Ogata, H. et al. (Eds.) (2015). Proceedings of the 23rd International Conference on Computers in Education. China: Asia-Pacific Society for Computers in Education

Tracing Problem-Posing Activity Sequences toward Detection of Trap-States in Thinking

Ahmad Afif SUPIANTO[a,b,*], Yusuke HAYASHI[a] and Tsukasa HIRASHIMA[a]

[a] *Graduate school of engineering, Hiroshima University, JAPAN*
[b] *Department of Informatics, Brawijaya University, INDONESIA*
*afif@lel.hiroshima-u.ac.jp

Abstract: Problem posing is well known as an effective activity to learn problem-solving methods. Although the activity is considered to contribute to understanding of the structure of problems, it is not clear how learners could understand it through the activity. This paper proposes detection of important learner's actions by tracing problem-posing activity sequences. The sequences are considered to represent thinking process of learners and reflect their understanding and misunderstanding about the structure of problems. This paper expects visualizations of the sequences to be helpful to infer learners' bottlenecks in thinking and misunderstanding behind them. The goal of this paper is to make it possible to analyze problem-posing activity sequence for enhancing the effectiveness in learning. As an example, this paper proposes detection of "trap-state" that is a combination of simple sentences many learners tend to make in problem-posing assignments.

Keywords: problem-posing process, problem state space, visualizations

1. Introduction

Problem posing is one of the key components of mathematical exploration. The development of problem-posing skills for students is one of the important aims of mathematics learning and it should occupy the center space in mathematical activities (Crespo, 2003). Moreover, problem-posing activities could provide us with important insights into children's understanding of mathematical concepts and processes, as well as their perceptions of, and attitudes towards, problem solving and mathematics in general (Brown and Walter, 1993). In order to improve students' learning in problem posing, it is important to develop an understanding about the developmental status of students' thinking and reasoning. The more information we can obtain about what students know and how they think, the more opportunities would be possible to create for student success (Cai, 2003).

In many cases, the analytics process would need to be transparent and enabling learners to respond with feedback that could be used to refine their thinking. Learners who used problem-posing learning environment changed their approach to pose problems after they had experienced posing the same type of story (Hasanah, Hayashi and Hirashima, 2015). Although practicing problem posing in learning environment is considered to contribute for understanding of the structure of problem posing, it is not clear how learners could understand it through the activity. Therefore, it is important to discovery learning and to generate inferences of learners' thinking from their behavior in learning environments. The discovery learning plays a role in increasing learners' motivation, and it creates more opportunities for learners to assess how well they could overcome obstacles, which may contribute towards learning (Reiser et. al, 1998).

This paper presents trap-states detected by tracing problem-posing activity sequences from the system's data-logs in problem-posing learning environment. (Ben-Naim, Marcus and Bain, 2008; Ben-Naim, Bain and Marcus, 2009) propose trap-states as pre-defined error answer and have detected actual trap-states as a solution trace based on the learners' answer. In addition to their definition of trap-states, we broad the definition to include the process of arranging the answer. With this definition, we would be able to analyze learners' understanding and then to provide feedback to them more effectively based on their understanding and offer adaptive learning.

2. Related Works

Many researchers have studied and used practically interactive learning environments for the problem posing. A new design of problem-posing learning environment using computer-based method is proposed as sentence-integration, named MONSAKUN (Hirashima et al., 2007). The use of sentence-integration method was proven to support learning by problem posing in the lower grades of elementary schools (Hirashima et al., 2008), higher grades of elementary schools (Hirashima and Kurayama, 2011), and was developed based on the "triplet structure model" (Hirashima, Yamamoto and Hayashi, 2014). Practical use of the environment was reported and showed that the practice to pose problems improved learners' ability not only in problem posing but also in problem solving (Yamamoto et al., 2012; Yamamoto et al., 2013). Several researches have specifically addressed the analysis of learning activities through the logs data (Hirashima et al., 2007; Hirashima et al., 2008; Hasanah, Hayashi and Hirashima, 2015). Even though many studies have analyzed the logs data, there are few studies to exploit the student behavior during learning activity.

