IOP Conf. Series: Journal of Physics: Conf. Series **812** (2017) 012004 doi:10.1088/1742-6596/812/1/012004

Process-based Assignment-Setting Change for Support of Overcoming Bottlenecks in Learning by Problem-Posing in Arithmetic Word Problems

A A Supianto[1, 2], Y Hayashi[1] and T Hirashima[1]

[1] Graduate School of Engineering, Hiroshima University, 1 Chome-3-2 Kagamiyama, Higashihiroshima, Hiroshima Prefecture 739-8511, Japan
[2] Department of Informatics/Computer Science, Universitas Brawijaya, Jl. Veteran No.8t, Malang 65145, Indonesia
Email: afif@lel.hiroshima-u.ac.jp

Abstract. Problem-posing is well known as an effective activity to learn problem-solving methods. Monsakun is an interactive problem-posing learning environment to facilitate arithmetic word problems learning for one operation of addition and subtraction. The characteristic of Monsakun is problem-posing as sentence-integration that lets learners make a problem of three sentences. Monsakun provides learners with five or six sentences including dummies, which are designed through careful considerations by an expert teacher as a meaningful distraction to the learners in order to learn the structure of arithmetic word problems. The results of the practical use of Monsakun in elementary schools show that many learners have difficulties in arranging the proper answer at the high level of assignments. The analysis of the problem-posing process of such learners found that their misconception of arithmetic word problems causes impasses in their thinking and mislead them to use dummies. This study proposes a method of changing assignments as a support for overcoming bottlenecks of thinking. In Monsakun, the bottlenecks are often detected as a frequently repeated use of a specific dummy. If such dummy can be detected, it is the key factor to support learners to overcome their difficulty. This paper discusses how to detect the bottlenecks and to realize such support in learning by problem-posing.

1. Introduction

Problem-posing is recognized as an important component in the nature of mathematical thinking [1]. Problem-posing involves generating new problems and questions aimed at exploring a given situation as well as reformulating a problem during the course of solving a related problem [2]. Recently, technology-enhanced approaches have been used to realize learning by problem-posing in a practical way. An interactive learning environment for problem-posing in arithmetic word problem as sentence-integration has been developed [3], named Monsakun. A long-term evaluation of the system was carried out, and the study confirmed that it was interesting and useful for learning [4]. In problem-posing via sentence integration, several simple sentences are provided to a learner. The learner, then, selects the necessary sentences and arranges them in an appropriate order. This approach makes simple and goal-oriented problem-posing tasks possible even for lower elementary students while maintaining its value as a viable learning method. Nevertheless, in some cases, the environment provides an assignment that makes learners feel difficulty so that it requires figuring out the assignment setting for helping learners posing meaningful problems. Thus, there is a need to support learners in problem-posing activities. This can be achieved by generating problem-posing assignments automatically based on the history of their activities, and hence, the assignments would meet the learners' level of understanding.

In this paper, we propose a method to help learners in problem-posing process on the learning environment called Monsakun by changing the assignment setting based on their bottlenecks of thinking. Supianto et al. [5] conducted detection of trap-states by tracing problem-posing activity sequences from the system's data-logs in a problem-posing learning environment. Moreover, the trap-state was found in the study is conducted by a technology in order to detect the bottlenecks of learners based on their activity, instead of pre-defined errors [6]. Detecting trap-state proposed by [5] was conducted in the group of learners. However, in this study, detecting trap-state is implemented for each learner in order to support individually based on the process of learning by problem-posing in arithmetic word problems. In our proposed method, the log data of learners' activity on Monsakun is collected and formulated. Next, trap-state as learners' bottlenecks in thinking is discovered. Finally, a new setting of the assignment is created according to the trap-state found. The provided assignment is expected to overcome from learners' bottlenecks in problem-posing activity and have a higher level of acceptance by both teachers and learners.

The paper is organized as follows. The next section presents problem-posing activity in Monsakun followed by describes the proposed mechanism support, which includes the detection of bottlenecks and changing assignment setting. Finally, we conclude by summarizing the paper and offer the suggestion for future work in the last section.

