learning by construction, cognitive model, production system, problem solving

1. Introduction

Science in recent decades has used two approaches to understand the natures of targets: an analytical approach through observation of targets, and a constructive approach through construction and simulation of target models. For example, cognitive science research adopted empirical studies of human behaviors and running computational models in understanding human mind (Schunn, Crowley and Okada, 1998).

Models are essential to the production, dissemination, and acceptance of scientific knowledge (Gilbert, 2004). As well as science research, science education uses models to have learners interpret scientific knowledge. Besides the model use, construction and simulation of models by learners has also been argued (Clement, 2000; Gilbert, 2004; Harrison, and Treagust, 1998). Model construction is promising as a learning activity in understanding complex or invisible targets, and computational environments are useful both for researchers and learners because they enable to instantly construct, test, and evaluate models. However, it can be a heavy task for novice learners to construct computational models because it requires considerable instruction and practice of programming languages (Penner, 2000). Therefore, several studies addressed support for model construction by learners (e.g., Basu, Dukeman, Kinnebrew, Biswas and Sengupta, 2014; Brady, Holbert, Soylu, Novak, and Wilensky, 2015; Hirashima, Imai, Horiguchi and Toumoto, 2009). Support by the studies allow learners to construct and simulate models by designing models abstractly describing the attributes or behaviors of targets. Instantiation of the models into computer-executable models is left to support systems. Here, the former models of abstract description of targets are referred to as *abstract models*, and the latter as *implementation models*. These studies successfully alerted misconceptions, produced conceptual changes, and deepened understanding in scientific phenomena through designing abstract models.

Models on which computer simulations are based correspond to both instructionally designed models and interfaces to guide learner model construction (Seel, and Blumschein, 2009). Thus, the support systems described above may be limited to targets which can be represented as models of interaction among agents and objects. Mental processes in problem solving by a person, for example, could not be properly modelled on an interface to arrange agents and objects. Therefore, learning of human mental processes with computational models must require a different approach.

We designed a basic framework for learning by model construction in the domain of cognitive science (Kojima, Miwa, Nakaike, Kanzaki, Terai, Morita, Saito, and Matsumuro, 2016). In this framework, a learner instantiates an abstract model initially given into an implementation model. Basically, abstract models are critical in learning by construction because they are externalized products in understanding of targets. On the other hand, implementation of models also plays a critical role in deepening understanding as demonstrated in history of cognitive science. One of the central keys in learning by construction is to receive feedback from actual or virtual worlds through instantiation of abstract models into implementation models (Nakashima, 2008). Our framework is intended to provide opportunities for learners to gain such benefits through model construction with lower load. In this paper, we reported a preliminary study to confirm the effect of experience construction of a cognitive model.

2. Support System for Learning by Constructing Cognitive Models

Figure 1 illustrates the framework for learning of human problem-solving processes by constructing cognitive models. In this framework, a learner is given an abstract model of problem solving, and processes of the problem solving produced by executing its implementation model. He or she is then engaged in instantiating it into the implementation model by himself/herself according to the processes. This activity allows to experience construction of a cognitive model with lower load, and is expected to deepen learner understanding of mental processes embedded in the abstract model (e.g., sophisticating a mental model of learners about a phenomenon the abstract model represents).

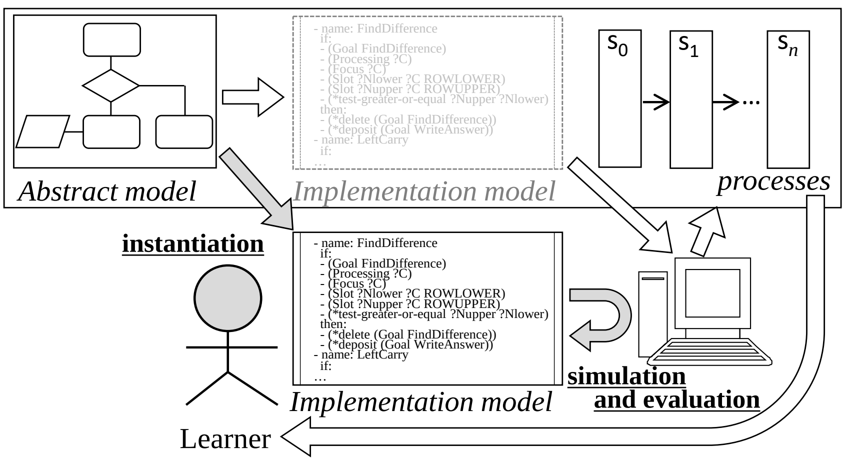

Figure 1. Framework for learning by constructing cognitive models

We implemented a support system for the framework, which adopts a production system as an architecture of implementation models. Actually, it uses DoCoPro (Nakaike, Miwa, Morita and Terai, 2009), a production system designed for learning by constructing models by novice learners. Before the system is given to learners, an instructor implements a cognitive model for an abstract model of human problem-solving on DoCoPro. The system executes the model and extracts its problem-solving processes. It then creates information indicating steps involved in the processes. This information includes explanation of a production rule fired and two states in the working memory before/after the rule firing for each step of the processes.