There had been considerable works analyzing learners' activities to get general view of learners' learning. Statistics and visualization information are the two main techniques that have been most widely used for this task (Romero and Ventura, 2010). For statistical techniques, Hadwin et al. (2007) examine logs of trace data affords opportunities to examine the intersection between what students perceive about their studying, and what they do when they study; and Zorrilla et al. (2005) exploring the learners behavior and time distribution of network traffic over time. While visualization information techniques, they oriented toward visualizing educational data such as: learner tracking data regarding social, cognitive and behavioral aspects of learners (Mazza and Dimitrova, 2003); and tracing learners' answer that reflect on their behavior in an adaptive tutorial (Ben-Naim, Bain and Marcus, 2009).

3. Problem-Posing Activity in MONSAKUN

MONSAKUN was designed as an interactive learning environment for problem posing as sentence integration based on "triplet structure" model (Hirashima, Yamamoto and Hayashi, 2014). This model defines an arithmetic word problem as a composition of three simple sentences with two known numbers and one unknown number and problem posing as ensuring consistency among a story composed from three simple sentences and numerical relation of known and unknown numbers. Basically, the learners have already learned problem-posing structure on the black board by using several sentence cards that are parts of problems (Yamamoto et al., 2012). In order to promote learning deeper, MONSAKUN used to exercise and receive lectures of problem structure as usual classes.

MONSAKUN provides more than three cards. This means the cards include ones not necessary to pose the required problem. We call such cards as "dummy cards". These cards are included intentionally and used by learners with supposed types of overlooking, misunderstanding and so on, for example, careless of story types or confusion of formulas for representing stories and for calculation to solve problems. The learner selects several sentence cards and arranges them to pose a problem in a proper order. Putting a sentence card into a card slot and removing out a sentence card from a card slot are basic actions of learner on MONSAKUN. MONSAKUN records learners' problem-posing activity as the results that are combinations of cards set in the card slots.

4. Tracing Problem-Posing Activity Sequences

The learner selects several sentence cards and arranges them to pose a problem in a proper order. Putting a sentence card into a card slot and removing out a sentence card from a card slot are basic actions of learner on MONSAKUN. MONSAKUN records learners' problem-posing activity as combinations of cards set in the card slots. An activity is a resultant combination of cards, which is called "state" of the problem learner try to make. Based on the model, all possible states can be defined (including state never performed by learners). All learners' action can be mapped to one defined state. All possible states obtained from combining all the available sentence cards, included a state when all slots are empty. We refer to all possible states as the "Problem State Space". The problem state space means range of basic unit of thinking. The example of all possible states from 6 available cards is shown in Figure 1.