2. Problem-posing activity in Monsakun

Monsakun was designed as an interactive learning environment for problem posing as sentence-integration. In the problem-posing activity by using Monsakun, learners do not create their own problem statements, however, they are required to interpret the sentence cards and integrate them into one problem. The interface of Monsakun is shown in Figure 1. A learner is required to integrate three sentence cards from five or six available sentence cards in order to build a problem in the three card slots based on the requirement for each assignment. There are more than three cards provided to the learners, which means that not all the cards included are necessary to complete the required problem. We call such cards "dummy cards," and they are intentionally included to test learners and check for their understanding. For example, an overlooking of the story type or confusion about the formula may lead to problems completing the exercise correctly. The learner selects several sentence cards and arranges them to pose a problem in the correct order. Putting and removing sentence cards are the basic actions of learners, and Monsakun records the result of actions, which are combinations of sentence cards.

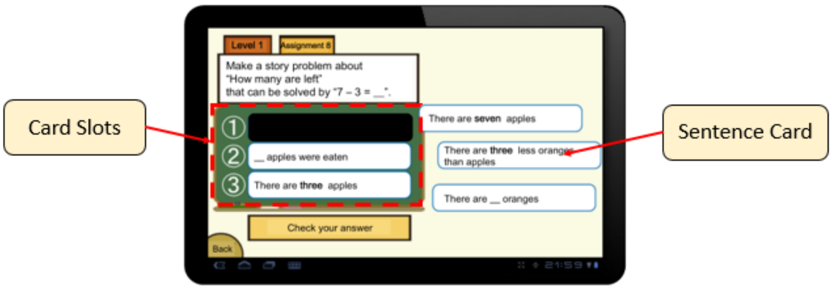

Figure 1. Interface of Monsakun for one operation of addition and subtraction.

The task model of problem posing as sentence integration on Monsakun has been developed based on the consideration of problem types in the triplet structure model [7]. This model is an arithmetical word problem model that is solved by one arithmetical operation composed of three quantities: operand, operand, and result quantity. In this model, all word problems are composed of two "existence sentences" and one "relational sentence." Although an existence sentence can be used in any story type, a relational sentence is used only in one specific story type. There are four types of story in arithmetic word problem of addition and subtraction: 1) combination, 2) increase, 3) decrease, and 4) comparison. Combination and comparison stories are story problems that the order of sentence cards in the slot is not necessary, while increase and decrease are story problem that the order of cards is restricted [8].

3. Method

In this section, we summarize the formulation of the problem before discussing the mechanism of support. Next, the bottlenecks in the process of problem-posing are formulated, and procedure of detection of bottlenecks and support system for overcoming them is elaborated.

3.1. Problem formulation

In this study, each composition of sentence cards is treated as a "state" in the problem-posing process by the learner. All possible states can be obtained by combining all the available sentence cards which divided into three categories: incomplete states, mistaken states, and correct answer state. Each state represents a basic unit of thinking, and a problem state space provides the range of thinking in a problem-posing assignment. Five main constraints to be satisfied by each posed problem have been devised [8], which are calculation, story type, number, objects, and sentence structure. When all five constraints are satisfied, the learner has succeeded in posing a correct problem according to the assignment requirements. When less than five constraints are satisfied, it shows that the learner has acquired a level of understanding in the structure of the arithmetic word problem; however, the final problem does not yet satisfy the requirements. If there are no constraints satisfied by the learner, it shows that the learner is unable to understand the structure of the arithmetic word problem. In order to cover the measurement of the validity of states, we define three values for each constraint: -1, 0, and 1. The value of -1 means the constraint is violated, the value of 0 means the constraint is not violated, and the value of 1 means the constraint is satisfied. The validity of a state is calculated by summing all constraints value, and the number of violated constraints is obtained by counting how many constraints are violated.

