Analysis of Problem-Posing Activity Sequences toward Modeling Thinking Process and Detection of Trap States

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

[a] *Graduate school of engineering, Hiroshima University, JAPAN*
[b] *Faculty of computer science , 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. The goal of this study is to make it possible to analyze problem-posing activity sequence for enhancing the effectiveness in learning. This paper proposes visualizations of problem-posing activity sequence in MONSAKUN, a learning environment for problem-posing of arithmetic word problems as sentence integration. This system requires that users pose problems not freely but as combinations of given simple sentences and logs problem-posing activity as sequences of them. The sequences are considered to represent the thinking process of learners and reflect their understanding and misunderstanding about the structure of problems. This study expects visualization of the sequences to be helpful to infer learners' bottlenecks in thinking and misunderstanding behind them. 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).

Educational Data Mining (EDM) is concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings in which they learn. Learning Analytics (LA) consists of measuring, collecting, analyzing, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs. EDM and LA both reflect the emergence of data-intensive approaches to education and improve the quality of analysis of large-scale educational data, to support both basic research and practice in education (Siemens and Baker, 2012). EDM focuses on the technical challenge, deals with the development of methods to extract value from data originating in an educational context (Romero and Ventura, 2010). LA focuses on the educational challenge, deals with the development of analytics for learning which focused on the perspectives of institutions needs such as grades and persistence, and this is a challenge to focus on the perspectives of learners related to their needs (Ferguson, 2012). In many cases, the analytics process would need to be transparent to enable 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 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. Therefore, this is important to generate inferences of learners' thinking from their behavior in learning environments when they receive feedback. Visualization is one approach which could be used to interpret the student's behavior.

The purpose of visualization is to "amplify cognition" about data (Card et al., 1999). Visualization could be fully leveraged to get better understanding from step-by-step data logs generated by learning environments. Anscombe (1973) suggested both calculations and graphs should be used by a computer, both sorts of output should be studied due to each of them would contribute to understanding. Visualization could help to avoid misinterpretation of data. Shneiderman (2002) claimed that integration of both data mining and information visualization to invent discovery tools could enable more effective exploration and promote responsibility. In this research, we present the design and evaluation of a set of visualizations that infer the learners' thinking from their behavior to detect the important actions. By this detection, we would be able to provide individualized feedback based on learners' 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. A long-term evaluation with the system was carried out and confirmed that it was interesting and useful for learning (Hirashima et al., 2008a). Moreover, the system also improved the problem solving ability of low performance students (Hirashima et al., 2008b). In 2011, a task model of problem-posing that dealt with not only the forward thinking problem but also the reverse thinking problem was proposed (Hirashima and Kurayama, 2011). Practical use of the environment focused on the first grade students was reported (Yamamoto et al., 2012) and developed with online connected media tablets (Yamamoto et al., 2013). The results showed that the practice to pose problems improved learners' ability not only in problem-posing but also in problem-solving. Finally, an interactive learning environment for learning by problem-posing based on the "triplet structure model" was developed and practically used (Hirashima, Yamamoto and Hayashi, 2014). In the practical use, it was confirmed that learning by problem-posing with MONSAKUN was a useful learning method.

Several researches have specifically addressed the analysis of learning activities in MONSAKUN. (Hirashima et al., 2007) have analyzed whether learners could pose problem based on the logs of the system. The number of posed problems and correct problems was shown and discussed. (Hirashima et al., 2008a) and (Hirashima and Kurayama, 2011) have analyzed the learning effect of MONSAKUN comparing with the score of pre-test and post-test on problem solving and posing. Further analysis has been conducted by (Hasanah, Hayashi and Hirashima, 2015), this study examines the way of learners' thinking based on the first selected sentence in assignments on MONSAKUN. Binomial test to the amount of each card being firstly chosen or not in each assignment was implemented to analyze the result, and found that the selection changed based on different type of approach, type of story and students' exercise experience. Even though many studies have analyzed the logs data from MONSAKUN, there are few studies using visualization to exploit the potential learning of activities in that environment.