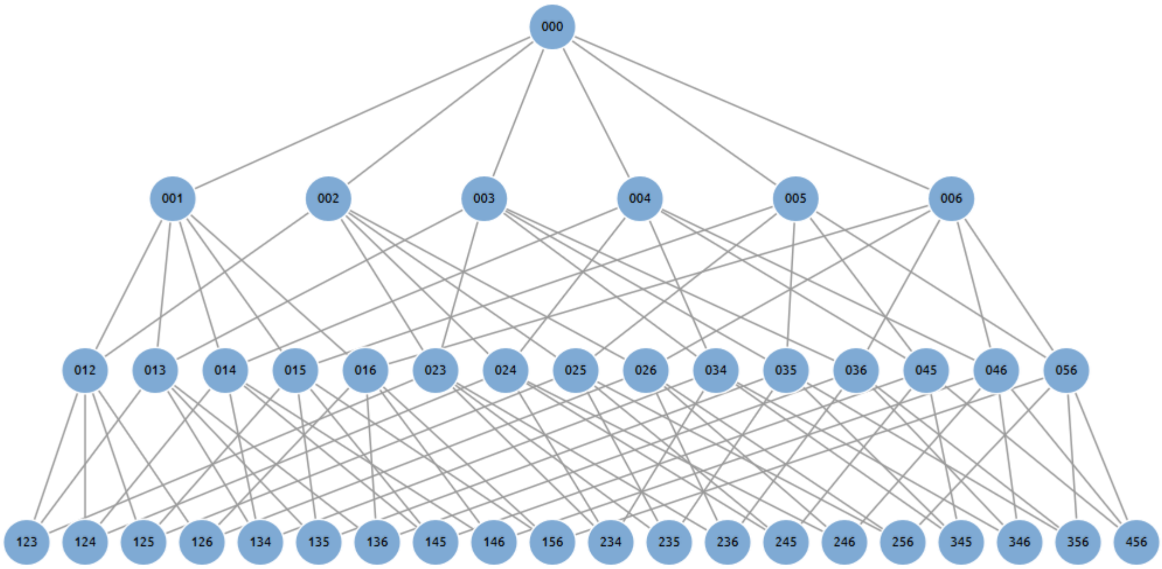

Figure 1. The graph of problem state space.

The possible combinations starting from state 000 which means that all empty slots (root-state), then proceed with one slot filled, two slots filed and all slots filled. There is a constraint that must be satisfied to generate all possible states. The card could only be used one time. Since the order of cards in the state is not important, we then combine state that has the same composition with different order into one state. The result of combining states, we get total state is 42 states. At last, we connect each state in accordance with the proper conditions where there is a relation between the situation before and after an action. For example, we connect a situation where all slots are empty with a situation where one card slot filled instead of two slots filled, because there is one situation that elapsed.

In order to complete an assignment, the learner tried various compositions of cards to generate a particular state according to what they thought. They continue to change the composition of cards until they reach the completed correct card composition. Every state that occurs on learners is stored by the system. Thus, we had an order of each state called "Sequences of States". A sequence of states is a collection of states arranged in the order of the learner's activities. This sequence reflects the way of learners' thinking.

A state that happens to learners is the result of their thinking. When students choose to put one card into one of the empty slot, it has a consequence. Similarly, when students tried to take out a card that has been installed in one slot, it will lead consequences too. The consequences could lead learners face the correct-state (state without dummy cards) or the error-state (state which contains dummy cards). The important thing is the consequences that cause learners getting away from the correct-state, even the correct answer. In this case, the learners are stuck in a condition where they would do more steps to reach the correct answer. In other words, the learners trapped in the state that distanced them from the correct answer. In addition, there are many learners who perform such state. Thus, we defined a state where it could lead learners do a lot of steps that distanced them to the correct answer and supported by many learners as "Trap-State".

In order to detect the trap-state, we describe two kinds of graphs: Support Graph and Distance Graph. A Support Graph displays the frequency of states appearing in learners' problem-posing activity. Support Graph is a graph where size of each node determined by how many times a state arranged by learners. This graph aimed to visualize states that have number of support shown by the size of the node. The node with a larger size has a number of supports more than the node with smaller sizes. A distance graph is a graph where size of nodes based on the far-close of the current-state to the correct answer state; it is called distance of state. The distance of state means the average number of steps of a state to the correct answer state. This graph aimed to visualize the distance value indicated by the size of the node. The node with a large-sized node has a long distance to the correct answer.

The sequences of state shown in Figure 2 are sequences of state that have been carried out by different learner. All sequences has the same state of correct answer, it is state 321 (state with green color). State 400 (state with orange color) occurs three times to achieve the correct answer, they are at the 1st step, the 7th step, and the 35th step which causes distance of 56 other states, 50 other states, and 22 other states respectively shown in Figure 2(a). We obtained the distance value of state 400 in this sequence by calculating the average of all distance values. State 400 in this sequence has a distance value of 42.67. Sequence shown in Figure 2(b) shows that state 400 occurs seven times and has a distance value of 30.86.

(a)

(b)

Figure 2. Three sequences with different states. (a) 57 states. (b) 67 states. (c) 50 states.

