A System that Facilitates Diverse Thinking in Problem Posing

Kazuaki Kojima, *Faculty of Human Sciences, Waseda University, Saitama, Japan.*
koj@aoni.waseda.jp
Kazuhisa Miwa, *Graduate School of Information Science, Nagoya University, Nagoya, Japan.*
miwa@cog.human.nagoya-u.ac.jp

Abstract. Problem posing is identified as an important activity in mathematics education. In problem posing, it is important but difficult for learners to generate diverse problems. In this study, we examine a method and implement a system for facilitating learners' diverse thinking in the task of posing mathematical word problems. Since we focus on an aspect of problem posing as a creative generation task, we utilize the presentation of problems as cases in supporting problem posing. We experimentally investigate the diversifying effect of presenting cases while controlling similarities and then implement a support system for problem posing that can present various cases. We also conduct experimental evaluations to verify the effectiveness of our system.

Keywords. Creative Generation Support, Case Presentation , Problem Posing, Word Problems , Diverse Thinking

INTRODUCTION

In general mathematical learning, students solve many problems provided by a teacher or textbooks. But, besides problem solving, problem posing has also been identified as an important activity in mathematics education. In fact, some mathematicians and mathematics educators have pointed out that problem posing lies at the heart of mathematical activity (e.g., English, 1997b; Polya, 1945; Silver; 1994).

Problem solving and problem posing are not entirely different cognitive activities but are closely related. It has been experimentally confirmed that problem solving ability and problem posing performance are correlated, and that problem posing has a positive influence on problem solving (e.g., Ellerton, 1986; Nikata, & Shimada, 2005; Silver, & Cai, 1996). Of course, problem posing also has some aspects different from those of problem solving where students solve given problems with instructed solutions. Students have to engage in more productive and autonomous activities in problem posing because it is ill-structured and it requires production. That is expected to improve students' attitudes for mathematics. Moreover, to formulate reasonable mathematical problems, students must be aware of facts and their relationships embedded in problem situations. That is expected to promote students' understanding of problem solutions. Many benefits are gained from problem posing, such as enhancing problem solving ability and grasp of mathematical concepts, generating diverse and flexible thinking, alerting both teachers and students to misunderstandings, and improving students' attitudes and confidence in mathematics (English, 1997b; Silver, 1994).

It is frequently argued that there is a relationship between problem posing and creativity, because problem posing has an aspect of a creative generation task. It is not useful to repeatedly generate similar problems in learning by problem posing. In such a task, learners are required to generate diverse problems. However, it has been confirmed that problems generated by novice learners lack diversity (English, 1998; Mestre, 2002), so that diverse problem posing is considered a difficult task. Therefore, it is crucial to develop approaches to support learners in generating diverse problems.

In this study, we examine a method and implement a system for facilitating learners' diverse thinking in the task of posing mathematical word problems. Since our study focuses on an aspect of problem posing as a creative generation task, we utilize presentation of problems as cases in supporting problem posing. Case presentation is one of the major strategies used in creative generation support.

In this paper, we first describe the theoretical background of our study in Section 2. We then show an experimental investigation of the effects of case presentation in problem posing in Section 3. Section 4 demonstrates implementation of a supporting system for problem posing, and Section 5 demonstrates experimental evaluations of the system. We draw general conclusions about our study in Section 6.

THEORETICAL BACKGROUND

Problem Posing as a Creative Generation Task

Problem posing refers to both generation of new problems and reformulation of given problems (Silver, 1994). In the first case, problem posing is a divergent task that has multiple possible answers (posed problems). Therefore, problem posing is considered to be a creative generation task that requires productive thinking. In fact, problem posing is used in some creative thinking tests to measure fluency, flexibility and originality of individuals' thinking (Sternberg, 1999), and the relationship between problem posing and creativity is frequently argued (Leung, 1997a; Silver, 1994). However, the extent of the relationship is still unknown. Leung and Silver (1997b) measured problem posing performance, mathematical knowledge and verbal creativity of prospective elementary school teachers and found that problem posing performance was related to mathematical knowledge but not to verbal creativity. The problem posing performance measured by them focused on the quality and complexity as mathematical problems. Few studies have addressed the aspect of problem posing as a creative generation task.

In learning by problem posing, it is not useful for learners to repeatedly generate similar problems (Hirashima, Yokoyama, Okamoto, & Takeuchi, 2006). To promote diverse and flexible thinking, it is critical for learners to generate diverse problems. However, it has been confirmed that problems generated by novice learners lack diversity. English (1998) had children generate word problems in formal contexts (from numerical formulas) and informal contexts (from stories and photographs) in a problem-posing program, with the results indicating that children generated an overall limited range of problems in both formal and informal contexts. In Mestre's experiments (2002), undergraduates were asked to generate physics problems from given information on contexts and physical concepts. He confirmed that the undergraduates generated problems like those commonly used in textbooks and lectures because of their narrow associations between contexts and concepts. Therefore, it is difficult for novice learners to flexibly generate diverse problems, and it is necessary to support their diverse thinking in problem posing.

In this study, diverse thinking is defined as the ability to generate various ideas or products. This ability is closely related to flexibility (the variety of idea generation) measured by the creative thinking tests. Thus, diverse thinking is a part of creative thinking. As described above, the researches regarding education of problem posing require students to generate a broader range of problems, because it is critical to widely combine situations with mathematical concepts or solution methods. However, problem posing by students lack diversity (flexibility). Therefore, we believe that it is a meaningful challenge to approach facilitation of diverse thinking in problem posing. Here, the facilitation of diverse thinking denotes to promote generation of problems different from example problems learned in text books or lectures. Our purpose is to support learners in generating diverse problems by combining various situations and solutions, along with the focus on the issue that students' problem posing tends to conform to the example problems. We don't intend to promote their generation of actually original problems which no one has ever thought of. Our study aims to support students in general education. That is also a fundamental limitation of our study.

In the domain of creative generation, case presentation is one of the major strategies used in support systems. People often utilize past experiences or existing examples as cases in creative generation, such as in designing new products or forming concepts for new projects. Actually, creative generation support systems that construct and retrieve libraries of past products or that generate examples as hints to novel ideas have been proposed and implemented in a variety of tasks (e.g., Domeshek, Kolodner, & Zimring, 1994; Maher, & Garza, 1996; Orihara, 1994; Restrepo, & Christiaans, 2005; Young, 1987). Since our focus is on problem posing as a creative generation task, we utilize case presentation as an approach to supporting problem posing.

Surface and Structural Features of Problems

Generally, not every problem is acceptable in a problem-posing task. Problem posers have to appropriately combine problem contexts with key concepts and structures in solutions along with constraints and requirements in the task. Hence, the posers must be well versed in how solutions apply across a wide range of contexts (Mestre, 2002).

In analogy studies in cognitive psychology, two attributes of problems are recognized as crucial: one is surface features such as contextual settings in problem texts, and the other is structural features such as mathematical structures of solutions (Gentner, 1983; Forbus, Gentner, & Law, 1995; Holyoak, & Thagard, 1995). We refer to those two attributes as *situations* and *solutions*. The influence of those attributes on human problem solving has been experimentally discussed also in the domain of mathematical word problems (e.g., see Novik, 1988; Reed, Dempster, & Ettinger, 1985; Ross, 1987). Thus, it is considered important to understand and control those two attributes, situations and solutions, in presenting problems as cases in problem posing.

Comparison between problems can help derive their critical features. Gick and Holyoak (1983) drastically demonstrated that similarity judgement between problems facilitated the induction of schemata (general information about key elements and their relationships in the problems) and the solution of an analogous problem. According to this insight, English (1997a) presented children with arithmetic word problems controlling situations and solutions and had the children judge similarities between them. She reported that the children improved their understanding of the problems as a result of schemata induction. In problem posing, it is important to identify key elements and their relationships embedded in problems (English, 1997b, Leung et al., 1997b). Therefore, it is useful in supporting problem posing to present learners with problems controlling similarities in the two attributes, situations and solutions, so that they can compare the problems. We assume that such support enables learners to be aware of the control of the two attributes and facilitates their diverse problem posing.

According to the theoretical background described above, we experimentally investigated the effect of presenting problems as cases while controlling similarities in situations and solutions.

The Task Format

In problem posing, we can design various task formats by specifying constraints or materials given to posers. For example, English (1998) designed tasks where children posed problems that can be solved by specified numerical formulas and ones where they posed problems by using information in photographs they were shown.

In this study, we adopt a task where posers are asked to generate new problems from an example. In this task, posers are presented with an example problem and asked to generate problems that will be available in exercises for students who have understood the example. Because new problems cannot be produced by using only information included in the example, the posers need to generate novel ideas. The generation of new ideas comprises the recall of situations and the operation of the structures of solutions. Whether or not diverse problems are generated can be evaluated by judging similarities between the example and the problems generated.

In our problem-posing task, three different types of problems are defined.

- An *example problem*

  It is presented to a poser as a base at the start of the task. The example problem also indicates a problem domain in the task.

- *Cases*

  Cases are presented to the poser in addition to the example problems. The presentation of cases aims to support the poser. Those two types of problems are considered as input.

- *Posed problems*

  Posed problems are generated from the example problem by the poser. Posed problems are considered as output.

In the evaluation of posed problems, categories that indicate similarities in situations and solutions are used. Figure 1 shows the categories. Category-I / I indicates problems almost the same as an example problem, category-D / I indicates those generated by altering a situation of the example problem, category-I / D indicates those generated by altering a solution, and category-D / D indicates those generated by combining both alterations. All problems in category-I / I have the identical situation and solution. Problems that belong to categories other than category-I / I have a variety of situations and/or solutions. Our aim is to promote learners' generation of problems other than in category-I / I.