Figure 2 depicts an illustrative example of processed log data of a learner's problem-posing activity using Monsakun in Level 5 Assignment 1. The story type of this assignment is about combination problem. The requirement of this assignment was: *Make a word problem about "How many are there in all" that can be solved by "8-3."* There are six sentence cards that could be used by the learner. The sentences for each card, from the first card index to the sixth, are: ***There are 3 white rabbits, There are _ black rabbits, There are 8 white and black rabbits altogether,*** *There are 8 white rabbits, There are 3 more white rabbits than black rabbits,* and *There are 3 brown rabbits*. In this assignment, the correct answer consists of sentence card #1, #2, and #3 (the sentences appearing in bold), while sentence card #4, #5, and #6 are identified as dummy cards. The example of learners' log data of this assignment, which is designed as combination story problem, will be used for explaining the procedure suggested in this paper.

The brown, red, and green rectangles are shown in Figure 2(a) indicate the incomplete states, the mistaken states, and the correct answer state, respectively. In this example, the total number of steps is 67 times and the total number of mistakes is 18 times, which indicate that the learner feels difficult to reach the correct answer, and hence, need to be supported. The blue nodes in Figure 2(b) indicate the states that had already been made by the learner. The yellow lines show the path of the learner's steps and the thickness of the link indicates how many steps from a node to another. Based on this information, it looks the learner got stuck in state 000, 004, 024, 045 and 245. The learner seems struggle to achieve the correct answer, state 123, although in the end, the learner is able to achieve it through state 023. Therefore, in order to help them out of the impasse, a system that can support them according to learner's bottlenecks is conducted. The proposed system presents different assignment setting for the learner based on the detected bottlenecks. This setting expects the learner could overcome the bottlenecks and could accomplish the assignment smoothly.

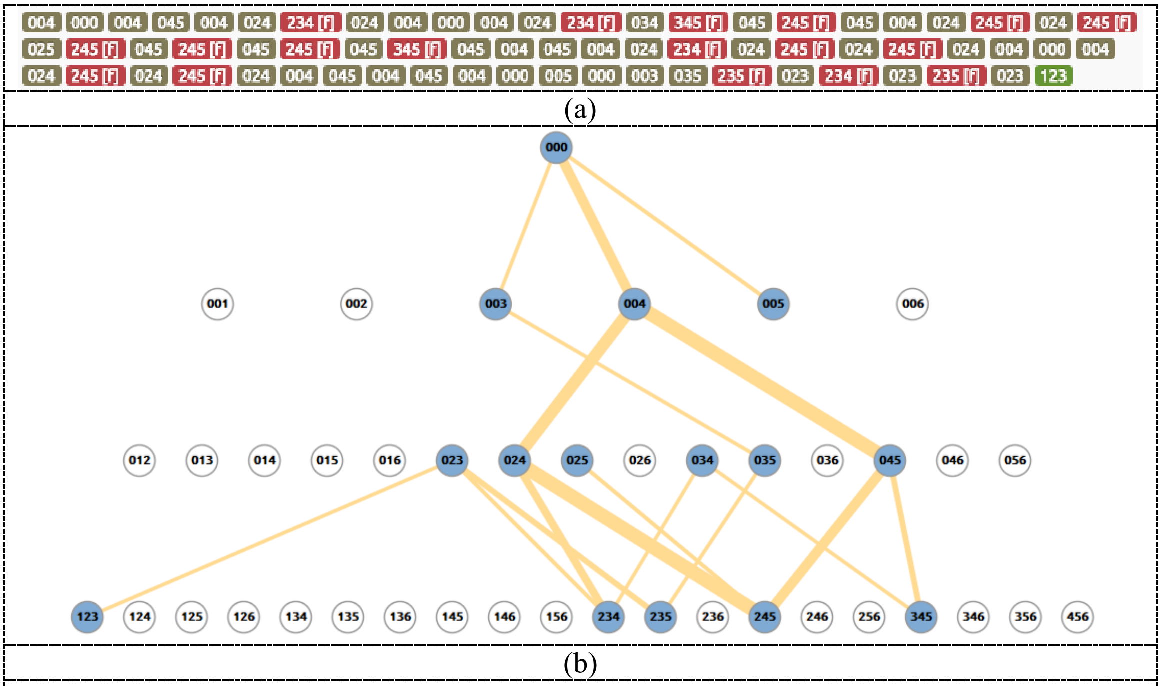

Figure 2. An example of log data of a learner's problem-posing activity in Monsakun. (a) The sequence of states. (b) Graphical representation.