The value of each state in the both types of graph is normalized by scaling 0 to 1. We discard the node that has a value of zero, which means the learners have never performed the state. We want to focus on the state that ever made by learners. We also implement two different colors for nodes. The first color is red; it is for nodes that have a value greater than or equal to 0.3 on the scale normalization. The second one is blue; it is for nodes that have a value of less than 0.3. We did it on the ground that the nodes that has a value greater than or equal to 0.3 are: (1) states that have high value support, and (2) states that have long distances from the correct answer. For that reason, we would like to focus on the red states to be further analyzed. We argue that using these two graphs, we could detect trap-states based on large-sized node in Support Graph and Distance Graph.

5. Result and Analysis

For the purpose of a preliminary study regarding its usefulness, we focus on the first assignment in the fifth level. The participants were 39 Japanese students of first grade of elementary school that aged 6 years old. There are four story types: combination problem, comparison problem, increase problem, and decrease problem. The first assignment is about combination problem. This assignment consists of 1818 actions. The learners are asked to combine three sentence cards in order to pose a problem. Requirement of the first assignment is: *Make a word problem about "How many are there overall" that can be solved by "8-3"*. There are 6 sentence cards that could be used by learner. The sentences for each card from the first card to the sixth card are ***There are 3 white rabbits***, ***There are _ black rabbits***, ***There are 8 white rabbits and black rabbit's altogether***, *There are 8 white rabbits*, *There are 3 more white rabbits than black rabbits*, and *There are 3 brown rabbits* respectively. The correct answer is consisted of card 1, card 2 and card 3 (sentence card with printed bold).

Figure 3 shows the result of support graph and distance graph. The support graph shown in Figure 3(a) has 8 red states, while the distance graph shown in Figure 3(b) has 4 red states. Combining distance graph with support graph, general trap-states are revealed. We found that state 014 highlighted in both graphs. It is means that, for many learners, they tend to do more steps and further away from the correct answer when they are on state 014. Thus, this could be said as a general trap-state. In other words, we could say that a general trap-state is a colored red state in Support and Distance Graph.

The difficulty in this assignment is that learners are confused about the gap between the required story type of combination and the numerical expression of subtraction (8-3). Although subtraction generally implies story type of decrease and comparison, in this case learners must pose a problem of combination. In addition, before this assignment, learners have done assignments in which learners could make the correct answers by arranging cards according to the order of numbers in the numerical expression. However, this is not valid to this assignment because the numerical expression expresses not story but solution to evaluate unknown number. Even if they make a strategy to arrange cards according to the numerical expression from previous assignments, it doesn't work on this assignment. Actually learners tend to make such a strategy (Hasanah, Hayashi and Hirashima, 2015). In order to complete this assignment, for example, learners need to transform the numerical expression, "8-3", into the numerical expression representing a combination story, "3+?=8". And then, learners could assign cards of existence sentences to "3" and "?". State 014 consists of sentence card 1 (*There are 3 white rabbits*) and sentence card 4 (*There are 8 white rabbits*). This is supposed that learners try to directly use the given numerical expression, "8-3", and to assign card 1 and 4 to "3" and "8", respectively. Based on available cards, it was reasonable that card 1 and card 4 had chosen instead of card 2 containing unknown number (*There are _ black rabbits*) and card 6 contains different story with others (*There are 3 brown rabbits*). In this situation, most of them have confused and stuck due to the correct answer was number 8 on the calculation expression should be number in relational sentence (*There are 8 white rabbits and black rabbit's altogether*). Thus, state 014 could also be explained as a trap-state based on "triplet structure" model. We will confirm that by using these visualizations, trap-state for learners could be detected.

