Title
Can an AI System Facilitate Human Creative Generation? An Experimental Investigation in Mathematical Problem Posing

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 29(29)

ISSN
1069-7977

Authors
Kojima, Kazuaki
Miwa, Kazuhisa

Publication Date
2007

Peer reviewed

Can an AI System Facilitate Human Creative Generation? An Experimental Investigation in Mathematical Problem Posing

Kazuaki Kojima (koj@aoni.waseda.jp)
Faculty of Human Sciences, Waseda University
2-579-15, Mikajima, Tokorozawa-City, Japan
Kazuhisa Miwa (miwa@cog.human.nagoya-u.ac.jp)
Graduate School of Information Science, Nagoya University
Furo-cho, Tikusa-ku, Nagoya-City, Japan

Abstract

In the domain of creative generation, case presentation is one of the major strategies used in AI support systems. However, it has been experimentally indicated that introducing cases can limit human creative generation. Therefore, it is important to investigate whether or not an AI support system that uses cases can actually facilitates human creative generation. In this study, we implemented an AI support system for creative generation that presents cases by controlling similarities in a task domain of posing mathematical word problems. We then experimentally investigated the effects of case presentation by our system, with the results indicating that our system can facilitate human creative generation when it presents diverse cases. We conclude that AI support systems must appropriately control cases.

Keywords: an AI system; creative generation; problem posing; mathematical word problems; case presentation

Introduction

In the domain of creative generation, case presentation is one of the major strategies used in AI 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 for a variety of tasks (e.g., Domeshek & Kolodner, 1994; Restrepo & Christiaans, 2005; Young 1987).

However, introducing cases can limit human creative generation. It has been experimentally indicated that human creative generation tends to conform to examples given prior to starting the generation tasks (Smith, Ward & Schumacher, 1993). That is referred to as conformity effects of examples. Such constraining effects of examples are somewhat undesirable, because the essential factor in creative generation is the production of novel output. On the other hand, it is also pointed out that imitation of examples can relax mental constraints in creative generation and facilitate the creativity of generated works. Okada and Ishibashi's experiment revealed that copying an example work facilitated copiers' creativity in a task domain of artistic drawing (Okada & Ishibashi, 2003). Based on the two viewpoints, our research question emerges: can an AI support system that uses cases actually facilitate human creative generation?

In this study, we implemented an AI system that supports human creative generation by presenting cases. We then experimentally investigated the effects of case presentation by our system on human creative generation.

In the current study, we selected a task domain of posing mathematical word problems. Problem posing is considered to be a creative generation task that requires productive thinking, and the relationship between problem posing and creativity is frequently argued (Leung, 1997; Silver, 1994). Problem posing by learners is identified as an important activity in mathematics education. Many benefits are gained from learning by problem posing, such as enhancing problem solving ability, generating diverse and flexible thinking, and improving students' attitudes and confidence in mathematics (English, 1997b; Silver, 1994).

Mathematical word problems have two attributes that are essential in problem solving; surface problem *situations* denoting contextual settings expressed in texts such as "*purchase of goods*" and "*transfer by vehicles*," and the mathematical structures of *solutions*. It has been experimentally discussed in cognitive psychology that human problem solving is influenced by similarities in these two attributes (e.g., Novick, 1988; Reed, Dempster & Ettinger, 1985; Ross, 1987). In the context of mathematics education, it has been recognized as an important issue to present students with various problems while controlling the similarities (English, 1997a). Moreover, in the context of problem-posing education, it has been pointed out as important but also difficult to generate diverse problems in both *situations* and *solutions* (English, 1998). Therefore, we propose an AI support system for problem posing that presents problems as cases by controlling the similarities in *situations* and *solutions*. Prior to the implementation of the system, we experimentally confirmed that such case presentation can affect human problem posing (Kojima & Miwa, 2006b).

AI Support System for Problem Posing

In this study, we implemented a system that gives learners as users a problem posing task where the learners generate new problems (*posed problems*) from a given problem as an example (an *example problem*).