There has also been considerable work exploring the importance of visualizations to externalize the activity of learners. Some of them have conducted design and visualize learning process in a computer supported collaborative learning environment (Janssen, Erkens and Kanselaar, 2007; Tan et al., 2008), visualize and externalize the activity of groups working together on collaborative learning participation (Janssen et al., 2007; Rabbany, Takaffoli and Zaïane, 2011), and visualize the learning interaction with respect to collaborative and learning attitudes of each participant (Hayashi, Ogawa and Nakano, 2013). On individual learning, systems which collect detailed real-time data on learner behavior and interpret those data by drawing on behavioral research have been developed (Macfadyen and Sorenson, 2010). In 2011, the adaptive learning environment has been developed (Anjewierden et al., 2011). This system could monitor learner behavior through the actions they perform and identify patterns that point to systematic behavior using visual representation. Moreover, the visualization uses a tree structure to provide an overview of class performance also have been developed to allow easy navigation and exploration of student behavior (Johnson et al., 2011).

3. Recording Learners' Problem-Posing Activity in MONSAKUN

3.1 Problem-Posing Activity on MONSAKUN

MONSAKUN was designed as an interactive learning environment for problem posing as sentence integration based on "triplet structure" model (Hirashima, Tamamoto 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. Based on this model, MONSAKUN interface consists of three components: problem-composition area, sentence cards, and diagnosis button as shown in Figure 1.

The problem composition area consists of calculation expression and three card slots, the area in the left side of the interface. Here, calculation expression is an arithmetic expression that becomes reference to pose a problem using sentence card by learner. The three card slots in the area are the ones to set sentence cards. Sentence cards are presented at right side of the interface. A learner can move the card by dragging and dropping it to a slot in the interface. MONSAKUN provides more than three cards. This means the cards include ones not necessary to pose the required problem. We call such cards "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 last component is a button located under the problem composition area called diagnosis button. Diagnosis button is used to check the answer of the combination of sentences cards posed by learner. 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.

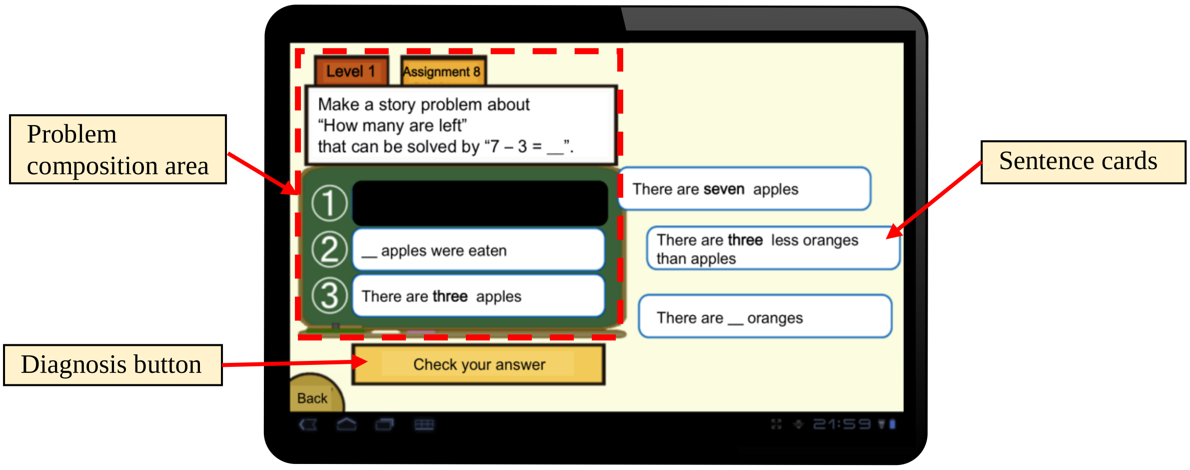

Figure 1. Interface of MONSAKUN.

