A Cognitive Simulator for Tutoring Causal Relation between Mental Operations and Behavior

Kazuhisa Miwa and Daisuke Nakazawa ({miwa, nakazawa}@is.nagoya-u.ac.jp)
Graduate School of Information Science, Nagoya University
Nagoya, 464-8601 Japan

Hitomi Saito (hsaito@auecc.aichi-edu.ac.jp)
Programs in Education for Information Sciences, Aichi University of Education
Kariya, 448-8542 Japan

Abstract

Problem solving is understood as a process through which states of problem solving are transferred from the initial state to the goal state by applying adequate operators. Within this framework, knowledge and strategies are given as operators for the search. One of the most important points of researchers' interest in the domain of problem solving is to explain the performance of problem solving behavior based on the knowledge and strategies that the problem solver has. We call the interplay between problem solvers' knowledge/strategies and their behavior the causal relation between mental operations and behavior. It is crucially important, we believe, for novice learners in this domain to understand the causal relation between mental operations and behavior. Based on this insight, we have constructed a learning system in which learners can control mental operations of a computational agent that solves a task, such as knowledge, heuristics, and cognitive capacity, and can observe its behavior. We also introduce this system to a university class, and discuss which findings were discovered by the participants.

Introduction

In the traditional theory of human problem solving, human behavior in solving a problem has been regarded as a search for a problem space. In this perspective, problem solving is understood as a process through which states of problem solving are transferred from the initial state to the goal state by applying adequate operators.

The utility of this theoretical framework, especially in the early stages of problem solving studies, was supported by many empirical studies in which simple puzzle-like experimental tasks were used. Following after these studies, this view was expanded into a more general framework, and has taken a central role in providing fundamental principles for explaining complex human cognition. For example, there are some representative studies such as: a series of Klahr's studies where scientific discovery is regarded as a search for dual problem spaces: rule and data spaces (Klahr & Dunbar, 1988, 2000, Kulkarni & Simon, 1988); studies on insight problem solving where insight is brought about by switching a problem space searched (Kaplan & Simon, 1990); studies on diagrammatic problem solving where the superiority of using diagrams has been very often analyzed from the viewpoint of the efficiency of the problem space search (Larkin & Simon, 1987); and studies on distributed cognition, where the efficiency of distributed cognition is explained based on the merging of multiple problem spaces represented in the internal and external world (Zhang & Norman, 1994).

Within this framework, knowledge and strategies are given as operators for the search. One of the most important points of researchers' interest in the domain of problem solving is to explain the performance of problem solving behavior based on the knowledge and strategies that the problem solver has. In the production system, the most popular computer architecture used in studies on human problem solving, operators are represented as production rules (e.g., Klahr, Langley, & Neches, 1987). There have been many trials in which the difference of rules implemented in a cognitive model tries to explain a variety of human problem solving behavior. In one landmark book, Human Problem Solving, published in 1972, Newell and Simon indicated that complex human behavior can be generated by repeated applications of a relatively small amount of simple operators, and a wide variety of behavior can be successfully explained based on the addition and elimination of a very small number of specific operators (Newell & Simon, 1972). These findings strongly influenced following studies on human problem solving. In this paper, we call the interplay between problem solvers' knowledge/strategies and their behavior the causal relation between mental operations and behavior.

Many introductory textbooks on cognitive science and cognitive psychology introduce the theoretical framework of problem solving mentioned above. It is crucially important, we believe, for novice learners in this domain to understand the causal relation between mental operations and behavior. Based on this insight, we have constructed a learning system in which learners can control mental operations of a computational agent that solves a task, such as knowledge, heuristics, and cognitive capacity, and can observe its behavior. The users are expected to notice many aspects of the causal relation between mental operations and behavior while simulating the human problem solving process using our system. We also introduce this system to a university class, and discuss which findings were discovered by the participants.

