

# Does Experience of Feedback Generation Promote Student Novel Problem Posing? An Empirical Study in a Database Course

Kazuaki KOJIMA

*Learning Technology Laboratory, Teikyo University, Japan*  
kojima@lt-lab.teikyo-u.ac.jp

**Abstract:** The use of problem posing to assess students' understanding and transfer of knowledge has been discussed. The author's previous study proposed a method for learning assessment where students created problems that could be solved with solutions of given problems by changing the contextual settings in texts (situations). However, some of the students' problems did not feature new situations. To promote novel idea generation in the transfer of solutions into novel situations, this study adopted the generation of feedback to problems that students learned in a course to involve them more deeply in the problems. It then empirically investigated the effects of feedback generation on the students' problem-posing performance.

**Keywords:** Problem posing, assessment, student-generated feedback

## 1. Introduction

In addition to solving problems provided by a teacher or textbook, students can participate in problem posing, which involves them creating problems on their own. This has been identified as an important learning activity. However, problem posing is not commonly used in general education for various reasons, such as cognitively demanding tasks (Cai, & Hwang, 2023; English, 1997) and the necessity of providing individual feedback (Hirashima, Yokoyama, Okamoto, & Takeuchi, 2007). This is despite that fact that its importance has been well-documented (e.g., Ellerton, 1986; English, 1997; Silver, 1994).

The use of problem posing to assess students' understanding and transfer of knowledge has also been discussed (e.g., Cai, & Hwang, 2023; Mestre, 2002; Papadopoulos, Patsiala, Baumanns, & Rott, 2022). It is a difficult task as students might fail to compose appropriately solvable problems, even when they can easily solve problems in the same task domain (Kojima, Miwa, & Matsui, 2013a; 2013b). Thus, it may be challenging for general education to adopt problem posing to measure students' learning achievement. The author's previous study (Kojima, 2020) proposed a method for assessing learning through problem posing by students who had not received training in creating problems. It empirically revealed that the performance of problem posing in the assessment exhibited similar trends to the scores of a conventional examination through solving problems. However, a certain number of students did not create problems that met the requirements presented.

This paper reports learning assessment through problem posing in a university course. The author investigated whether students' problems were changed by a learning activity that allowed them to involve problems that appeared in the course more deeply. Here, the generation of feedback on problems was adopted as the activity.

## 2. Theoretical Background

Problem posing is a promising method for assessing students' understanding of concepts and procedures and testing their transfer to novel problem-solving situations. Several studies have

addressed assessment by having students pose their own problems; however, difficulties appear to exist in implementing problem posing in general education. Cai et al. (2012) measured the effect of a middle-school math curriculum on high school students through problem-posing tasks and observed that only a small number of them created valid math problems. Furthermore, Mishra and Iyer (2015) explored the potential of problem posing as an assessment tool in a computer science course and reported the relationship between problem-posing performance and traditional assessment (answering quizzes). However, they let the students generate problems in pairs to facilitate such performance.

The author proposed a task that combined solution-based and problem-based problem posing as an assessment tool (Kojima, 2020). Solution-based problem posing requires students to compose a problem that can be solved with a solution specified and problem-based problem posing to generate a new problem by changing a given one (Hirashima, et al., 2007). In the task, students are given a problem (base) before creating their own problems that can be solved with the solution of the base by changing the contextual setting in the text (the *situation*; e.g., “customers order goods when doing online shopping”). Table 1 illustrates the problems used in the author’s previous study, which were in the database (relational algebra) domain.

Table 1. Example of D/I problems.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (a) A table <i>employee</i> has columns of <i>employee_id</i> , <i>name</i> , <i>office_id</i> , and <i>service_years</i> . Describe a relational-algebra expression that can perform operations the same as the following SQL statement.<br>SELECT name, office_id FROM employee WHERE office_id IN (“E1”, “E2”, “E3”)<br>Solution) $\Pi_{name, office\_id}(\sigma_{(office\_id = "E1") \vee (office\_id = "E2") \vee (office\_id = "E3")} employee)$                                                |
| (b) A table <i>student</i> has columns of <i>student_number</i> , <i>name</i> , <i>department_id</i> , and <i>gpa</i> . Describe a relational-algebra expression that can perform operations the same as the following SQL statement.<br>SELECT name, department_id FROM student WHERE student_number IN (“19E221”, “19E222”, “19E223”)<br>Solution) $\Pi_{name, student\_number}(\sigma_{(student\_number = "19E221") \vee (student\_number = "19E222") \vee (student\_number = "19E223")} student)$ |
| (c) A table <i>order</i> has columns of <i>order_id</i> , <i>user_id</i> , <i>item_id</i> , and <i>number</i> . Describe a relational-algebra expression that can perform operations the same as the following SQL statement.<br>SELECT user_id, item_id FROM order WHERE item_id IN (“S500CK1”, “S500CK2”, “S500CK3”)<br>Solution) $\Pi_{user\_id, item\_id}(\sigma_{(item\_id = "S500CK1") \vee (item\_id = "S500CK2") \vee (item\_id = "S500CK3")} order)$                                         |