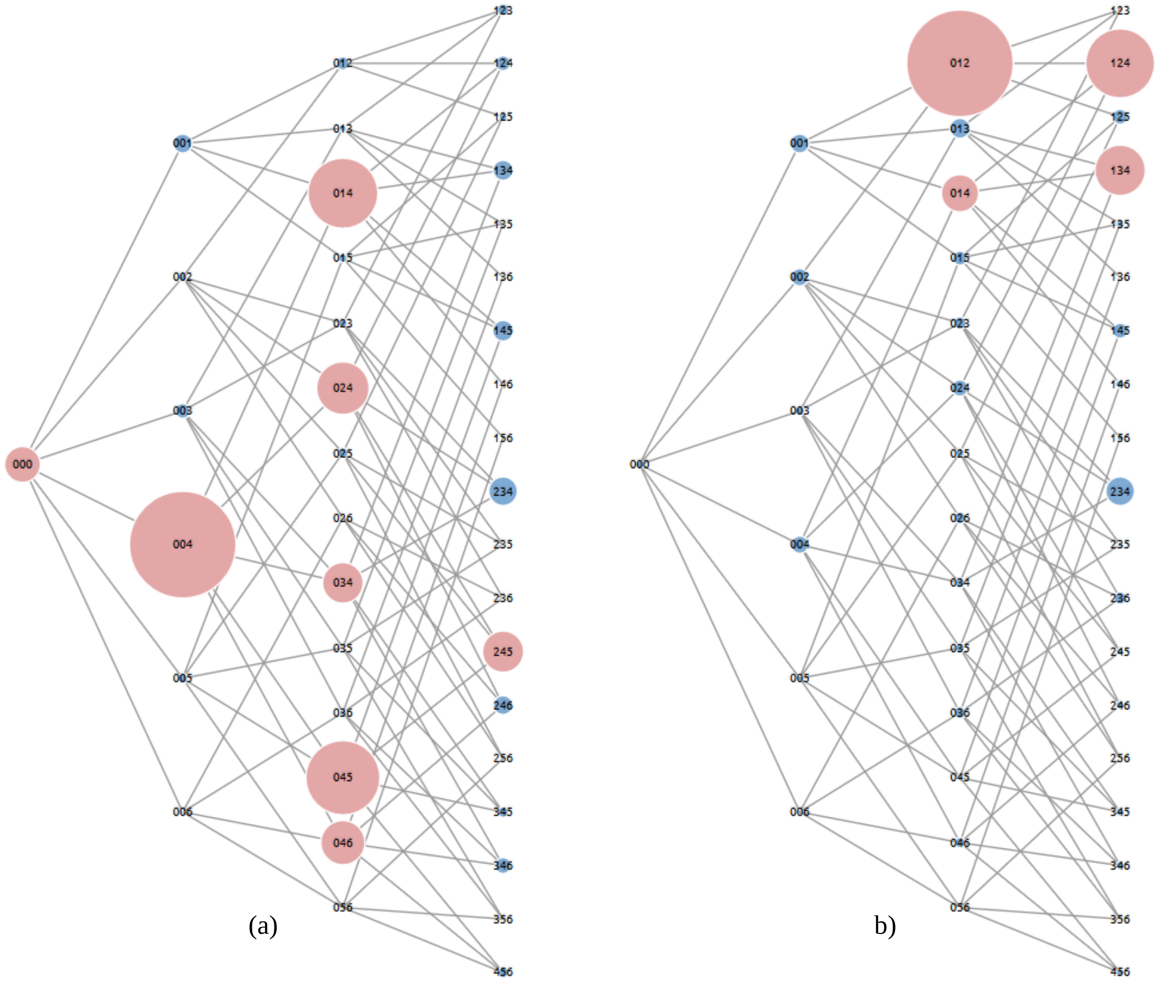

Figure 3. (a) Support Graph. (b) Distance Graph.

6. Conclusion and Future Work

We have conducted modeling data from system's data-logs that infer the learners' thinking from their behavior. We trace learners' activity sequences to detect the trap-states in problem-posing learning environment. As the result, we present two visualizations that externalize the activity of learners, support graph and distance graph. These visualizations trace different aspects of learners' activity, and combination from both of them could detect trap situation for learners. By this detection, the system could give support to learners during the learning process, especially when they are confused due to errors in choosing the sentence cards. Thus, learners could learn adaptively.