Model and System

Our system functions as a hypothesis-deduction system where it infers what behavior emerges when a set of mental operations are assumed. Learners repeatedly observe sets of an assumption of operations - deduction - behavior while being supported by the system's deduction function. Learners are guided to understand heuristically the causal relation between mental operators and behavior by using this system. Figure 1 shows a schematic illustration of the overall framework employed by our system.

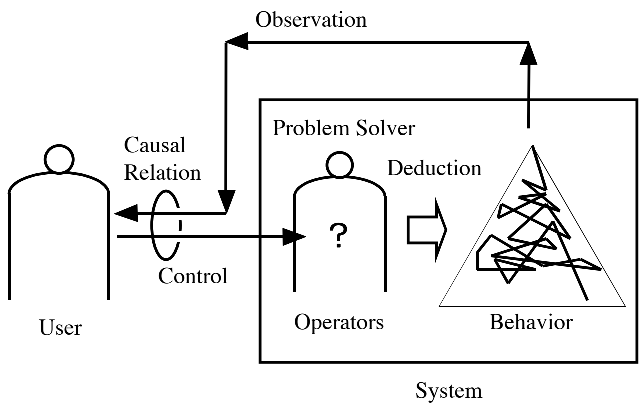

Figure 1 Basic framework employed by our system.

Task

We used the cryptarithmetic task as an experimental task. The following is an example problem.

```
  DONALD     D=5
 +GERALD
  ROBERT
```

The problem is to assign digits (0, 1, 2, …, 9) to the letters (A, B, D, E, G, L, N, O, R, and T), so that when the letters are replaced by their corresponding digits, the sum is satisfied. Here the information D=5 is given in the initial statement of the problem.

The reasons for using this task are: (1) the cryptarithmetic task has a characteristic that problem solving behavior is remarkably influenced by changes of mental operations able to be used, (2) a computational cognitive model (problem solver) solving this task has been already proposed, and (3) reliable psychological data on processes of solving this task have been provided (Newell & Simon, 1972).

Summary of the model's behavior

Figure 2 shows an example behavior of the problem solver. The behavior of the problem solver can be organized as Episodes, each of which comprises a sequence of inferences. An episode starts either with assigning one of the digits to one of the letters, or with selecting a column to be processed. For example, the first episode, Episode 1, begins with an assignment, in which the digit 5 is assigned to the letter D. (This assignment is a special case because the information D=5 is given by the experimenter.) The experimenter does not give the problem solver any further information, so it tries to test assignments systematically, which sometimes produces trial-and-error behavior. Another way an episode can begin is with selecting a column. For example, the fifth column, O+E=O, can be worked on independently without the other columns. In this example, the problem solver selects this column in the initial stage of Episode 2, and can directly infer from it E=0 or E=9, without any arbitrary assignments or other information obtained from previous episodes. Either type of episode continues until all obvious information has been inferred based on the assignment of digits to letters or the selection and examination of a column. After an episode ends, the next episode begins with another assignment or column selection. A detailed description of the model can be seen in Miwa & Simon (1993) and Miwa (1999).

Episode 1 D□ 5 $\xrightarrow{\text{Da1}}$ T=0, Carry2=1 $\xrightarrow{\text{Pb1}}$ R=odd $\xrightarrow{\text{Ia2}}$ R>4, G<5 $\xrightarrow{\text{CDa}}$ R=5 or 7 or 9 $\xrightarrow{\text{CDd}}$ R=7 or 9

Episode 2 $\xrightarrow{\text{Dc1}}$ E=0 or 9 $\xrightarrow{\text{CDc}}$ E=9 $\xrightarrow{\text{CDb}}$ R=7 $\xrightarrow{\text{Dc2}}$ Carry5=1, Carry6=1 $\xrightarrow{\text{Da1}}$ G=1, Carry7=0 $\xrightarrow{\text{Db2}}$ A=4, Carry3=1, Carry4=0 $\xrightarrow{\text{Db1}}$ L=8, Carry2=1 $\xrightarrow{\text{Ia1}}$ N>2, B<7 $\xrightarrow{\text{CDd}}$ N=3 or 6

Episode 3 N□ 6 $\xrightarrow{\text{Da1}}$ B=3, Carry5=1

Episode 4 O□ 2

Figure 2 Behavior of the complete model.