The three problems in Table 1 have different situations but share an identical-solution structure. Solution-based problem posing can create (a) or (b) as output when (c) is given as the base. Problems such as (a) and (b) for (c) are referred to as different-situation/identical-solution (D/I) problems. The creation of D/I problems can be effectively facilitated by presenting students with some examples (Kojima, & Miwa, 2008) as hints for their own idea generation.

The author’s previous study (Kojima, 2020) engaged students in the problem-posing task after presenting the three problems in Table 1 as examples to illustrate the concept of D/I problems. They were instructed that they had to use new situations they had not seen before in any course materials. Despite this instruction, a certain number of their problems involved situations that they had seen elsewhere in the course. To assess students’ understanding, it would be effective to have them adapt what they have learned in a situation to a novel one. Hence, it would be necessary to promote the idea generation of new situations away from familiar ones. The students in the previous study had solved many problems with various situations before they encountered the problem-posing task, even though they might not have involved each problem to a sufficient extent. To facilitate generative performance, problems should be learned through a generative task rather than problem solving.

A simple approach to such promotion is to provide students with experience of problem posing in a course. However, since problem posing by novice students occasionally requires much more time than problem solving, incorporating it into course instructional design is not necessarily easy. An alternative approach may be to generate feedback (Yu, & Liu, 2016) to problems posed in the course. Yu and Wu (2022) investigated students’ feedback generation to questions they had generated and noted its benefits, which included the promotion of question-generation quality. Although it is not necessarily clear whether feedback generation

can promote generative performance like novel idea generation about situations, it may lead students to more active, deeper involvement in problems. Therefore, in the next session, the author incorporated feedback generation into a computer science course and investigated its effects on the problem-posing task from bases.

### 3. Method

#### 3.1 Course Overview

The investigation was conducted in a database course held each semester of 2022–24 in the department of information and electronic engineering at the author's university. Each 15-week course included 13 lectures, practice performing database operations with SQL, and an assignment of the problem-posing task. Each lecture in 2022 and 2023 engaged students in a worksheet and test on a learning management system (LMS). Each worksheet included questions that students were to answer by referring to a textbook. Each test included 5–10 problems, each of which was randomly selected from a set of 3–6 problems. The problems in Table 1 formed one of the sets used in the LMS tests, which included different problems for each student. All problems had no feedback from the instructor but gave only correct/incorrect feedback. Thirteen tests in the lectures contained a total of 97 problem sets. The students were instructed to take each test repeatedly until they answered at least 80% of the problems correctly as a requirement for receiving course credit. They were also told that the number of tests in which they scored 100% would be added to their final course score.

Lectures except for the first of 2024 were different in some ways. In even-numbered lectures, a worksheet integrating a pair of lectures (e.g., the second and third lectures) in 2022–23 was given to the students. In odd-numbered lectures, students were randomly assigned into one of 3–4-person groups and received a set of 3–4 problems used in the test with the correct answers. Mostly, problems in a single set were D/I problems of each other. The students were instructed to describe their justification for selecting the answer (Yu, & Wu, 2022) as feedback for each problem. Every student had to generate feedback for one problem and check the feedback by one other group member. They then shared their feedback for the problems with other groups on the LMS before taking the test. There were 8–13 problem sets in each of seven tests in 2024, 3–8 problems in each set, and 71 sets in all tests.