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 users' diverse thinking. Our system supports users by evaluating problems posed by them based on similarities in *situations* and *solutions*, and by presenting various problems (*cases*) controlling the similarities. Thus, it needs a database that has a variety of problems 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; 2006a). Figure 1 shows the relation between the generation and support systems. The generation system is also an AI system, which produces a problem database containing various problems (c in Figure 1) 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 (b). 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. It 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).

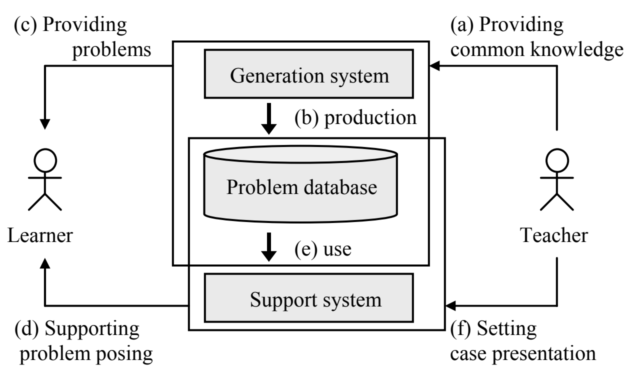

Figure 1: Relation between generation and support systems

Construction of the Support System

Our 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 evaluation of the *posed problems*. The system incorporates two components of the generation system, a dictionary database that provides the word knowledge needed in identifying words in texts of the *posed problems* and a problem database that stores problem data presented as *cases*.

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 1.

Understanding of *posed problems*

In the support system, a user is first given an *example problem* 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 2 shows a screenshot of the *problem-input interface*. In step 3, the problem-input interface solves the equations and requires the user to revise them if they are unsolvable. In step 4, the *problem-input interface* lists necessary keywords and numeric values so that the user can copy them into the problem text (5 in Figure 2). Such support in inputting *posed problems* aims to prevent any inappropriate problems from being accepted.

The user's *posed problem* is represented in the generation system's data format, where the *solution* is represented in operational procedures needed to evaluate the answer and the *situation* is represented by a label estimated from the words in the problem text, such as "*purchase*." Since the support system automatically constructs the data of the *posed problem*, the user doesn't need to know the data representation.

To estimate labels denoting *situations*, our system uses *situation-estimating models*, each of which is constructed from independent words in the texts of problems in the problem database comprising identical *situation*. However, the *situation-estimating models* can never identify novel *situations* that are not included in problems in the problem database. Thus, our 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 3 shows a screenshot of the *feedback interface*.

The *feedback interface* indicates evaluation of the *posed problem* (1 in Figure 3). In the evaluation of *posed problems*, categories that indicate similarities in *situations* and *solutions* are used. Figure 4 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.

The support system can present various *cases* by controlling similarities in *situations* and *solutions* (2 in Figure 3), 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. We assume that a teacher sets up the *case* presentation (f in Figure 1).

As described above, the support system repeatedly evaluates *posed problems* devised by users 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.