Process Column

The core operations for inferences in our problem solver are the Process Column (PC) operation and the Coordination (CD) operation. For example, in Figure 2, the inferences in Episode 1 develop as follows: (1) D=5, (2) T=0 & carry2=1 (the carry into the second column equals 1), (3) R=odd, (4) R>4 & G<5, (5) R=5, R=7, or R=9, and (6) R=7 or R=9.

After obtaining the initial assignment in (1), the following three inferences in (2), (3), and (4) are drawn by the PC operation. The PC operation infers new information from other assignments by processing each column. For example, in (2), T=0 & carry2=1 are obtained from D=5 by processing the first column; in (3), R=odd is obtained from the second column; and in (4), R>4 & G<5 are obtained from the sixth column.

Table 1 Core operations that the problem solver can use. (Carry means a carry at the processed column, and Carry+ means a carry into the left-side column.)

(a) PC operations

	Type of Information	Status of Column	Known Information	Inferred Information	An example: Known Information	An example: Inferred Information
Da1	Digit	A+B=C	Two letters and Carry	The other letter, Carry+	A=5, C=2, Carry=1	B=6, Carry+=1
Da2	Digit	A+B=C	Two letters	The other letter, Carry+	A=5, C=2	B=6 or 7, Carry+=1
Da3	Digit	A+B=C	Three letters	Carry, Carry+	A=5, B=6, C=2	Carry=1, Carry+=1
Db1	Digit	A+A=B	Bottom letter and Carry+	Top (= Middle) letter, Carry	B=2, Carry+=1	A=6, Carry=0
Db2	Digit	A+A=B	Bottom letter	Top (= Middle) letter, Carry	B=2	A=1 or 6, Carry=0
Dc1	Digit	A+B=A	Nothing	Middle letter		B=0 or 9
Dc2	Digit	A+B=A	Middle letter	Carry, Carry+	B=9	Carry=1, Carry+=1
Pa	Parity	A+B=C	Two letters and Carry	The other letter	A=odd, C=odd, Carry=0	B=even
Pb	Parity	A+A=B	Carry	Bottom letter	Carry=1	B=odd
Ia1	In-equality	A+B=C	Top (or Middle) letter, Carry, and Carry+	Middle (or Top) letter, Bottom letter	B=6, Carry=1, Carry+=0	C>6, A<3
Ia2	In-equality	A+B=C	Top (or Middle) letter, Carry+	Middle (or Top) letter, Bottom letter	B=6, Carry+=0	C>5, A<4
Ib	In-equality	A+A=B	Carry+	Top (= Middle) letter	Carry+=1	A>5

(b) CD operations

	Processing	New info.	Known info.	Inferred info.
CDa	If a new ambiguous assignment is inferred, and merging the assignment in an already-known ambiguous assignment restricts the ambiguity, Then do so.	A=3 or 5 A>5	A=5 or 6 A=even	A=5 A=6 or 8
CDb	If a decisive assignment is newly inferred, and the assignment is one of two alternatives already known. Then restrict the two alternatives to a decisive assignment.	A=6	B=6 or 8	B=8
CDc	If two alternatives are newly inferred, and one of the two alternatives is already assigned. Then restrict the two alternatives to a decisive assignment.	A=6 or 8	B=8	A=6
CDd	If a new ambiguous assignment is inferred, and merging the assignment in digits left over restricts the ambiguity. Then do so.	A=odd	Digits 3, 6, 8, 9 are left over.	A=3 or 9

In our problem solver, twelve different PC operations are installed. Seven operations infer digit information (three of the seven operations, Da1, Da2, and Da3, process a column whose arrangement of letters is A+B=C, two operations, Db1 and Db2, process a column whose arrangement of letters is A+A=B, and other two operators, Dc1 and Dc2, process a column whose arrangement of letters is A+B=A.) Two operations, Pa (used in A+B=C) and Pb (in A+A=B), infer parity information, and three operations, Ia1, Ia2 (in A+B=C), and Ib (in A+A=B), infer inequality information. Details of the functions of these operations are given in Table 1 (a).