Figure 2 shows a screenshot of the support system. As the left side of the figure indicates, the system provides information of each step in the problem-solving processes. For every step of the processes, the learner composes a production rule which can change the before-state to the after-state with the editor of the right side. The learner can check his/her rule on each step through comparison between the after-state and the result from firing the rule. Construction of the implementation model is completed through composition of rules for all steps. Although learners cannot experience design of problem representation in this framework, it enables the learners who are not familiar with programming to experience instantiation of an abstract model and receiving of feedback from the instantiation.

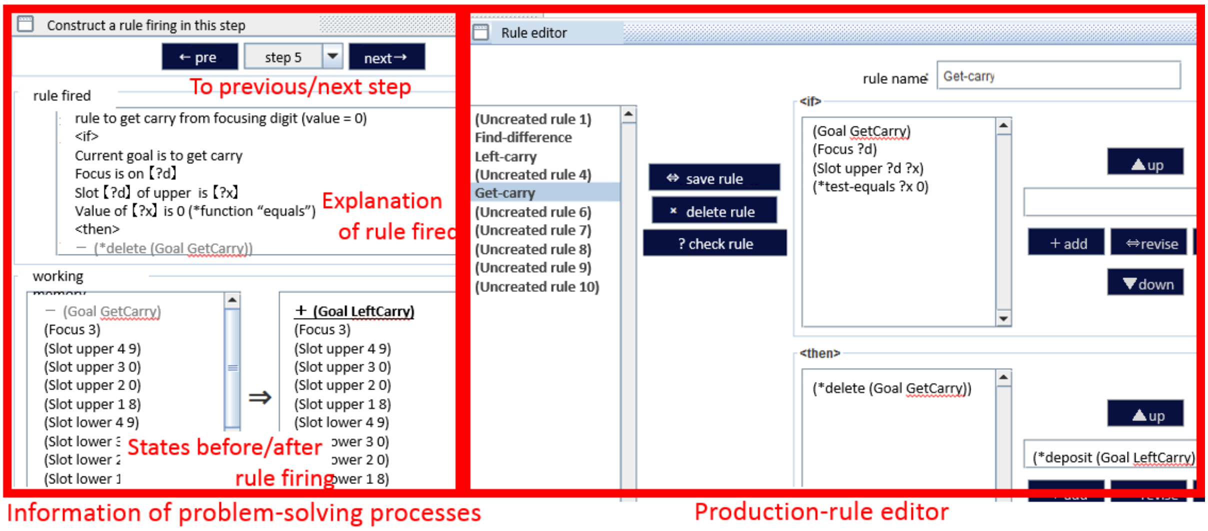

Figure 2. Screenshot of support system

3. Preliminary Study of the Effect of Learning by Constructing a Cognitive Model

We empirically studied whether experience of model construction with the support system had the learning effect. We used a model of subtraction requiring regrouping, which was used in a practice of our previous study (Kanzaki, Miwa, Terai, Kojima, Nakaike, Morita, and Saito, 2015). Everyone can easily solve problems of subtraction, but do it implicitly with procedural knowledge. Such problem solving is suitable as a learning target because construction of its model requires deep understanding of implicit mental processes automatically performed.

In the practice of previous study, undergraduates trained model construction on DoCoPro in a 90-minutes class, which was followed by three classes where they constructed a subtraction model and a bug model producing incorrect answers because of bugs in rules. This model construction was supported by a function visually representing states in the working memory.

3.1. Method

Eight undergraduates who had not experienced in training computer programming participated in this study. Prior to the study, they learned model construction on DoCoPro with instructional contents used in the previous study. In this study, they first responded to a pretest. This test asked the participants to solve a subtraction problem 317 – 98," describe general procedures to perform subtraction, and infer what made incorrect answers to two problems "9008 – 3149 = 5959" and "806303 – 182465 = 623938." These answers occurred because the solver merely changed 0 into 9 when digits to borrow a number were 0. The first subtraction problem was not intended to test the participants, but to bring procedures of regrouping to their attention before describing subtraction procedures.

Second, the participants learned procedures to compose a model according to processes given from the support system with instructional video. They then were given two sheet of paper which described an abstract model of subtraction in a state-transition diagram, and explanation of predicates used in implementing a model. After the instructions, they actually instantiated the abstract model into an implementation model.

Finally, the participants responded to a posttest including the tasks to describe subtraction procedures and infer bugs in the two problems, which were identical to those of the pretest. They were then asked to report what they had learned in the instantiation of the model.