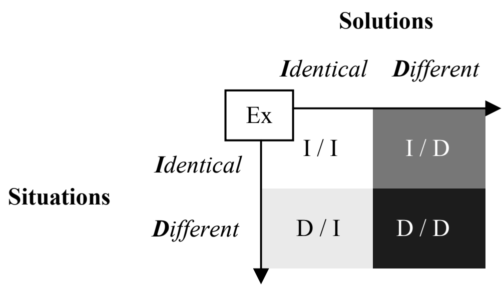

Fig.1. Categories for evaluating posed problems

In the current study, we select a problem domain of word problems solved by simultaneous equations. We use the following problem *Ex A* as an example problem.

Ex A. I bought some 60-yen oranges and 120-yen apples for 1020 yen. The total number of oranges and apples was 12. How many oranges and apples did I buy?
Solution.
Let x denote the number of oranges and y denote the number of apples.

$x+y=12$

$60x+120y=1020$

According to the equations above, $x=7$, $y=5$.

Figure 2 shows examples of problems in each category in the task where Ex A is adopted as an example problem. (Those problems in the figure are used as cases in our investigation described in the next session.)

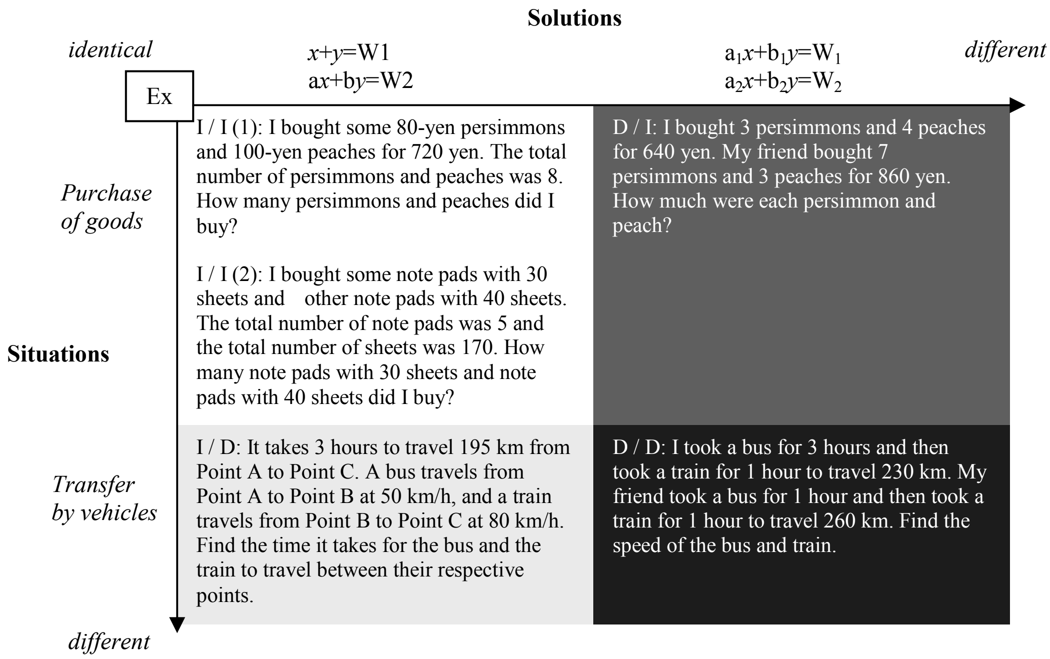

Fig. 2. Examples of cases for each condition group

EXPERIMENTAL INVESTIGATION

We experimentally investigated the effect of presenting cases in problem posing. In our investigation, participants were asked to generate new problems from Ex A. In the task of problem posing, they were presented cases that were systematically produced while controlling similarities in two attributes, situations and solutions. The participants' posed problems were categorized by judging similarities between them and Ex A. We verified the effects of case presentation based on the relation between the categories of the participants' posed problems and the cases.

Method

One hundred forty-five undergraduates participated in our investigation conducted in a class of psychology for design in which topics about creative problem generation were taught. The experimental procedures were as follows.

1. Presenting Ex A

   The participants were presented with Ex A and were asked to solve it. They were then shown the solution to Ex A.

2. Generating a problem (first)

   Every participant was asked to generate a problem (a posed problem) from Ex A. They were required to write the texts and provide the solutions to the posed problems on the sheets provided.

3. Presenting cases

   The participants were presented with three cases and were asked to solve one of them, specified by the experimenter. They were then shown the solution to the case.

4. Generating a problem (second)

   Every participant was asked to generate another problem from Ex A. In the second round of problem generation, the participants could refer to the cases presented in Procedure 3. However, they were not given any instructions encouraging them to use those cases in their generation.

The participants' posed problems were evaluated according to the categories shown in Figure 1.

Condition Groups

Each participant was randomly assigned to one of six condition groups. Every group was given a different set of cases in Procedure 3. The groups and the sets of cases for each group were as follows.

- I / I (1) case group: cases whose situations and solutions were identical to Ex A
- I / I (2) case group: cases whose situations and solutions were identical to Ex A but some of the parameters in the texts were different from Ex A
- D / I case group: cases whose situations were different from and whose solutions were identical to Ex A
- I / D case group: cases whose situations were identical to and whose solutions were different from Ex A
- D / D case group: cases whose situations and solutions were different from Ex A
- Control group: geometry problems that had no attributes similar to Ex A

Figure 2 shows examples of cases in each of the condition groups.

We verified the effects of case presentation based on a comparison between the participants' posed problems in the first and second rounds of problem generation. The effects of case presentation by controlling similarities in situations and solutions were studied based on the differences in the posed problems in the second round of problem generation. To confirm the influence of performing problem generation twice without having viewed any of the cases, we set up a control group in which the participants were given non-worded problems.

Results

In the following results, we have excluded 13 participants who were late for the class, 5 who could not complete the problem posing tasks, 12 who generated problems that were mathematically inappropriate or had inconsistent texts and solutions[1], and 3 who generated problems in other domains such as those including inequalities. The numbers of evaluated participants were 18 in the I / I (1) case

[1] In this case, the solution described by a participant was inconsistent with the problem text that they generated. Thus, the problem that the participant tried to generate was unclear.

group, 20 in the I / I (2) case group, 16 in the D / I case group, 23 in the I / D case group, 17 in the D / D case group and 18 in the control group.

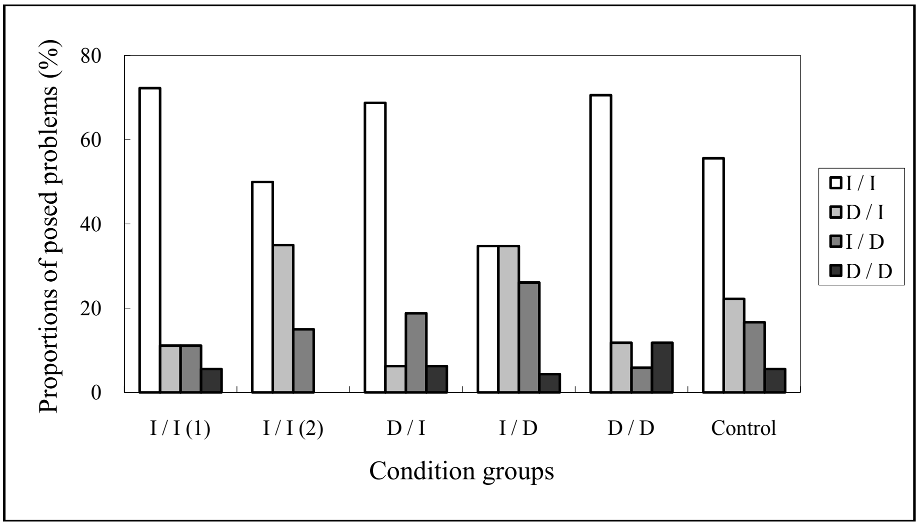

Fig. 3. Proportions of posed problems in each category (first round)

Figure 3 indicates the proportions of posed problems in each category in the first round of problem generation. As the figure indicates, more than half of the posed problems were in category-I / I; only in the I / D case group was the proportion of posed problems in category I / I less than 50% and equal to that in category D / I. Thus, overall results indicated that the majority of the participants generated problems whose situations and solutions were identical to Ex A when they did not see any cases.