The ultimate goal of this line of research is placed in the context of exploring and mining data in problem-posing learning environment to get useful information for supporting learners. This research still preliminary and we believe that this research promises many further analysis such evaluating all assignments to detect trap-state. We also would like to explore ways to identify the other significant actions. We also plan to include data mining method for discovering learners' activities, for example, sequential data mining and clustering method for grouping learners.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15H02931.

References

Ben-Naim, D., Marcus, N., & Bain, M. (2008). Visualization and analysis of student interaction in an adaptive exploratory learning environment. *In International Workshop on Intelligent Support for Exploratory Environment*, EC-TEL (Vol. 8).

Ben-Naim, D., Bain, M., & Marcus, N. (2009). A user-driven and data-driven approach for supporting teachers in reflection and adaptation of adaptive tutorials. In *Educational Data Mining*.

Brown, S. I. & Walter, M. I. (1993). Problem Posing: Reflections and Applications, *Lawrence Erlbaum Associates*, Hillsdale, New Jersey.

Cai, J. (2003). Singaporean students' mathematical thinking in problem solving and problem posing: an exploratory study. *Intl. Journal of Mathematical Education in Science and Technology*, *34*(5), 719-737.

Crespo, S. (2003). Learning to pose mathematical problems: Exploring changes in preservice teachers' practices. *Educational Studies in Mathematics*, *52*(3), 243-270.

Hadwin, A. F., Nesbit, J. C., Jamieson-Noel, D., Code, J., & Winne, P. H. (2007). Examining trace data to explore self-regulated learning. Metacognition and Learning, 2(2-3), 107-124.

Hasanah, N., Hayashi, Y., & Hirashima, T. (2015). Analysis of Problem-Posing Process of Arithmetical Word Problem as Sentence Integration: Viewpoint of First Selected Sentence. In *Emerging Issues in Smart Learning* (pp. 85-88). Springer Berlin Heidelberg.

Hirashima, T., & Kurayama, M. (2011). Learning by problem-posing for reverse-thinking problems. In *Artificial Intelligence in Education* (pp. 123-130). Springer Berlin Heidelberg.

Hirashima, T., Yamamoto, S., & Hayashi, Y. (2014). Triplet structure model of arithmetical word problems for learning by problem-posing. In *Human Interface and the Management of Information. Information and Knowledge in Applications and Services* (pp. 42-50). Springer International Publishing.

Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi, A. (2007). Learning by problem-posing as sentence-integration and experimental use. In*AIED* (Vol. 2007, pp. 254-261).

Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi, A. (2008). Long-term use of learning environment for problem-posing in arithmetical word problems. In *Proc. of ICCE* (pp. 817-824).

Mazza, R., & Dimitrova, V. (2003). Informing the design of a course data visualisator: an empirical study. *In 5th International Conference on New Educational Environments* (ICNEE 2003) (pp. 215-220).

Reiser, B. J., Copen, W. A., Ranney, M., Hamid, A., & Kimberg, D. Y. (1998). Cognitive and motivational consequences of tutoring and discovery learning. *Northwestern univ evanston il*.

Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions* on, 40(6), 601-618.

Yamamoto, S., Kanbe, T., Yoshida, Y., Maeda, K., & Hirashima, T. (2012). A case study of learning by problem-posing in introductory phase of arithmetic word problems. In *Proceedings of the 20th International Conference on Computers in Education* (pp. 25-32).

Yamamoto, S., Kanbe, T., Yoshida, Y., Maeda, K., & Hirashima, T. (2013). Learning by problem-posing with online connected media tablets. In *Human Interface and the Management of Information. Information and Interaction for Learning, Culture, Collaboration and Business*, (pp. 165-174). Springer Berlin Heidelberg.

Zorrilla, M. E., Menasalvas, E., Marin, D., Mora, E., & Segovia, J. (2005). Web usage mining project for improving web-based learning sites. *In Computer Aided Systems Theory–EUROCAST* (pp. 205-210). Springer Berlin Heidelberg.