3.2 Formulation of Problem Posing Process

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, shown in Table 1. Type 1 represents incomplete states as there are still empty card slots that are represented as zero. This type consist of card number and at least one zero number (state with brown color). Type 2 represents complete states before the student submitted the solution by pushing the diagnosis button. This type is coded by three combination of card number and followed by string '[u]' (state with black color). Type 3 represents an incorrect solution: the learner specified all three cards, but the solution is wrong. This type is coded by three combination of card number and followed by string '[f]' (state with red color). The last one (Type 4) represents correct solutions, the completed state and gets correct answer when pushed the diagnosis button. This type is coded by three combination of card number without followed by any other string code.

Table 1: The example of each state type.

Type	Definition	Example	Description
1	incomplete state	010	Slot 1 is empty, slot 2 is occupied by card 1, slot 3 is empty
2	complete state	413 [u]	Slot 1 is occupied by card 4, slot 2 is occupied by card 1, slot 3 is occupied by card 3, and without check the answer
3	wrong solution	315 [f]	Slot 1 is occupied by card 3, slot 2 is occupied by card 1, slot 3 is occupied by card 5, and check the answer then gets fail
4	correct solution	312	Slot 1 is occupied by card 3, slot 2 is occupied by card 1, slot 3 is occupied by card 2, and check the answer then gets success

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 state obtained from combining all the available sentence cards, including the empty slot. 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 2. The possible combinations starting from state 000 which means that all empty slots (root state), then proceed with the state 100, 200, 300... 010, 020, 030... 001, 002, 003, and so on. 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. For instance, state 013 is a combination of state 013, 031, 103, 130, 301, and 310. There is a constraint that must be satisfied to generate all possible states. The card could only be used one time. For example, it is impossible to create the state 121, which means that the first card is used twice, at the first slot and the third slot. However, it becomes possible to make a combination of empty slot by appearing more than once. The result of combining states, we get total state is 42 states.

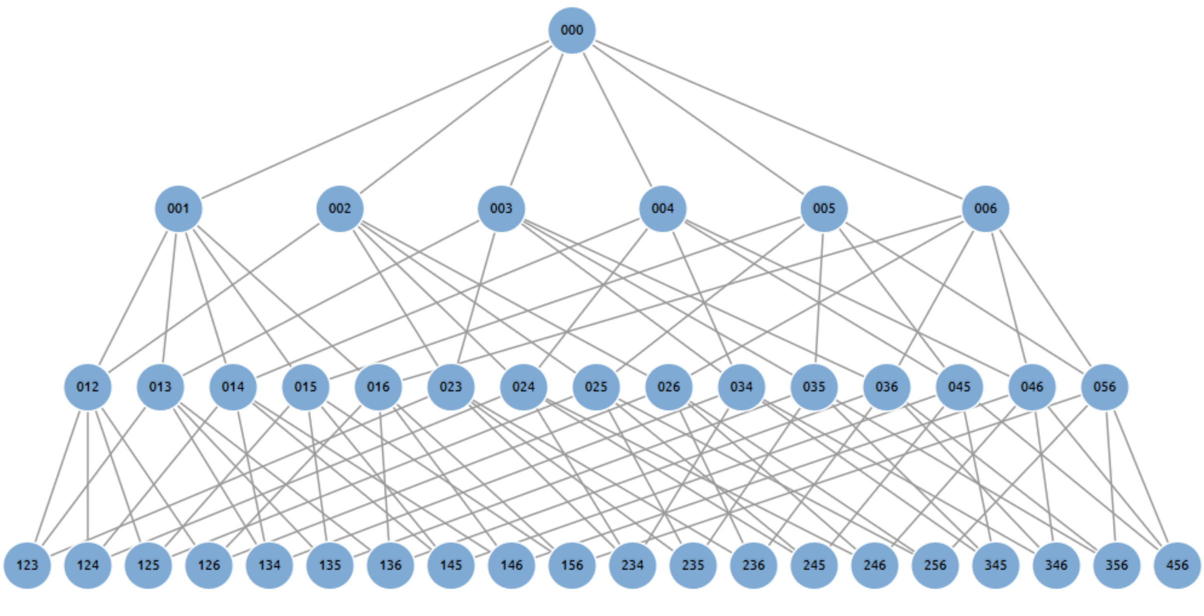

Figure 2. The graph of problem state space.