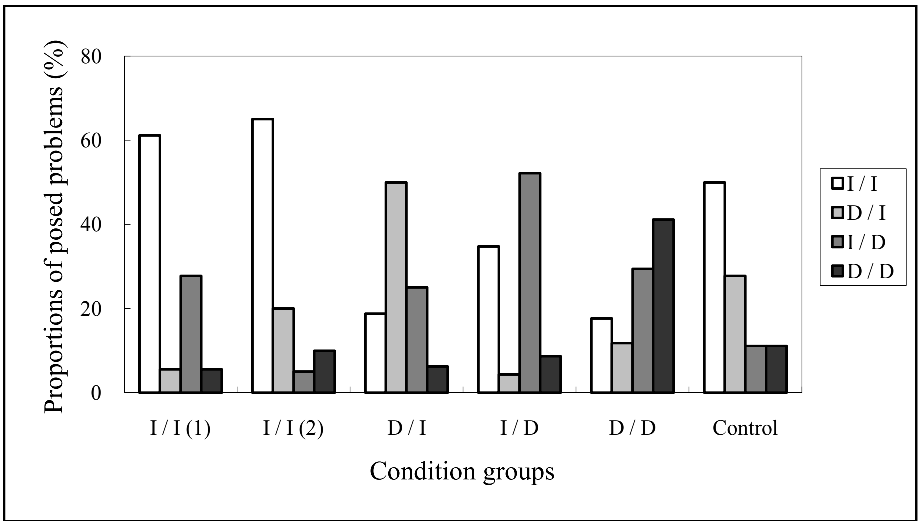

Fig. 4. Proportions of posed problems in each category (second round)

Figure 4 shows the proportions of posed problems in each category in the second round of generation where the participants had already examined cases. Similar to the first round, more than half of the posed problems were in category-I / I in the I / I (1) case group, the I / I (2) case group and the control group in the second round of problem generation. On the other hand, the proportions in the D / I case group, I / D case group and D / D case group in the second round were different from those in the first round. We examined differences of posed problems in each category in each group between the first and second rounds of generation by the chi-square test, with the results indicating that there were significant differences in the D / I case group ($\chi^2(3)=10.16$, $p<.05$) and the D / D case group ($\chi^2(3)= 10.84$, $p<.05$) and a moderate significant difference in the I / D case group ($\chi^2(3)=7.78$, $p<.10$). Furthermore, the results of residual analysis indicated that the number of posed problems in category-I / I in the first round and that in category-D / I in the second round were significantly high and in category-D / I in the first round and in category-I / I in the second round were significantly low in the D / I case group; the number of posed problems in category-D / I in the first round was significantly high and in the second round was significantly low in the I / D case group; and the number of posed problems in category-I / I in the first round was significantly high and in the second round was significantly low in the D / D case group. There were no differences in the I / I (1) case group ($\chi^2(3)= 1.79$, *n.s.*), I / I (2) case group ($\chi^2(3)= 4.21$, *n.s.*), and control group ($\chi^2(3)= 0.70$, *n.s.*).

These results indicate that case presentation actually influenced problem posing and that the presentation of different cases had different effects on problem posing.

Some examples of the participants' posed problems are shown in Appendix A.

Discussion

The results shown in Figures 3 and 4 confirmed that presentation of different cases had different effects on problem posing. In the D / I cases, I / D cases, and D / D case groups, posed problems in the categories the same as those of the presented cases in each group increased in the second round of generation. Thus, case presentation might have had an effect that allowed the participants to learn the relation between Ex A and the cases, even though they were not told to do so.

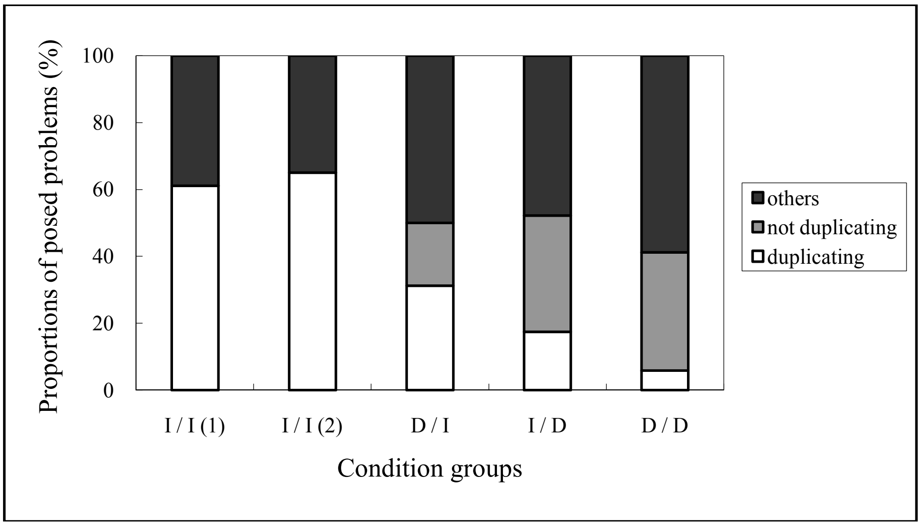

Fig. 5. Proportions of posed problems generated by duplicating cases

Or, they might have decided to simply generate problems similar to the cases by duplicating the cases. For example, in the D / I case group, a posed problem whose situation and solution were identical to any of the cases was evaluated as in category-D / I. We basically assume that posed problems in category-I / I were generated without adding any new ideas, and those in other categories were novel in some sense. However, if the participants' posed problems were duplications of the cases, the cases did not facilitate diverse problem posing because the participants presented no novel ideas in their posed problems. To verify this viewpoint, we compared the participants' posed problems to the cases. Figure 5 illustrates the proportions of posed problems generated by duplicating cases. In the figure, “duplicating” means posed problems whose situations and solutions were identical to the presented cases, and “not duplicating” means posed problems whose categories were identical to the cases, but whose situations and/or solutions were different from the cases. As the figure shows, some of the posed problems were mere duplications of the cases they had learned. However, participants in the D / I case, I / D case, and D / D case groups also generated problems that did not simply duplicate the cases, whereas the I/I group did not generate any non-duplicating cases.

The results of our investigation imply that presenting cases by controlling similarities in two attributes, situations and solutions, can facilitate diverse thinking in problem posing. Therefore, we propose a support system for problem posing that presents cases by controlling similarities. In the next section, we describe our system in detail.

IMPLEMENTATION OF THE SUPPORT SYSTEM FOR PROBLEM POSING

In this study, we propose a system that gives learners as users the problem-posing task described in Section 2 and assists them in generating diverse problems.

Some intelligent learning environments for problem posing were proposed and developed in studies by Hirashima and his colleagues. They had defined and implemented computational representations of problems to understand features of situations and solutions (Hirashima, Niitu, Hirose, Kashihara, & Toyoda, 1994; Hirashima, Umeda, & Takeuchi, 2001). Based on the representation, they developed systems that give tasks to generate problems solved by specified solutions (Nakano, Hirashima, & Takeuchi, 1999; 2002, Hirashima et al., 2006). Their studies mainly focused on promotion of learners' mastering solution methods, understanding of the relation between concepts and numerical relations, and integration of information in problem texts through posing word problems; and they experimentally confirmed that problem-posing activities by using their systems improved learners’ problem solving performance. However, learners can pose only a limited range of problems by using templates or cards of texts prepared in the systems because researchers simplified the tasks to suit elementary school children.

Our study focuses especially on the aspect of problem posing as a creative generation task in which the production of new and various problems is critical, and aims to facilitate learners’ diverse thinking. Therefore, we have to implement a system that can assist users as learners in generating diverse problems and that can accept and evaluate a broader range of problems that the users generate with few constraints.

Our system supports users by evaluating their posed problems based on the similarities in the two attributes, situations and solutions, and by presenting various cases controlling for the similarities. Thus, the support system needs a database of problems varying in situations and solutions. The support system mounts and uses a problem database produced by an automatic generation system for word problems that we had implemented in previous studies (Kojima, & Miwa, 2005; 2006). Figure 6 shows the relation between the generation and support systems. The generation system produces a problem database containing various problems (b in Figure 6) by propagating new problems from those initially stored in the system through interactions with a user as a teacher to acquire common knowledge (a). It provides learners with problems for learning by problem solving (c). On the other hand, the support system proposed in the current study does not have a function to produce new problems. The support system has functions to automatically understand posed problems that users input and to give feedback including case presentation to the users. The system supports the users’ learning by problem posing (d) by using those functions. The support system can present various cases by using the problem database produced by the generation system (e). We assume that configurations of the case presentation are set up by a teacher (f).

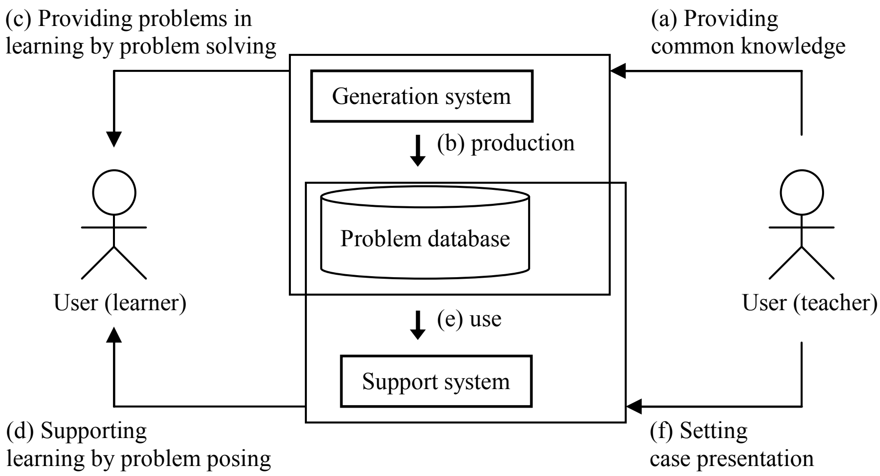

Fig. 6. Relation between generation and support systems

The problem domain that the systems deal with is word problems solved by simultaneous equations, as mentioned in Section 2. Although specific interfaces to accept problems are needed depending on domains, we believe that our approach and design of the support systems are feasible in various domains of problems that have texts denoting situations and solutions including equations or formulas.

The Automatic Generation System for Word Problems

Here we briefly explain the generation system. The system stores initial mathematical problems and generates new problems from them. First, it acquires problem data and common knowledge through interactions with a user, which are then used in problem generation.

Basic idea of problem generation

The generation system forms *episodes* and uses them to generate new problems from initial ones stored in the system. Each episode is knowledge comprising a single base example problem (*base*) and a single new analogical instance (*new instance*). In the generation system, each problem is regarded as a case because it is the system's product. Thus, an episode is regarded as a meta case where a new instance is generated from a base. In problem generation, a new problem (*output*) is generated from a given example problem (*input*) by mapping relationships between a base and a new instance in an episode.