In the last week of the course in each semester of 2022–24, students were presented with the assignment along with its instructions. The assignment included seven bases, each of which was created by slightly altering one of the problems used in the tests. Seven bases for the three years were different from each other because some students failed in the previous year and retook the course. Nevertheless, they shared basic concepts in their solutions (e.g., the third base had a subquery, while the sixth had normalization). The students were asked to pose a new D/I problem from each of the seven bases. They first learned the idea of D/I problems with the three presented in Table 1. The students were then informed that the mere duplication of any problems from the tests or the textbook was not acceptable. They were then presented with a list of seven situations that appeared in the course materials and told that using any of them was prohibited. They were explicitly required to adapt ideas in each base to a novel situation and create a suitable database. Each of them described their problems in a document file within a week and then submitted it using the LMS assignment function. This function enables an instructor to mark a student's response using a rubric and show the rubric to the student before submission and after marking. This rubric directly listed the seven situations prohibited to prevent the students from using any of them.

#### 3.2 Data Analysis

In the data analysis, a situation from each student problem was categorized into one of the following three categories:

- *Identical*: The situation was the same as one of the seven prohibited situations.

- *Similar*: The situation shared part of entities or relations in one of the seven situations; for example, one of the situations comprised *shops/stores*, *customers/clients*, *goods/products*, and *order/delivery*. Thus, the situation “a bakery shop manages ingredient stocks” was categorized as similar, whereas “customers order bread in a bakery shop” was categorized as identical.
  - *Different*: The situation was different from all of the seven prohibited situations.
- Students’ solutions were then categorized into the following three categories:
- *Failure*: including any misconceptions, inconsistencies, or errors;
  - *Simple*: a structure simpler than the base;
  - *Success*: an identical structure to the base or a more complex one.
- The analysis of the solutions was separately conducted for problems of the situation categories above. Please see the Appendix for examples of a base and student problems.

## 4. Results and Discussion

In the database courses, 55, 41, and 60 students submitted problems to the assignment in 2022, 2023, and 2024, respectively. The data of two students who did not complete the seven problems, two who retok the course and created problems from the previous year’s bases, and three who did not create their own problems were excluded from the analysis. The results below include data of 55 students in 2022, 38 in 2023, and 56 in 2024.

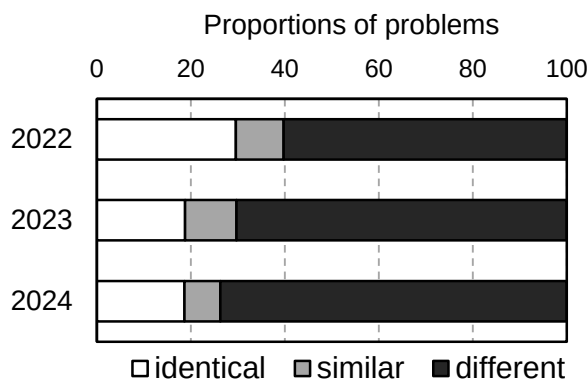


Figure 1. Proportions of Student Problems in Each Situation Category.

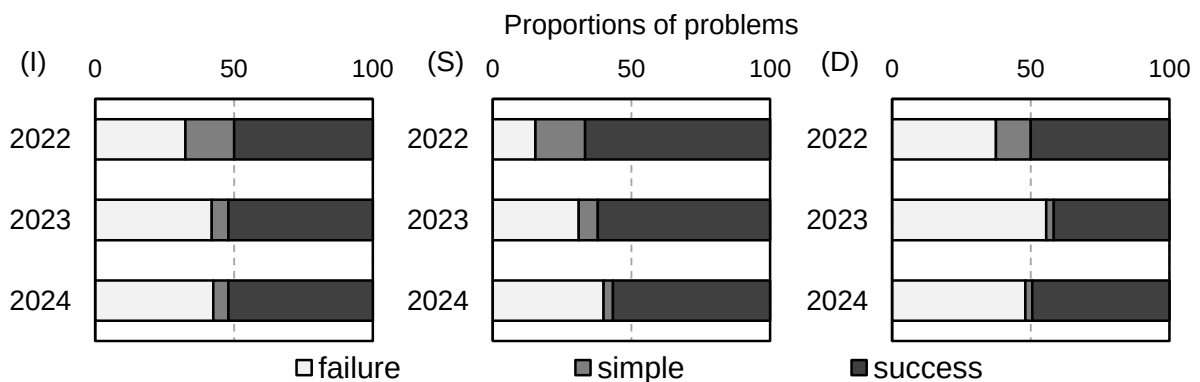


Figure 2. Proportions of Student Problems in Each Solution Category of Identical (I), Similar (S), and Different (D).