Coordination

The other inferences in (5) and (6) are obtained by the CD operation. After obtaining new information through the PC operation, the problem solver tries to coordinate the new information with other information that has been already obtained. For example, after R>4 in (5) is obtained, the problem solver infers R=5, R=7, or R=9 by coordinating R>4, which is newly inferred, with R=odd which is already known. In (6), the problem solver also restricts this undecided information into R=7 or R=9 by checking all information already known and noticing that the digit 5 has already been used for the letter D.

Four kinds of CD operation, CDa, CDb, CDc, and CDd, are installed in our model. Details of the functions of these operations are presented in Table 1(b).

Interface of the learning system

The interface of our learning system that contains the problem solver explained above consists of two main windows: the operation window and the behavior window.

The operation window: In the operation window, users control the knowledge and heuristics of the problem solver. Each of the PC and CD operations is easily plugged in or removed from the problem solver by clicking a check box corresponding to each operation. Other factors that users can control are: (1) a strategy for selecting a column processed; (2) a strategy for selecting a letter when assigning a digit; (3) a strategy for selecting a digit for the assignment; and (4) memory capacities of the problem solver (i.e., to what degree the problem solver can activate information that has been already inferred from its memory).

The behavior window: In the behavior window, users can observe the problem solver's behavior when a certain set of operations is implemented. The behavior is displayed as inference sequences such as the ones shown in Figure 2.

Behavior

Figure 2 showed the behavior of the complete problem solver into which all sets of PC and CD operations are plugged in. In Figure 2, the types of operation used in each step of the inferences are concretely indicated. Figure 3 shows overall patterns of representative behaviors of other transformed problem solvers where some of the operations are removed. In Figure 3, the circle nodes indicate information drawn by the PC operations, while the square nodes indicate information by the CD operations. The letters indicated at the beginning of each episode denote letters to which digits were assigned in the trial-and-error manner.

The overall characteristics described above reproduced the results of hand-simulations conducted in Newell and Simon's Human Problem Solving. The most interesting finding is that the replacement of a very small amount of operations drastically influences the behavior. For example, as can be seen in Figure 3(d), the absence of a single operation, Dc1, greatly lengthened the problem solving path. The problem solver repeated the trial-and-error behavior until it reached the solution because of the lack of an ability to process the fifth column, O+E=O.

Additionally, the order of assignments was substantially varied in each case. When no more information can be drawn, the problem solver begins to move to trial-and-error behavior; i.e., the problem solver assigns a digit to each letter systematically. Basically, the order of assignments in Figure 3 followed the alphabetical order. However, to let the length of problem solving path minimum, a letter whose constraint of assignment is stronger is first tested. For example, in Figure 2, at the beginning of Episode 3, the letter N was first tested because in the previous episode the information N=3 or 6 was obtained. Many kinds of ambiguous information, such as