Figure 7 shows our basic idea of problem generation by using an episode. In Figure 7, the vertical axis represents the features in situations and the horizontal axis represents the features in solutions. Each cell (e.g., A-1 and A-2) represents a category of problems that have the same situation and solution. Each category is called a *problem pattern*. Initially, the system has four problems in problem patterns A-1, B-1, A-2 and A-3. Suppose that a valid episode *E* is formed from a problem in A-1 as a base and a problem in B-1 as a new instance. If relationships in *E* can be adapted to a problem in A-2 as input, then a new problem in B-2 is generated as output. Therefore, problem generation by our system is regarded as the generation of output from input through the solving of the four-term analogy "base: new instance = input: output." In the same way, a problem in B-3 is generated from a problem in A-3 by using *E*. Our system generates problems in B-2 and B-3 that have different situations and the same solution; this means that the system adds two new problem patterns. With this basic idea, the generation system increases the number of problems and expands their variety by forming and using episodes.

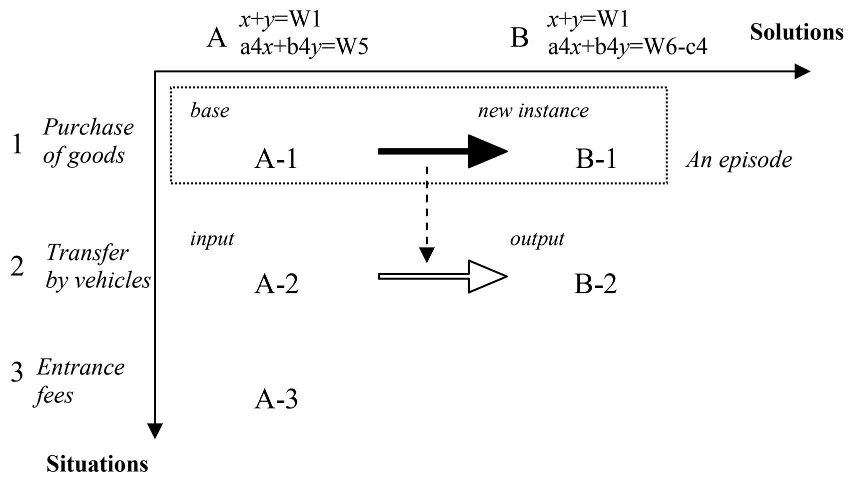

Fig. 7. Basic idea of problem generation

Problem representation

The generation system represents indexes of problem data in the form of a table. Table 1 shows an example of problem data. In this table, *numeric parameters* correspond to coefficients in the equations of the solution, *objects* indicate entity elements appearing in the problem text such as "pencils," and *properties* are attributes of the objects. Thus, relationships among objects, properties, and numeric parameters are indicated in the table. The first row in the table indicates answers (x and y), the second and third rows indicate coefficients included in the upper equation of the solution, and the forth and fifth rows indicate coefficients in the lower equation. In the table, the vertical relationships between numeric parameters correspond to multiplication (row 1 × row 2 = row 3 and row 1 × row 4 = row 5), and the horizontal relationships correspond to addition (column a + column b = column W). Numeric parameters are omitted when coefficients of x and y are one. Thus, the second and third rows in this table have only empty cells. Numeric parameters in the sixth row and column c don't appear in the equations. Relationships about such parameters are represented by indexes of "operations." The problem data includes other information, such as a *verb* denoting the situation, a label of the situation (corresponding to a superordinate of the verb), *equations* of the solution, and *operations* needed to evaluate numeric parameters included in the equations but not in the problem text.

The problem data also includes a *text template* that is formed from a problem text by replacing objects, properties and numeric parameters in the text with empty slots. The text template is used to create texts of new problems generated by the system.

Based on the representation, similarities in situations are identified by indexes of "situation" in the table such as in Table 1, and similarities in solutions are identified by "equations" and sets of "operations." For example, if a problem in Table 1 is compared to Ex A, it is identified as category-I / D because it has an identical situation (purchase) and a different solution (an operation is added).

Table. 1. Example of problem data

I bought some 80-yen pencils, 120-yen pens, and a 500-yen pencil box for 1,060 yen. The total number of pencils and pens was 6. How many pencils and pens did I buy?
Solution.
Let x denote the number of pencils and y denote the number of pens.

$x+y=6$

$80x+120y=560$

According to the equations above, $x=5$ and $y=3$

		objects			
		a: pencils	b: pens	W: (whole)	C: pencil box
	1: (how many)	x	y	6	
	2:				
properties	3:				
	4: yen	80	120		500
	5: yen			560	
	6: yen			1060	

verb	buy
situation	purchase
equations	$x+y$=W1 a4x+b4y=W5
operation	W5=W6-c4

Construction of the generation system

The generation system comprises four main components: an *input-analysis interface*, a *dictionary database*, a *casebase*, and a *production engine*. The input-analysis interface constructs problem data from problems input by a user. The dictionary database has a *conceptual dictionary* with a thesaurus and an *ontology database* that contains knowledge of words used in past word problems. The dictionary provides the word knowledge needed in problem generation. The casebase consists of two different databases a *problem database* and an *episode database* that stores knowledge used to generate new problems. The production engine generates new problems using the dictionary database and the casebase. Figure 8 shows the architecture of the generation system.

The system acquires word knowledge in the ontology database, and problem data in the problem database through interactions with a user. An increase of such knowledge means an increase in the number of episodes formed validly, words used as entity objects in problem texts, and text templates. Each text template in problem data is knowledge formed from a problem text, which is used to create texts of new problems generated by the system. The increase of knowledge thus expands the variety of problems generated by our system.

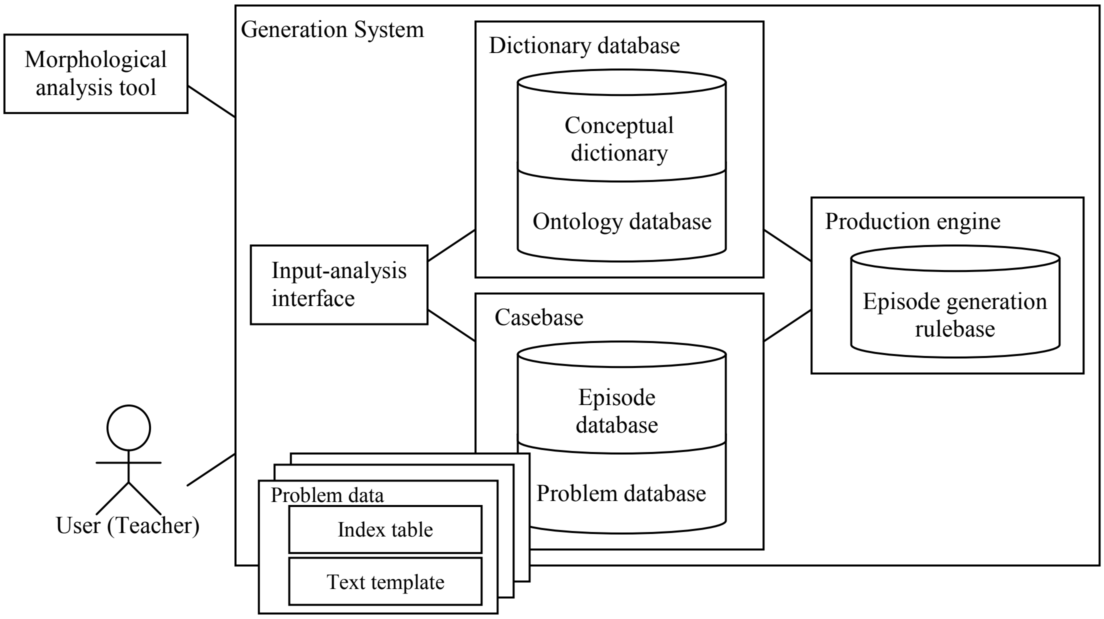

Fig. 8. Architecture of generation system

The procedures of the generation system

The summary of the generation system's problem generation through interactions with a user is as follows.

1. Storing problem data: In problem generation by the system, a user first stores problem data in the system. The user inputs a problem text and its solution in plain-text format. The *input analysis interface* then analyzes them and constructs problem data. After that, the system requires the user to add new information or modify the problem data that the system cannot understand correctly.
2. Forming episodes: Each episode is formed from two problems that a user selects as the base and the new instance from problems stored in the system. The production engine forms episodes by providing procedures called *altering actions* that describe which indexes of problem data are altered and how in problem generation based on comparisons between indexes of the base and the new instance.
3. Problem generation: The system generates new problems (output) from given problems (input) by using the episodes. The basic idea of problem generation has already been illustrated in Figure 6. The production engine generates output by adapting altering actions in each episode to a given input. The system presents output to a user and simultaneously requires the user to revise it, if needed. Since any problem can be generated through interactions with a user, every new generated problem is always evaluated and revised by the user.

For details on those procedures of the generation system, see (Kojima, & Miwa, 2005; 2006).

Construction of the Support System

The support system comprises two main components: a *problem-input interface* and a *feedback interface*. The problem-input interface analyzes posed problems input by a user, whereas the feedback interface presents cases while indicating similarities between Ex A and the posed problems. *Situation-estimating models* are used to estimate situations of the posed problems, and the *equation parser* is used to evaluate their solutions. The support system incorporates two components of the generation system, the dictionary database that provides the word knowledge needed in identifying words in texts of the posed problems and the casebase that stores problem data presented as cases. Figure 9 shows the architecture of the support system.