Figure 1 indicates the proportions of student problems in each of the three situation categories. A comparison of the three categories among three years using a chi-square test indicated a significant difference ( $\chi^2(4) = 20.37, p < .01$ ). The residual analysis revealed that the numbers of identical problems in 2022 ( $p < .01$ ) and different problems in 2024 ( $p < .01$ ) were high, those of different problems in 2022 ( $p < .01$ ) and identical problems in 2024 ( $p$

$< .05$ ) were low, and those of identical problems in 2023 ( $p < .10$ ) were moderately low. Therefore, identical problems decreased and different problems increased in 2024.

Figure 2 indicates the proportions of student problems in each of the three solution categories. We could not compare the differences among the years because the bases were not the same; however, only half of the student problems whose situations were identical or different were a success in all years.

As Figure 1 indicates, identical situations decreased in 2024. Each student in 2024 experienced the generation of feedback for some problems in the LMS tests as well as took the tests. This experience might have drawn students' attention to information about the problems, including the situations. However, 20% of their problems still had identical situations. The effect of promoting problem posing in a new situation was not necessarily sufficient for meeting the task's requirements.

In addition, identical situations decreased in 2023. Although the reason for this is unclear, a possible explanation is that students in 2023 learned more from the LMS tests. The average number of LMS tests in which they scored 100% was 9.42 out of 13, whereas the figure was 7.85 in 2022. A Mann–Whitney U-test revealed a significant difference between the averages ( $z = 1.17$ ,  $p < .05$ ). The number of the tests with a score of 100% was 2.61 out of 7 in 2024.

However, the effect of improving student solutions was not observed. The author could not measure this effect because whether the difficulty levels of the assignments differed between years was unclear. This is one limitation of this study. However, even problems in the identical situations failed to have appropriate solutions composed. Problem posing might have been challenging enough to the students even when they did not attempt to transfer the bases to novel situations.

One direction for future work is to conduct further analysis. In particular, studies could examine student-generated feedback in 2024.

## References

- Cai, J., Moyer, J. C., Wang, N., Hwang, S. Nie, B., & Garber, T. (2012). Mathematical Problem Posing as a Measure of Curricular Effect on Students' Learning. *Educational Studies in Mathematics*, 83(1), 57–69.
- Cai, J., & Hwang, S. (2023). Making Mathematics Challenging through Problem Posing in the Classroom. In R. Leikin (Ed), *Mathematical Challenges for All*, (pp. 115–145). Springer Nature.
- Ellerton, N. F. (1986). Children's Made-Up Mathematics Problems: A New Perspective on Talented Mathematicians. *Educational Studies in Mathematics*, 17(3), 261–271.
- English, L. D. (1997). Promoting a Problem-Posing Classroom. *Teaching Children Mathematics*, 4(3), 172–179.
- Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi, A. (2007). Learning by Problem-Posing as Sentence-Integration and Experimental Use. In R. Luckin, K. R. Koedinger, & J. Greer (Eds), *Proceedings of AIED2007* (pp. 254–261). IOS Press.
- Hirashima, T., Yamamoto, S., & Hayashi, Y. (2014). Triplet Structure Model of Arithmetical Word Problems for Learning by Problem-Posing. In S. Yamamoto (Ed), *Proceedings of International Conference on Human Interface and the Management of Information* (pp. 42–50). Springer-Verlag.
- Kojima, K., & Miwa, K. (2008). A System that Facilitates Diverse Thinking in Problem Posing. *International Journal of Artificial Intelligence in Education*, 18(3), 209–236.
- Kojima, K., Miwa, K., & Matsui, T. (2013a). Supporting Mathematical Problem Posing with a System for Learning Generation Processes through Examples. *International Journal of Artificial Intelligence in Education*, 22(4), 161–190.
- Kojima, K., Miwa, K., & Matsui, T. (2013b). Empirical Study on Errors of Mathematical Word Problems Posed by Learners. In S. C. Tan, Y. T. Wu, T. W. Apoko, L. H. Wong, C. C. Liu, T. Hirashima, P. Sumedi, M. Lukman (Eds), *Workshop Proceedings of ICCE2013* (pp. 187–191). UHAMKA Press.
- Kojima, K. (2020). Preliminary Study on Learning Assessment by Using Problem Posing in an Online Course as an Alternative Method of a Conventional Examination. In H. J. So, M. M. Rodrigo, J. Mason, A. Mitrovic (Eds), *Workshop Proceedings of ICCE2020* (pp. 93–98). APSCE.