3.2. Proposed procedure of process-based support system

In order to promote adaptive learning, we must constantly detect the learner's bottlenecks and possibly redirect the assignment setting accordingly. Our goal is to provide assignment setting in the system based on previous learner's steps recorded in the log data. These learner's steps are known as states. A state contained at least one dummy card where it could lead the learner to do many steps to get to the correct answer and reached by many actions as a "trap-state." In this paper, such trap-state is defined as learner's bottlenecks.

The proposed support system consists of two functional modules: *detection of bottleneck* and *recommendation setting*. The detection of bottleneck module responsible to find the learner's bottlenecks. It processes the log data of learner's activities and provides trap-state(s). The recommendation setting module generates new assignment setting to a learner based on the trap-state(s) found in the previous stage. As the result, we remove the frequently repeated use of a specific dummy card that makes learner stuck in posing the problem and build new assignment by omitting this dummy. The detail proposed procedure of process-based support system is given as follows:

- Formulation the problem-posing process: the sentence cards are encoded with indexing number. When the slot is still empty, index = 0 is implemented. For instance, when the learner poses the problem by selecting sentence card #1 and arranges it to the second slot, state 010 is generated.
- Tracing learner's activity: Every state stored by the system sequentially produce an order of each state called a "sequence of states." The sequence is then used in the next processes to detect the bottlenecks. An example of a sequence that consists of 67 states is shown in Figure 2(a).
- Finding trap-state: A state found as trap-state when it has the highest trap-value obtained from support value and distance value [5]. When the highest score happens in more than one state, then all states with the highest score will be selected as trap-states.

- Selecting the trap-card (the dummy): the system will select the trap-card in the trap-state(s). If there are more than one trap-cards, the system will sum-up the trap-value of the dummies and select the dummy having the highest score.
- Build new assignment by omitting the dummy obtained from the previous step. This new assignment setting will be inserted to the Monsakun.

4. Changing assignment setting

Removing the dummy, which is from the beginning intended as a distractor and eventually becomes the cause of the bottlenecks, might be able to overcome their impasse. We propose a method to change the characteristic of assignment based on the learners' bottlenecks. The changes of the assignment are generated from the previous one in order to guide the learners. The characteristic of current assignment has caused learners difficult to get the correct answer. Therefore, by changing the characteristic of assignment to be less difficult, having a good understanding of the problem's structure is expected. In this study, the characteristic of the assignment is defined as the violation of the constraints for all states generated in the states space

Figure 3 shows the proportion of the number of states that violate the constraints of Level 5 Assignment 1. It can be seen that the new setting when sentence card #4 is removed will help learners having difficulty in calculation constraint. The number of states in the calculation constraint of this setting is lowest than the other new settings. Thus, this setting is fit for learners that have bottlenecks in states related to the calculation constraints. On the other hand, the new setting when sentence card #5 is removed will help learners having difficulty in the story constraint because of the possibility of the violation of constraint about the story is decreased. While when the new setting in which sentence card #6 is removed will support learners that have difficulty in objects constraint because of the possibility of the violation of constraint about objects is decreased. Therefore, depending on the characteristic of the difficulty that faced by learners, the new setting would be generated.