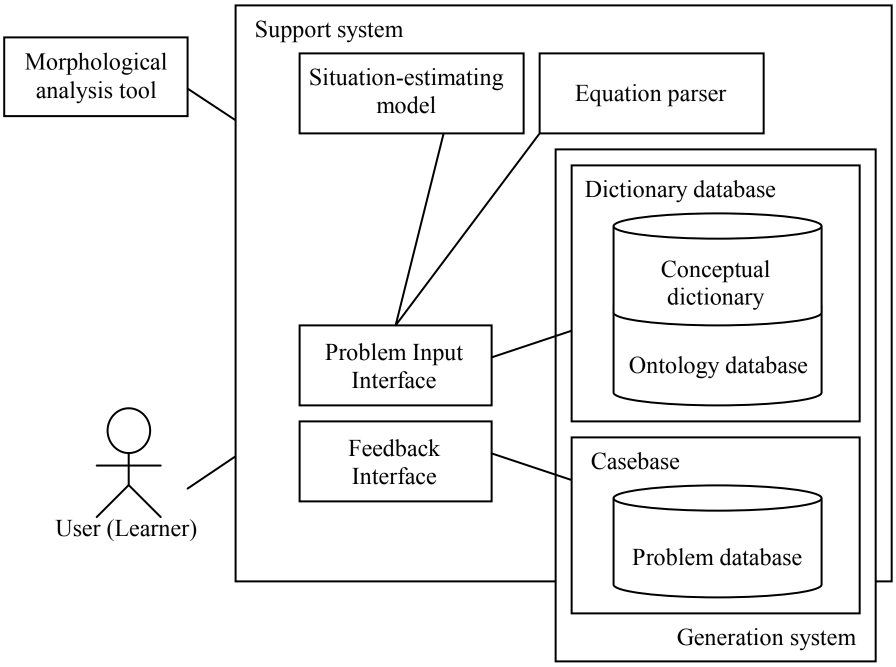

Fig. 9. Architecture of support system

Procedures of the Support System

Since the system requires a variety of problems to function properly, it is assumed that the system stores many problems propagated in advance by the generation system. This procedure corresponds to (b) in Figure 6.

Understanding of posed problems

In the support system, a user is first given Ex A and prompted to generate a new problem from it. The user generates and inputs a posed problem into the problem-input interface. In this phase, the problem-input interface in turn requires (1) objects to appear in the problem text (such as pencils), (2) numeric values to be included in the text for solving the posed problem, (3) equations for solving the posed problem, and (4) the problem text itself. Figure 10 shows a screenshot of the problem-input interface. After the user inputs the equations (3), the problem-input interface solves the equations by using the equation parser and requires the user to revise them if they are unsolvable[2]. When the user inputs the problem text (4), the problem-input interface lists necessary keywords and numeric values so that the

[2] Unsolvable equations have an arbitrary value as the answer or have no answer.

user can copy them into the problem text (5 in Figure 10). Such support in inputting posed problems aims to prevent any inappropriate problems from being accepted.

The problem-input interface understands a posed problem input by the user by representing it in the generation system's data format as shown in Table 1. “Equations” and “operations” in the table of the problem data are obtained by parsing equations input by the user, and a label of “situation” in the table is inferred from words in the text input by the user. Since the support system automatically constructs the data of the posed problem, the user doesn’t need to know the data representation. Values of "equations" in the table of the problem data are obtained by transforming equations input by the user into the standard form (ax+by=c). If operations to evaluate coefficients or constant terms in the standard form are needed in the transformation, the system records processes of the operations and then fills values of "operations" in the table. The table of the problem data is constructed based on the structures of the standard form.

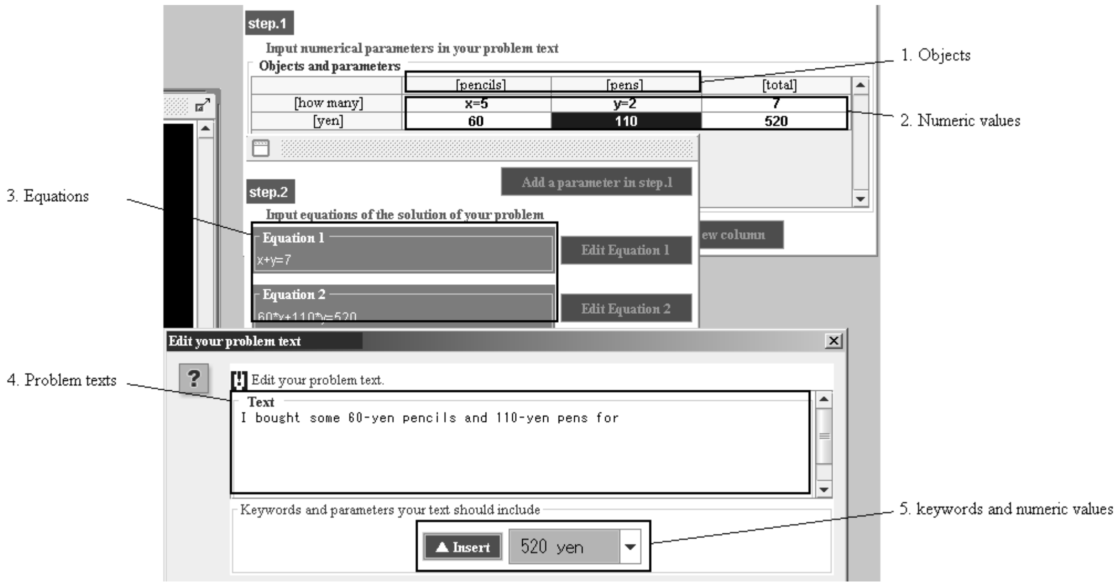

Fig. 10. Part of a screenshot of problem-input interface

To estimate a label denoting a situation, the problem-input interface uses the situation-estimating models, each of which is constructed from independent words in the texts of problems in the casebase comprising identical situations. To extract independent words from problem texts, the system uses *Chasen*, a morphological analysis tool for Japanese sentences. Chasen can separate words in sentences[3] and can specify features of the words, such as parts of speech and conjugations. A situation of the user's posed problem is inferred by evaluating similarity scores between the problem text and situations in the models. The similarity scores are evaluated by using a method to compute weight values, which is developed by transforming tf/idf.

In the situation-estimating models, each situation is represented as follows:

$$S_i = (m_{i1}, m_{i2}, \ldots, m_{iL}).$$

Here, L denotes the total number of words included in the situation-estimating models, and m_{ij} denotes the number of problems that have S_i as their situation and a word T_j in their texts. Between a posed problem P and S_i, their similarity score $sim(P, S_i)$ is computed by the following formula:

[3] In Japanese sentences, words are not separated from each other.

$$sim(P, S_i) = \sum_{j=1}^{L} w_{Pj} \times w_{ij}.$$

When T_j is included in the text of P, w_{Pj} is 1; otherwise, w_{Pj} is 0. w_{ij} denotes a weight value of T_j in S_i, and is given as follows:

$$w_{ij} = \frac{m_{ij}}{M} \times \log_{10} \frac{N}{n_j}.$$

N denotes the number of situations in the situation-estimating models, n_j denotes the number of situations where m_{ij} is more than 0 (j=1,2,...,N), and M denotes the number of problems that have S_i as their situation. The system evaluates similarity scores between a posed problem and situations in the situation-estimating models, and then estimates a situation whose similarity score takes the max value as the one for the posed problem.

However, the situation-estimating models can never identify novel situations that are not included in problems in the casebase. Thus, the support system can basically estimate only situations that are seen as typical and well-known problems; it supposes that other situations are novel in some way. The system shows the estimated situation to a user and requires the user to set up a correct situation if the estimated one is inappropriate or the system fails in estimation.

Feedback

After the support system understands a user's posed problem, it then gives feedback to the user. Figure 11 shows a screenshot of the feedback interface. The feedback interface indicates evaluation of the posed problem (1 in Figure 11), indicating similarities between Ex A and the posed problem, using the same representation as in Figure 1. Simultaneously, it retrieves and presents some cases (2 in Figure 11). Categories of cases are also indicated, as well as that of the posed problem. The feedback aims to promote the user's understanding of the relation among Ex A, the posed problem, and cases. The user can generate another problem after reading through the cases.

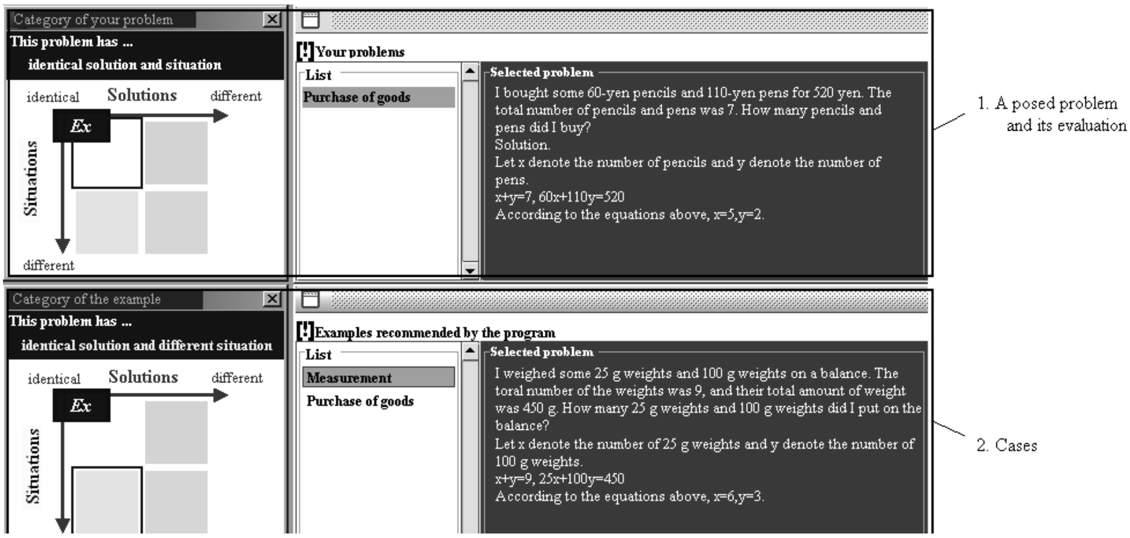

Fig. 11. Part of a screenshot of feedback interface

The support system can present various cases by controlling similarities in situations and solutions, such as presenting cases whose solutions are identical to and whose situations are different from the user's posed problem. The case presentation can be arbitrarily controlled by changing configurations of the system. It is configured by specifying the following three conditions.