- Lan, Y. F., & Lin, P. C. (2011). Evaluation and Improvement of Student's Question-Posing Ability in a Web-Based Learning Environment. *Australasian Journal of Educational Technology*, 27(4), 581–599.
- Mestre, J. P. (2002). Probing Adults' Conceptual Understanding and Transfer of Learning via Problem Posing. *Journal of Applied Developmental Psychology*, 23(1), 9–50.
- Mishra, S., & Iyer, S. (2015). An Exploration of Problem Posing-Based Activities as an Assessment Tool and as an Instructional Strategy. *Research and Practice in Technology Enhanced Learning*, 10, 5.
- Papadopoulos, I., Patsiala, N., Baumanns, L., Rott, B. (2022). Multiple Approaches to Problem Posing: Theoretical Considerations Regarding its Definition, Conceptualisation, and Implementation. *CEPS Journal*, 12(1), 13-34.
- Silver, E. A. (1994). On Mathematical Problem Posing. *For the Learning of Mathematics*, 14, 19-28.
- Yu, F., Liu, Y., & Chan, T. (2005). A Web-based Learning System for Question-Posing and Peer Assessment. *Innovations in Education and Teaching International*, 42(4), 337–348.
- Yu, F. Y., & Liu, Y. H. (2016). Development and Evaluation of Student-Generated Feedback in an Online Student-Generated Multiple-Choice Questions Learning Space. In T. Supnithi, A. F. Mohd Ayub, M. Mavinkurve, T. Kojiri, W. Chen, J. C. Yang, S. Murthy, S. L. Wong, S. Iyer (Eds), *Workshop Proceedings of ICCE2016* (pp. 413–418). APSCE.
- Yu, F. Y., & Wu, W. S. (2020). Effects of Student-Generated Feedback Corresponding to Answers to Online Student-Generated Questions on Learning: What, Why, and How? *Computers & Education*, 145, 103723.

## Appendix

**(The third base in 2024)** The following figure presents a database for managing the employees of a company. Some of the employees work at multiple offices.

| employee    |      |         | office      |          | service_years |             |       |
|-------------|------|---------|-------------|----------|---------------|-------------|-------|
| employee_id | name | address | office_name | location | employee_id   | office_name | years |

You have investigate the service years of employees who work at offices in Saitama Prefecture. The following is an incomplete SQL description for creating a table with two columns of employee\_id and years.

```
SELECT employee_id, years FROM service_years WHERE
```

Solution: 

```
SELECT employee_id, years FROM service_years WHERE office_name IN (SELECT office_name FROM office WHERE location='Saitama')
```

**(A student problem with a different situation and failure solution)** The following figure presents a database for managing films. Some of the actors have appeared in multiple films.

| film    |       |          |      | actor    |      |               | appearance |          |      |
|---------|-------|----------|------|----------|------|---------------|------------|----------|------|
| film_id | title | director | year | actor_id | name | date_of_birth | film_id    | actor_id | part |

The following is an incomplete SQL description for creating a table of actors who have appeared in films released in the 1950s with two columns of names and date\_of\_birth.

```
SELECT names, date_of_birth FROM actors WHERE
```

Solution: 

```
SELECT names, date_of_birth FROM actors WHERE year BETWEEN 1950 AND 1959
```

**(A student problem with a similar situation and success solution)** The following figure presents a database for managing the devices of a company.

| device        |             |                  | office    |             |          |
|---------------|-------------|------------------|-----------|-------------|----------|
| device_id     | device_name | date_of_purchase | office_id | office_name | location |
| status_of use |             |                  |           |             |          |
| office_id     | device_id   | years            |           |             |          |

You have to investigate the years of devices that are used in offices in Tokyo. The following is an incomplete SQL description for creating a table with two columns of device\_id and years.

```
SELECT device_id, years FROM status_of_use WHERE
```

Solution) 

```
SELECT device_id, years FROM status_of_use WHERE office_id IN (SELECT office_id FROM office WHERE location='Tokyo')
```