The next step, we connect each state in accordance with the proper conditions. The proper condition is a condition 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 is filled. It was impossible to connect a situation where all slot is empty with a situation in which two slots filled with cards, because there is one situation that elapsed. As a concrete example, we could not connect state 000 to state 014, because there is one step that elapsed before the state 014. The state that may be done before state 014 is state 001 or state 004.

3.3 Tracing the Way of Thinking: Sequence of States

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 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. Figure 3 shows three example sequences.

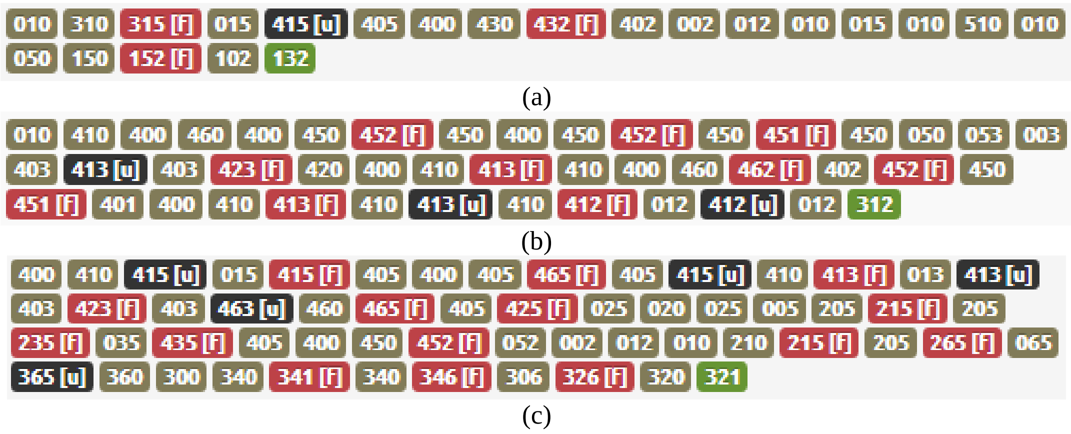

Figure 3. Sequence of states.

The first sequence has 22 states shown in Figure 3(a); which means that the sequence comprises 22 steps to reach the correct answer. The second and third sequences had 45 steps shown in Figure 3(b) and 57 steps shown in Figure 3(c) respectively. Four example steps in the first sequence are shown in Figure 4. In the second sequence begins with the state 010; this means that the learner put the first card in the second slot. State 010 is shown in Figure 4(a). The next state is the state 310 shown in Figure 4(b). In this state, learner put a third card into the first slot. The next state is learner put the fifth card in the third slot, followed by pressing the diagnosis button and learner get an error. Representation of the state is 315[f] shown in Figure 4(c). Due to an error found, then the learner tried to correct it by taking the third card from the first slot, this condition makes the state turned into state 015. The condition of the state 015 is shown in Figure 4(d). The complete steps of the first sequence shown in Figure 3(a) could be mapped on the problem state space shown in Figure 5. The blue nodes show visited states. The yellow links show relation between the states and the thickness represents how many steps the link is followed. On the other hand, the gray nodes mean states that never be arranged by learner. This represents what the learner consider before he get to the correct answer.

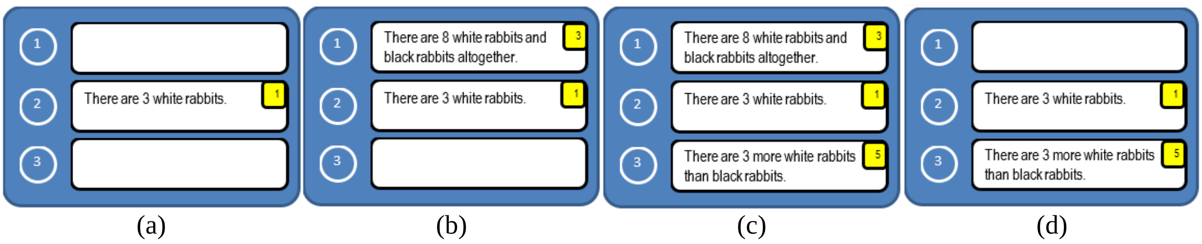

Figure 4. Part of a sequence. (a) State 010. (b) State 310. (c) State 315[f]. (d) State 015.