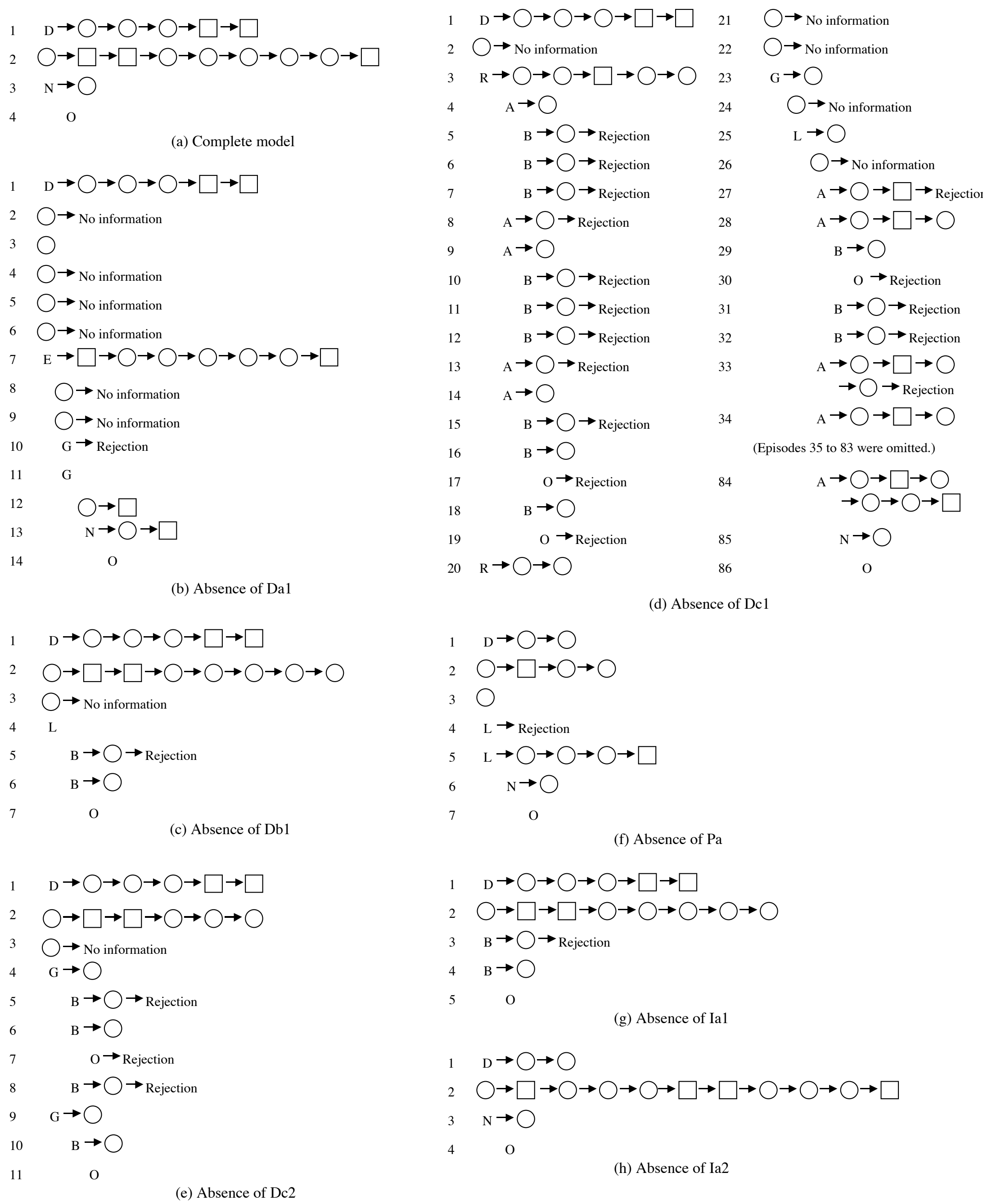

Figure 3 Representative patterns of the problem solver's behavior

parity and inequality information, generated in the process of problem solving, are drawn, and the intermediate information is determined by the operations the problem solver can use. This factor produces the variety of patterns of problem solving paths shown in Figure 3.

Evaluation

Procedure

To test the utility of this system, we actually had graduate students use it, and then evaluated the results. This experiment was conducted in a cognitive science course opened in the first author's university. The purposes of the experiment were: (1) to confirm whether or not the participants' understanding of the problem solver's operations and behavior deepens by using this system, and (2) to discuss what types of findings are discovered on the causal relation between mental operations and behavior.