1. An antecedent

   This specifies a condition for presenting some case type. Actually, it is configured by selecting one from five candidates such as "if a posed problem is in category-I / I", "if category-D / I", "if category-I / D", "if category-D / D", or "always."

2. A base problem

   This specifies a problem used to judge categories of problems in the problem database, one of which is presented as a case. Actually, it is either of the example problem or a posed problem input by a user.

3. A category of a presented case

   This specifies a category of a problem presented as a case. Actually, it is selected from category-I / I, category-D / I, category-I / D, or category-D / D.

We assume that a teacher sets up the case presentation (f in Figure 6).

As described above, the support system repeatedly evaluates either the posed problems devised by the user or the example problem, and presents cases based on the evaluations. We believe that problem posing by the users with case presentation by the support system facilitates the users' diverse thinking and diversifies their problem posing.

EXPERIMENTAL EVALUATIONS

The support system was evaluated from the following viewpoints: (1) whether it can understand users' posed problems (applicability), and (2) whether it can facilitate users' diverse problem posing (effectiveness). We also examined the relation between the control of case presentation and effectiveness.

METHOD

Prior to the evaluations, we had the generation system produce a problem database from 24 initial problems obtained from three general workbooks. The first author interacted with the system as a teacher. The produced database contained 75 problems and 12 situations.

Sixty-eight undergraduates participated in the evaluations individually. Prior to starting experimental tasks, we gave the participants an instruction as follows.

> "Today, we'd like you to be a test user for our software program to make mathematical word problems. We need a number and a variety of problems in mathematics learning. Your task is to pose problems that will be available in the learning."

Each participant worked in a small room. The experimental procedures were as follows.

1. Pre-test

   The participants were presented with *Ex B*, which is a word problem solved by a unitary equation, as an example problem.

Ex B. I give candy to some children. If I give 5 pieces of candy to each child, then I have 3 pieces left. If I give 6 pieces to each child, then I need 5 pieces more. How many children are there?
Solution.
Let x denote the number of children.

$$5x+3=6x-5$$

According to the equation above, $x=8$

Every participant was then asked to generate two problems from Ex B.

2. Problem posing while using the support system

   Every participant was asked to generate two problems from Ex A with the support system. Prior to the generation, he or she learned to operate the system by inputting Ex A into the system.

3. Post-test

   Every participant was asked to generate two more problems from Ex B.

4. Questionnaires

   Every participant was asked to evaluate how difficult operation of the system and problem posing were on a five-point scale where 1 denotes difficult and 5 denotes easy.

We verified the effectiveness of problem posing using the support system based on a comparison between the participants' posed problems in the pre- and post-tests. In the pre- and post-tests, we adopted Ex B that belonged to a domain different from the one in problem posing with the support system (Ex A). Therefore, problem posing by merely duplicating cases presented by the system was prevented in the post-test. The participants needed to transfer learning from problem posing with the system to the post-test. That enabled us to verify learning effect given by the system.

The participants' posed problems were evaluated according to the categories shown in Figure 1.

Condition Groups

We hypothesized that if participants were presented similar cases, then their diverse thinking was blocked and their posed problems were fixed. We also hypothesized that if participants were presented cases different in some way, then their diverse thinking was facilitated. According to those hypotheses, we designed three condition groups by controlling the support system's feedback in step 2 of the procedure. Each participant was randomly assigned to one of those condition groups. The configurations in each group feedback were as follows.

1. Control (no case) group

   The support system gave no feedback to participants.

2. Convergent case group

   The support system indicated a category for each posed problem and presented a case in the category identical to that of the posed problem. Thus, the system's feedback always agreed with the participants' problem posing.

3. Diverse case group

   The support system indicated a category for each posed problem and presented a case in a category different from that of the posed problem. More precisely, it presented a case in category-D / I when the posed problem was in category-I / I or category-I / D, and one in category-I / D when in category-D / I or category-D / D. The presented cases were always different from Ex A in either situations or solutions. Thus, the system's feedback was diversified. Although case presentation by the system could be controlled by various ways, we could not test every possible case presentation because of limitations in conducting the experiment. First of all, we decided to test the case presentation showing cases that were different from users' posed problems, and that were different in either of situations or solutions from the example problem.

Results

Applicability

In step 2 of the procedure, 133 posed problems were input into the system by 68 participants. Eleven (8.3%) of the posed problems had unknown situations not included in the system's problem database, which were impossible for the system to understand.
The other 122 problems had situations included in the problem database. The system succeeded in identifying 105 (86.1%) of them and failed on 17 (13.9%). The failures occurred due to errors in estimation of situations. It never failed in extraction of features from solutions. Figure 12 shows the result of understanding participants' posed problems.

In cases where the system could not understand posed problems and failed, the participants input appropriate situations according to instructions by the first author. Hence, every participant could correctly receive feedback from the support system.

Fig. 12. Result of understanding participants' posed problems

Effectiveness

In the following results, we excluded one participant who could not generate a problem at all in the pre-test. The numbers of evaluated participants were 22 in the control group, 22 in the convergent case group, and 23 in diverse case group. None of the posed problems was mathematically inappropriate or in other domains.

Figure 13 indicates the proportions of posed problems in each category in the pre-test. As the figure indicates, more than half of the posed problems were in category-I / I in every group. Thus, overall results indicate that the majority of the participants generated problems whose situations and solutions were identical to Ex B as an initial problem when they did not use the support system.

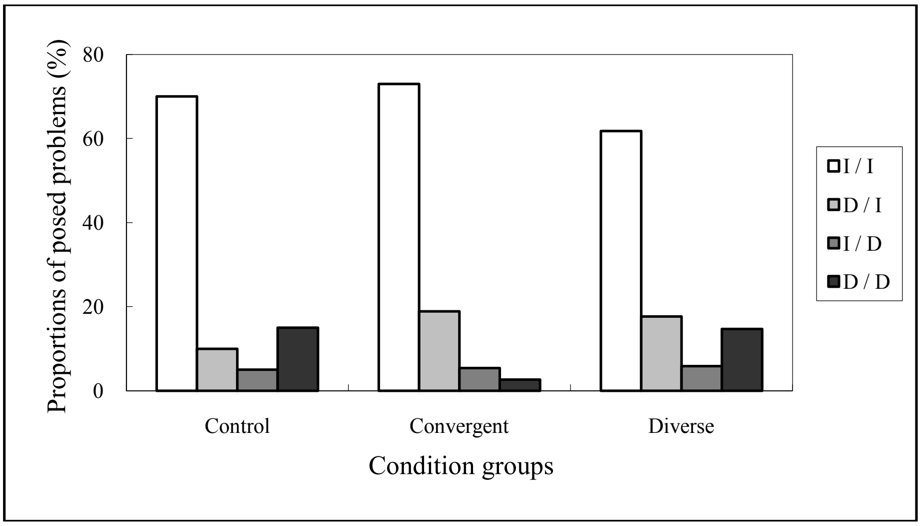

Fig. 13. Proportions of posed problems in each category (pre-test)

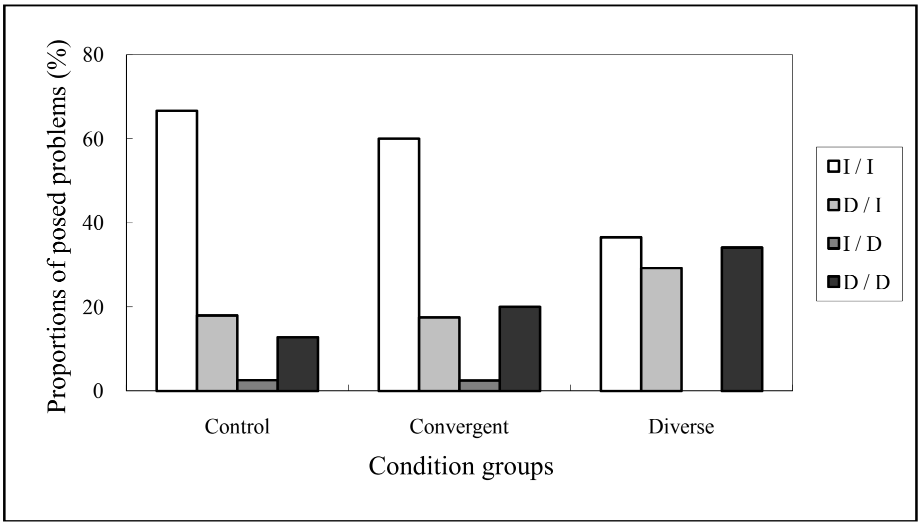

Fig.14. Proportions of posed problems in each category (post-test)

Figure 14 indicates the proportions of posed problems in each category in the post-test where the participants had already used the system. Similar to the pre-test, more than half were in category-I / I problems in the control group and convergent case group in the post-test. On the other hand, the proportions in the diverse case group in the post-test were different from those in the pre-test. We examined differences of posed problems in each category in each group between the pre- and post-tests by the chi-square test, with the results indicating that there was a significant difference in the diverse case group ($\chi^2(3)= 8.685$, $p<.05$). Furthermore, the results of residual analysis indicated that the number of posed problems in category-I / I in the pre-test was significantly high and in the post-test significantly low in the diverse case group. There were no differences in the control group ($\chi^2(3)$= 1.304, n.s.) and convergent case group ($\chi^2(3)$= 5.846, n.s.). These results indicate that use of the support system influenced problem posing in the divergent case group in the post-test. In the post-test, no posed problems were duplication of cases presented by the system. That was because the domains adopted in the post-test and problem posing using the system were not identical.