In the analysis of the subtraction-procedures task, we checked whether participants' descriptions included information corresponding to ten rules comprising the implementation model. For each rule, participants' descriptions were categorized into *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 information of the ten rules was as follows.

FindDifference1 If the minuend is equal to or greater than the subtrahend in the digit to perform subtraction (*processing digit*) (then move to WriteAnswer)

WriteAnswer Write the difference between the minuend and subtrahend

ShiftColumn Shift the processing digit to the left column

Completed Finish when the difference in the far-left column is written

FindDifference2 If the minuend is smaller than the subtrahend in the processing digit (then move to **LeftCarry**)

LeftCarry Shift the digit to borrow 1 (*focus digit*) to the left column

GetCarry1 If the minuend in the focus digit is not zero, then subtract 1 from it and shift the focus digit to the right

PutCarry1 Add 10 to the focus digit (and then move to FindDifference)

GetCarry2 If the minuend in the focus digit is zero (then move to LeftCarry)

PutCarry2 Add 10 to the focus digit (and then move to GetCarry)

For example of PutCarry2, descriptions such as "add 10 to the one's place" and "add 10 to the right digit" were categorized incomplete because some subtraction problems are not correctly solved with these operations.

Because the bug inference task was used in the previous study mentioned above, we scored the participants' responses in the same way: two points if the bug causing the incorrect answers to the two problems was appropriately described, one point if factors causing the two incorrect answers were described with a single consistent rule but not appropriate, and zero point if only the phenomena were described or factors were described with two different rules.

3.2. *Results*

All of the participants successfully instantiated the implementation model on the support system. The average time it took them to finish the tests and model construction was about 60 minutes.

The participants successfully solved the subtraction problem in the pre-test. Figure 3 indicates the categories for each rule in the subtraction procedures task in the pre- and posttests. The participants' descriptions in the pretest included much incomplete information or no information about the lower five rules. These rules are corresponding to procedures to borrow a number in regrouping. Descriptions including such information increased, on the other hand, in the posttest.

The average score of the bug inference task was 0.88 in the pre-test, and 1.13 in the post-test. Actually, only two of the eight participants scored higher in the post-test than in the pre-test.

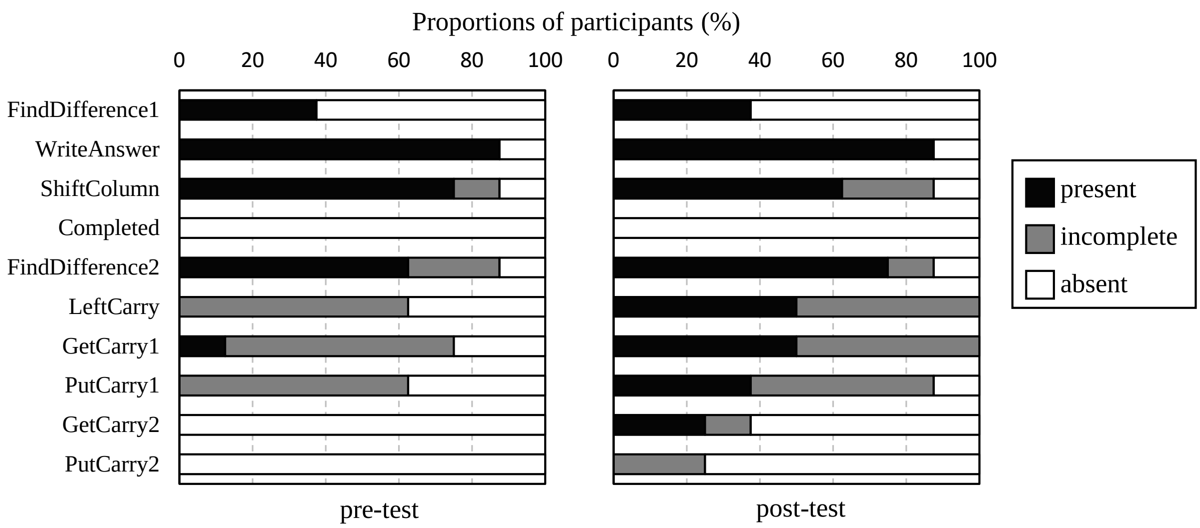

Figure 3. Categories for each rule in the subtraction procedures task

3.3. *Discussion*

Figure 3 revealed that the participants' descriptions about subtraction procedures were improved through experience of instantiation into the implementation model on the support system. In the pretest, their descriptions omitted much information about regrouping procedures. Although they could easily perform subtraction procedures, they could not exactly explain them. The information was expanded in the posttest. In the posttest, five out of the eight participants reported findings about their implicit, automatized mental processes, such as "the process was complex than I had expected, although I perform subtraction in everyday life" and "I found I usually omit some steps when I explain procedures of subtraction to someone." Those facts confirm the possibility that support system improved their understanding of mental processes behind the model they learned. On the other hand, information about some procedures did not change such as FindDifference 1 and Completed. They are conditions to perform subtraction in a column and finish entire subtraction. Perhaps that was because of the difficulty in externalizing automatized mental processes. And this difficulty had not been overcome thoroughly.

