Trap-States Found in Problem-Posing Activity Sequences Based on Triplet Structure Model

Ahmad Afif Supianto 1,2, Yusuke Hayashi 2, and Tsukasa Hirashima 3

1 Information Engineering, Hiroshima University, JAPAN
2 Department of Informatics, Brawijaya University, INDONESIA

Abstract: Problem-posing activities can provide a significant insights into learners' understanding about structure of problem. Finding an interesting pattern in a problem-posing learning environment is crucial to identify an important situation that learner may have difficulty to complete an assignment. This paper expects visualizations of the activity sequences to finding turning points where learners lose a way to reach the goal of an assignment. The activity sequences are considered to represent thinking process of learners and reflect their understanding and misunderstanding about the structure of problems. This paper proposes detection of “trap-states” that is an intermediate state of thinking in which learners have difficulty in achieving to the correct answer. As the results from an exercise detection of trap-states from real data, trap-states have found.

1 Introduction

Learning by problem posing can provide us with important insights into children’s understanding of mathematical concepts and processes [1]. Moreover, 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 [2]. Although learning by problem posing has been suggested as an important way to promote learner's understanding about the structure of problems [3, 4], it is still not clear how learners could understand it through the activity. Two techniques that have been used to explore and analyze learners’ activities are Educational Data Mining (EDM) and Learning Analytics (LA).

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 [5]. EDM focuses on the technical challenge, deals with the development of methods to extract value from data originating in an educational context [6]. 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 [7]. 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.

The general purpose of this study is to develop a method to analyze learners' thinking in problem posing. In general, problem-posing exercise where learners pose problems freely, it is difficult for students to pose problems and for teachers to analyze problems posed by students. The learning environment for problem-posing exercise MONSAKUN proposed as sentence integration [8] clears up the difficulty of the problem-posing method. In this method, learners pose problems meeting some sort of requirements by combining three simple sentences from given sentences. These make it easy for learners to increase the opportunity to pose problems and for teachers to assess the validity of posed problems by checking combinations of sentences and to provide feedback to learners according to their mistakes.

From the results of previous studies with MONSAKUN, lectures and exercise with it improve not only learners' problem posing but also their problem solving [9]. In addition to that, from the preliminary analysis of sentence selection in problem posing, learners changed their approach to pose problems after they had
experienced posing the same type of story [10]. 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 [11].

This paper presents trap-states detected by tracing problem-posing activity sequences from the system’s data-logs in problem-posing learning environment. [12, 13] 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 Research Methodology

2.1 Data Collection Tool

MONSAKUN was designed as interactive learning environment for problem posing as sentence integration based on “triplet structure” model [14]. This model defines an arithmetic word problem as a composition of three simple sentences with two known number 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 card, and diagnosis button as shown in Figure 1.

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 could move the card by drag and drop freely to set ones in the slots on the interface. Cards provided learners are more than three. 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.

There are four story types: combination problem, comparison problem, increase problem, and decrease problem. The first assignment is about combination problem. The learners asked to combine three sentence cards for posed the problem. They had to select sentence card, move to the available slot, and complete all three slots. 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 six 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). We would like to focus on this assignment. 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) which made learners confused to pose the correct answer.
2.2 Research Design

This study explores the log data of learners’ activities collected from MONSAKUN as Interactive Learning Environment for problem posing activity. The general purpose of this study is to develop a method to analyze learners’ thinking in problem posing. Specifically, this study try to discover important things during learning by problem-posing. Figure 2 shows the general approach to reach our goal.

In order to achieve our research goal, firstly, we collect log data of learners' activity from MONSAKUN. Then, we design a formulation to encode the problem-posing process. This stage transform from sentence card domain to the number representation domain. After that, we tracing problem-posing activity sequences by tracing activity records in database and convert it to the number representation domain. Then, we calculate two kind of values, support value and distance value. These two value used to measure how many learners perform an activity and to measure how far learners from current answer to the correct answer respectively. We would explain more detail about these values at the next section. The next stage, we visualize the values as graph representation. At the last, we find trap-states based on the visualization.

2.3 Research Questions

The objective of this study is discover important things during learning by problem-posing, especially in condition where learners have difficulty in achieving to the correct answer. It is mean that learners have misunderstanding behind the structure of problem and faces bottlenecks in thinking. The primary research question is: Do we could detect a condition where learners have difficulty in achieving to the correct answer by using visualization of problem-posing activity sequences? By this research question, this study aims to confirm that our approach can be used to discover a valuable situation where learners have difficulty to reach the correct answer.