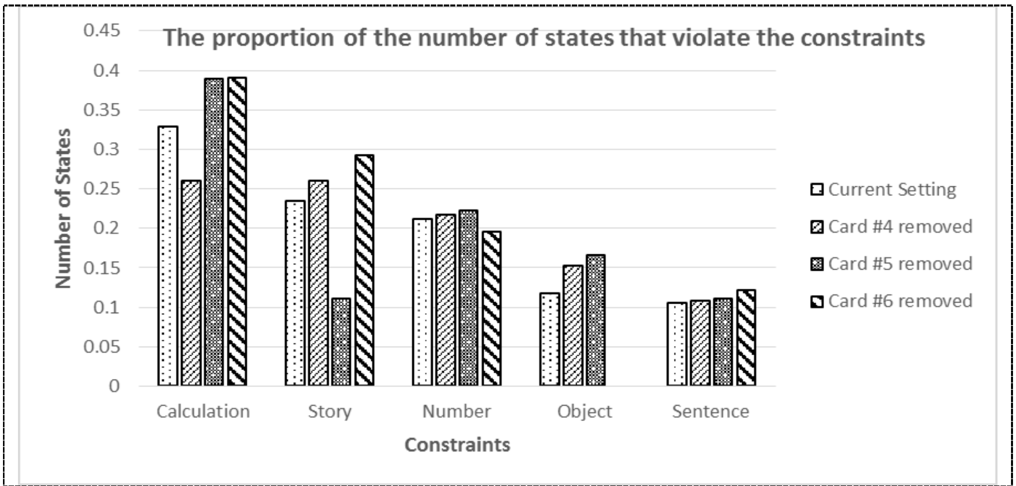

Figure 3. The chart of proportion of the number of states that violate the constraints of Level 5 assignment 1.

Based on the detection of the bottlenecks, the learner is trapped in state 024 during pose the problem in Level 5 Assignment 1. The trap-values of top-5 trap-states are state 024, 004, 245, 045, and 234 with trap-values are 144, 130, 90, 80, and 76, respectively. The difficulty in this assignment is that the learner confused about the gap between the required story type of combination and the numerical expression of subtraction (8–3). Although subtraction implies the story type of decrease and comparison, in this case, the learner must pose a problem of the combination story type. To complete this assignment, the learner needs to transform the numerical expression, "8–3," into the calculation expression representing a combination story, such as "3 + ? = 8," or "? + 3 = 8." Thus, the learner could assign existence sentence cards to “3” and “?”. According to the trap-state 024, the learner seems kept trying to use sentence card #4 (*There are 8 white rabbits*). There was a decided tendency based on some sort of thinking that the number “8” in the numerical expression was considered to be an existence sentence, instead of the relational sentence. In this situation, the learner confused and stuck because the correct answer with the number “8” is the number in the relational sentence, sentence card #3 (*There are 8 white and black rabbits altogether*). Such kind of difficulty, it would be helpful for learner when the assignment is changed to the new setting when sentence card #4 is removed because the number of states in the calculation constraint of this setting is lowest than the other new settings.

5. Conclusion and future work

In this paper, we present a method of the detection of learner’s bottlenecks in problem-posing activity on the learning environment and adaptively change the assignment setting based on the bottlenecks in order to support for overcoming them. We propose a method to change the characteristic of assignment which generated from the previous one based on the learners’ bottlenecks in order to guide the learner to have a good understanding of the problem’s structure. The method can generate recommendations of assignment-setting and it would be inserted into an existing problem-posing learning environment. The proposed method responsible for formulating problem-posing process from log data of learner’s activities, finding a trap condition that made learners stuck in a condition where they would have to do more steps to reach the correct answer and building new assignments to the learner. For the future work, we plan to implement the proposed setting of assignments in the practical use and analyze the result.

6. References

[1] Kilpatrick J 1987 *Cognitive Science and Mathematics Education* ed A H Schoenfeld and Hillsdale (NJ: Lawrence Erlbaum) pp 123–147
[2] Silver E A 1994 *For the Learning of Mathematics* **14** 19
[3] Hirashima T, Yokoyama T, Okamoto M and Takeuchi A 2007 *Proc. of the 2007 Conf. on Artificial Intelligence in Education: Building Technology Rich Contexts that Work (Los Angeles)* (Amsterdam: IOSPress) p 254-261
[4] Hirashima T, Yokoyama T, Okamoto M and Takeuchi A 2008 *Proc. of ICCE 2008* p 817 *(Taiwan)*
[5] Supianto A A, Hayashi Y and Hirashima T 2015 *Proc. of the 23rd Int. Conf. on Computers in Education (Hangzhou)* (New York: Springer) p 85
[6] Ben-Naim D, Bain M and Marcus N 2009 *Int. Conf. on Educational Data Mining (Raleigh)* p 21
[7] Hirashima T, Yamamoto S and Hayashi Y 2014 *Int. Conf. on Human Interface and the Management of Information (crete)* (New York: Springer) p 42
[8] Supianto A A, Hayashi Y and Hirashima T 2016 *Research and Practice in Technology Enhanced Learning* **11**

Acknowledgments

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