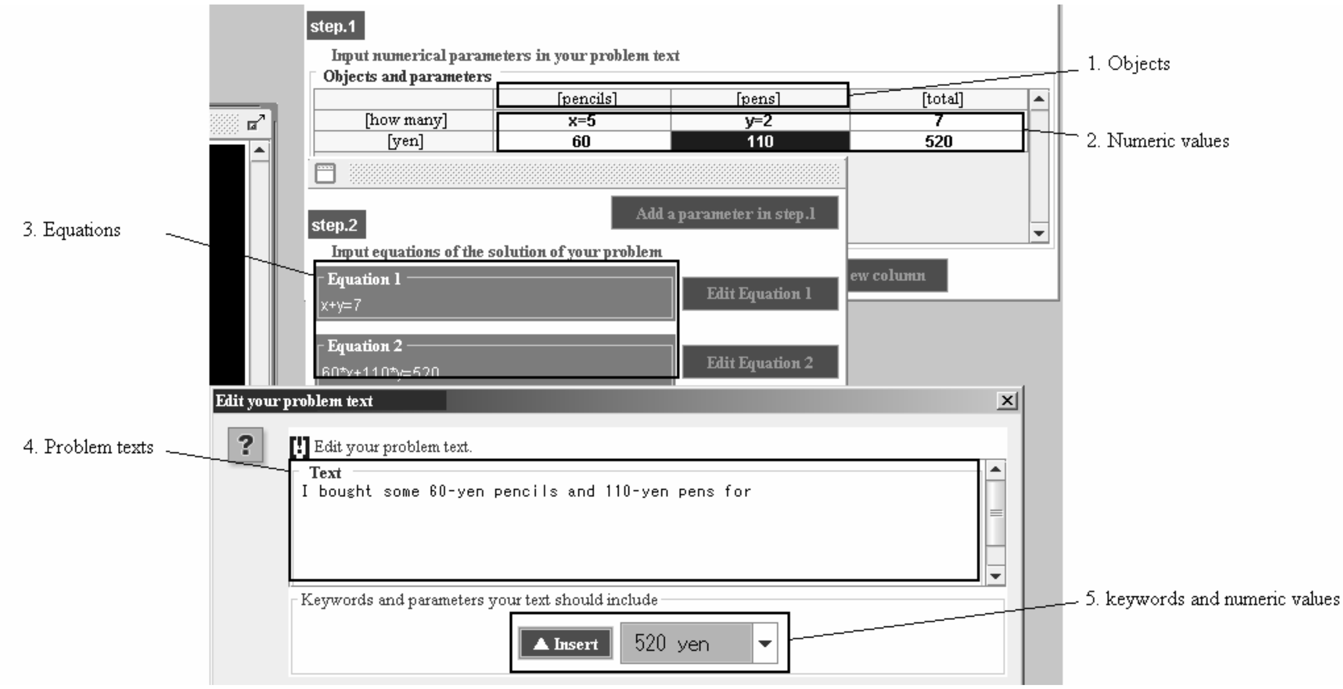

Figure 2: Part of a screenshot of the *problem-input interface*

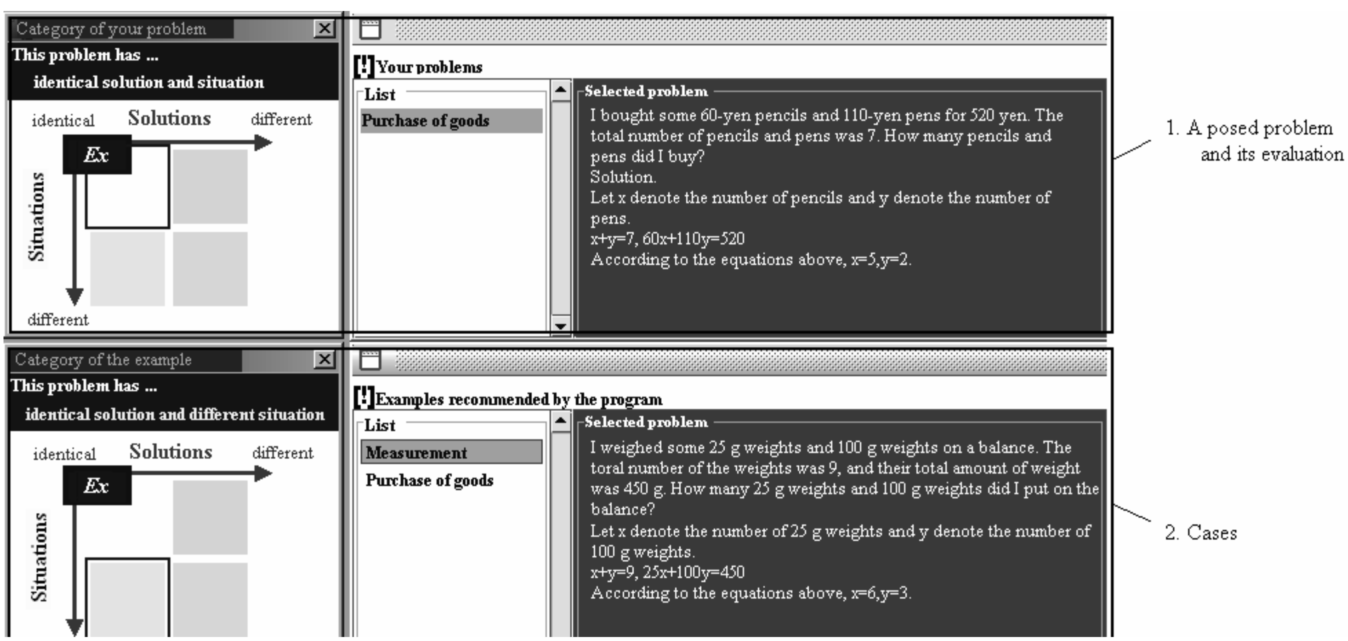

Figure 3: Part of a screenshot of the *feedback interface*

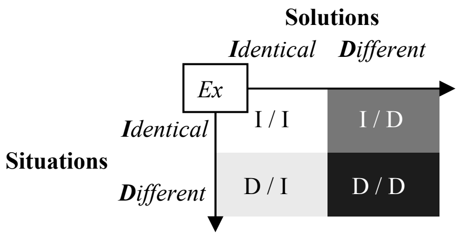

Figure 4: Categories for evaluating *posed problems*

Experimental Investigation

We experimentally investigated whether our support system can facilitate users' diverse problem posing. We also examined the relation between the control of *case* presentation and its effects on users' problem posing.

Method

Sixty-eight undergraduates participated in our investigation individually or in groups of two or three. 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 with the support system. In Procedure 2, *Ex A* was used 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, $x=7$, $y=5$.
   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.

The participants' *posed problems* were evaluated according to the categories shown in Figure 4. We verified the effects of problem posing using the support system based on a comparison between the participants' *posed problems* in the pre- and post-tests.