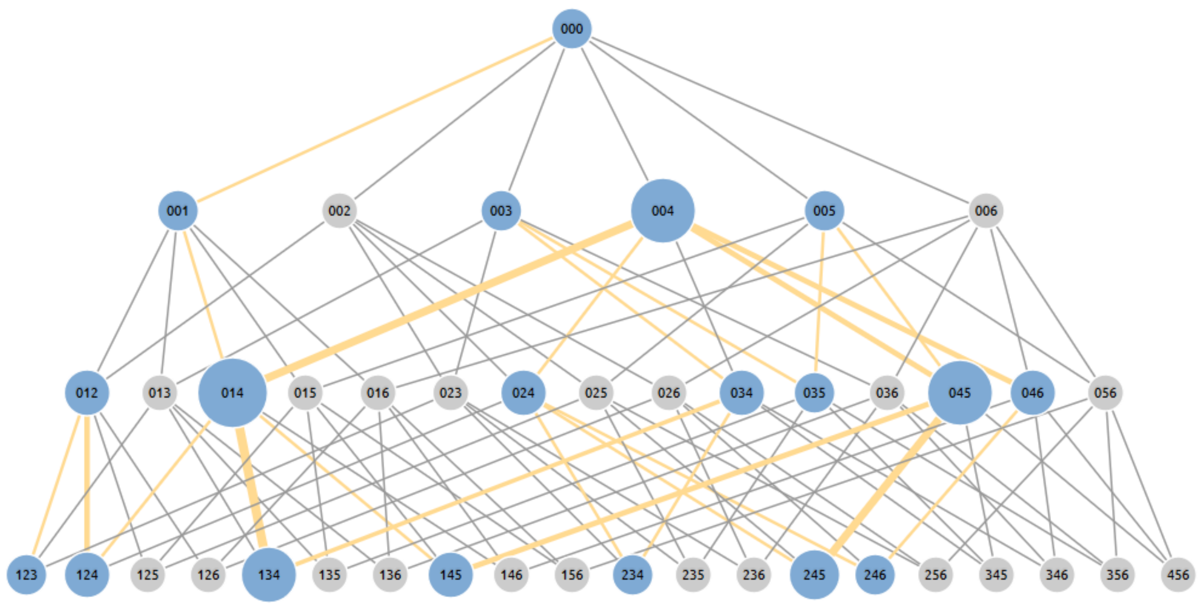

Figure 5. An example of mapped a sequence to the problem state space.

4. Visualizations of Problem-Posing Process

4.1 Data Logged in MONSAKUN

The participants were Japanese students of first grade of elementary school which aged 6 years old. Basically, the learners have already learned problem structure on the black board by using several sentence cards that are parts of problems (Yamamoto et al., 2012). These cards are provided to the learners as a request of problem posing. In order to promote learning deeper, MONSAKUN used as interactive learning environment to exercise and receive lectures of problem structure as usual classes. Every learner's action on MONSAKUN was logged into database. Each student was asked to create story problems using sentence card based on calculation expression. They had to select sentence card, move to the available slot, and complete all three slots. There are five or six cards provided by system. When learners finished posing the problem, they could push a diagnosis button under the problem-composition area. Then the system diagnoses the combination of sentences, and shows the results of the diagnosis and message to help the learner's problem-posing on another window.

This study collected data from learners' activity on MONSAKUN who involve 39 participants, and we focus on the first assignment in the fifth level. There are four story types: combination problem, comparison problem, increase problem, and decrease problem. The first assignment is about combination problem. 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:

① **There are 3 white rabbits**.
② **There are _ black rabbits**.
③ **There are 8 white rabbits and black rabbit's altogether**.
④ There are 8 white rabbits.
⑤ There are 3 more white rabbits than black rabbits.
⑥ There are 3 brown rabbits.