2.4 Participants

The participants were Japanese students of first grade of elementary school, which aged 6 years old. Basically, learners were already learned problem structure on the black board by using several sentence cards that are parts of problems [9]. 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.

3 Finding Trap-States

3.1 Collecting Log Data

Every learner’s action on MONSAKUN was logged into database. This study collected data from learners’ activity on MONSAKUN who involve 39 participants. All participants tried to pose the problem three times. The assignment consisted of 1818 actions. The raw data was coded as a series of Events, where Event= \{id, lv, asg, trial, stp, slt1, slt2, slt3, jdg\}. "id" shows learner ID. "lv" is difficulty of problem-posing task and "asg" is number of assignment. "trial" is learners' exercise and "stp" shows sequence number of actions. slt1 “,” slt2 and "slt3" is location of sentence card of first place, second place, and third place respectively. We present a sample of log data from learners’ action shown in Table 1.

3.2 Formulating 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. 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.

| Table 1: Example of log data from learners' action. |
|---|---|---|---|---|---|---|
| id | lv | asg | trial | stp | slt1 | slt2 | slt3 |
| 1 | 5 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 5 | 1 | 1 | 2 | 4 | 1 | 0 |
| 1 | 5 | 1 | 1 | 3 | 4 | 0 | 0 |
| 1 | 5 | 1 | 1 | 4 | 4 | 6 | 0 |
| 1 | 5 | 1 | 1 | 5 | 4 | 0 | 0 |
| 1 | 5 | 1 | 1 | 6 | 4 | 5 | 0 |
| 1 | 5 | 1 | 1 | 6 | 4 | 5 | 2 |
| 1 | 5 | 1 | 1 | 6 | 4 | 5 | 0 |
| 1 | 5 | 1 | 1 | 7 | 4 | 0 | 2 |

The possible combinations of states starting from state 000 which means that all empty slots, then proceed with one slot filled, two slots filed and all slots filled. Since the order of cards in this assignment is not necessary, we then combine state that has the same composition with different order into one state. Moreover, there is a constraint that must be satisfied to generate all possible state. The card could only be used one time. The result of combining states, we get total state is 42 states.

The next step, 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. For instance, we could not connect state 000 to state 012, because there is one step that elapsed before the state 012; it is state 001 or state 002.

3.3 Tracing Activity Sequences

In order to complete an assignment, the learner tried varying composition of the card so that 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”. Sequence of states is collections of state are sorted based on the sequence of learners’ activity. This sequence reflects the way of learners’ thinking. The example of a sequence is shown in Figure 4.

There are two examples of sequences that have a different number of states that shown in Figure 4. The first sequence has 57 states shown in Figure 4(a); it is means that the sequence comprises 57 steps to reach the correct answer. While the second has 67 steps to reach the correct answer shown in Figure 4(b).

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 closer or farther to the correct answer. The important thing is the consequences that cause learners getting away from the correct answer. In this case, the learners are stuck in a condition where they would do more steps to reach it. 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 general “Trap-State”.

![Figure 4: Sequence of states.](a)

3.4 Calculating Support and Distance

In order to detect the trap-state, we propose two kinds of calculation: calculation of support value and calculation of distance value. A support value is obtained by calculate the frequency of learners that perform the state. This value aims to show how many learners that support the state. A distance value is obtained by
calculate the average number of steps of a state to the correct answer state. This value aims to show how far-close of the state to the correct answer state; it is called distance of state.

The sequences of state shown in Figure 4 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 4(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 4(b) shows that state 400 occurs seven times and has a distance value of 30.86.

3.5 Visualizing using Graph

A sequence has several states as objects that linked by ordered 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 this study, such in [15], we design graph where each node represents a state and each link represents 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 based on two values proposed and discussed on previous subsection: Support Graph and Distance Graph. Support Graph displays the frequency of state appearing in learners’ problem posing process. Figure 5 and Figure 6 show an example of Support Graph and Distance Graph of the assignment respectively. Support Graph is a graph where size of each node determined by support value. 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 distance value. This graph aimed to visualize the average number of steps of a state to the correct answer 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.