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 Procedure 2. 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.

Results

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 5 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 almost all participants generated problems whose *situations* and *solutions* were identical to *Ex B* as an initial problem when they did not use the support system.

Figure 6 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 in 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 ($\chi2(3)=8.685$, $p<.05$). 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 ($\chi2(3)=1.304$, n.s.) and convergent *case* group ($\chi2(3)=5.846$, n.s.). These results indicate that use of the support system influenced problem posing in the divergent *case* group in the post-test.

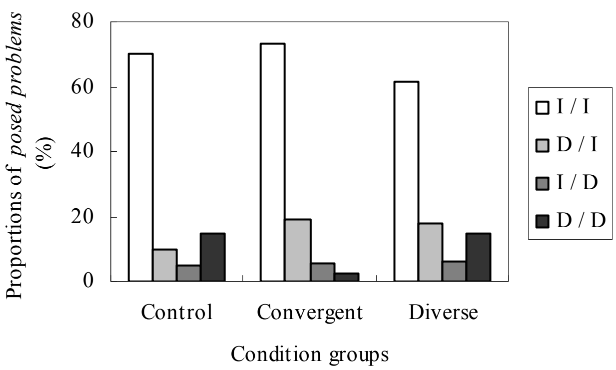

Figure 5: Proportions of *posed problems* in each category (pre-test)

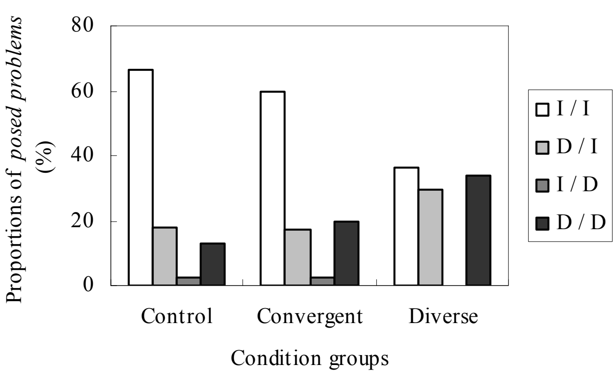

Figure 6: Proportions of *posed problems* in each category (post-test)

Figure 7 indicates the proportions of participants who generated only problems in category-I / I[1]. 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.