At this assignment, the correct state is consisted of card 1, card 2 and card 3 (sentence card with printed bold).

This assignment consists of 1818 actions. The raw data was coded as a series of Events, where Event= {id, lv, asg, stp, slt1, slt2, slt3, jdg}. "id" shows learner ID. "lv" is difficulty of problem-posing task and "asg" is number of assignment. "stp" shows sequence number of actions. slt1 "," slt2 and "slt3" is location of sentence card of first place, second place, and third place respectively. The last code is "jdg" which shows type of action, for example incomplete slot action, failed action, or successful action. We present a sample of log data from learners' action shown in Figure 6.

id	lv	asg	stp	slt1	slt2	slt3	jdg
1	5	1	1	0	1	0	n
1	5	1	2	4	1	0	n
1	5	1	3	4	0	0	n
1	5	1	4	4	6	0	n
1	5	1	5	4	0	0	n
1	5	1	6	4	5	0	n
1	5	1	7	4	5	2	f
1	5	1	8	4	5	0	n
1	5	1	9	4	0	0	n

Figure 6. Example of log data from learners' action.

4.2 Support Graph and Distance Graph

A sequence has several states as objects representing the learner's steps. The first step linked to the second step, the second step linked to the third one, and so on. For this reason, we propose the graph visualization, which shows the states and its relations. In addition, information visualization is best represented in graph structures which act as bridge between the visualization and graph drawing field (Gröller, 2002). Rabbany et al. (2011) use graphs to visualize overall snapshots of the students' participation in the discussion forums and gives the instructor a quick view of what is under discussion in online courses. In this study, such in Johnson et al. (2011), we design a graph where each node represents a state and each link an action that takes learner from one state to the next. The graph gives an overview visualization of all relations between the previous state and the next state in a sequence.

We describe two kinds of graphs: Support Graph and Distance Graph. A Support Graph displays the frequency of states appearing in learners' problem posing process. Figure 7 and Figure 8 show an example of Support Graph and Distance Graph respectively. Support Graph is a graph where size of each node is determined by how many times a state arranged by learners. This graph aimed to visualize states which 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 state; it is called distance of states. This graph aimed to visualize the average number of steps of a state to correct state indicated by the size of the node. The node with a larger size has an average number of steps more than node with a smaller size. It is means that a large-sized node has a long distance to the correct state.

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 state has never been done by the learners. 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 node which has a value greater than or equal to 30% are: (1) states that have high value support as shown in Figure 7, and (2) states that have long distances from the correct answer as shown in Figure 8. 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. Trap State: A Finding about Characteristic State in Problem-Posing

5.1 Characteristic Behavior of Learners

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 cause learners difficult to get 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 state. 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 to the correct state and supported by many learners as “Trap State”.