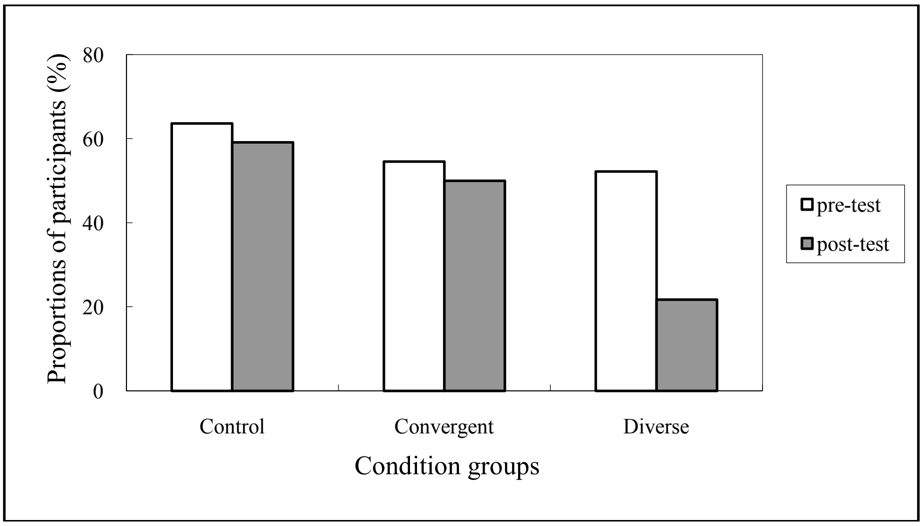

Fig. 15. Proportions of participants who only generated problems in category-I / I

Figure 15 indicates the proportions of participants who generated only problems in category-I / I[4]. As the figure indicates, more than half of the participants in every group generated only problems in category-I / I in the pre-tests. In the post-test, the number of such participants decreased only in the diverse case group. We examined differences in the numbers of participants in each group between in the pre- and post-tests by Fisher's one-way exact test, with the results indicating that there was a significant difference only in the diverse case group (p=.033). These results also indicate the effect of using the support system. Some examples of the participants' posed problems are shown in Appendix B.

Discussion

Applicability

In the evaluations, the support system used a problem database produced from initial problems obtained from general workbooks by the generation system. It is assumed that the set of initial problems contained only typical situations. Thus, the system could correctly understand only problems that had the typical situations. However, there were a few posed problems that had no typical situations. Although the system failed in understanding some typical problems, we confirmed that the system is relatively feasible even when a small set of problems is installed.

[4] Each of such participants generated two category-I / I problems, or generated only one category-I / I problem. Although each participant was asked to generate two problems in each task, some could generate only one.

Effectiveness

The results shown in Figures 13, 14, and 15 indicated that use of the support system influenced on problem posing only in the diverse case group. Therefore, we confirmed that the system can facilitate users' diverse problem posing.

Problems in category-D / D increased in the diverse case group in the post-test, although participants were not presented any cases in category-D / D. This is an important point as evidence that the participants learned variation of problem posing. On the other hand, no problems in category-I / D were generated. We haven't found any reasonable explanation for this fact. One possible explanation may be that generation of problems in category-I / D is more difficult than in other categories in the domain of the pre and post-tests. We believe that these results successfully indicated evidence supporting the utility of our system, even though it still has considerable limitations.

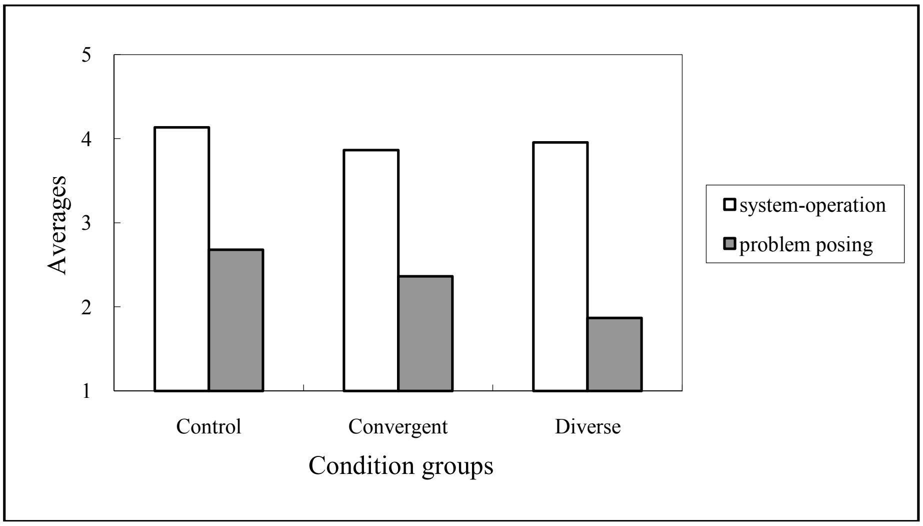

Fig. 16. Averages of participants' responses to two questionnaires

In the post-test, posed problems in the diverse case group were different from those in other groups. Responses to the questionnaires in step 4 of the procedure were also different among the groups. Figure 16 shows the averages of participants' responses to the questionnaires. “System-operation” was estimated as easy and did not differ over groups. On the other hand, “problem posing” was estimated as more difficult in the diverse case group than in other groups. We conducted 1 (each questionnaire) by 3 (condition groups) ANOVA tests, with the results indicating that there was a moderate significant difference in the averages of problem posing ($F(2,66)=3.01$, $p<.10$) but no difference in that of system-operation ($F(2,66)=0.54$, *n.s.*). The results of multiple comparisons indicate that the average of problem posing in the diverse case group was significantly lower than that in the control group. Thus, participants in the diverse case group may have tried difficult problem posing because the support system facilitated their diverse thinking.

The support system did not influence problem posing by participants in the control and convergent case groups, although they also used it. The convergent case group received feedback from the system. Thus, the feedback might have had no influence on, or it might have blocked their problem posing. Smith et al. (1993) experimentally demonstrated the conformity effects of examples that introducing similar examples can limit human creative generation. If the results in the convergent case group can be interpreted by the conformity effect, then the results of the diverse case group would demonstrate that reference to examples that have different features is effective in creative generation.

The participants in the diverse case group learned to pose different types of problems by understanding the relationships between the example problem and the cases. However, there is an alternative explanation for the effect of seeing cases. The participants did not learn but they were already able to pose diverse problems from the beginning; only the diverse case group were aware that they were expected to pose the types of problems same to the presented cases. The participants in the control and convergent case groups might not simply have realized that they had to pose problems different from the example problem. We haven't obtained any evidence to verify whether the participants could not diversify their problem posing or they did not merely notice the necessity to diversify. In the post-test, more than half of participants asked questions about problem posing in all groups, such as "Should I make problems different from previous ones?" The experimenter simply answered "yes." We had not expected that we received such questions many times. They may have felt it unnatural to generate same problems repeatedly. However, not all participant who asked such questions successfully generated problems different from the example problem. Although that is informal evidence, the participants must have thought that they were required to pose various problems. We have to address the consciousness of the necessity for diversifying ideas in further studies.

According to the results described above, we have verified the effectiveness of the support system. We also found that the system must appropriately present cases to facilitate users' diverse thinking.

CONCLUSIONS

In this study, we examined a method and implemented a system for facilitating learners' diverse thinking in the task of posing mathematical word problems. Since we focused on an aspect of problem posing as a creative generation task, we utilized presentation of cases in supporting problem posing. We experimentally confirmed the diversifying effect of presenting cases controlling similarities in situations and solutions, and then implemented a support system for problem posing that presents cases by controlling for the similarities. We conducted experimental evaluations of the system and verified its effectiveness.

One important future work is to improve the estimation of situations in understanding posed problems by the system. The support system estimates the situations by using a statistical method and problems produced by the generation system. Therefore, the system often fails in understanding problems whose texts contain words uncommonly used in textbooks, even if their situations are typical. We have to propose complementary methods to reinforce the estimation.

Another important work is further study of supporting problem posing. Although we confirmed that the support system can diversify learners' problem posing, we still have some work to do. For example, none of participants in the diverse case group generated problems in category-I / D in the post-test, even though they generated more diverse problems in the post-test than in the pre-test. In the domain of Ex B, generation of problems in category-I / D may be more difficult than in other categories. It would be more desirable for them to generate problems in all categories with complete control. In the current study, we focused on the control of case presentation and verified its effect. We have to study further approach to support users, while focussing on the users' learning from cases. We need to pursue more effective support for problem posing as a task for generating learners' creative thinking.