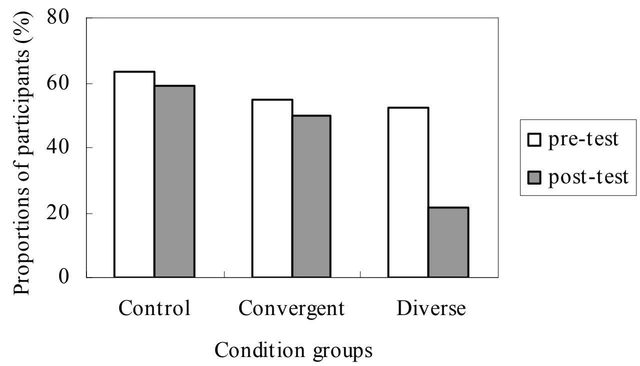

Figure 7: Proportions of participants who only generated problems in category-I / I

Discussion and Conclusions

The results shown in Figures 5, 6, and 7 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.

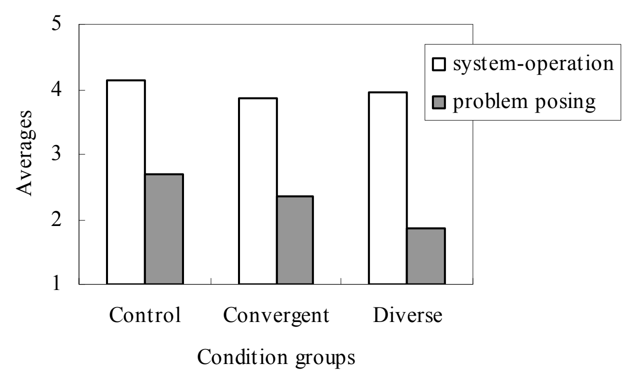

Figure 8: Averages of participants' responses to two questionnaires

In the post-test, *posed problems* in the diverse *case* group were different from those in other groups. Responses to the questionnaires in Procedure 4 were also different among the groups. Figure 8 shows the averages of participants' responses to the questionnaires. "System-operation" was

[1] 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.

estimated as easy and did not differ in every group. 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. In the latter case, problem posing in the convergent *case* group may have been constrained by *cases* presented by the system, as creative generation by subjects who were shown examples in the experiment in Smith et al. (1993) was. If the results in the convergent *case* group can be interpreted by the conformity effects, then the results of the diverse *case* group would demonstrate that reference to examples that have different features is effective in creative generation.

According to the results described above, we have verified the effectiveness of the support system. We also demonstrated that AI support systems must appropriately present *cases* to facilitate users' creative generation.

References

Domeshek, E. A., Kolodner, J. L., & Zimring, C. (1994). The design of a tool kit for case-based design aids. *Proceedings of International Conference on Artificial Intelligence in Design* (pp. 109-126). Norwell, MA: Kluwer Academic Pub.

English, L. D. (1997a). Children's reasoning processes in classifying and solving computational word problems. In English, L. D. (Ed.), *Mathematical Reasoning: Analogies, Metaphors, and Images* (pp. 191-220). Mahwah, 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.

Kojima, K., & Miwa, K. (2005). A system that generates word problems using problem generation episodes. *Proceedings of International Conference on Computers in Education* (pp. 193-200). Amsterdam, The Netherlands: IOS Press.

Kojima, K., & Miwa, K. (2006a). Evaluation of a system that generates word problems through interactions with a user. *Proceedings of Intelligent Tutoring Systems* (pp. 124-133). Berlin, Germany: Springer-Verlag.

Kojima, K., & Miwa, K. (2006b). Experimental investigation and implementation of support for problem generation by presenting cases. *Proceedings of International Conference on Computers in Education* (pp. 123-126). IOS Press.

Leung, S. S. (1997) On the role of creative thinking in problem posing. *International Reviews on Mathematical Education*, 29, 81-85.

Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14, 510-520.

Okada, T., & Ishibashi, K. (2003). Creativity by copying: How examples inspire innovation. *Proceedings of International Symposium on Cognitive Approaches to Creative Processes, SIGLAL 2003-1* (pp. 36-40).

Reed, S. K., Dempster, A., & Ettinger, M. (1985). Usefulness of analogous solutions for solving algebra word poblems. *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. *Proceedings of Creativity and Cognition* (pp. 195-204).

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.

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.

Young, L. (1987). The metaphor machine: A database method for creativity support. *Decision Support Systems*, 3, 309-317.