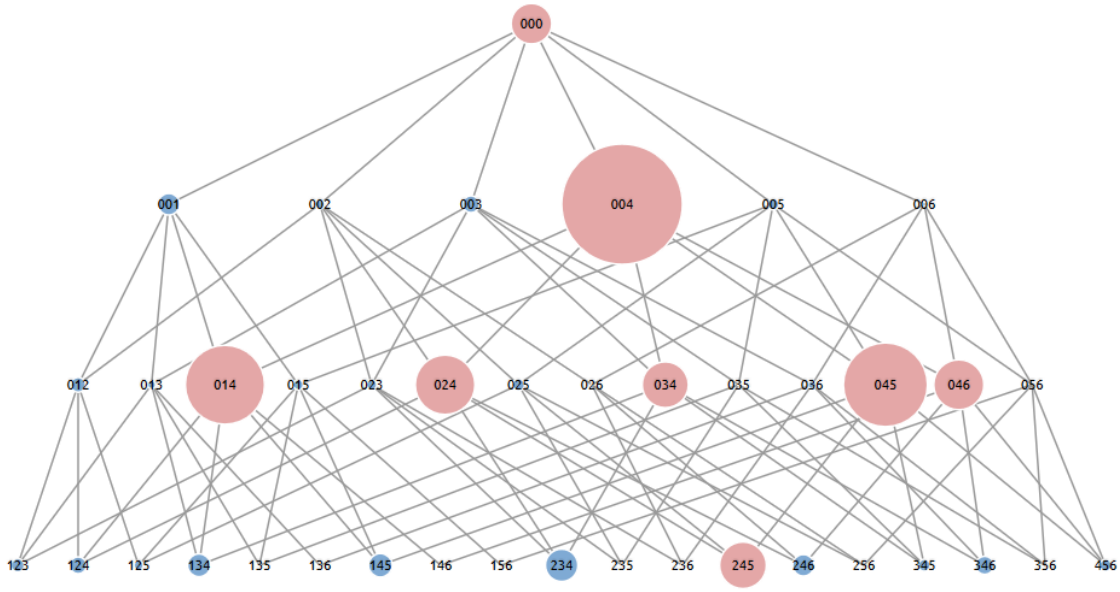

Figure 7. Support Graph of an assignment.

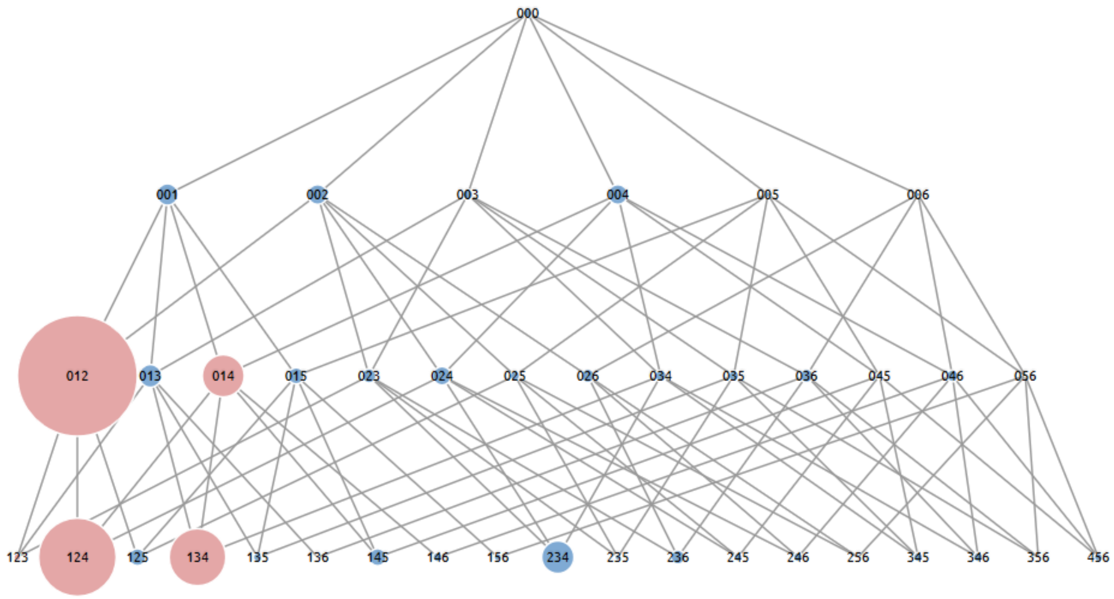

Figure 8. Distance Graph of an assignment.

5.2 An Example of Trap State

The support graph shown in Figure 7 has 8 red states. Based on the value of its support, the states are 004, 045, 014, 024, 034, 046, 245, and 000 which have a value of support 179, 125, 118, 89, 69, 75, 70, and 61 respectively. This means that the state 004 made by learners 179 times, state 045 made 125 times, and so on. On the other graph, distance graph shown in Figure 8 has four red states. The states are 012, 124, 134, and 014 which have the average distance to the correct state 63, 41, 30, and 22 respectively. This means that when learners were on state 012, they took 63 steps to reach the correct answer on average. Similarly with state 124 that required 41 steps, state 134 required 30 steps and state 014 required 22 steps. Basically, highlighted states on distance graph are strong candidate of trap states.