Two class hours were assigned for evaluation. In the first class hour, the participants actually solved a cryptarithmetic task identical to the one the problem solver solved. They then learned the basic specifications of the problem solver, such as the basic flow of the model's behavior and the PC and CD operations that the participants manipulate. After instruction, the pre test was conducted. In the pre test, another type of cryptarithmetic task, CROSS+ROADS=DANGER: where S=3, was used. The pre test consisted of two categories of items. For the first category, after the participants were presented with assignments that had already been decided, they were required to identify which operation is used and what information is inferred. For the second category, the participants were required to indicate sequences of inferences with a form presented in Figure 2 from the initial state until the problem is solved.

The second class hour was conducted one week after the first class. The participants were required to trace the complete problem solver's behavior using this system for treatment. At every step of each problem solving sequence, they were first required to identify which operation is used and what information is inferred, then they confirmed whether their prediction was correct by having one step of inferences of the problem solver proceed. After this treatment, a post test was conducted. The post test was identical to the pre test.

After the post test was conducted, the participants were allowed to use the system as they like. They explored changes in the problem solver's behavior while manipulating the operations that the problem solver can use. While using the system, they were required to make notes of their findings on the causal relation between mental operations and behavior.

Result

Figure 4 shows a comparison of the results of the pre and post tests. The results of the nine participants who took part in both of the two class hours were analyzed. A 2 (pre/post tests) x 2 (categories of test) ANOVA revealed that a main effect of the pre/post tests reached significance ($p < 0.01$, $F(1, 32)=17.16$) and the interaction between the two factors was not significant ($F < 1$). This analysis shows that the participants learned to identify operations that should be used in a certain situation and to trace the problem solver's behavior more accurately. This result indicates that the participants successfully learned to understand the problem solver's operations and behavior more deeply through using this system.

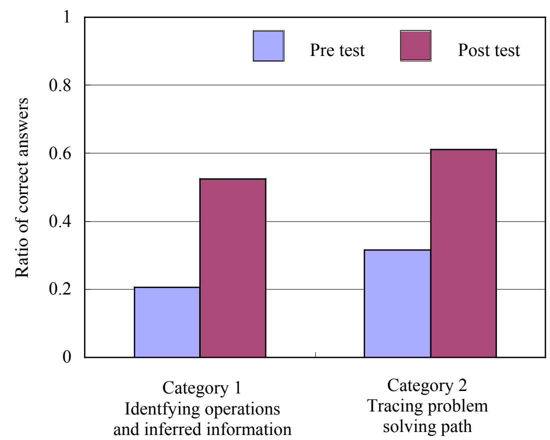

Figure 4 The results of the pre and post tests.

Table 2 shows representative findings discovered by the twelve participants who participated in the second class hours. Eight of the twelve participants mentioned findings relating to the causal relation between mental operations and behavior. Furthermore, many participants successfully noticed important aspects of influences of the system's cognitive abilities, such as mental operations that can be used, on the problem solving behavior. Some of the aspects had also been focused on in traditional literatures on human problem solving, such as Human Problem Solving in 1972.

Discussion and Conclusions

There are some other trials where a cognitive model as a hypothesis-deduction system is used in cognitive science education. For example, COGENT provides learners with an educational environment in which they are allowed to construct computational models while setting up various types of mental operation and simulate the behavior by executing the models on a computer (Cooper, 2002).

COGENT is a cognitive architecture on which students construct individual models working various domains by themselves. COGENT seems to be designed for advanced graduate students who are specifically interested in cognitive modeling, since to handle COGENT effectively a considerable number of hours are generally needed for training. Moreover, the students must be familiar with basic

Table 2 Representative findings on the relation between mental operations and behavior.