REFERENCES

Domeshek, E. A., Kolodner, J. L., & Zimring, C. (1994). The Design of a Tool Kit for Case-Based Design Aids. In J. S. Gero & F. Sudweeks (Eds.) *Proceedings of International Conference on Artificial Intelligence in Design* (pp. 109-126). MA: Kluwer Academic Pub.

Ellerton, N. F. (1986). Children's Made up Mathematics Problems: A New Perspective on Talented Mathematicians. *Educational Studies in Mathematics*, 17, 261-271.

English, L. D. (1997a). Children's Reasoning Processes in Classifying and Solving Computational Word Problems. In L. D. English (Ed.) *Mathematical Reasoning: Analogies, Metaphors, and Images* (pp. 191-220). NJ: Lawrence Erlbaum Associates.

English, L. D. (1997b). Promoting a Problem-posing Classroom. *Teaching Children Mathematics*, 4, 172-179.

English, L. D. (1998). Children's Problem Posing within Formal and Informal Contexts. *Journal for Research in Mathematics Education*, 29, 83-106.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A Model of Similarity-Based Retrieval. *Cognitive Science*, 19, 141-205.

Gentner, D. (1983). Structure-mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7, 155-170.

Gick, M. L., & Holyoak, K. J. (1983). Schema Induction and Analogical Transfer. *Cognitive Psychology*, 15, 1-38.

Hirashima, T., Niitu, T., Hirose, K., Kashihara, A., & Toyoda, J. (1994). An Indexing Framework for Adaptive Arrangement of Mechanics Problems for ITS, *IEICE Transactions on Information and Systems*, E77-D, 19-26.

Hirashima, T., Umeda, T., & Takeuchi, A (2001). An Intelligent Problem-Sharing Environment on Web. In *Proceedings of International Conference on Computers in Education* (pp. 449-456), International Conference on Computers in Education 2001, Soul, Korea.

Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi, A (2006). Interactive Learning Environment by Posing Arithmetical Word Problems as Sentence-Integration. In *ICCE2006 Workshop Proceedings of Problem-Authoring, -Generation and -Posing in a Computer-Based Learning Environment* (pp. 1-8), International Conference on Computers in Education 2006, Beijing, China.

Holyoak, K. J., & Thagard, P. (1995) *Mental Leaps: Analogy in Creative Thought*. Cambridge, MA: MIT Press.

Kojima, K., & Miwa, K. (2005). A System that Generates Word Problems Using Problem Generation Episodes. In C-K Looi, D. Jonassen & M. Ikeda (Eds.) *Proceedings of International Conference on Computers in Education* (pp. 193-200). Amsterdam : IOS Press.

Kojima, K., & Miwa, K. (2006). Evaluation of a System that Generates Word Problems through Interactions with a User. In M. Ikeda, K. D. Ashley & T-W Chan (Eds.) *Proceedings of Intelligent Tutoring Systems* (pp. 124-133). Springer-Verlag.

Nakano, A., Hirashima, T., & Takeuchi, A. (1999). Problem-Making Practice to Master Solution-Methods in Intelligent Learning Environment, In *Proceedings of International Conference on Computers in Education* (pp. 891-898), International Conference on Computers in Education 1999, Chiba, Japan.

Nakano, A., Hirashima, T., & Takeuchi, A. (2002). An Evaluation of Intelligent Learning Environment for Problem Posing, In S. A. Cerri, G. Gouarderes & F. Paraguacu (Eds.) *Proceedings of Intelligent Tutoring Systems* (pp. 124-133). Springer-Verlag.

Novick, L. R. (1988). Analogical Transfer, Problem Similarity, and Expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14, 510-520.

Leung, S. S. (1997a) On the Role of Creative Thinking in Problem Posing. *International Reviews on Mathematical Education*, 29, 81-85.

Leung, S. S., & Silver, E. A. (1997b). The Role of Task Format, Mathematics Knowledge, and Creative Thinking on the Arithmetic Problem Posing of Prospective Elementary School Teachers. *Mathematics Education Research Journal*, 9, 5-24.

Maher, M. L., & Garza, A. G. (1996). Developing Case-Based Reasoning for Structural Design. *IEEE Expert*, 11, 42-52.

Mestre, J. P. (2002). Probing Adults' Conceptual Understanding and Transfer of Learning via Problem Posing. *Journal of Applied Developmental Psychology*, 23, 9-50.

Nikata, K., & Shimada, H. (2005). Facilitation of Analogical Transfer by Posing an Analogous Problem for Oneself. *Japanese Journal of Educational Psychology*, 53, 381-392.
Orihara, R. (1994). A System for Creativity Support "Chie-no-izumi(R)". *Transactions of Japanese Society for Artificial Intelligence*, 9, 248-257.
Polya, G. (1945). *How to Solve it*. Princeton University Press.
Reed, S. K., Dempster, A., & Ettinger, M. (1985). Usefulness of Analogous Solutions for Solving Algebra Word Problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11, 106-125.
Restrepo, J., & Christiaans, H. (2005). From Function to Context to Form: Precedents and Focus Shifts in the Form Creation Process. In *Proceedings of Creativity and Cognition* (pp. 195-204), Creativity and Cognition 2005, London, U.K.
Ross, B. H. (1987). This is Like that: The Use of Earlier Problems and the Separation of Similarity Effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13, 629-639.
Silver, E. A. (1994). On Mathematical Problem Posing. *For the Learning of Mathematics*, 14, 19-28.
Silver, E. A., & Cai, J (1996). An Analysis of Arithmetic Problem Posing by Middle School Students. *Journal for Research in Mathematics Education*, 27, 521-539.
Smith, S. M., Ward, T. B., & Schumacher. J. S. (1993). Constraining Effects of Examples in a Creative Generation Task. *Memory and Cognition*, 21, 837-845.
Sternberg, R. J. & O'hara, L. A. (1999). Creativity and Intelligence. In R. J. Sternberg (Ed.) Handbook *of Creativity* (pp.251-272). Cambridge University Press.
Young, L. (1987). The Metaphor Machine: A Database Method for Creativity Support. *Decision Support Systems*, 3, 309-317.

APPENDIX A: EXAMPLES OF POSED PROBLEMS IN EXPERIMENTAL INVESTIGATIONS

I / I (1) *case* group in the second round
I bought 180-yen red flowers and 210-yen white flowers for 1380 yen. The total number of red flowers and white flowers was 7. How many red flowers and white flowers did I buy?
Solution.
Let x denote the number of red flowers and y denote the number of white flowers.

$$x+y=7$$
$$180x+210y=1380$$

According to the equations above, $x=3$, $y=4$.
(In category-I / I)

D / I *case* group in the second round
I mixed bottles of 150-g milk and bottles of 200-g coffee to produce 1500-g cafe au lait. The total number of bottles was 8. How many bottles of milk and coffee did I prepare?
Solution.
Let x denote the number of bottles of milk and y denote the number of bottles of coffee.

$$x+y=8$$
$$150x+200y=1500$$

According to the equations above, $x=2$, $y=6$.
(In category-D / I)

I / D *case* group in the second round
I always buy 6 cakes and 6 puddings for 2400 yen. Today, I bought 3 more cakes than usual because cakes were discounted 30%. I bought them for 2475 yen. Find the usual prices of a cake and a pudding.
Solution.
Let x denote the price of a cake and y denote the price of a pudding.

$6x+6y=2400$

$(6+3)x\times(100-30)\div100+6y=2475$

According to the equations above, $x=250$, $y=150$.
(In category-I / D)

D / D *case* group in the second round
Mr. B went to play billiards with his friends. There were 6 persons. Playing billiards cost 350 yen every 30 minutes. Some of them had discount coupons. If one had the coupon, she or he could play for 250 yen every 30 minutes. Each coupon was valid for one person. They played 3 hours and paid 10,800 yen. Find the number of discount coupons they had.
Solution.
Let x denote the number of persons who did not have the coupon and let y denote the number of persons who did have it.

$x+y=6$

$(350x+250y)\times(60\div30)=10800$

According to the equations above, $x=3$, $y=3$.
(In category-D / D)

APPENDIX B: EXAMPLES OF POSED PROBLEMS IN EXPERIMENTAL EVALUATIONS

Control group in the post-test
I deal cards to play a game. If I give 5 cards to each person, then I have 3 cards left. If I give 6 cards to each person, then I need 5 more cards. How many persons are there?
Solution.
Let x denote the number of persons.

$10x+3=11x-2$

According to the equations above, $x=5$
(In category-I / I)

Diverse *case* group in the post-test
I put water in a bowl by using a glass. If I put water in the bowl from the glass 7 times, then I need 10 ml more to fill the bowl. If I put water 9 times, then 130 ml overflows. Find the volume of the glass.
Solution.
Let x denote the volume of the glass.

$7x+10=9x-130$

According to the equations above, $x=70$
(In category-D / I)

My father is 6 years older than 3 times my age. He is as old as 6 times my sister's age. She is 5 years younger than I am. Find my age.
Solution.
Let x denote your age.

$3x+6=6(x-5)$

According to the equations above, $x=12$
(In category-D / D)

Convergent *case* group in the pre-test.

I give 80-yen notebooks to some children. If I give 5 notebooks to each child, then I have 100 yen left. If I give 7 notebooks to each child, then I need 700 yen more. How many children are there?

Solution.

Let x denote the number of children.

$$(5\times100)x+100=(7\times100)x-700$$

According to the equations above, x=4

(In category-I / D)