However, this is not enough to identify that a state is a trap state. Combining distance graph with support graph, general trap states are revealed.

The most distinct state in the distance graph is state 012 (the largest node shown in Figure 8). This situation makes the state 012 could potentially be a trap state for many learners. However, when we look at the support value, this state is only supported by a few learners (a little blue node shown in Figure 7). Although state 012 is a state that has required a lot of steps to reach the correct answer in this data, but there are not many learners' action arrange this state. In this situation, this state is not a trap state in general. The same thing occurred on state 124 and state 134. Both of them are also not supported by many learners.

The rest state with red color shown in Figure 8 is a state 014. When learners were in this state, they were required 22 steps to reach the correct answer on average. Moreover, this state is also supported by many actions with colored red in Figure 7. 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, by using visualization 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 a solution rather than a story to evaluate 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.

6. Conclusion and Future Work

We have presented visualizations that externalize the activity of learners at problem-posing learning environment to pose the problem based on requirement of an assignment. The Support Graph illustrates the number of states visited by learners. The Distance Graph depicts the number of steps to the correct answer. These visualizations trace different aspects of learners' activity, and combination of both visualizations 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 card. 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 these visualizations for 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.

References

Anjewierden, A., Gijlers, H., Saab, N., & de Hoog, R. (2011). Brick: Mining Pedagogically Interesting Sequential Patterns. In *EDM* (pp. 341-342).

Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, *27*(1), 17-21.

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. *International Journal of Mathematical Education in Science and Technology*, *34*(5), 719-737.

Card, S. J., Mackinlay, & Shneiderman, B. (1999). Readings in Information Visualization. *Morgan Kaufmann*.

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

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges.*International Journal of Technology Enhanced Learning*, *4*(5), 304-317.

Gröller, E. (2002). Insight into data through visualization. In *Graph Drawing* (pp. 352-366). Springer Berlin Heidelberg.

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.

Hayashi, Y., Ogawa, Y., & Nakano, Y. I. (2013). Visualization System for Analyzing Collaborative Learning Interaction. *Procedia Computer Science*, *22*, 1379-1387.

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. (2008a). Long-term use of learning environment for problem-posing in arithmetical word problems. In *Proc. of ICCE* (pp. 817-824).

Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi, A. (2008b). An experimental use of learning environment for problem-posing as sentence-integration in arithmetical word problems. In *Intelligent Tutoring Systems* (pp. 687-689). Springer Berlin Heidelberg.

Janssen, J., Erkens, G., & Kanselaar, G. (2007). Visualization of agreement and discussion processes during computer-supported collaborative learning.*Computers in Human Behavior*, *23*(3), 1105-1125.

Janssen, J., Erkens, G., Kanselaar, G., & Jaspers, J. (2007). Visualization of participation: Does it contribute to successful computer-supported collaborative learning?. *Computers & Education*, *49*(4), 1037-1065.

Johnson, M., Eagle, M., Joseph, L., & Barnes, T. (2011). The EDM Vis Tool. In*EDM* (pp. 349-350).

Macfadyen, L., & Sorenson, P. (2010). Using LiMS (the Learner Interaction Monitoring System) to Track Online Learner Engagement and Evaluate Course Design. In *EDM* (pp. 301-302).

Rabbany, R., Takaffoli, M., & Zaïane, O. R. (2011). Analyzing participation of students in online courses using social network analysis techniques. In *Proceedings of educational data mining* (pp. 21–30).

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.

Shneiderman, B. (2002). Inventing discovery tools: combining information visualization with data mining. *Information visualization*, *1*(1), 5-12.

Siemens, G., & d Baker, R. S. (2012). Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 252-254). ACM.

Tan, W., Lin, S., Yang, Y., & Zhao, X. (2008). Design on collaborative virtual learning community and learning process visualization. In *Advances in Web Based Learning-ICWL 2008* (pp. 424-433). Springer Berlin Heidelberg.

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.