Participant 2 Knowledge for processing parity and inequality information was more difficult for me to understand than knowledge for processing digit information. So I tried to eliminate these types of knowledge and then simulated the behavior. As a result, many contradictions appeared, enormous trial and error processes emerged, and the length of the problem solving path became much longer. Even though humans can use unconscious knowledge in solving a problem, it may be difficult to identify the knowledge explicitly.
Participant 3 The amount of knowledge and the problem solving performance did not necessarily correlate. Even if knowledge used many times was absent, the performance did not become so different.
Participant 6 Knowledge is essential to solve a problem. However when certain specific knowledge was absent, other certain knowledge sometimes prevented the problem solver from solving the problem effectively. When solving a problem, all knowledge relating to the problem solving is not necessarily applied.
Participant 7 Complete knowledge is not necessarily needed to solve a problem. However if certain knowledge cannot be used, the length of problem solving path becomes much longer.
Participant 8 Dc1 is the knowledge that humans are most likely to forget. When this knowledge was absent, the problem solver tried to test many assignments arbitrarily and faced enormous contradictions. Even though this type of crucial knowledge is absent, the problem solver can reach the solution; however, the time for acquiring the solution becomes much longer.
Participant 10 When I solved the problem, I often forgot to coordinate information. So I eliminated the CD operations from the problem solver. As a result, the length of problem solving path became two or three times longer. It was not so impressive to me because coordination of information is crucial for solving this problem.
Participant 11 When I solved the problem, the only knowledge I could not imagine was only Dc1 and Dc2. In the complete problem solver, these pieces of knowledge fired only twice. I was surprised, however, that 94 episodes were needed to solve the problem when the problem solver could not apply these pieces.
Participant 12 When the CD operations were absent, the number of arbitrary assignments became greater. An episode moves to another episode when a contradiction is identified. When the problem solver accidentally hit a correct assignment, the problem solving progressed suddenly. Even though a part of the knowledge is absent, the problem can be solved by an exhaustive search.

techniques of artificial intelligence. In contrast, our system has been developed for introductory students in cognitive science/psychology, who may not be very interested in computational modeling. Neither specific knowledge nor training is needed to use our system. Without specific preparations, the learners are guided to explore various aspects of the relationship between mental operations and behavior, which is crucially important for introductory students.

Contrary to COGENT, it is important to note a strong constraint that in our system a task and operations for solving the task were initially determined. However, the findings discovered by the participants, as shown in Table 2, are relatively general and task-independent, which relate to many domains of problem solving. Additionally, it should be noted that some of the participants, such as Participants 2, 8, 10, and 11, designed their simulation based on their own experience of problem solving that was engaged in the initial stage of the first class hour, and they tried to interpret the problem solver's behavior while connected with their own problem solving activities. This implies that our system could function as an environment for experiencing meta cognitive activities where the participants are guided to perform self-reflective activities with their own problem solving processes.

We believe that these activities are important for introductory students who are interested in studies dealing with human/machine intelligence, and can be brought about by the learners' self-organized exploration of "an assumption of operations - deduction – behavior" cycles in the framework shown in Figure 1.

References

Cooper, R. P. (2002). Modeling high-level cognitive processes. LEA.

Kaplan, C. A., & Simon, H. A. (1999). In Search of Insight. Cognitive Psychology, 22, 374_419.

Klahr, D. (2000). Exploring science: The cognition and development of discovery processes. Cambridge, Mass.: MIT Press.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. Cognitive Science, 12, 1-48.

Klahr, D., Langley, P., & Neches, R. (Eds.), (1987). Production system models of learning and development. MIT press.

Kulkarni, D., & Simon, H. A. (1988). The process of scientific discovery: The strategy of experimentation. Cognitive Science, 13, 139-176.

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, 11, 65-99.

Miwa, K. (1999). Deductive and Inductive Ways of Investigation on Human Problem Solving. Journal of Japanese Society for Artificial Intelligence, 14, 1165-1176.

Miwa, K., & Simon, H. (1993). Production System Modeling to Represent Individual Differences: Tradeoff between Simplicity and Accuracy in Simulation of Behavior. Proceedings of Artificial Intelligence and Simulation of Behavior Conference, 158_167.

Newell, A., & Simon, H. (1972). Human problem solving. Prentice-Hall, Englewood Cliffs, NJ.

Zhang, J., & Norman, D. (1994). Representations in distributed cognitive tasks. Cognitive Science, 18, 87-122.