![Figure 5: Support graph of the assignment.](image)

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 done the state. We would like 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 0.6 and 0.3 on the scale normalization for support graph and distance graph respectively. Otherwise, nodes will be colored with the second color, it is blue. We did it on the ground that the red node are: (1) states that have high value support as shown in Figure 5, and (2) states that have long distances from the correct answer as shown in Figure 6. 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.
Furthermore, we present another visualization called “Trap Graph”. Trap graph was obtained by multiplying support value and distance value. We call this multiplication result as trap value. We also normalize the trap value by scaling 0 to 1. An example of trap graph shown in Figure 7 provides information about the general trap-state of the assignment. General trap-state is displayed by a red node in the graph.

3.6 Finding Trap-States

The support graph shown in Figure 5 has six red states. Based on the value of its support, the states are 004, 045, 014, 245, 024, and 034, which have a value of support 37, 32, 27, 27, 26, and 25 respectively. On the other graph, distance graph shown in Figure 6 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 due to learners have stuck and do a lot of step to reach the correct state. However, this is not enough to identify that a state is a trap-state. As defined before that trap-state is state that not only lead learners do a lot of steps, but also have to be supported by many learners. Therefore, combining distance graph with support graph, general trap-states are revealed.

Figure 6: Distance graph of the assignment.

Figure 7: Trap graph of the assignment.
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 5). 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 arrange this state shown in Table 2. In this situation, this state is not a trap-state in general because only 9 learners arrange this state. The same thing occurred on state 124 and state 134 which are arranged by 11 and 17 learners respectively shown in Table 2. Both of them are also not supported by many learners.

Table 2: Properties of four red states in the distance graph.

<table>
<thead>
<tr>
<th>State</th>
<th>Number of Actions</th>
<th>Number of Learners</th>
<th>Average Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>012</td>
<td>23</td>
<td>9</td>
<td>63</td>
</tr>
<tr>
<td>124</td>
<td>26</td>
<td>11</td>
<td>41</td>
</tr>
<tr>
<td>134</td>
<td>35</td>
<td>17</td>
<td>30</td>
</tr>
<tr>
<td>014</td>
<td>118</td>
<td>27</td>
<td>22</td>
</tr>
</tbody>
</table>

The rest state with red color shown in distance graph is 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 learners with colored red in support graph. 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. As shown in Table 2, this state is supported by 118 actions which are arranged by 27 learners. In addition, state 014 contains dummy card. Thus, this state 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.

Based on the trap graph, we also obtained the result of general trap-state was state 014 at the assignment. The result of general trap-state in the trap graph showed the same result when we used support and distance graph. We argue that trap graph could be used to represents both support graph and distance graph.

4 Discussions

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 [10]. 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 "?".

General trap-state at this assignment that makes learners tend to do more steps and distanced from the correct state is state 014. 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 general trap-state based on “triplet structure” model. We will confirm that by using these visualizations, general trap-state for learners could be detected. It is mean that our approach can be used to discover a valuable situation where learners have difficulty to reach the correct answer.

5 Conclusions 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.

Acknowledgment

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

References

and Applications, Lawrence Erlbaum Associates,
Hillsdale, New Jersey (1993)

Exploring changes in preservice teachers’
practices, Educational Studies in Mathematics,

Problems: A New Perspective on Talented
Mathematicians, Educational Studies in Mathematics,

method, Princeton University Press, (1957)

[5] Siemens, G., & d Baker, R. S.: 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 (2012)

review of the state of the art, Systems, Man, and
Cybernetics, Part C: Applications and Reviews, IEEE

and challenges, International Journal of Technology

[8] Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi,
A.: Learning by problem-posing as sentence-integration
and experimental use, AJED, Vol. 2007, pp. 254-261,
(2007)

Hirashima, T.: 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,
(2012)

Problem-Posing Process of Arithmetrical Word Problem as
Sentence Integration: Viewpoint of First Selected
Sentence, In Emerging Issues in Smart Learning, Springer

Kimberg, D. Y.: Cognitive and motivational consequences
of tutoring and discovery learning, Northwestern univ
evanston il, (1998)

[12] Ben-Naim, D., Marcus, N., & Bain, M.: Visualization and
analysis of student interaction in an adaptive exploratory
learning environment. In International Workshop on
Intelligent Support for Exploratory Environment,

[13] Ben-Naim, D., Bain, M., & Marcus, N.: A user-driven and
data-driven approach for supporting teachers in reflection
and adaptation of adaptive tutorials. In International
Conference on Educational Data Mining, Cordoba, Spain,
np. 21-30, (2009)

[14] Hirashima, T., Yamamoto, S., & Hayashi, Y.: 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, Springer International

EDM Vis Tool, In International Conference on
Educational Data Mining, Eindhoven, Netherlands, pp.
349-350, (2011)