The scores in the bug inference task did not changed in the pre- and posttests. Because the participants did not experience construction of any bug model, performance in this task to infer thinking processes by other persons was not improved.

The support system recorded 63 errors in log files when the eight participants operated it. Twenty of the errors were due to an uninformed specification[1] of DoCoPro. Nineteen out of the remaining 43 were semantic errors because of positions of variables in predicates, such as inputting “(Leftof R L)” in a line which must have “(Leftof L R)[2]” in the implementation model. The learning activity on the support system does not include design of problem representation. The participants were only given texts explaining problem representation in the sheets provided. Actually in the posttest, some of the participants reported difficulty in comprehending the problem representation, such as “Task of programming was heavy, so I did not afford to learn things about the model construction” and “I wanted graphical information to understand the processes.” This indicates necessity to expand the learning activity in the framework of the support system in terms of comprehending problem representation.

The participants’ descriptions about subtraction procedures were incomplete, even though, they might be sufficient as explanation for people. People would unconsciously complete the missing condition “the minuend must be equals to or greater than the subtrahend when performing subtraction (FindDifference1)” if the operation “borrow one from the left digit when the minuend is smaller than the subtrahend (FindDifference2)” is presented. For computers, however, such incomplete descriptions are not acceptable. Therefore, having learners examine their own mental processes with construction of cognitive models may help in facilitating certain thinking, such as computational thinking. Recent research on science education has paid much 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). Our framework to provide opportunities to construct cognitive models for non-information engineering students might contribute development of computational thinking.

Acknowledgements

This research was partially supported by Grant-in-Aid for Challenging Exploratory Research 15H02927 of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Basu, S., Dukeman, A., Kinnebrew, J., Biswas, G., & Sengupta, P. (2014). Investigating student generated computational models of science. Proceedings of ICLS2014 (pp. 1097-1101). Boulder, CO: International Society of the Learning Sciences.

Brady, C. Holbert, H., Soylu, F., Novak, M., & Wilensky, U. (2015). Sandboxes for model-based inquiry. Journal of Science Education and Technology, 24(2-3), 265-286.

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.

Clement, J. (2000). Model based learning as a key research area for science education. International Journal of Science Education, 22(9), 1041-1053.

Gilbert, J. K. (2004). Models and modelling: routes to more authentic science education. International Journal of Science and Mathematics Education, 2(2), 115-130.

Harrison, A. G., & Treagust, D. F. (1998). Modelling in science lessons: are there better ways to learn with models. School Science and Mathematics, 98(8), 420-479.

[1] A syntax error occurs in DoCoPro when a rule name includes some specific multi-byte characters.

[2] It is a predicate representing a fact “Column L is located at the left of column R (L and R are variables).”

Hirashima, H., Imai, I., Horiguchi, T., & Toumoto, T. (2009). Error-based simulation to promote awareness of errors in elementary mechanics and its evaluation. Proceedings of AIED2009 (pp. 409-416). Amsterdam, Netherlands: IOS Press.

Kanzaki, N., Miwa, K., Terai, H., Kojima, K., Nakaike, R., Morita, J., & Saito, H. (2015). A Class Practice and Its Evaluation for Understanding Cognitive Information Processing by Constructing Computational Cognitive Models. Transactions of the Japanese Society for Artificial Intelligence, 30(3), 536-546

Kojima, K., Miwa, K., Nakaike, R., Kanzaki, N., Terai, H., Morita, J., Saito, H., & Matsumuro, M. (2016). Basic framework for learning by constructing cognitive models based on problem-solving processes. Workshop Proceedings of ICCE2016 (pp. 451-453). Taoyuan, Taiwan: APSCE.

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). Jhongli, Taiwan: Asia-Pacific Society for Computers in Education.

Nakashima, H. (2008). Methodology and a discipline for synthetic research. Synthesiology, 1(4), 305-313.

Penner, D. E. (2000). Cognition, computers, and synthetic science: building knowledge and meaning through modeling. Review of Research in Education, 25, 1-35.

Schunn, C. D., Crowley, K., & Okada, T. (1998). The growth of multidisciplinarity in the Cognitive Science Society. Cognitive Science, 22(1), 107-130.

Seel, N. M., & Blumschein, P. (2009). Modeling and simulation in learning and instruction: a theoretical perspective. In P. Blumschein, W. Hung, & D. Jonassen (Eds) Model-Based Approaches to Learning (pp. 3-16). Rotterdam, Netherlands: Sense Publishers.

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).